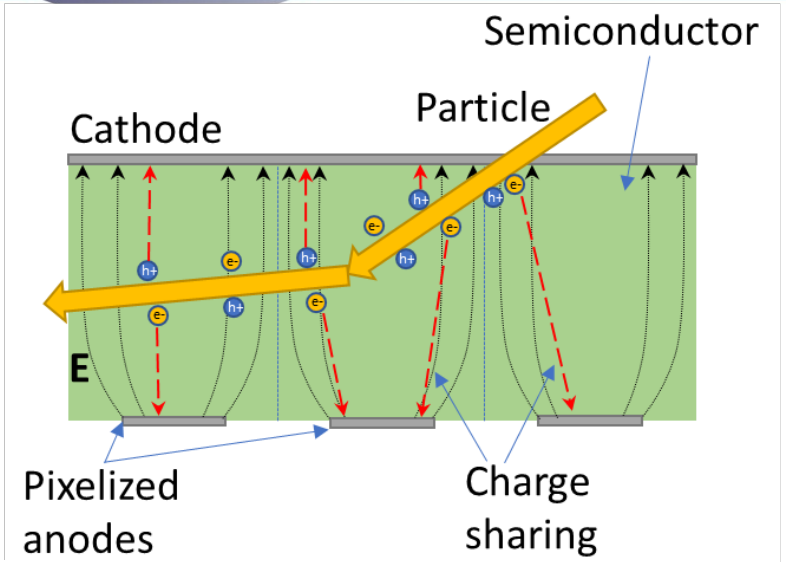
Development of Clustering Algorithm for Pixel Detctors for FPGA

