

# An Efficient, FPGA-Based, Cluster Detection Algorithm Implementation for a Strip Detector Readout System in a Time Projection Chamber Polarimeter

Kyle J. Gregory<sup>a</sup>, Joanne E. Hill<sup>a</sup>, J. Kevin Black<sup>a,b</sup>, Wayne H. Baumgartner<sup>a,c</sup>, Keith Jahoda<sup>a</sup>

<sup>a</sup>NASA, Goddard Space Flight Center, 8800 Greenbelt Rd, Greenbelt, USA;

<sup>b</sup>Rock Creek Scientific, 1400 East-West Hwy, Silver Spring, MD, 20910, USA

<sup>c</sup>University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD, USA

## ABSTRACT

A fundamental challenge in a spaceborne application of a gas-based Time Projection Chamber (TPC) for observation of X-ray polarization is handling the large amount of data collected. The TPC polarimeter described uses the APV-25 Application Specific Integrated Circuit (ASIC) to readout a strip detector. Two dimensional photoelectron track images are created with a time projection technique and used to determine the polarization of the incident X-rays. The detector produces a 128x30 pixel image per photon interaction with each pixel registering 12 bits of collected charge. This creates challenging requirements for data storage and downlink bandwidth with only a modest incidence of photons and can have a significant impact on the overall mission cost. An approach is described for locating and isolating the photoelectron track within the detector image, yielding a much smaller data product, typically between 8x8 pixels and 20x20 pixels. This approach is implemented using a Microsemi RT-ProASIC3-3000 Field-Programmable Gate Array (FPGA), clocked at 20 MHz and utilizing 10.7k logic gates (14% of FPGA), 20 Block RAMs (17% of FPGA), and no external RAM. Results will be presented, demonstrating successful photoelectron track cluster detection with minimal impact to detector dead-time.

**Keywords:** Cluster Detection, FPGA, Strip Detector, Time Projection Chamber, photoelectron APV25, x-ray, polarization, polarimeter

## 1. INTRODUCTION

The gas-based TPC has been demonstrated as an effective method for measuring small fractional polarizations across the 2-10 keV band.<sup>1-4</sup> This technology now enables such measurements to be taken from a Small Explorer class mission.<sup>5</sup> This raises a new challenge of handling the large amount of data that a gas-based TPC can generate.

In order to effectively limit the data rate, a digital algorithm has been developed and implemented in an FPGA for locating and isolating the photo-electron track within the detector image. Since this is intended for a spaceborne mission in low earth orbit, a radiation tolerant FPGA is used, which has the advantages of being low-power and suitable for the radiation environment along with the disadvantage of having limited performance capability when compared to commercial FPGAs. Furthermore the FPGA is responsible for performing several other functions and handling four sets of detectors simultaneously. In addition, the TPC polarimeter needs to process hundreds of photo-electron tracks per second. Therefore, the image processing time has a direct impact on dead-time, or the fraction of time when the detector is unable to detect additional photo-electron interactions. For these reasons, an implementation was developed to minimize FPGA resource utilization and computation time.

---

Further author information:

K.Gregory.: E-mail:kyle.j.gregory@nasa.gov, Telephone: 1 301 286 4813

J.Hill: E-mail: joanne.e.hill@nasa.gov, Telephone: 1 301 286 0572

## 1.1 Time Projection Chamber Polarimeter

The Micropattern TPC<sup>1,3,6,7</sup> polarimeter, originally planned for the Gravitational and Extreme Magnetism Small Explorer (GEMS) mission,<sup>6</sup> operates in the 2-10 keV X-ray band, and consists of a gas proportional counter Filled with 190 Torr Di-Methyl Ether (DME). A cross section diagram of a GEMS polarimeter is shown in Figure 1. The photons arrive from the left, interact with the gas atoms in the detector depositing charge, which then drifts down to the gas electron multiplier (GEM) where the charge is multiplied, collected by the strips, and read by one of the four APV25 ASICs.<sup>8</sup> The sampled charge data from the ASIC is used to form an image of the track, also called a cluster, from 30 consecutive time samples collected on each of the 128 strips. Figure 3 illustrates how the charge read out sequentially in this fashion forms a two dimensional image of the initial photo-electron track. A resulting modulation curve such as in Figure 2 is formed based on post-processing of the track images.

