1 September 2021
ryuji.moriya@glasgow.ac.uk

Simulating Hexagonal Pixel Sensor Design in Allpix^2

Ryuji Moriya
CERN Summer Student 2021, University of Glasgow

Keywords: Hexagonal pixel geometry, Allpix Squared, detector model, MIMOSA26, Simulation, Monte Carlo, semiconductor detector, particle physics detector

Summary

A hexagonal pixel grid is an alternative pixel sensor geometry that has several potential benefits compared to the rectangular/square pixel sensor geometry, such as improved timing resolution and better efficiency. In order to understand the performance of this design, a completely new detector class has been implemented within the Allpix^2 framework. Specifically, the getPixelIndex() function within the new detector class allows the program to successfully extract the index of the pixel based on the position of the hit. Finally, basic simulations involving a hexagonal pixel sensor grid were performed and plots such as hit map, cluster size, cluster charge, and cluster seed charge were extracted.

Contents

1 Introduction

Semiconductor detectors are common in a variety of scientific domains, such as high energy physics, nuclear physics, and observational astronomy. In many of these fields, the rectangular or squared pixel geometry is the most commonly used, for example, the planned upgrade of the ATLAS detector for the High-Luminosity Large Hadron Collider (HL-LHC) is expected to have an all-silicon inner tracker that will utilize square pixels[3].

However, an alternative pixel geometry exists that uses hexagonal pixels as part of the detector system. In this document, the hexagonal pixel geometry design will be studied using the Allpix^2 framework by implementing a new sensor geometry class and comparing the simulation results with that of the traditional square pixel geometry design.

1.1 Hexagonal Pixel Geometry

The hexagonal Pixel sensor design, as suggested by the name, is a semiconductor detector design that uses hexagonal pixels. There are a multitude of benefits that are associated with hexagonal pixels.

Firstly, for the same pixel area, the distance between the center electrode to the edge region is about 88% compared to that of the square pixel (see figure 1). The result is that the drift time of the electrons will be reduced for signal charge generated at the edge region and hence the time resolution of the detector is improved [2].

Secondly, at the edge region of the hexagonal pixel, there will be less charge sharing due to less pixel neighbors compared to a rectangular/square pixel. This is shown in figure 1. Due to less charge sharing, more charges are collected by the individual electrodes and therefore lead to greater detector efficiency [2].

1.2 Allpix^2

The simulation of a hexagonal pixel geometry is done with Allpix^2. Allpix^2 is an open source Monte Carlo simulation framework for semiconductor pixel detectors[4]. Specifically, it allows both simple and complex setup of particle physics detectors to be established and tested for their performance. Here, the simulation is done in a flexible fashion, where each step of the simulation from irradiation to digitization can be configured with each step in what is known as 'modules'.

Simulating square pixels with predefined detector models such as Timepix and Medipix 3 is easily performed due to being already available as existing models; however, this is not the case for semiconductor detectors such as FASTPIX[2]. This is due to the lack of functionality for simulating detectors with hexagonal pixels within Allpix^2.

2 Implementation of Hexagonal Pixel Sensor Class in Allpix^2

As described in the previous section, the lack of functionality for simulating detectors with hexagonal pixels makes it impossible to simulate devices such as FASTPIX. Hence, to over-

