A FPGA-based Cluster Finder for CMOS Monolithic Active Pixel Sensors of the MIMOSA-26 Family

Qiyan Li\textsuperscript{1,2}, S. Amar-Youcef\textsuperscript{1}, D. Doering\textsuperscript{1}, M. Deveaux\textsuperscript{1}, I. Fröhlich\textsuperscript{1}, M. Koziel\textsuperscript{1}, E. Krebs\textsuperscript{1}, B. Linnik\textsuperscript{1}, J. Michel\textsuperscript{1}, B. Milanovic\textsuperscript{1}, C. Müntz\textsuperscript{1}, J. Stroth\textsuperscript{1,3}, and T. Tischler\textsuperscript{1}

\textsuperscript{1}Goethe University Frankfurt, Max-von-Laue-Str. 1, 60438 Frankfurt/M, Germany
\textsuperscript{2}Central China Normal University, Luoyulu.152, 430079 Wuhan, China
\textsuperscript{3}GSI Helmholtzzentrum für Schwerionenforschung, 64291 Darmstadt, Germany

E-mail: Li.Q@physik.uni-frankfurt.de

Abstract. CMOS Monolithic Active Pixel Sensors (MAPS) demonstrated excellent performances in the field of charged particle tracking. Among their strong points are an single point resolution few $\mu$m, a light material budget of 0.05\% X$_0$ in combination with a good radiation tolerance and high rate capability. Those features make the sensors a valuable technology for vertex detectors of various experiments in heavy ion and particle physics. To reduce the load on the event builders and future mass storage systems, we have developed algorithms suited for preprocessing and reducing the data streams generated by the MAPS. This real-time processing employs remaining free resources of the FPGAs of the readout controllers of the detector and complements the on-chip data reduction circuits of the MAPS.

1. Introduction

Traditional DAQ-systems of experiments in heavy ion and particle physics rely on multiple trigger systems. Triggers reject experimental data unless a well-defined condition is met. This reduces the data rates efficiently provided the time needed to obtain a trigger decision is in the order of the time between two events. Future experiments like the CBM at FAIR do not fulfill this condition as one aims at combining very high data rates with complex trigger conditions.

To avoid excessive dead-times, the CBM-experiment will rely on a free streaming DAQ system. This system attaches a time stamp to the information on each fired detector cell and streams the data to a processor farm, which scans the data for complex patterns like secondary decay vertices. This approach eases the handling of latencies in the processing as the data is rapidly evacuated from the aggressive radiation environment of the cave and hereafter stored in relatively cheap off-the-shelf RAM until a trigger decision arrives. On the other hand, it requires enormous network bandwidths and computing resources, which defines a significant cost factor. Those costs can be reduced by minimizing the data stream of the detector systems and by reducing the number of processing steps to be carried out on the processor farm. To do so, we studied the option to preprocess the data of the Micro Vertex Detector of CBM already on the FPGAs of its readout controller boards (ROC). The study was carried out based on the MAPS prototype MIMOSA-26, which is considered as a realistic precursor of the final sensor chip for CBM. Moreover, MIMOSA-26 or compatible devices are being used in the STAR-HFT, the EUDet telescope.
2. Fundamental approach for data preprocessing

2.1. MIMOSA-26 and its native data processing

MIMOSA-26 [1] hosts a pixel matrix of 1152 columns with 576 pixels each. The pixel pitch is 18.4 \( \mu \)m. The sensor is readout with a column parallel rolling shutter: the signals of one line of pixels is send to an array of 1152 discriminators, which is located on the same chip aside the sensor matrix. After discriminating the pixel signals, an on-chip data zero suppression logic scans the digital pattern obtained for groups of up to four consecutive fired pixels. The groups are encoded into dedicated 16-bit data words and stored in a block memory, which handles up to 6 data words originating from blocks of 64 neighboring pixels. In a next step, the words are moved to a line memory holding up to 9 words and hereafter transported to the global output buffer. The total processing time for one line is 200 ns, which turns into a 115.2 \( \mu \)s frame readout time. Once the processing of a frame is completed, the related data is pushed via two digital links of 80 Mbps each to a controlling FPGA.