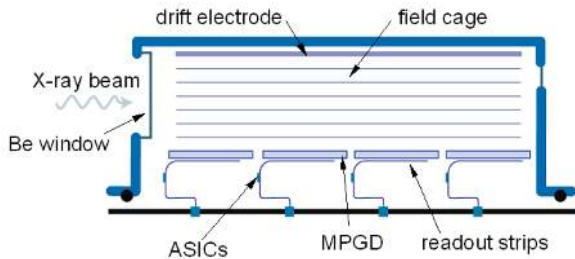


Figure 1. Diagram of the GEMS Micropattern Time Projection Chamber polarimeter. The MPGD is also known as the gas electron multiplier, or GEM.<sup>1,4,7</sup>

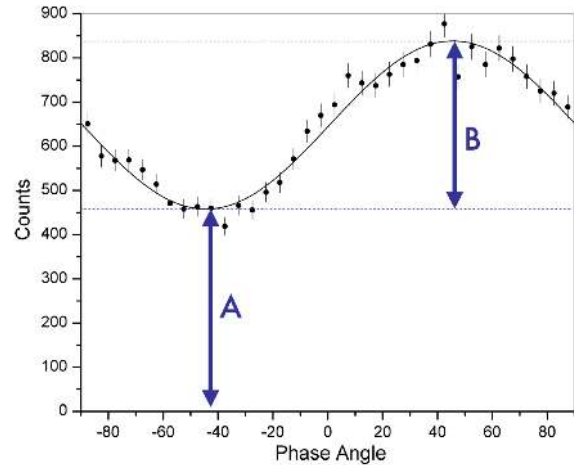


Figure 2. Modulation curve formed from a histogram of the photo-electron track directions measured in a single observation. The x-axis is the derived polarization, and the modulation factor is given by  $\mu = \frac{B}{2A+B}$ .<sup>1,4,7</sup>

The drift direction in the detector is perpendicular to the incident direction of the photons, which allows for very long photon path lengths in the detector and high quantum efficiency.<sup>7</sup> Multiplication of the charge deposited by the initial photo-electron is provided by a Gas Electron Multiplier, and the signal is read out by a set of 128 one dimensional strips coupled to the GEM with their long directions parallel to the initial photon direction. When regular time samples of the signals present on the strips are taken, a two-dimensional image can be made of the charge track deposited by the initial photo-electron. The initial direction of the photo-electron is closely correlated with the polarization of the incident photon, and analysis of many events can yield information on the polarization of the X-ray source. An algorithm is used to determine the initial direction of the photo-electron generated by an X-ray interaction.<sup>7,9</sup> For space-based observations, reconstruction would be done on the ground, which requires the photo-electron track images to be telemetered for each photoelectric event.

## 1.2 Photo-electron Track Images and Data Rates

The photo-electron track images are read out from a strip detector and sampled at 20 MHz by an APV25 ASIC. The raw image is always 128x30 pixels in size. Pixels are digitized with 12-bits on an offset binary encoding scheme, yielding a full-scale range of 0 to 4095 ADC units for each pixel. A few sample track images are shown in Figure 4, these were developed using a physics simulation.<sup>7</sup> The track itself can vary in size from as small as 8x8 pixels to 30x30 or larger, with larger tracks corresponding to higher energy photons.<sup>7</sup> A separate photon detection and triggering technique is used to trigger the detector when photon interaction has occurred. This

