ABSTRACT

Proton Computed Tomography and Constructing Tracker Boards

By

Gatlin Bredeson

Scientists and engineers at the Santa Cruz Institute for Particle Physics have been commissioned by Loma Linda Medical University to build tracker boards to assist in creating a system that seeks to replace X-ray computed tomography (xCT) with proton computed tomography (pCT). During a CT scan, protons pass through a phantom and can be deflected by Multiple Coulomb Scattering, thus decreasing the image resolution of the scanner. During development, tracker boards are needed to trace and reconstruct the paths of incident protons to better understand and engineer the pCT equipment to its optimal efficiency. This paper documents the methods used to troubleshoot and configure tracker boards that will be used in the pCT testing process. Pulsing the channels of the electronics with a charge confirms the proper functionality of the data input channels. Employing multiple pulse values with many threshold limits allow us to construct gain and response curves that can be used to determine a charge or energy threshold. This threshold is calibrated to be well below the most probable particle energy, but well above electronic noise magnitudes. Finally, using one-dimensional event count histograms (with a channel per bin), we can construct a two-dimensional position profile tracking where particles were incident on the boards. I found that the coding for the independent trigger may be responsible for anomalies during these tests, and that the coincidence trigger is reliable.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First I would like to thank Hartmut Sadrozinski for being so willing to give people like me an opportunity to learn something and be productive in his lab. I'm also thankful for his patience in times when I had trouble understanding something, and fervor for progress. The spirit made it exciting get to work and produce results.

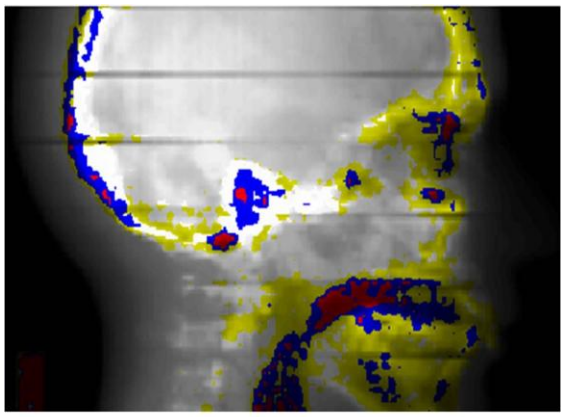
I'd like to thank Daniele Fusi for being an effective, understanding teacher and colleague. He taught me the troubleshooting methods used in this paper, and no doubt had to repeat himself a few times. I'm appreciative of his patience and willingness to work with me.

Finally, I'd like to thank Ford Hurley for being such a fount of knowledge. He usually helped me through any hardship I faced when needing to learn some aspect of the process, and always did so in a very friendly fashion.

All of these people made it a very comfortable and encouraging learning environment throughout the research process!

1. Introduction

Computed tomography (also known as a CT scan) is a method of scanning and imaging the human body that has become widely used in modern medicine. Using this method, medical technicians take pictures of "slices" of a patient's body in order to identify possible abnormalities.



Alderson Head Phantom

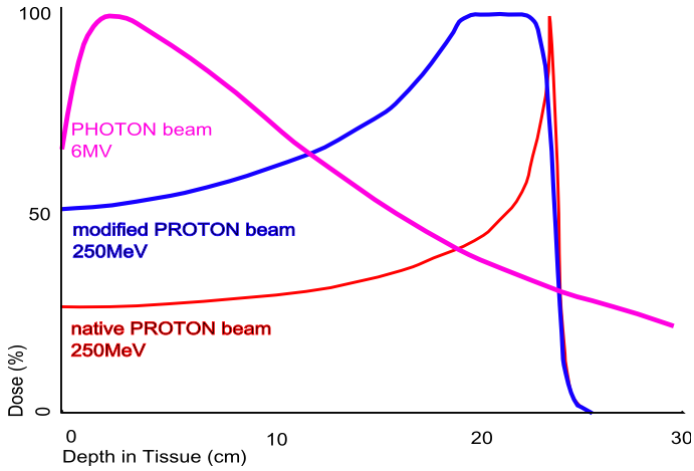
Traditionally, X-ray photons of a known energy are fired through the patient's body to be intercepted on the other side, where their final energy is measured [1]. This energy loss can be integrated with the stopping power *S* of the material to find the density through which the photons traveled by the following equation:

