Development of clustering algorithm for pixel detectors for FPGA

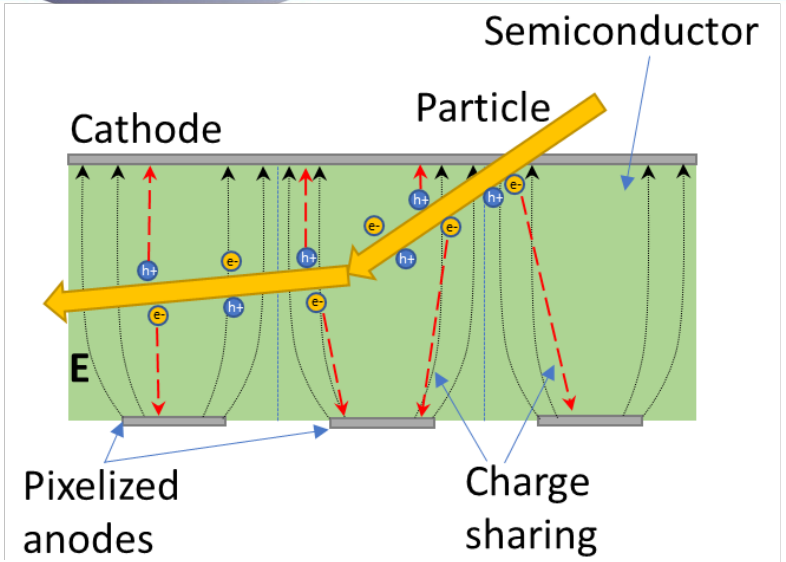
Aleksandr Lapkin (JINR, DLNP)

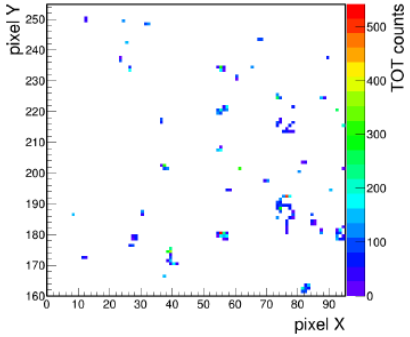
The XXVII International Scientific Conference of Young Scientists and Specialists (AYSS-2023)

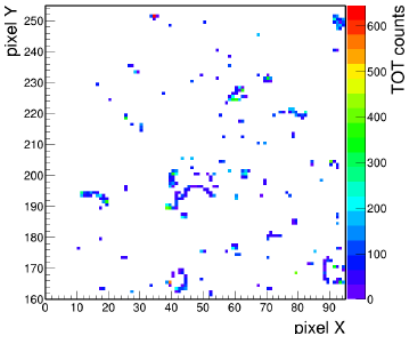
31 Oct 2023

Charge sharing

- When the particle and the sensor material interact, free charge carriers arise.
- Charge carriers induce a signal to the pixel electrodes
- Charge sharing - collecting charge from one particle in different pixels
- Causes of the charge sharing:
  - Interaction of a particle with the detector material opposite the pixel boundary
  - Long track of a particle in a semiconductor
  - Diffusion of a cloud of charge carriers
  - Inhomogeneity of the electric field in a semiconductor (near the electrodes, near the edge of the detector)
  - Semiconductor defects
  - Fluorescence