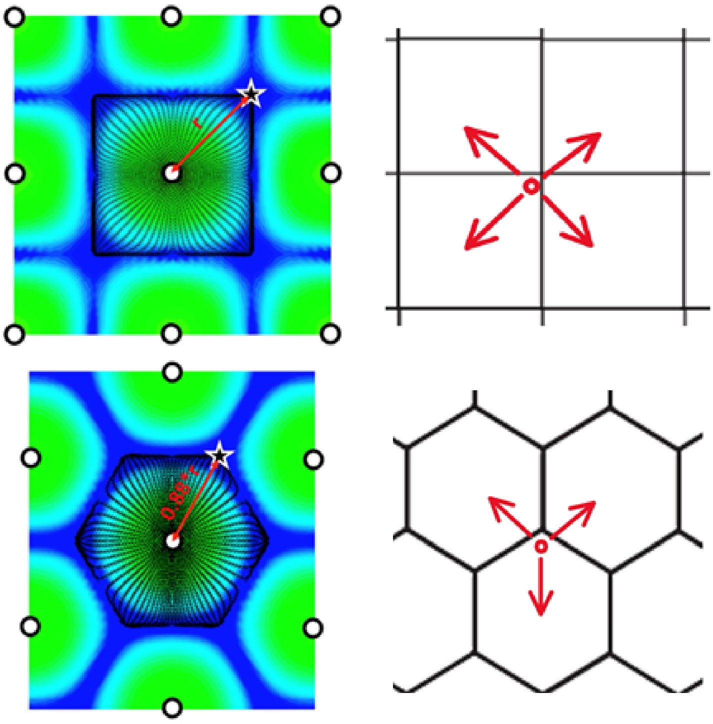

Figure 1: Comparison of rectangular and hexagonal geometry. Left: the top shows distance r, the distance connecting the center electrode to the edge in the square pixel. If this is compared to that of the hexagonal pixel with the same pixel area, then the distance is only 88% of the distance r. Right: the possible direction the charge could diffuse. It can be noted that for square pixels, electrons can diffuse to four pixel neighbors while only three for hexagonal pixels. Figure reproduced from [2].

come this problem, a new detector model class has been implemented within Allpix^2. Within this class, the local coordinate system is shown in figure 2 and as convention, the index of the pixel starts at 0 for both x and y coordinates. It is worth noting that unlike rectangular/square pixels, the pixels along the y-axis are not stacked on top of each other but alternate in a zig-zag pattern.

2.1 getPixelIndex()

The main function implemented within the new detector class is known as the getPixelIndex() function. The function takes the position of the hit and calculates the corresponding pixel index within the local coordinate system. It is worth noting that unlike in a square or rectangular pixel grid, the sides are not parallel to the x-axis nor the y-axis and therefore getting the pixel index simply by dividing the position by the pixel pitch will not yield the correct index.