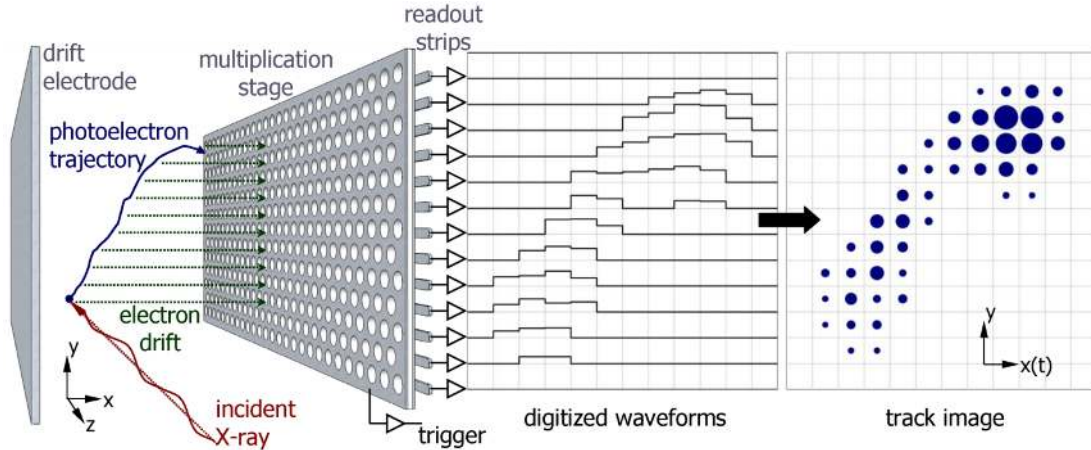


Figure 3. The time-projection chamber technique creates pixel images from a 1-dimensional readout. By using the strip number for one dimension and the arrival time multiplied by the drift velocity for the orthogonal dimension the photoelectron track can be reconstructed into an image (right). Drift distance (hence diffusion) is largely independent of active depth. Imaging of the photo-electron track allows one to estimate the initial direction and thus the polarization of the incident photon.<sup>1-4,7</sup>

technique ensures that a track is centered vertically within each image.<sup>7</sup> Outside of the track, the image data represents noise and a static pedestal, which can vary over time and from detector to detector.

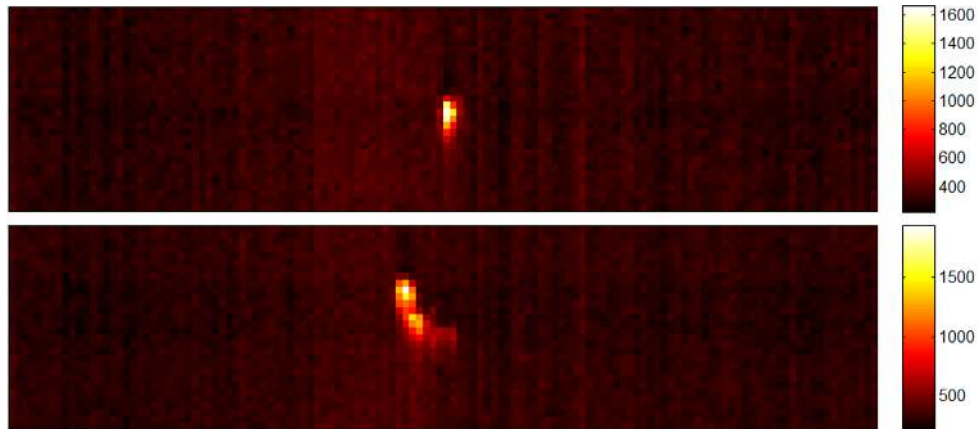


Figure 4. Simulated photo-electron track images are shown.<sup>7</sup> The top image is from a 3.5 keV interaction, bottom is 8.0 keV interaction. Images are digitized with 12-bits. Units are in ADU, 0 to 4095 full-scale.

Polarization is determined on the ground by analysis of each track image.<sup>1,3</sup> The pedestal must be known, but does not need to be included in every image. Periodic dark images will allow measurement of the pedestal and characterization of the noise. The data processing must also incorporate knowledge of the APV25 shaping time, which blurs the raw images in the time domain (vertical dimension). Ground processing must have sufficient knowledge to deconvolve this effect. The FPGA processing described in this paper does not remove or correct for this.

The count rate for the Crab nebula is expected to be  $200 \frac{ct}{s}$  per telescope.<sup>6</sup> With a digital resolution of 12 bits per pixel, taking full images yields as an average data rate for raw images of roughly 9.2 Mbps. A small explorer in Low Earth Orbit can expect to downlink 2Gbit/pass with an S-band contact. Operating two telescopes and ignoring housekeeping data and overhead at this rate would only allow for 100 seconds of observing time in between downlinks. Assuming these images can be reduced to an average size 18x18 pixels, there is a potential

for a 90% reduction in the data rate by discarding excess track image on-board the spacecraft. The reduced data rate would allow for close to 20 minutes of Crab observations in between S-Band contacts. While, this may not sound like a lot of observation time, the majority of the GEMS targets are much fainter than the Crab and will produce much less data.<sup>5,6</sup>

## 2. TRACK DETECTION AND ISOLATION

The basic principal for locating and isolating the photo-electron track is simple. The track will always be the brightest spot in the image. The goal is to simply draw a rectangular box around the track with a few pixels of buffer space on each side to ensure that the entire track is enclosed. Figure 5 illustrates the major steps in the cluster detection process. This section describes the design details that enable this to be done reliably, quickly, and with minimal use of computing resources.