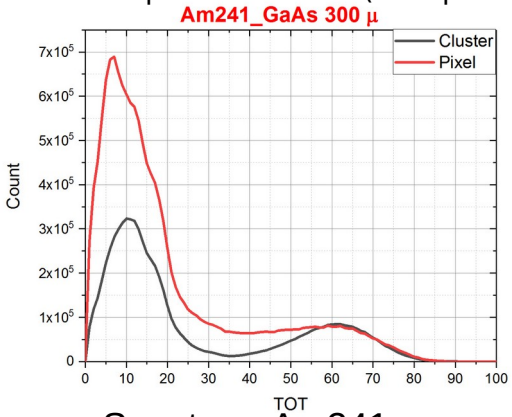




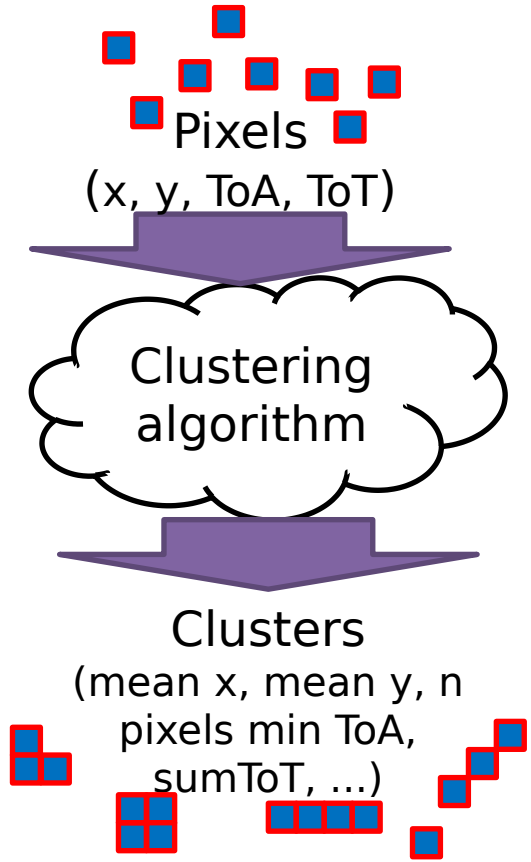


Clustering

- Clustering is the process of combining neighboring non-zero pixels.
- A cluster is a group of non-zero pixels with common boundaries, surrounded on all sides by pixels with zero data.
- Non-zero pixels that have a common angle are considered to belong to the same cluster.
- Clustering is necessary for energy measurement
- Clustering allow to improve spatial resolution
- FPGA clustering faster and can be done during readout
- It is applied for:
  - Big expariments as part of track reconstruction and triggers systems: LHCB, SPD(expected)
  - Standalone pixel detectors (Timepix family). For example, for tomography.



Spectrum Am241

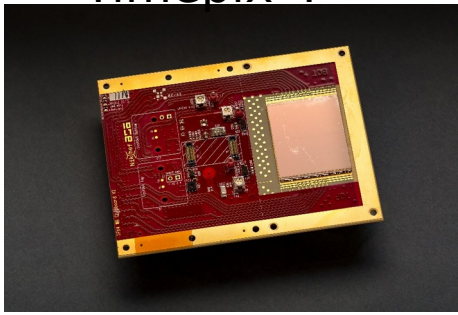


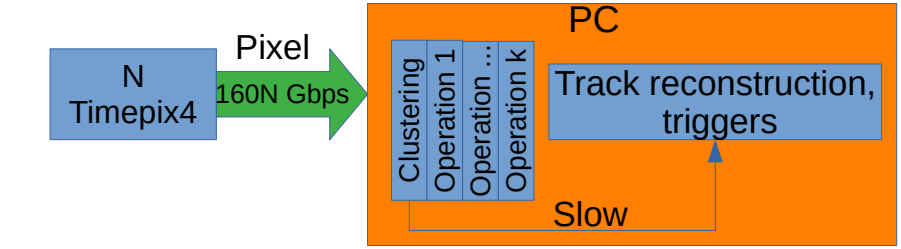
Timepix family detectors

- Timepix — semiconductor hybrid pixel detectors developed by Medipix colobaration (CERN)
- It is needed to reduce dataflow from Timepix 4 because it is too high (up to **163,84 Gbps**)
- Reducing data flow is neccessary for detectors made of several Timepix 4
- Clustering is needed for energy measurement in any case
- FPGAs usually are base of readout system