$$\int_{Ein}^{Eout} \frac{dE}{S(I_{mat}, E)} = \int_{L} \rho_e(r)dl$$

At this point, an image can be constructed based on the densities calculated [2]. Another application for this system is the treatment of tumors. When a particle passes through solid matter, energy is deposited along this path. When the particle finally loses enough energy to be stopped by the material, it will deposit the rest of its kinetic energy, causing a "spike" in the energy deposited versus depth traveled graph. The peak of this spike is known as the Bragg Peak [3].

Currently, medical technicians manipulate an X-Ray beam so that the Bragg Peak coincides with the tumor within the patient's body, delivering a high dose of harmful energy to the tumor. Unfortunately the Bragg curve for X-rays is rather smooth, which results in energy being

unnecessarily deposited to healthy tissue around the tumor as well. But there is an alternative! This thesis project is part of a collaborative effort to implement technology using proton beams, instead of X-rays, in the technical medical field.



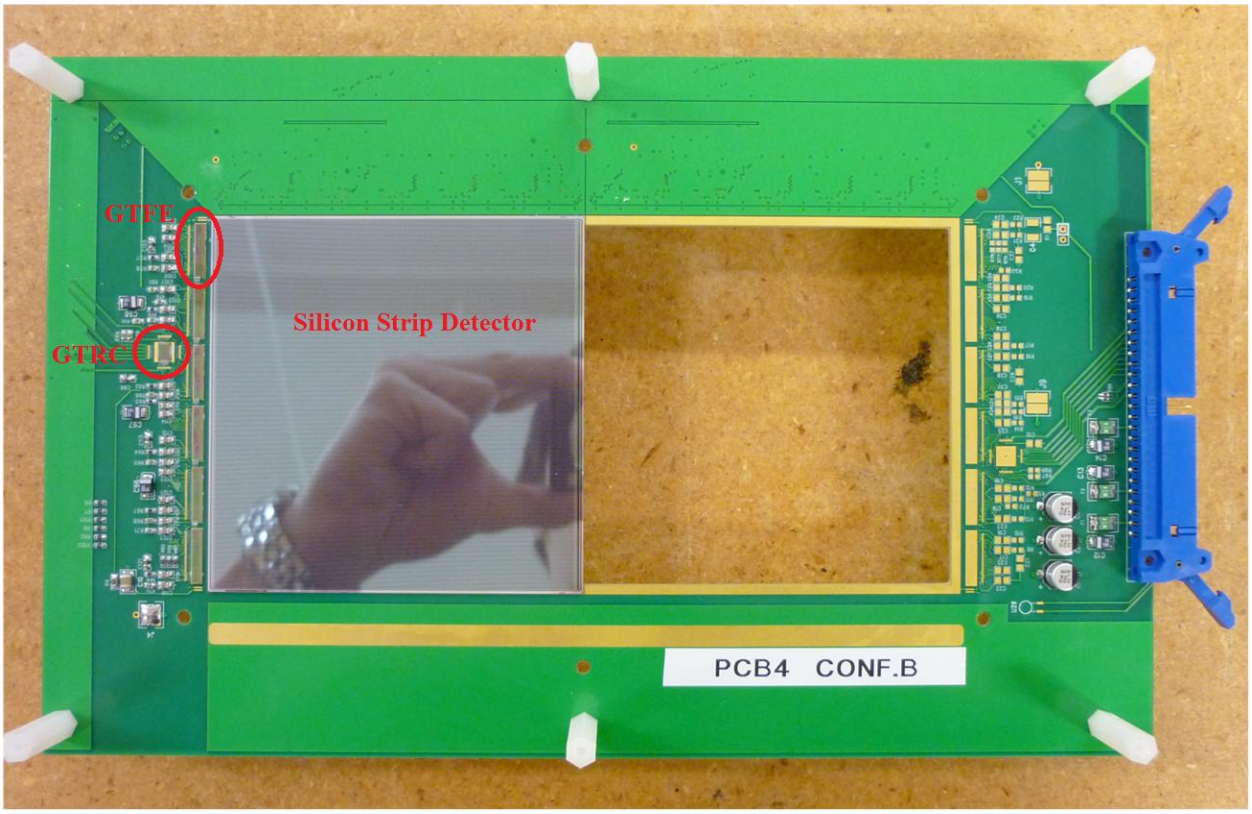
This will result in higher-resolution CT imaging, allowing us to see muscle and organ tissue in addition to bone. Using protons will also yield more precise radiation treatment for internal abnormalities. The Bragg Peak for a proton is much sharper than that of the X-ray, which will ultimately deliver much less unwanted radiation to healthy tissue within the patient's body.