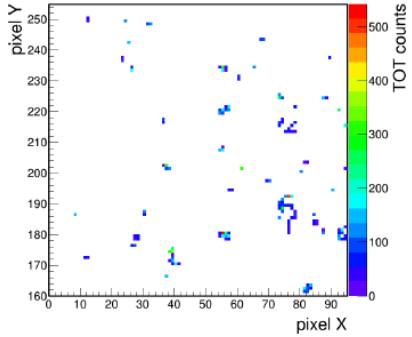
AYSS Conference Alushta-2024

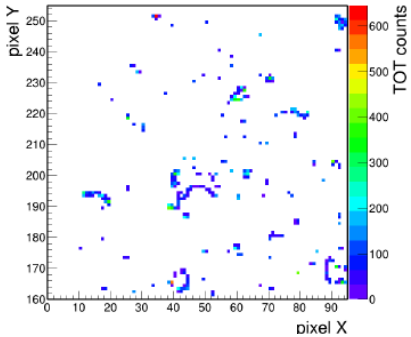
Aleksandr Lapkin
JINR DLNP

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

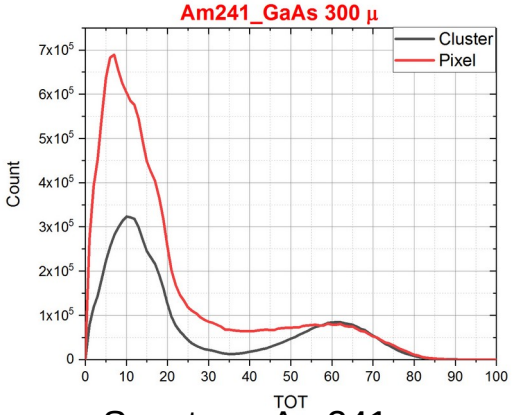




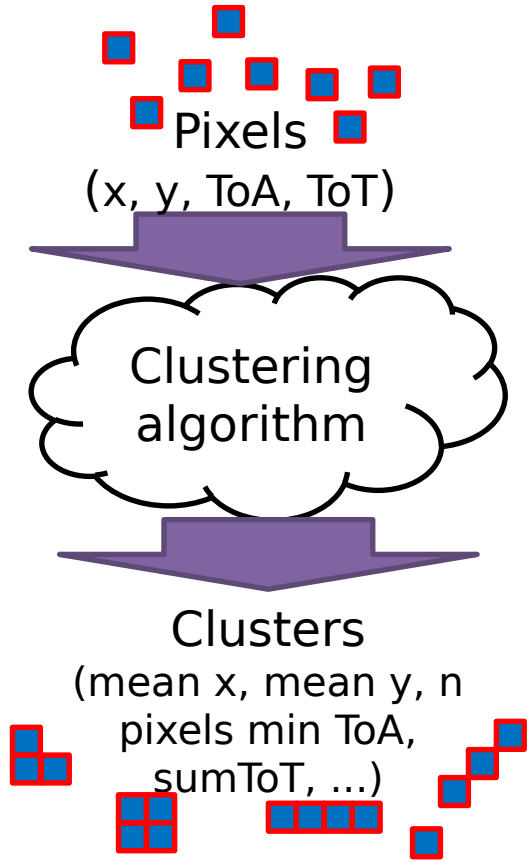


Clustering

- 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.