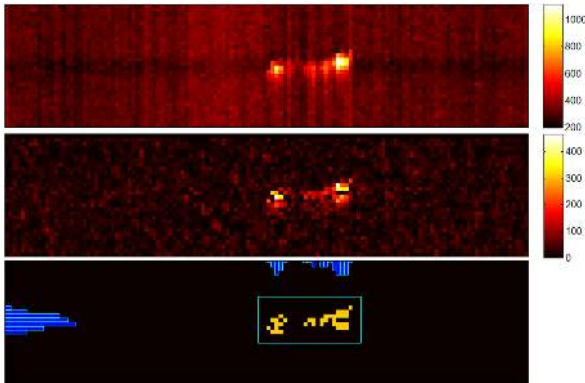


Figure 5. Three stages of processing one track image. The top image is the raw data from a simulated 8.0 keV interaction.<sup>7</sup> The middle image shows the result of the subtraction operation. The bottom image shows hit pixels superimposed with the row and column histograms and a border showing the resulting boundaries. Units are in ADU, 0 to 4095 full-scale.

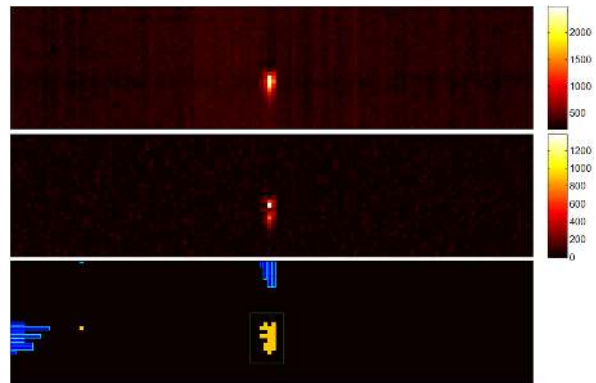


Figure 6. Three stages of processing one track image with a noise outlier pixel. The top image is the raw data from a simulated 4.5 keV interaction.<sup>7</sup> The middle image shows the result of the subtraction operation. The bottom image illustrates the identified hit pixels superimposed with the row and column histograms and a border showing the resulting boundaries. Units are in ADU, 0 to 4095 full-scale.

### 2.1 Hit Pixel Detection

The first challenge to be addressed is removing the DC offset, a.k.a. 'pedestal', from the image. The simplest approach to this would be to store a 12-bit pedestal value for each strip of the APV25 (i.e. each vertical column of the image), however, this would not be robust against short-term drift in the pedestal value. Instead, an approach was chosen where successive samples in the time dimension are subtracted and the absolute value is taken (an approximate time-series derivative). This approach has the additional benefit of eliminating the need to store and periodically update a set of pedestal values.

The subtraction operation produces an image that has no pedestal and emphasizes the top and bottom edges of the photon track. With the pedestal removed, the track can be found by simply setting a threshold just above the noise floor, yielding a binary 'hit' or 'no-hit' result for each pixel. With this information, the coordinates of the a rectangular boundary can be identified with a few cells of padding being added around the outer-most hits. This process is illustrated in Figure 6

### 2.2 Dealing with Noise

One additional affect to be considered is that of noise outliers. An example of a noise outlier can be seen in Figure 6. A single noise outlier can potentially add hundreds of unneeded bytes into the spacecraft downlink by

expanding a track boundary well beyond what is needed. These can be eliminated by exploiting the fact that they will almost always be a single lonely pixel and often far away from the actual track. A maximum gap size parameter is set to define the distance that a noise outlier can be from the main track before it is ignored. This is done by computing the median coordinate of the pixel hits and then working outward until the edges of the track are found and the maximum gap size is reached.

### 2.3 Additional Meta-data

After throwing out all of this excess data through cluster detection, there is some additional meta-data that must be included with the reduced image. First, we would like to have information about where the track was within the full scale image. Therefore, we need to include coordinates for the boundaries. This will require a total of 24-bits (5 bits for each horizontal border and 7 bits for each vertical border). In addition, we need to know the common mode values across all of the strips in order to do the common-mode correction in the post-processing.<sup>1</sup> For this, sum up all of the pixels in each row and encode the result into a 16-bit value for each of the 30 rows. Altogether, this amounts to 500 additional bits of meta-data needed, which is negligible on the scale of the data we are able to discard as a result of this operation.