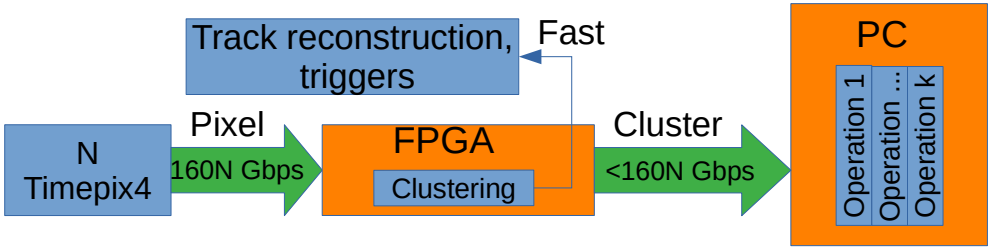
Feature	Timepix 1	Timepix 3	Timepix 4
Technology	55 um	55 um	55 um
Pixel size	250 nm	130 nm	65 nm
Range x/y	256/256	256/256	448/512
Readout modes	Frame	Frame/ Data driven	Frame/ Data driven
Max count rate (M counts /mm² /s)	<3200 fps	0,43	3,58
Bandwidth	< 3,2 Gbps	< 5,12 Gbps	**< 163,84 Gbps**

Timepix 4





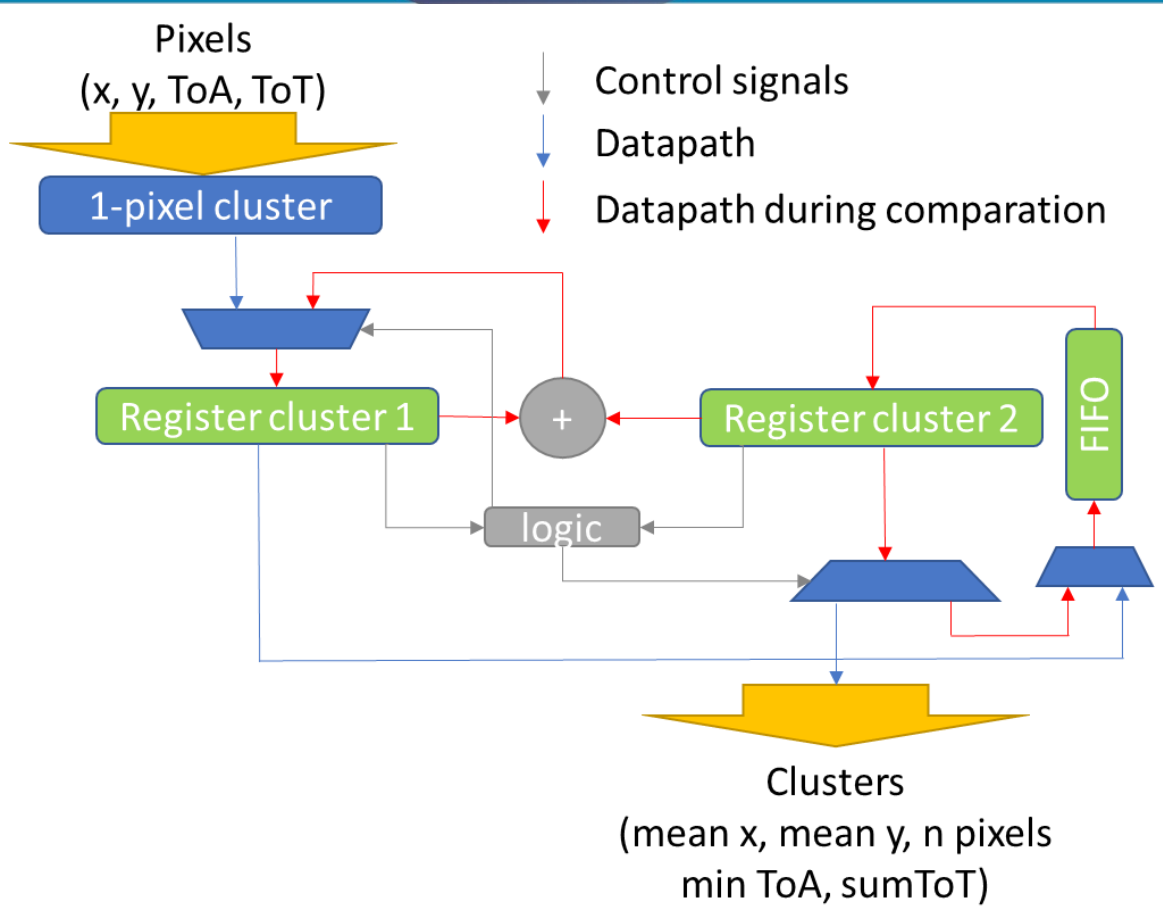
Program clustering



FPGA clustering

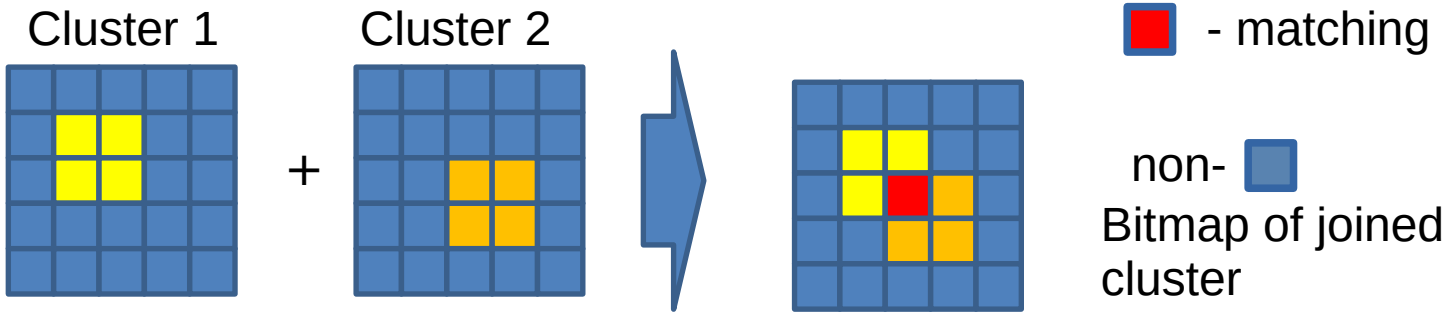
The algorithm

- The simplest way - brute force and pairwise matching
- Each new pixel is converted in cluster
- Each new cluster is matched with all cluster in memory