As with any scientific research and technology development, there are challenges. When protons fired from the proton beams get too close to other charged particles, they will be deflected. This phenomenon is called Multiple Coulomb Scattering. The deflection of protons will result in a more blurred image, so steps must be taken to account for the protons' deviation. These steps consist of building ten tracker boards, each with two layers of silicon strip detectors (to track X and Y axes), and building a calorimeter to intercept protons and measure their energies. The boards will then be aligned between the proton beam and the calorimeter. When the beam is fired, information received from the tracker boards should allow us to reconstruct and analyze the proton's path [2].

An artificial charge (pulse) of a known value can be inserted into the tracker board, and the output of the electronics can be recorded. Analyzing the output of this process allows us to compute the gain and response of the electronics, which can then be used to compute the input charge during a beam test. This gain and response should be relatively constant across the

electronics, so deviations allow us to identify corrupted board components. We can also use a radioactive source to test the board's X and Y particle tracking capabilities. These tests are the most important because they determine whether or not the board will yield accurate or reasonable tracking results when a beam test is conducted. This thesis will describe the process and results of testing and troubleshooting the tracker boards unto completion. Section 2 will catalogue the experimental apparatus, Section 3 will describe the procedure for experimentation, Section 4 will display results, Section 5 is an analysis and discussion of those results, and Section 6 is the conclusion. Section 7 is a list of references.

2. Apparatus



Each tracker board consisted of three main parts: a total of two silicon strip detectors (SSDs), two GLAST tracker read-out controllers (GTRCs), and 12 GLAST tracker front-end electronics chips (GTFEs).

Each SSD is 10 x 10 cm in size, and contains 384 individual strips. The detector is capable of detecting incoming sub-atomic particles in the form of a charge. This charge is turned into a digital signal through a charge amplifier; each charge that is registered is called a “hit” or an “event”.

Six GTFEs are assigned to each SSD. As there are six GTFEs per SSD, and each SSD has 384 strips, each individual GTFE therefore manages the data of 64 strips. The GTFE acts as a data collector and organizer when data is received from the SSD.

There is one GTRC to govern the six GTFEs of each layer. The GTRC acts as a mediator between the GTFEs and the FPGA (which will be described later). For example, The GTFEs will only save data in the buffer or send the data if the GTRC commands it to do so.

There are two layers per tracker board. Each layer consists of one detector, its six GTFEs, and one GTRC. There are two layers because the detectors are oriented orthogonally to one another, so that one has its strips running in the X direction (known as layer 1), and the other has its strips running in the Y direction (layer 0).

A single field-programmable gate array (FPGA) is connected to the tracker board for each test. The FPGA acts as the overseer of the GTRCs, and is the "authority" for giving commands. The GTRCs only communicate with the GTFEs when the FPGA commands them to do so. Ultimately, the FPGA is responsible for the order and timing of data formation and transmission, and is controlled by a user through a set of code. Within the code are multiple tools essential for testing and troubleshooting the tracker boards. Among these include pulsing, threshold selection, layer selection, and trigger mode selection.

It is possible to use the FPGA code with voltage sources to "pulse" the tracker boards with an "artificial" charge. What this allows us to do is simulate a charged particle passing through the SSD. The magnitude of these inserted charges can be manipulated through the code itself, so the values are known. A voltage source provides a voltage for a DAC (Digital-to-Analog-Converter) to manipulate. DAC values can then be manipulated in the FPGA code to put forth a controlled voltage. These values each correspond to certain voltages. For the pulse, the DAC inserted voltage passes through a capacitor in the GTFE that converts to a charge according to $Q = CV$. The capacitance of the capacitor within each GTFE is 46 fF.