## 3. HARDWARE IMPLEMENTATION

The hardware architecture for the cluster detection makes use of the Block RAM internal to the FPGA for short-term data storage as well as logic-flip-flops for storing process variables and functional parameters. The entire process is managed by a state machine. Figure 7 provides an overview of how the hardware is organized. This design is organized in such a way so as to maximize the number of computations that can be computed in parallel so as to minimize the computational time. Figure 8 shows an activity diagram of the processing that is done to help illustrate the computations that are performed in parallel. It should be noted that this function includes several optional diagnostic modes and error conditions that are not described for the sake of simplicity.

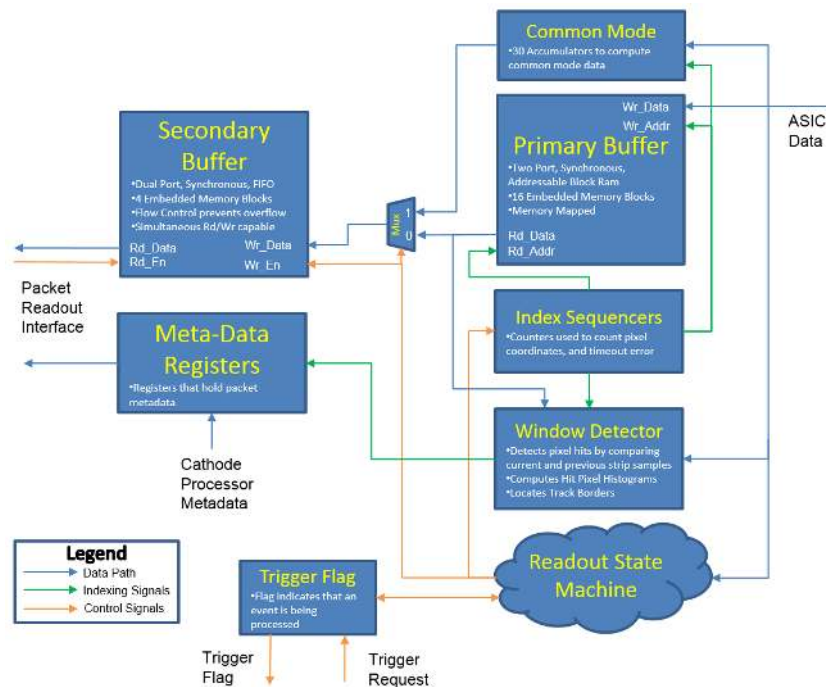


Figure 7. A block diagram illustrating a high-level overview of the computing hardware organization for cluster detection.

While this article focuses on the cluster detection, there are several other important functions that are managed by the TPC readout FPGA. First and foremost, the FPGA is responsible for detecting the photon

interactions, measuring their energy & length (in time), and rejecting background interactions using a process called cathode processing.<sup>1,7</sup> The FPGA also controls a pulsed x-ray source for calibration.<sup>4</sup> In addition, the FPGA accepts commands that can set performance parameters both for these algorithms and for the APV25 ASICs. It should also be noted that, as discussed earlier in Section 1.1 and Figure 1, one FPGA is responsible for reading out all four detectors.<sup>1,4</sup> The four readouts run concurrently and mostly independate except that they share an outgoing data pipe, which can get backed up in high-event-rate scenarios, leading to additional dead-time.