- If clusters are parts of one cluster they are joined. The resulting cluster is still matched with remain clusters in memory
- All clusters in memory were matched with each other
- Time consumption n^2 where n — number pixels



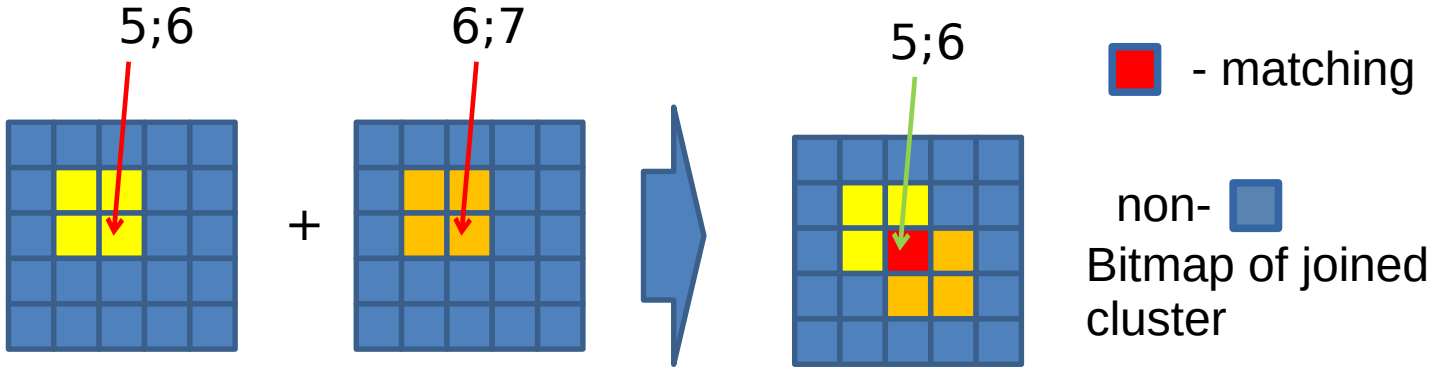
Matching mechanism

- A simple quick mechenism is requred sor cluster matching
- Bitmap — bit array. Each pixel correspond the sertain pixel of the detector
- If pixel (x, y) is belonging to the cluster, then bits (x, y),(x+1, y),(x+1, y+1),(x, y+1) is equal to 1 in bitmap.
- Matching ($|(bitmap_1 \ \& \ bitmap_2)$)
- Bitmap of joined cluster ($bitmap_1 \mid bitmap_2$)
- Bitmap require 8kB memory. Too much.



