Status of the digitization
Paolo Andreetto, Alessio Gianelle
INFN Padova
Logical steps

- Track simulation: from simulated hits to charge distribution over the sensors
- Sensor simulation: from charge distribution to matrixes of pixels (end of digitization)
- Clustering: from matrix of pixels to cluster of pixels (reconstructed hits)
Track simulation

Starting point: CMS wiki page on digitization

Pipeline:
• Ionization of the track
• Drift to the surface
• Diffusion of the charge
• Noise and thresholds

Final results: Simone’s presentation
Sensor simulation (first implementation)

Chipset RD53A

Complete simulation of the threshold mechanism

Pixel pile-up

Buffering and external trigger are excluded

No linear time operations in the code

References from paper on optimal use of charge information
Sensor simulation (implementation for tests)

Trivial sensor

Total charge for all simulated hits is delivered at the end of the clock step

No pixel pile-up

Used to bypass the complexity of a real sensor
Clustering

Algorithms

Hoshen-Kopelman (thresholding algorithm)

- Linear scan of the sensor, Find-Union pattern

- Variants implemented: HK-4 and HK-8

Radial sweep (contour algorithm)

- Not used, for future developments
Performance and results

Test - 100 events (signal), full BIB – total amount of time:

- RD53A chipset with HK8: ~5000 s/evt
- Trivial sensor with HK8: ~100 s/evt

Effective time related to the number of CPUs (up to the number of ladders)

Max hit reduction: ~15%

Tests run in the CloudVeneto infrastructure: VM with 32 cores 340 GB
Analysis

SimHit - RecoHit distance

Signal Hit offset

SignalHitDistance
Entries: 470
Mean: 0.008616
Std Dev: 0.01672

BIB Hit offset

BIBHitDistance
Entries: 4.60786E+07
Mean: 0.011141
Std Dev: 0.00304
Analysis

Relation: SimHits merged in a cluster

HK8 effects for tail (to be investigated)
Analysis

Relation: Clusters produced by a SimHit

Up to 5 clusters:
Tracks on multiple sensors

Above 5 clusters:
To be investigated

March 08, 2022
The proposed digitization is not yet competitive with the previous version.

Cluster analysis and cuts can be a valuable solution.
Reject clusters with size > 7 pixels:

- Hit reduction comparable with previous solution
- Signal hits lost ~5%
Consideration and next steps

The code is stable but further analysis of the behavior of the digitizer is required.

The processor is specific for the Vertex barrel, porting to other modules is the next step.

New techniques for the clustering:

- Evaluate new clustering algorithms
- Clustering analysis (shape, contour)
- Introducing ANN in the digitizer (integrating tensorflow lite)

The code is in github (branch experimental); it’s not yet integrated in the framework.
Strange behavior of BIB Near the origin
To be investigated