Besides a header, which contains amongst others data from an internal frame counter, the native output data format of MIMOSA-26 consists of two different data words, which will be referred as states hereafter. The first kind of state, the line state, encodes the line number in which groups of firing pixels were identified. The second kind of states, the cluster state, holds the information on the individual groups of pixels. The details on the data format of the states according to the manual of MIMOSA-26 [2] are shown in figure 1 and figure 2.

![Figure 1. State encoding the line address.](image1.png)

![Figure 2. State encoding the column address.](image2.png)

2.2. Requirements on an external preprocessing of the MIMOSA-26 data stream

MAPS like MIMOSA-26 collect their signal charge by means of thermal diffusion, which causes a significant charge sharing. One individual particle hit creates therefore typically 2-dimensional clusters formed from several firing pixels. The 1-dimensional cluster finder of the chip, which acts on lines only, encodes the clusters into several states. Performing an additional, 2-dimensional cluster finding already on so far unused FPGA resources upstream the processor farm was considered as a promising strategy to reduce the load of this farm. Besides the need to fit to the free FPGA resources, the algorithm was required to reduce the data volume by at the same time retaining the full cluster information.

To match the requirement on the data rates, we aimed at encoding the full cluster information into one single data word. Finding a suited encoding requires a detailed understanding of the nature and shape of the clusters. They were studied based on the response of MIMOSA-26AHR prototypes (with 15 \( \mu \)m thick epitaxial layer, see [1] for details) to photons of a \( ^{55} \)Fe-source and pions of the CERN-SPS. Data was taken for various sensor temperatures, particle inclination angles, and discriminator thresholds and the width \( W_x \) and \( W_y \) of the clusters in row- and column-direction was analyzed as shown in figure 3. We concluded that in all cases observed, \( W_x \) and \( W_y \) remains smaller than 10, and maximum of cluster size (\( W_x \cdot W_y \)) is smaller than 28. However, as shown in figure 4, the average clusters size is substantially smaller. Note that the number of pixels in a cluster is lower than known from earlier studies like [4]. This is plausible as the partially depleted sensor of MIMOSA-26AHR concentrates the signal charge into fewer pixels than the undepleted sensors of earlier prototypes.
2.3. Cluster shape encoding

Based on this information, we concluded that information on the clusters shape can be encoded lossless in a 32 bit word, which holds the width ($W_x \leq 9$) of the cluster in 4 bits and the status of the $\leq 28$ pixels (active or passive) per cluster with one bit per pixel. This data word has to be complemented with another $\geq 21$ bit data word encoding the position of cluster on the pixel matrix. This turns into a total of $\geq 53$ bits per cluster, which would, for reasons of convenience, most plausibly be embedded in 64-bit words at some point. If so, the encoding expands the data for all clusters, which were initially encoded with less than four native 16-bit states.

A more efficient encoding could be reached if the information on the position and the shape of the clusters was fitted in a single 32-bit word. Knowing that 21 bits have to be reserved for the position information, this leaves $\leq 11$ bit for shape information. In order to use this space most efficiently, we inspired ourselves by the ASCII-code. The 8-bit word of this code is insufficient to encode all possible symbols. However, each of the 256 entries is associated by means of a look-up table to the few symbols, which are of practical relevance for encoding texts composed from Latin symbols. In analogy to this concept, we tested if the numbers of relevant cluster shapes might stay below 1024 (10-bit). We assumed that the encoding remains lossless if the number of clusters not fitting into this encoding scheme remains substantially smaller than the $\sim 0.1\%$ inefficiency of the sensors.
Figure 6. The proposed format for encoding clusters in 32-bit.

Figure 7. Comparison of the data volume generated by SPS beam test data encoded in proposed format and the native data format of MIMOSA-26.

The results of the test are shown in figure 5, which sorts the shape of the clusters according to the probability of their occurrence. We find that only a very limited number of cluster shapes appear regularly and that substantially less then 0.01% of all clusters generate exceptions if the above mentioned coding concept is applied.