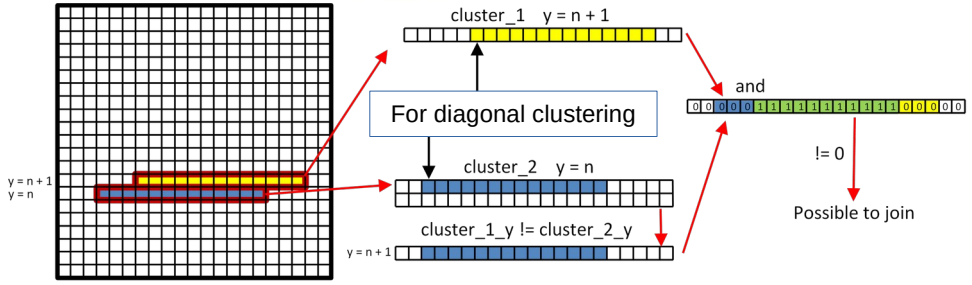
Matching mechanism

- A partial bitmap - bit array. Each pixel correspond the sertain pixel of the detector area
- Additional data is requared. For example coordinates of the center of the bitmap
- Matching and joining requare aligment of the cluster bitmaps
- Reliability is not guaranteed

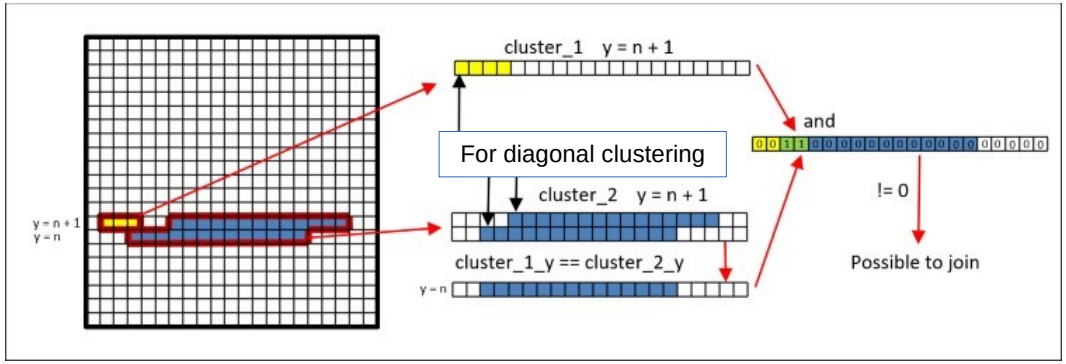


Sorted pixel flow

- Pixel sorting can be easyly done on FPGA. Some pixel detectors already have sorted output pixel flow
- Clusters grow only from the side of large y
- Bitmap — two rows (columns), additional data — coordinate y of the top cluster row.
- Creteria of output ready cluster. If y of a new cluster more than y+1 of cluster from memory, cluster from memory is ready.
- Reducing the cluster memory volume. No more than two row clusters are placed in memory
- Improve speed. It is not needed to wait end of clustering to received cluster data.
- Reliability is not lost