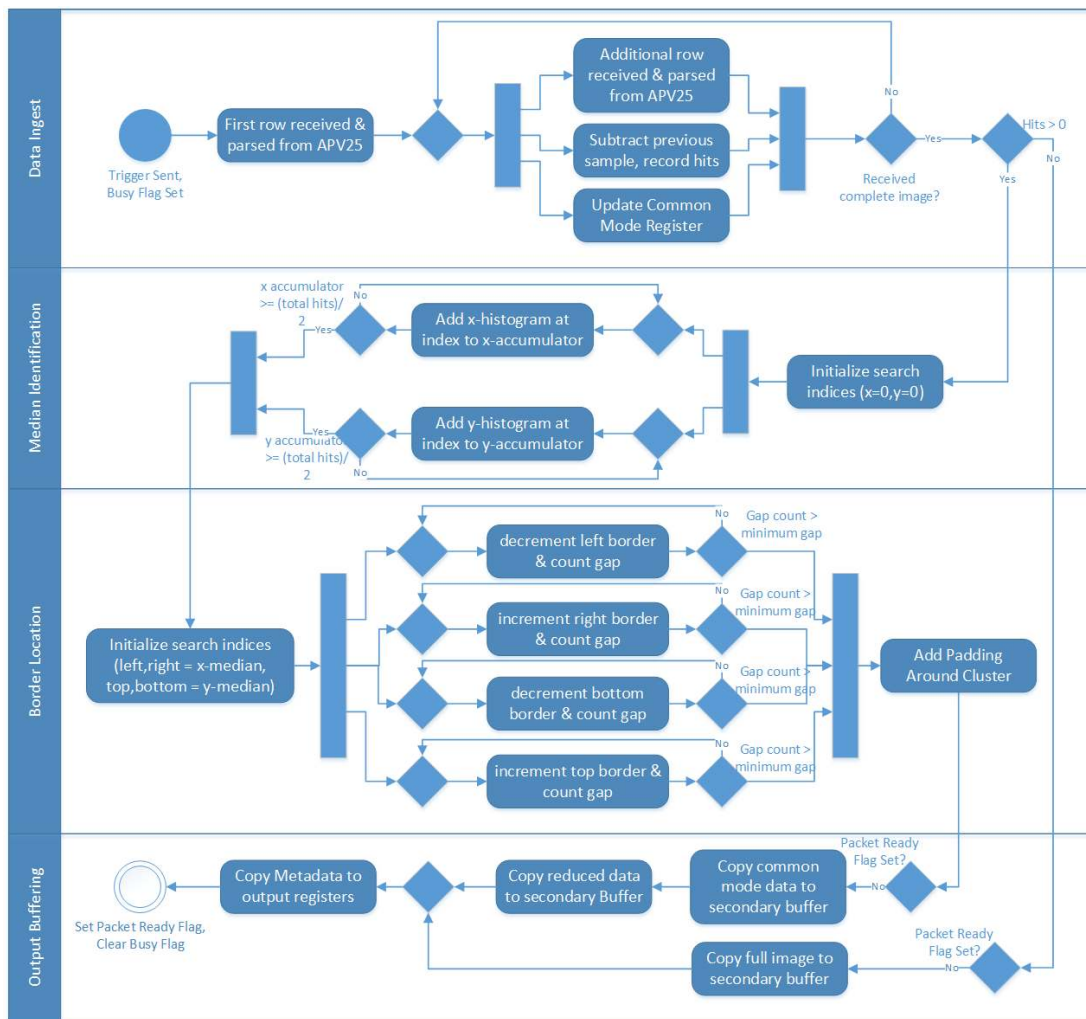


Figure 8. An activity diagram illustrating the hardware process for cluster detection & isolation.

### 3.1 Image Acquisition

The detection of a photo-electron event is performed by the cathode processor and a trigger request signal is received, indicating that the APV25 ASIC has been triggered for readout. This sets the track detection processing in motion. The readout system indicates back to the cathode processor that it is busy by setting a trigger flag signal while it collects and processes the image.

A two-stage organization of the block RAM is used. The two sections are referred to as 'Primary' and 'Secondary' buffers. The primary buffer allows for the storage of one entire image while the photo-electron track can be identified. The secondary buffer provides local storage for the reduced image data until it can be transmitted out through the shared serial communication channel.

Several registers, counters, accumulators, multiplexers and other logic devices are employed in parsing the data and performing the various computations that are described. For the sake of presenting a high-level overview, these devices have been grouped together under five categories: Common Mode, Indexing Sequencers, Window Detector, Meta-Data Registers, and Trigger Flags. Not shown are an array of registers that store operating parameters, such as the gap parameter, border padding, and noise threshold. These registers enable tuning of the cluster detection process at any point during operation.

The state machine first searches for a header pattern before each row of the image is received.<sup>8</sup> The pixel data samples are received serially. As each new sample arrives, the corresponding sample from the previous row is read back from the Primary Buffer so that the subtraction can be performed. The result of the subtraction is immediately compared with the programmable threshold value to determine if the pixel is a 'hit'. A positive hit result is used to increment three counters: one corresponding to the row, one corresponding to the column, and one for the overall image. Therefore, no storage is required to hold the subtracted image. Once the entire raw image has been read into the primary buffer, the state machine can proceed into median and border finding.

