



Contribution ID: 1

Type: **Oral presentation**

# A Parallel FPGA Implementation for Real-Time 2D Pixel Clustering for the ATLAS Fast Tracker (FTK) Processor

*Wednesday, 14 May 2014 09:30 (30 minutes)*

The parallel 2D pixel clustering FPGA implementation used for the input system of the ATLAS Fast Tracker (FTK) processor is presented. The input system for the FTK processor will receive data from the Pixel and micro-strip detectors from inner ATLAS read out drivers (RODs) at full rate, for total of 760Gbs, as sent by the RODs after level1 triggers. Clustering serves two purposes, the first is to reduce the high rate of the received data before further processing, the second is to determine the cluster centroid to obtain the best spatial measurement. For the pixel detectors the clustering is implemented by using a 2D-clustering algorithm that takes advantage of a moving window technique to minimize the logic required for cluster identification. The cluster detection window size can be adjusted for optimizing the cluster identification process. Additionally, the implementation can be parallelized by instantiating multiple cores to identify different clusters independently thus exploiting more FPGA resources. This flexibility makes the implementation suitable for a variety of demanding image processing applications. The implementation is robust against bit errors in the input data stream and drops all data that cannot be identified. In the unlikely event of missing control words, the implementation will ensure stable data processing by inserting the missing control words in the data stream. The 2D pixel clustering implementation is developed and tested in both single flow and parallel versions. The first parallel version with 16 parallel cluster identification engines is presented. The input data from the RODs are received through S-Links and the processing units that follow the clustering implementation also require a single data stream, therefore data parallelizing (demultiplexing) and serializing (multiplexing) modules are introduced in order to accommodate the parallelized version and restore the data stream afterwards. The results of the first hardware tests of the single flow implementation on the custom FTK input mezzanine (IM) board are presented. We report on the integration of 16 parallel engines in the same FPGA and the resulting performances. The parallel 2D-clustering implementation has sufficient processing power to meet the specification for the Pixel layers of ATLAS, for up to 80 overlapping pp collisions that correspond to the maximum LHC luminosity planned until 2022.

## Summary

The ATLAS Fast Tracker (FTK) processor is a custom electronics system that will rapidly reconstruct tracks in the inner-detector Pixel and micro-strip layers from every event that passes the level-1 trigger. The input system for the FTK processor will receive data from the ATLAS inner track detectors read out drivers (RODs) at full rate, for total of 760Gbs. This massive amount of data throughput requires a data reduction technique with as little loss of useful data as possible.

A 2D-clustering FPGA implementation was developed to achieve this for the input of the Pixel detector. The role of the 2D-clustering implementation is combined: a) reduce the high rate of the received data, b) determine the cluster centroid to obtain the best spatial measurement.

The 2D-clustering implementation consists of three modules: a) the hit decoder module, b) the grid clustering module and c) the centroid calculation module. The hit decoder transforms the pixel hit data in a format useful for the clustering identification. It is the module that ensures that all the data are properly identified and even in the unlikely event of missing control words, it will ensure stable data processing inserting the missing

control words in the data stream. It's main operation however is to realign the pixel hits in order to be in the proper sequence for the clustering identification. The ATLAS Pixel modules [1] have 344x128 pixels which are read out by 16 front end chips (FEs). The FEs however are read out in an anticlockwise sequence which leads to half of the pixels being read out in the opposite direction than the other half. The hit decoder restores the proper pixel read out sequence by storing the reversed half of the hits in a LIFO and propagating out the hits in an increasing column number order.

The grid clustering module is the most computationally intensive part of the implementation and the one that performs the actual cluster identification. A moving window technique is used to identify the clusters. The first hit that arrives serves as a reference hit and is placed in the middle row and leftmost column of the window. The hits are read from the input until a hit with a column number outside the detection window arrives. The hits whose coordinates are not part of the detection window are stored in a circular buffer. To identify the cluster the reference hit then serves as a "seed" which propagates a "select" signal to change the state of all hits neighboring it. The "selected" hits are part of a cluster and are read out one by one and the hits which are in the detection window but don't belong to the cluster are stored in the circular buffer. In the next run the leftmost hit stored in the circular buffer is chosen as a reference hit, the detection window is filled first by hits from the circular buffer and then the input and the process is repeated until the pixel module is all read out and the circular buffer is empty. The detection window size is generic and can be adapted for different applications. For the ATLAS Pixel module a size of 8 columns x 21 rows was chosen.

The centroid calculation module is where each cluster is replaced by a set of coordinates, the centroid coordinates then corrected by a variable calculated by taking into account the absolute pixel position as well as the charge imbalance (using the measured Time-Over-Threshold for each pixel hit).

One fundamental characteristic of the 2D-clustering implementation is that different clustering engines can work independently and in parallel to identify different clusters, therefore increasing performance while exploiting more FPGA resources. However, the pixel data are received through S-Links [2] and the processing units that follow the clustering implementation also require a single data stream, therefore data parallelizing (demultiplexing) and serializing (multiplexing) modules are introduced in order to accommodate the parallelized version and restore the data stream afterwards. Each engine processes the data from one Pixel module. A parallel distributor module was developed that splits the data stream into the different engines by choosing the less busy one to propagate the next module into. The LVL1ids of the processed events are stored in a FIFO so that the same sequence of events can be recovered when the data stream is serialized again. A data merger module is used to serialize the data output in the same data sequence.

The single flow 2D-clustering will be tested on the custom FTK input mezzanine (IM) board using an 80MHz clock. Post place and route simulations with 80 overlapping pp collisions files have demonstrated a worst case estimate of 10 cycles / data word processing time. The x16 implementation has achieved a 65MHz maximum clock frequency and occupies 40% of a Spartan 6 lx150T FPGA device. Pixel data is received at a maximum 40MHz word rate. By using a x16 parallelization the 2D-clustering implementation will significantly exceed the processing power required for the Pixel detector. The 2D-clustering operation has been overlapping pp collisions that correspond to the maximum LHC luminosity planned until 2022.

#### References

1. G. Aad et al., "ATLAS pixel detector electronics and sensors", JINST 3(2008) P07007.
2. E. Van der Bij, R. McLaren, and Z. Meggyesi. "S-LINK: A Prototype of the ATLAS Read-out Link."

**Primary authors:** Dr SOTIROPOULOU, Calliope-Louisa T. (Aristotle University of Thessaloniki); GKAITATZIS, Stamatios (CERN)

**Co-authors:** Dr ANNOVI, A. (INFN-LNF); Dr PETRIDOU, C. (Aristotle University of Thessaloniki); Dr VOLPI, G. (University of Pisa); Dr KORDAS, K. (Aristotle University of Thessaloniki); Dr BERETTA, M. (INFN-LNF); Dr NIKOLAIDIS, S. (Aristotle University of Thessaloniki)

**Presenter:** GKAITATZIS, Stamatios (CERN)