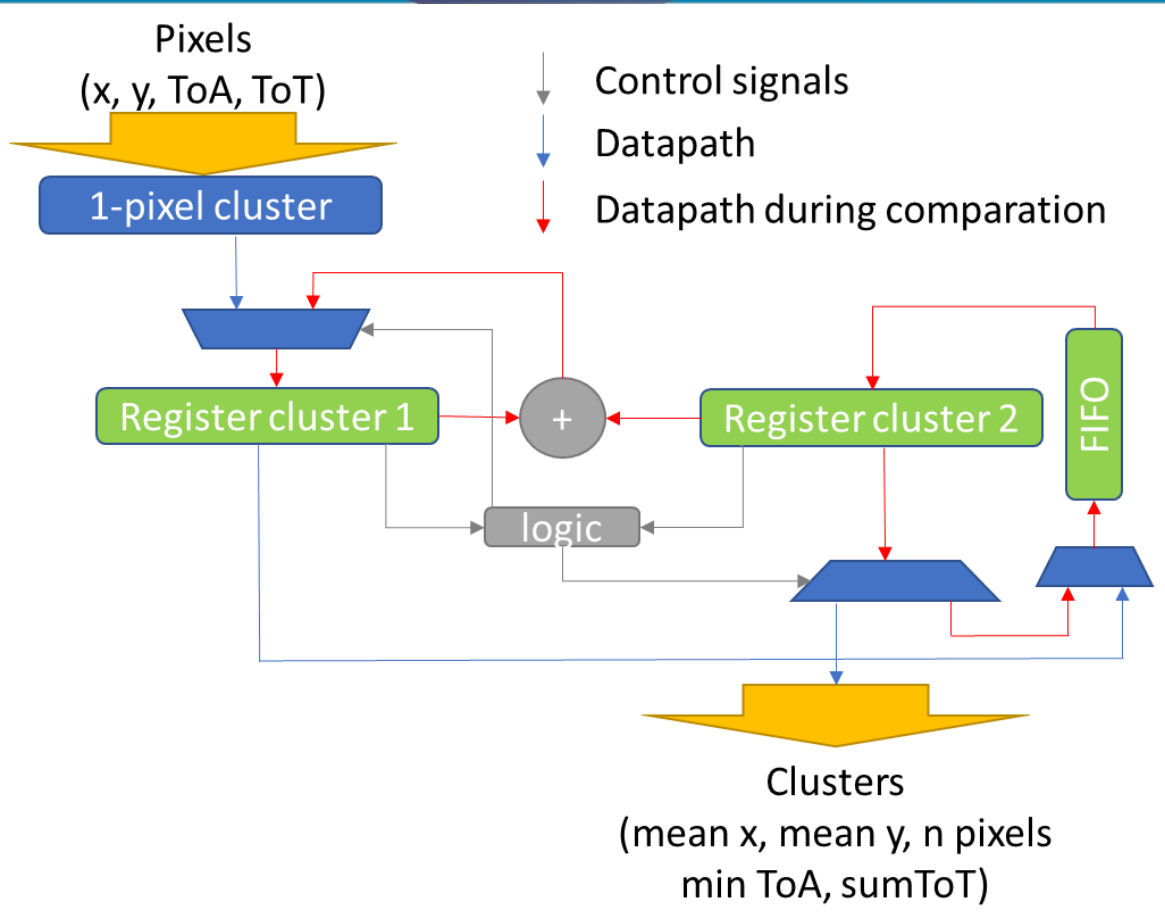


Spectrum Am241



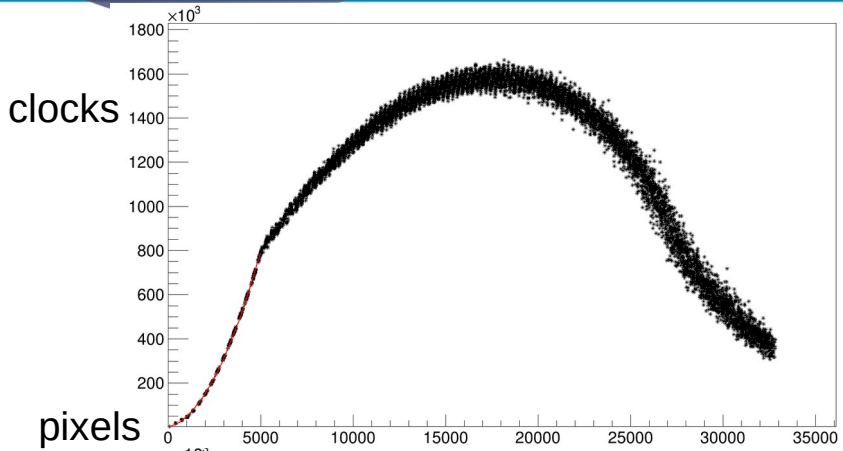
The algorithm

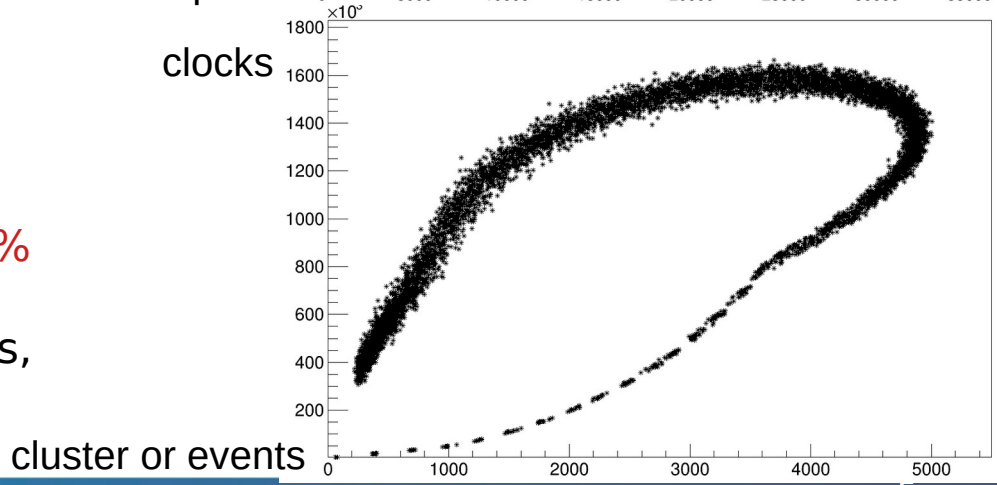
- The simplest way - brute force and pairwise matching
- Each new cluster is matched with all cluster in memory
- If clusters are parts of one cluster they are joined. The resulting cluster is still matched with remain clusters in memory
- All clusters in memory were matched with each other
- With a sorted input pixel stream it is possible to reduce complicated and make the algorithm faster