### 3.2 Border Identification

To locate the row and column position of the median track image pixel, the state machine starts by initializing a pair of index pointers to column zero and row one (because a subtraction can not be performed on row zero). The values of each of the histograms at these locations are added into a pair of accumulators, corresponding to rows and to columns. The state machine repeats this process, incrementing the each pointer until the associated accumulator has exceeded half of the total number of hits. This position is used as the median cell. The state machine can then move on to border detection.

Border detection is performed by starting at the median point and walking outwards in all four directions until there are no more hits. A minimum gap parameter is used, specifying the minimum number of rows/columns that must be free of hits before the border is identified. Once the gap limit is reached, the last hit row/column is identified as the border. A number of additional rows and columns are added on each side, which is given by a padding parameter. This was done so that the minimum gap and padding do not need to have the same value.

After the border has been identified, the relevant data can be transferred into the secondary buffer and the busy flag is cleared. This is done to free up the processing resources to process another photon interaction. At this stage, some meta-data is also gathered from the cathode processing algorithm so that all of the data associated with the event is kept together.

## 4. CONCLUSIONS AND NEXT STEPS

In evaluating the performance of this implementation, a simulated set of photo-electron tracks were used.<sup>7</sup> The data set included 500 tracks from each of the following energy levels: 2.0 keV, 2.5 keV, 3.5 keV, 4.5 keV, 6.4 keV, 8.0 keV, and 10.0 keV. These data were fed into an RTL simulation of our logic design and the results were compared with expected border solutions. In addition, performance of the design was analyzed in terms of FPGA resource utilization and detector dead-time.

The results of these analysis draw very positive conclusions about the viability and usefulness of this technique in a spaceborne application of the TPC polarimeter.

### 4.1 FPGA Resource Utilization

The FPGA resource utilization for this design is shown in Table 9. Since this design is replicated four times in order to readout the four ASICs as described in Section 1.1, the majority of the FPGA is dedicated towards performing cluster detection and isolation. This leaves just enough resources available for the other functions that need to be performed, namely cathode processing, calibration, and housekeeping. The entire FPGA design for the readout system typically places and routes with a maximum clock speed estimate of 26-28 MHz. Since the design will be clocked at 20 MHz, this provides a healthy timing margin.

It should be noted that the APV25 ASIC requires an I2C interface for setting its operational parameters. The I2C controller is not included with the cluster detection function and therefore does not show up in this resource report, but it would be required for operating an APV25 ASIC.

The conclusion to be drawn based upon this resource utilization is that this capability comes at very little cost to the design of the TPC polarimeter as a whole. Without cluster detection, the readout electronics need all of the same hardware outside of the FPGA and the FPGA would be needed to acquire the strip image data and perform the other functions regardless of whether cluster detection is to be performed. The resources utilized for this technique are minimal enough to not justify the need for an additional FPGA. Therefore, the two primary costs of utilizing this technique are the increase in power consumed by the FPGA and the new complexity in the system, both of which are minor.

Resource	Used	Available	Usage
Core Cells	10954	75264	15%
Block RAM	20	112	17%

Figure 9. FPGA resource utilization for RT-ProASIC3-3000 FPGA as reported by synthesis.

## 4.2 Detector Dead-Time

An estimate of detector dead time is given in Table 10. This estimate has been produced by a combination of laboratory measurements and analysis. This estimate shows very good margins compared with our requirement of 900  $\mu s$  dead-time for GEMS.

Source	Cause	Typical Dead Time
APV25 ASIC	Trigger Response	8 $\mu s$
	Readout Time	210 $\mu s$
Cluster Detection	Median Finding	$3.25 \pm 2.5 \mu s$
	Border Finding	$0.75 \pm 0.25 \mu s$
	Transfer results	$17.7 \pm 10 \mu s$
	Miscellaneous Delays	0.3 $\mu s$
<b>Total</b>		<b>240 <math>\mu s</math></b>
<b>Cluster Detection contribution to deadtime (<math>\mu s</math>)</b>		<b>22 <math>\mu s</math></b>
<b>Cluster Detection contribution to deadtime (%)</b>		<b>9.20%</b>

Figure 10. Detector dead-time contributions by source.

This estimate takes advantage of the fact that the drift process in the detector and the cathode processing can both take place in parallel while the cluster detection is performed. Those processes are expected to take on the order of 10  $\mu s$ , which is much smaller than the time needed for cluster detection. Therefore, those processes are not important with respect to detector dead-time.