2.4. Offline test of the algorithm
The encoding concept was tested by means of a C++ software, which acted as an after-burner to the software analyzing the data obtained from the CBM-MVD prototype [3]. The 32-bit data words were implemented as shown in figure 6. An overflow bit was added. This bit is to indicate (i) an overflow of the on-chip data encoding as indicated by the MIMOSA-26AHR sensor and (ii) an overflow or exception of the cluster finding and cluster encoding. The data volume generated by the novel encoding algorithm was compared with the data volume of the initial data of the sensors. A representative result of the comparison is shown in figure 7, which displays the data volume needed to encode a sensor frame in multiples of 16 bit. One observes that the novel encoding scheme reduces the data by a factor of two. Moreover, the fluctuations are reduced, which helps to balance the load of the readout network.

3. Status of the implementation in FPGA
Our preprocessing algorithm will be implemented in the FPGAs of the TRBv3-boards[5], which are foreseen as ROCs of the MVD. Figure 8 (a) shows the modules of the current ROC logic. The data received from the sensors are checked for potential errors and synchronization problems. Idle bits are removed from the data stream. Hereafter, the data are stored in a frame buffer, which serves as input for the novel cluster finder. The output of this cluster finder is transferred to a readout buffer and shipped forward via the TRB-net.

Figure 8 (b) displays the main logic structure of cluster finder. The data of each new row arriving at the Read Row - module are compared with the potentially incomplete clusters known from previous rows. In case a new state matches geometrically a known cluster, it is added to this cluster. Otherwise, a new cluster is created. Once a cluster does not find new neighbors, it is considered as completed and sent to the Shape Coder-module. Here, the cluster shape is first encoded according to the 53-bit encoding scheme discussed in section 2.3. The result of this encoding is translated into the final 32-bit encoding by means of a look-up table.

To accelerate this translation, we apply a staged concept for accessing the lookup table: A fast coder is used to encode the 8 most abundant shapes and all fully symmetrical shapes.
without accessing the table. Moreover, the look-up table was subdivided into groups of shapes with identical numbers of fired pixels and each group was ordered such that the most abundant cluster shape is encoded with shortest access times. According to our FPGA-simulations, this optimization accelerated the access by a factor of two.

A first version of the VHDL code was implemented in a stand-alone FPGA and recently tested with the test patterns. The tests were successful in the sense that the algorithm recognized the right number of clusters within the anticipated processing time. In a next step, the code will be exposed to real data and finally integrated into the real time data processing chain of the TRBv3-board.

4. Summary and conclusion
The free streaming DAQ concept of the future CBM-experiment introduces harsh requirements on the performances of the processor farm, which is to select interesting events in real time. To reduce the load of this computer system, we studied the option to perform a real time cluster finding and encoding already in the FPGAs steering the MAPS of the CBM-MVD. This processing is of use if it does not expanded the data rate beyond the efficient high-level data protocol of the MAPS.

To match this requirement, we encode the most abundant cluster shapes within a 10-bit space for cluster shapes. We find that this solution allows for encoding more than 99.99% of all clusters recorded in the beam test into a 32-bit word, which reduces the data volume by a factor of more than two as compared to the native data format of state-of-the-art MIMOSA-26AHR sensors. Once the study was completed by means of off-line data processing and analysis, we have started to migrate the algorithm to a VHDL-code. First and preliminary tests suggest that a cluster finding can indeed be carried out on so-far un-used resources of the TRBv3-readout controller boards foreseen for the MVD readout.

Acknowledgements
This work was supported by HIC for FAIR and BMBF (06FY9099I and 05P12RFFC7).

References
[1] M. Deveaux et al., 2011 JINST 6 C02004
[2] http://www.iphc.cnrs.fr/IMG/pdf/M26_UserManual_light.pdf
[3] M. Koziel et al., NIM-A, http://dx.doi.org/10.1016/j.nima.2013.07.041
[4] C. Dritsa et al., PoS (BORMIO2010) 015.
[5] M. Traxler et al., 2011 JINST 6 C12004