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

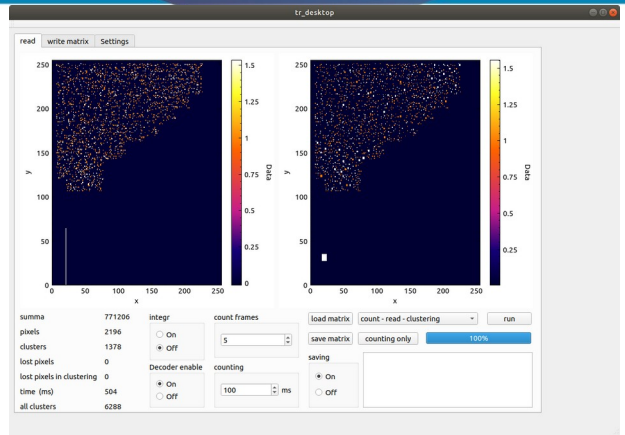




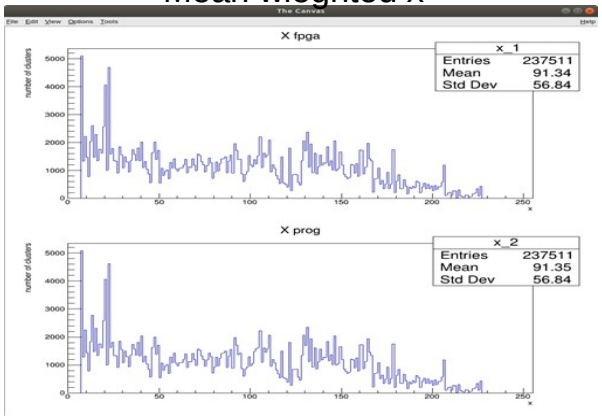
Readout test

- 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

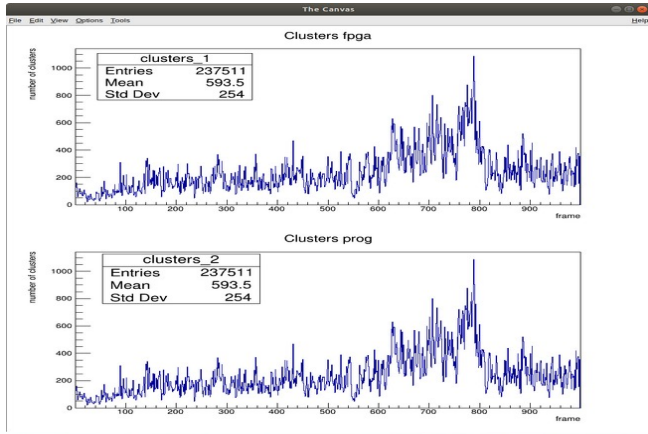
Software



Mean wieghted x

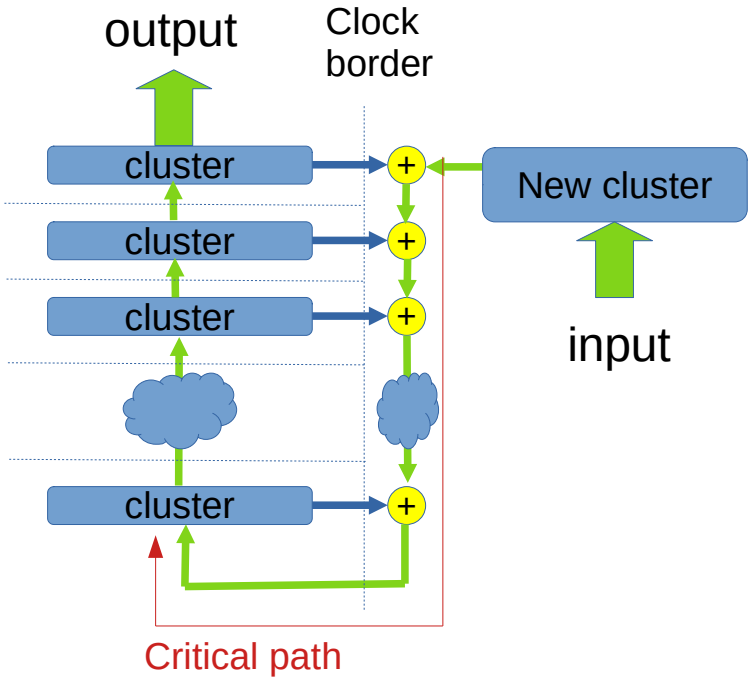


Cluster number



Parallel clustering

- Parallel clustering
- Each cluster go through registers pipeline
- Each pipeline register has a summing module and a matching module
- All cluster in the pipeline are compared with a new cluster
- A new cluster is compared with all clusters in the pipeline and is summed with matched clusters
- If a cluster in a register pipeline is match with the new cluster, it is removed. Else it is written in the next register in a pipeline
- Number of clusters = max clusters in a row
- Clustering time is constant
- Strict time conditions
- Large FPGA resources consumption
- The algorithm was debuged and simulated, but attempt to use it with randem generated data fell



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!