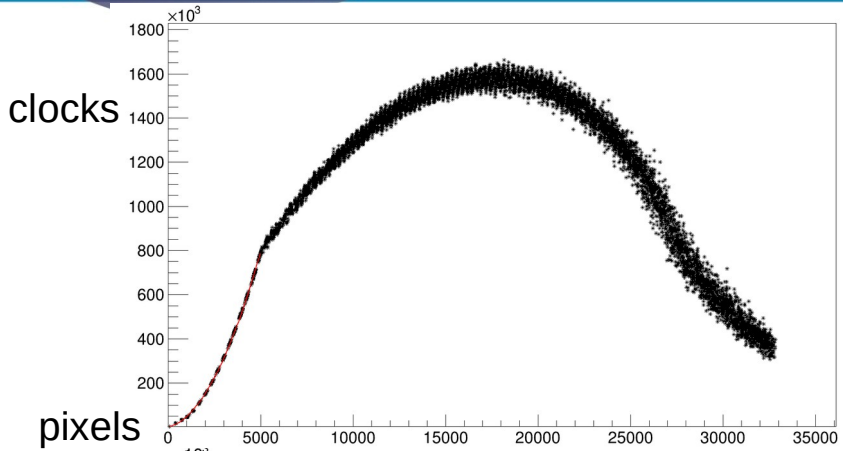
Different rows case

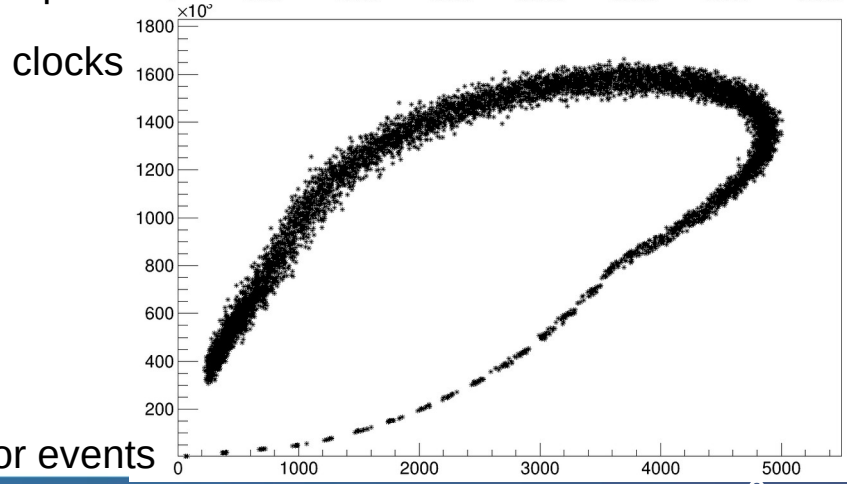


Same rows case

Simulation

- There was a simulation of the algoritm
- The simulation was on SAPHIR Data center
- Acceleration cards AMD Alveo U200 were used
- The simulation was done on generated random frames
- Each pixel in a frame had equal probability to have data
- 10000 frames were generated with 100 different probabilities for pixels to have data
- Errors (incorrect clustered frames) 3,86% (errors in transmitiion infrastructure)
- Time grows quadratic before 5000 pixels, after slowly

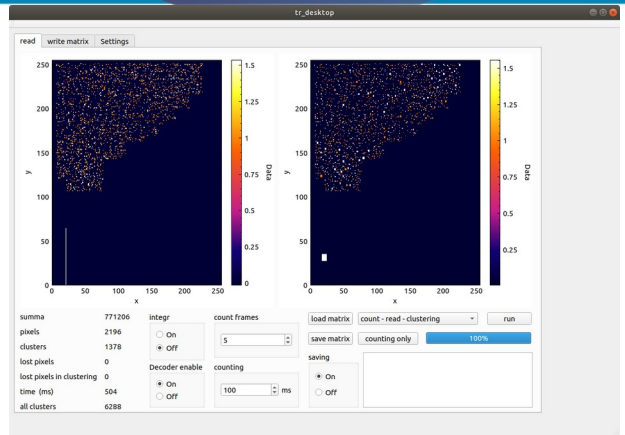




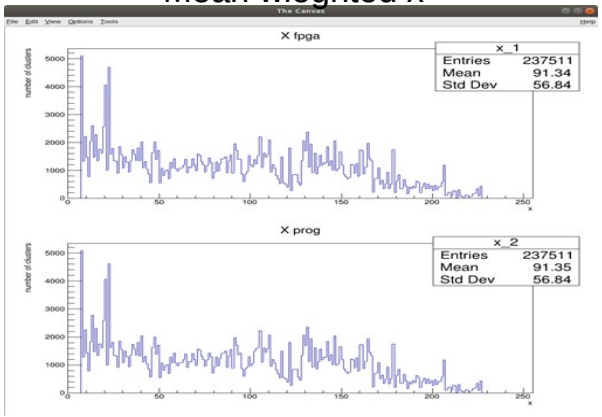
Readout 1

- Based on Arrow SoCKit Development Kit
- Outputs pixels and cluster data
- 40 MHz clock
- Detector output sorted pixel flow
- The results of PC and FPGA clustering are indistinguishable
- The distingtion in mean weighted x. It is rounding error