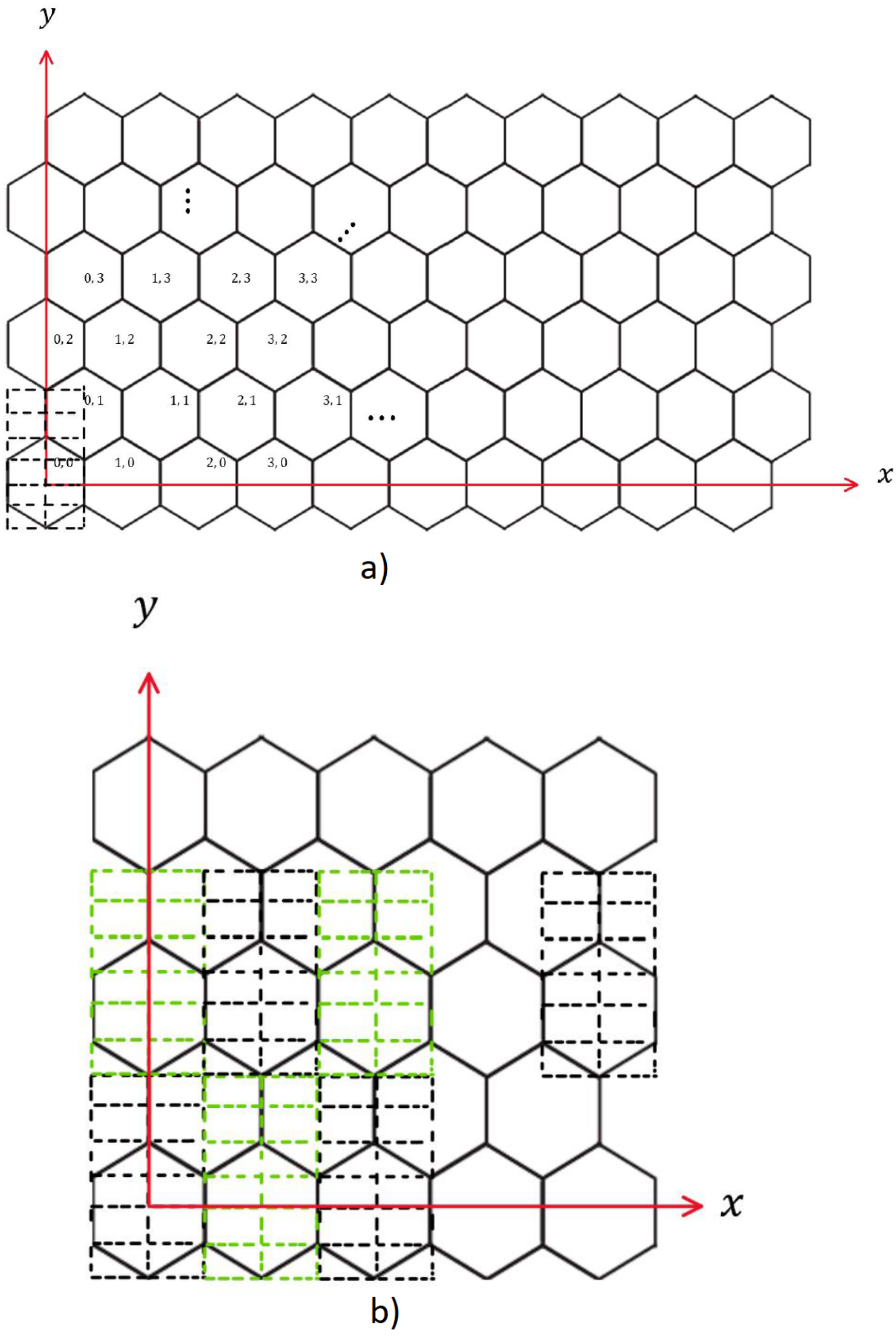

Figure 2: The hexagonal grid in local coordinate system. a) shows the hexagonal grid in the local coordinate system, where the origin is centered at the $(0, 0)$ pixel. b) shows the unit cell that is repeated throughout the hexagonal grid. Notice that the unit cell itself can be divided into 12 different regions to then assign a pixel index accordingly.

To correctly define the pixel index in the case of a hexagonal grid, it is necessary to exploit the periodicity of the hexagonal grid. Hence, a unit 'cell' was defined, as shown in figure 2. This cell is repeated throughout the grid as seen in the figure. The idea is to first treat the grid as a rectangular pixel and to calculate the pixel index. Then, within the pixel grid, a switch statement within C++ can be used to divide the cell into 12 different regions. The index of the hexagonal pixel can then be assigned correctly by considering which of the 12 regions the hit position corresponds to and subtract or add the x, y indices accordingly. One problem that arises from this method is how to calculate the index for 4 of the regions where the side length runs from corner to corner within the region. In this case, a linear equation was defined and an if statement was used to assign a pixel index depending on the position relative to the side length. The method presented here assumes an infinitely extending grid of hexagonal pixels and hence also works for indexing pixels with negative indices. However, this is nonphysical in this case and hence a separate function will test whether the indices is within or outside the hexagonal grid as specified by the user.

3 Basic Simulation of Hexagonal Pixel Sensor Grid

With the introduction of the hexagonal pixel detector class within the source code of Allpix^2, the first basic simulation comparing hexagonal pixels and square pixels is presented. In the simulation, the MIMOSA-26 sensor was used; MIMOSA-26 is a monolithic semiconductor detector that has square pixels of 1152×576 with a pixel pitch of 18.4 μm [1].