The largest contributor to dead-time is the time needed to receive the image data from the APV25. While the cluster detection method can do nothing to reduce that time, it does make use of the time to compute the difference image as the data arrives. This leaves only the median finding, border finding, and data packetization as contributors to deadtime. The time needed for these operations is a function of the location and size of the cluster due to the sequential nature of the searching functions, therefore a typical range has been given.

## ACKNOWLEDGMENTS

I would like to acknowledge the entire GEMS team, in particular Joe Hill, Kevin Black, Wayne Baumgartner, Keith Jahoda, Richard Szymkiewicz, Ed James, Jerry Cote, Norm Dobson, Robert Baker, Craig Markwardt, Ken Wagner, and many others for their support throughout GEMS X-ray polarimeter development. Without this team, this work would not have been possible.



## REFERENCES

1. J. K. Black, P. Deines-Jones, J. E. Hill, T. Iwahashi, K. Jahoda, P. Kaaret, T. R. Kallman, C. J. Martoff, Z. Prieskorn, J. Swank, and T. Tamagawa, "The gems photoelectric x-ray polarimeters," in *Space Telescopes and Instrumentation: Ultraviolet to Gamma Ray, Proc. of SPIE* **7732**, 2010.
2. J. E. Hill, J. K. Black, T. J. Emmett, T. Enoto, K. M. Jahoda, P. Kaaret, D. S. Nolan, and T. Tamagawa
3. J. K. Black, R. G. Baker, P. Deines-Jones, J. E. Hill, and K. Jahoda, "X-ray polarimetry with a micropattern tpc," in *Nuclear Instruments and Methods in Physics Research A*, **581**, pp. 755–760, 2007.
4. J. Hill, R. Baker, J. Black, M. Browne, W.H.Baumgartner, E. Caldwell, J. Cantwell, A. Davies, A. Desai, P. Dickens, N. Dobson, R. Foxwell, A. Francomacaro, D. Gall, K. Gregory, S. G. and A. Hayato, R. Hampshire, T. Hwang, M. Jhabvala, K. Jahoda, P. Kaaret, S. Lehtonen, N. Martin, J. Mohammed, K. M. de Garcia, A. Morell, D. Nolan, R. Russel, M. Sampson, J. Sanders, K. Simms, M. Singer, J. Swank, T. Tamagawa, A. Weaver, S. Yerushalmi, and J. Xu, "The design and qualification of the gems x-ray polarimeters," in *Space Telescopes and Instrumentation 2012: Ultraviolet to Gamma Ray, Proc. of SPIE* **8443**, 2012.
5. K. Jahoda, J. Black, J. Hill, T. Kallman, P. Kaaret, C. Markwardt, T. Okajima, R. Petre, Y. Soong, T. Strohmayer, T. Tamagawa, and Y. Tawara, "X-ray polarization capabilities of a small explorer mission," in *Space Telescopes and Instrumentation 2014: Ultraviolet to Gamma Ray, Proc. of SPIE* **9144**, 2014.
6. K. Jahoda, "The gravity and extreme magnetism small explorer," in *Space Telescopes and Instrumentation 2010: Ultraviolet to Gamma Ray, Proc. of SPIE* **7732**, 2010.
7. W. Baumgartner, T. Strohmayer, T. Kallman, J. K. Black, J. E. Hill, and J. Swank, "Gems x-ray polarimeter performance simulations," in *Space Telescopes and Instrumentation 2012: Ultraviolet to Gamma Ray, Proc. of SPIE* **8443**, 2012.
8. M. French, L.L.Jones, Q. Morrissey, A. Neviani, R. Turchetta, J. Fulcher, G. Hall, E. Noah, M. Raymond, G. Cervelli, P. Moreira, and G. Marseguerra, "Design and results from the apv25, a deep sub-micron cmos front-end chip for the cms tracker," in *Nuclear Instruments and Methods in Physics Research A*, **466**, pp. 359–365, 2001.
9. L. Pacciani, E. Costa, G. D. Persio, M. Feroci, P. Soffitta, L. Baldini, R. Bellazzini, A. Brez, N. Lumb, and G. Spandre, "The sensitivity of a photoelectric x-ray polarimeter for astronomy: The impact of gs mixture and pressure," in *Polarimetry in Astronomy*, S. Fineschi, ed., *Proc. of SPIE* **4843**, 2003.