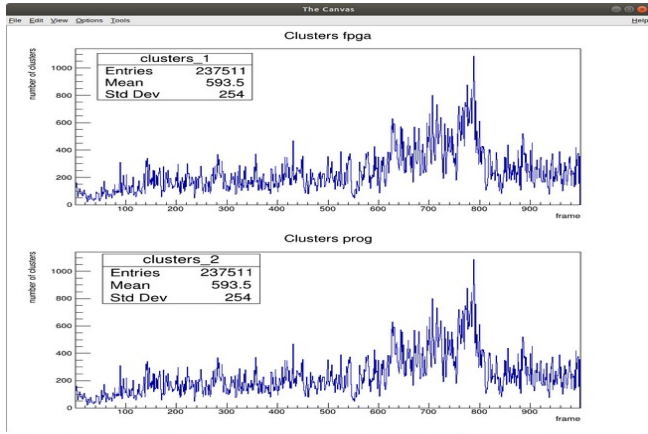
Software



Mean wieghted x

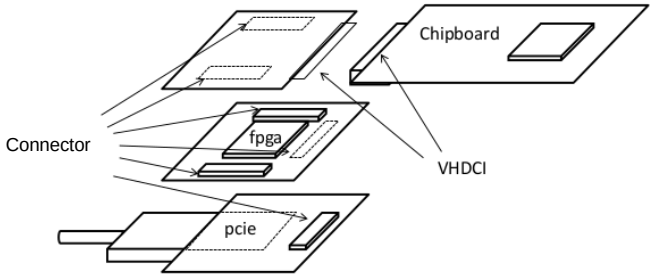


Cluster number



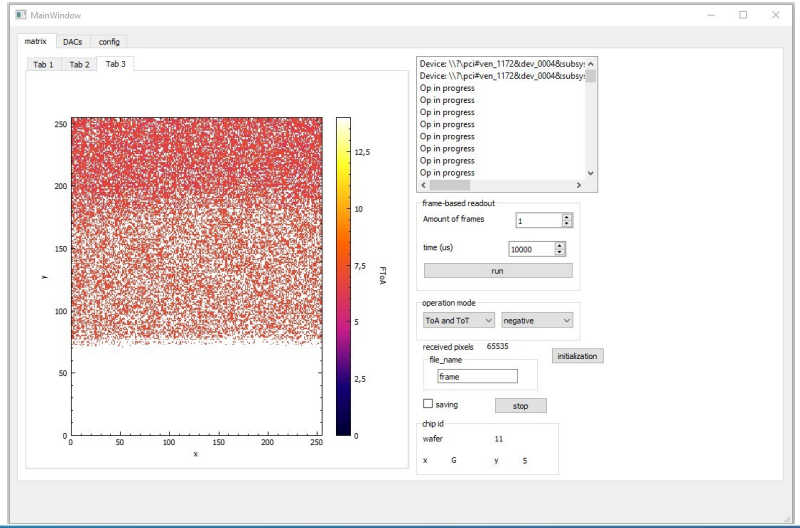
Readout 2

- Designed board based on Cyclone V GT
- Outputs pixels and clusters data
- 100 MHz clock
- Detector has 8 output without sertain order of pixels.
- FPGA serialize and sorte input flow
- Clustering tested only for frame based readout
- The results of PC and FPGA clustering are indistinguishable
- Equalization



Readout scheme

Software



Conclusion

- An FPGA clustering algorithm for pixel detector reading systems was presented.
- Its functionality was substantiated and confirmed.
- Using simulation, the dependences of the calculation time on the amount of input data were obtained.
- The presented algorithm was built into existing reading systems.
- The operation of the algorithm in reading systems was tested and its performance was proven.

Thank you
for your attention!