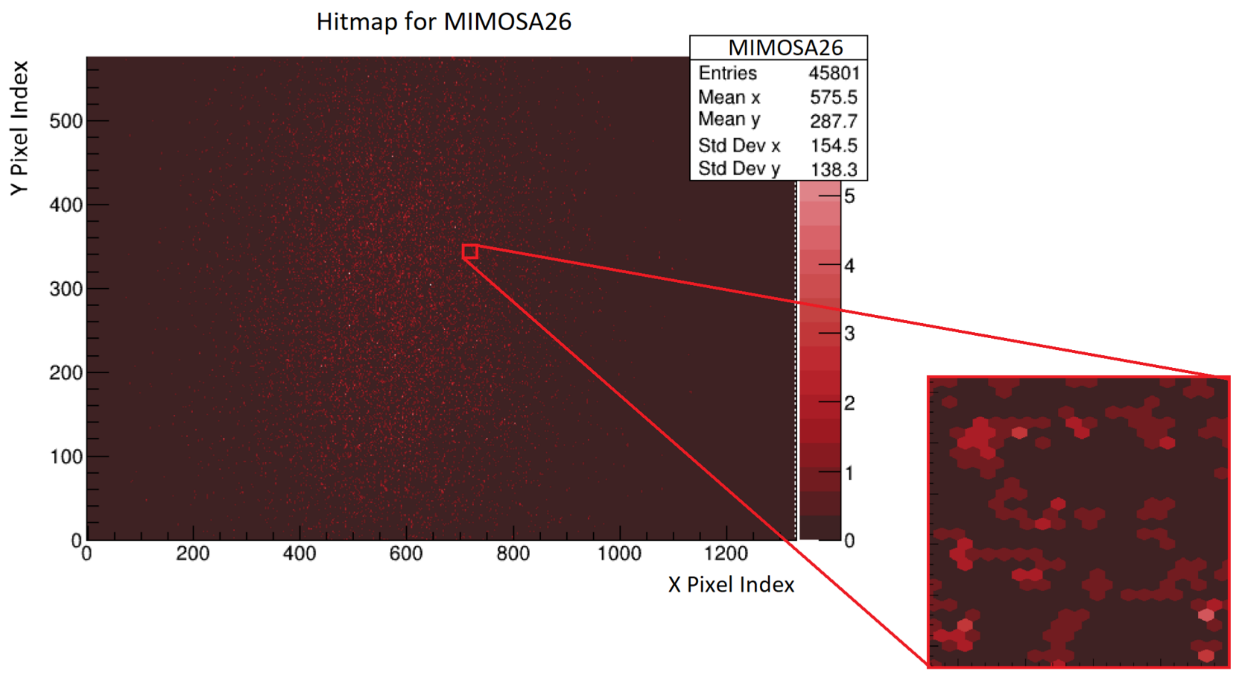

Figure 3: The hit map of MIMOSA-26 with hexagonal pixels. It is worth noting that beyond 1152 along the x-direction and 576 along the y-direction, no hits are registered, as hits beyond this will be out of the sensor grid.

For a fair comparison, only a few changes were introduced for the modified MIMOSA-26 model featuring hexagonal pixel cells, with all other parameters remaining the same. The first change is to the pixel pitch of the hexagonal pixel, which is defined to be the flat-to-flat distance. This was altered to a pixel pitch of 19.6 μm to ensure that both square and hexagonal pixels have the same active area.

It is worth noting that the comparison between the two detectors assumes effects purely due to geometrical effects. That is, only the simple linear electric field model was used and hence no custom electric field for hexagonal pixels was used for the simulation.

3.1 Hit Map

The hit map is a 2D map of the pixel grid that indicates which pixels in the sensor registered a hit. The simulation uses 15,000 events, but multiple hits may be registered with each event, e.g. through the production of delta-rays. The hit map for MIMOSA-26 with hexagonal pixels is shown in figure 3.

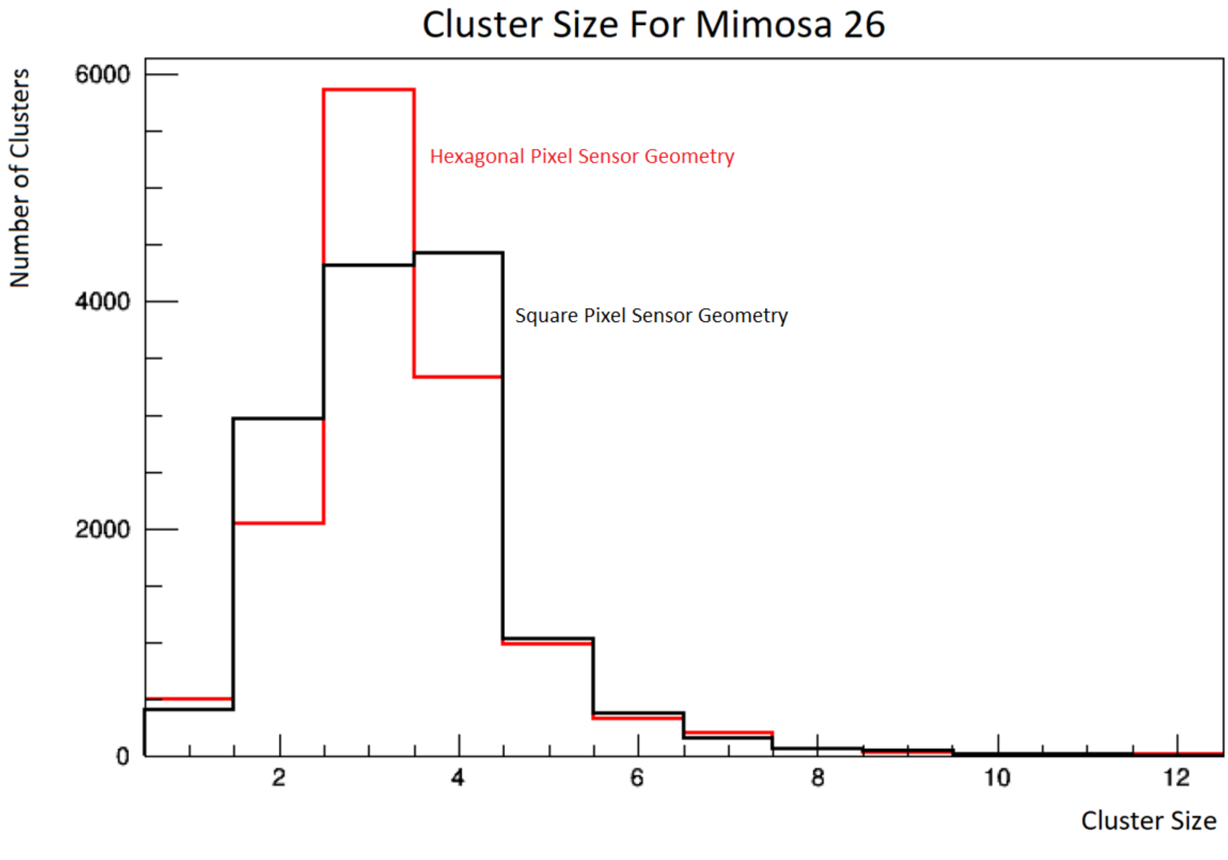

Figure 4: The cluster size of square pixels and hexagonal pixels superimposed on each other.

3.2 Cluster Size

The cluster size is defined as the number of associated hits in a cluster and ranges from 1 to 4+. The cluster size will depend on several factors such as the angle under which the particle hit the sensor, the threshold of the readout, the energy of the incident particle, position of the hit within the pixel, etc. For example, for a high threshold or particles hitting the center of the pixel, cluster size of one will tend to dominate.

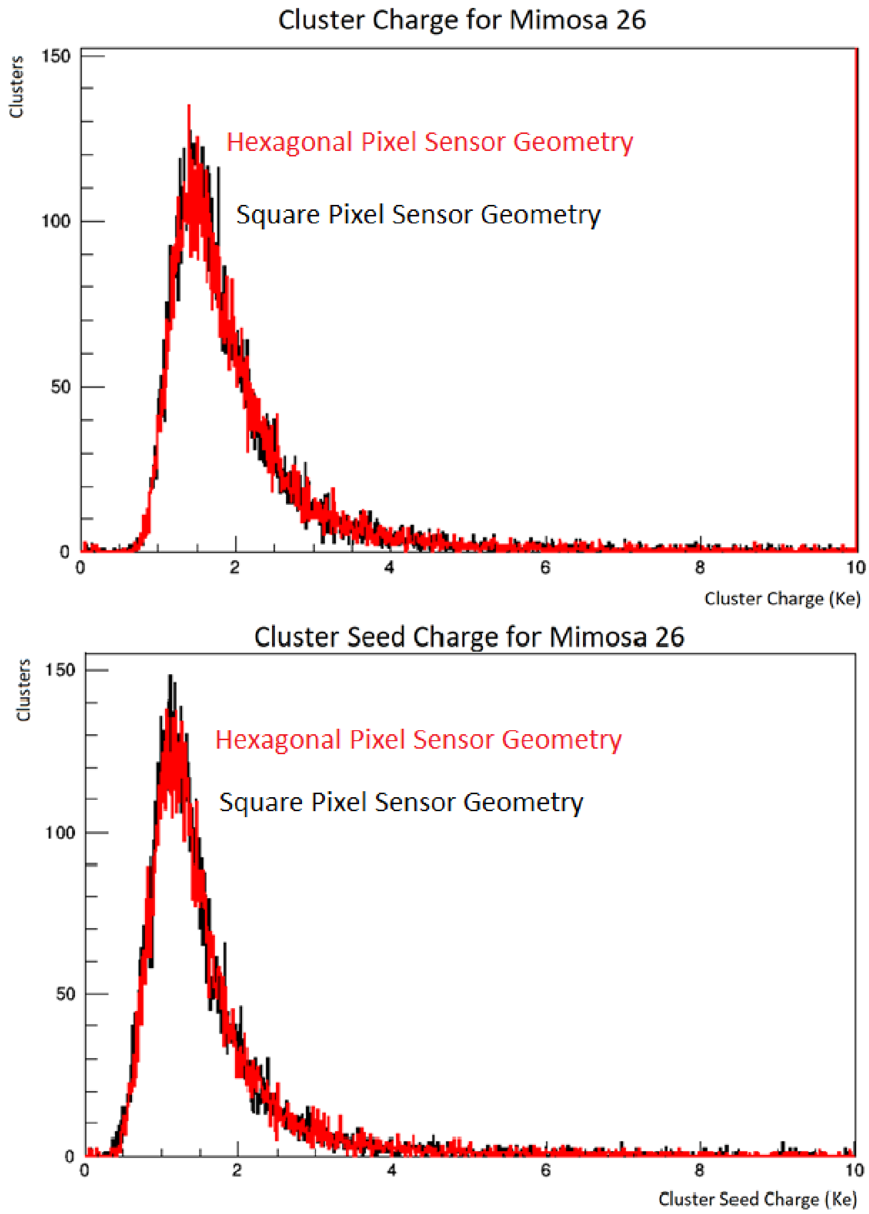

Figure 5: Top: cluster charge for the square pixel sensor and hexagonal pixel sensor. Bottom: cluster seed charge for the square pixel sensor and hexagonal pixel sensor.

When comparing the cluster sizes of square and hexagonal pixels, a clear difference is observable, as seen in 4, the difference is clear. Here, cluster size 4 dominates in a hexagonal pixel grid, while a cluster size of 3 and 4 is approximately the same in a square pixel grid. This is due to charge sharing with neighbors as seen in figure 1, where at the edge of a hexagonal pixel, the charge will be shared with three different pixels while for a square pixel, the charge will be shared with four different pixels due to geometry.

3.3 Cluster Charge and Cluster Seed Charge

Cluster charge is the sum of all single pixel charges inside a given cluster and follows a Landau distribution. On the other hand, the cluster seed charge is the highest single pixel charge within a given cluster.

Both cluster charge and cluster seed charge superimposed with that of square pixel sensor and hexagonal pixel sensor are shown in figure 5. The results are unexpected, since due to less charge sharing, the peak of the seed charge distribution for the hexagonal pixel sensor will be higher than the square pixel sensor. However, the results indicate that they are practically the same and will be subject to further study.

4 Outlook and Conclusion

Basic elements for simulating hexagonal pixels has been implemented within Allpix^2 and further development is currently ongoing. One such ongoing development is the functionality to import custom field maps from TCAD for hexagonal pixels; this requires to consider the field potential of adjacent pixels of an activated pixel. Hence, a new function will be created that identifies the indices of the neighouring pixel cells, given a pixel index.

In summary, the document reported about the potential benefits of a hexagonal pixel geometry for a semiconductor detector; these benefits include better timing resolution and detector efficiency. Then, the Allpix^2 framework was introduced and changes to the source code was made to allow for the simulation of hexagonal pixel sensors. Specifically, this involve creating the getPixelIndex() function that accepts the hit position as a parameter and returns the corresponding pixel index associated to the hit position. Finally, a basic simulation was performed using the MIMOSA-26 detector to compare square pixels and hexagonal pixels. Specifically, the simulation looked at the hit map, cluster size, cluster charge, and cluster seed charge. The simulation result showed reduced charge sharing for a sensor with a hexagon pixel grid.

References

[1] J Baudot, G Bertolone, A Brogna, G Claus, C Colledani, Y Degerli, R De Masi, A Dorokhov, G Doziere, W Dulinski, et al. First test results of MIMOSA-26, a fast cmos sensor with integrated zero suppression and digitized output. In *2009 IEEE Nuclear Science Symposium Conference Record (NSS/MIC)*, pages 1169–1173. IEEE, 2009.

[2] Thanushan Kugathasan, Taeko Ando, Dominik Dannheim, Takeharu Goji Etoh, Magdalena Munker, Heinz Pernegger, Angelo Rivetti, Kazuhiro Shimonomura, and Walter Snoeys. Monolithic cmos sensors for sub-nanosecond timing. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 979:164461, 2020.

[3] Lingxin Meng. Atlas itk pixel detector overview. *arXiv preprint arXiv:2105.10367*, 2021.

[4] S. Spannagel, K. Wolters, D. Hynds, N. Alipour Tehrani, M. Benoit, D. Dannheim, N. Gauvin, A. Nürnberg, P. Schütze, and M. Vicente. Allpix2: A modular simulation

framework for silicon detectors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 901:164–172, 2018.