The threshold is a voltage value manipulated through the FPGA code in the form of DAC values. When a particle hits the SSD, it passes into the electronics in the form of a charge. Essentially, this charge is converted to a voltage via a charge sensitive preamplifier, and compared to the threshold voltage (or just the "threshold"). If the voltage exceeds the threshold, it is considered a hit. The purpose of the threshold is to eliminate unwanted noise from the data stream.

The layer select allows us to choose which layers of the tracker board we currently wish to work with. Being able to select specific parts of the tracker board to read data from makes troubleshooting easier in that it is possible to pursue problems to specific areas of the electronics.

The trigger selection mode only has significance when doing measurements with the radioactive source. The first of the two modes is the coincidence mode. When this mode is activated, a particle will be required to be detected on both layers (using AND logic) to enter the data stream and be registered as an event. When the independent mode is activated, a particle need only register on a single strip of either layer (using OR logic) to be registered as an event.

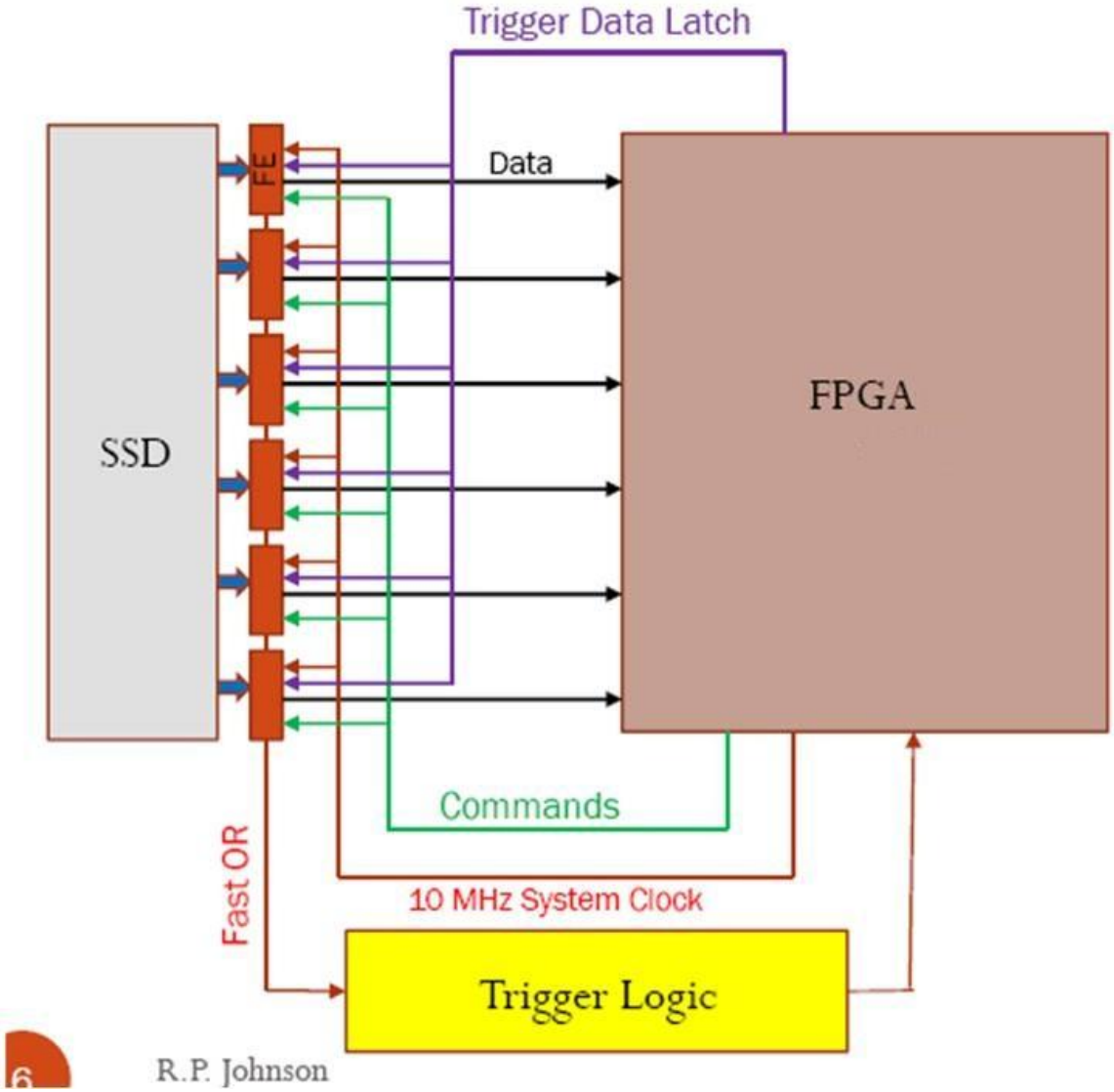


Figure 1: Detailed schematic of command and data flow between electronics. [4]

3. Procedure

I. Configuration

Each board is docked in a protective case. The case has room for the tracker board connector to be exposed, so that it can be connected to the FPGA while protected. Setting up the tests was simple. A set of cables put together into a single band connects into the tracker board. This band is connected to the FPGA as well as a two voltage sources. Then, another voltage source is plugged directly into the FPGA, which is connected to a PC by an Ethernet cable.

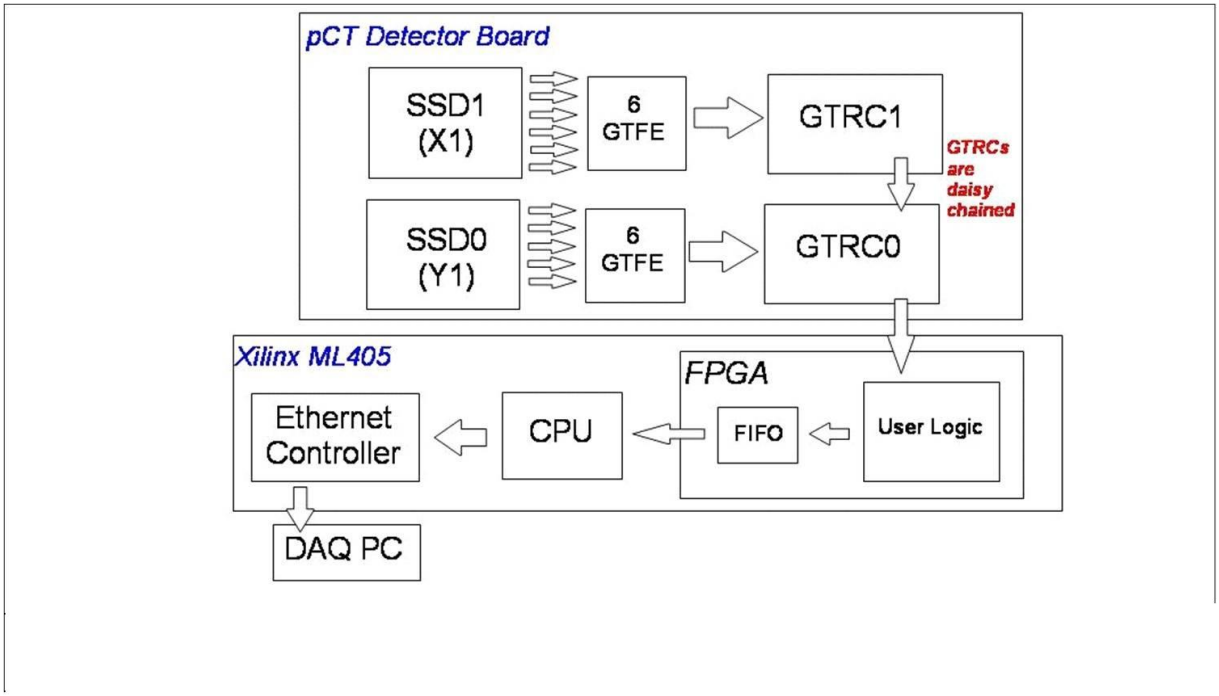


Figure 2: Simplified schematics of the data flow in the present tracker system. [5]

II. Testing Procedure

The first test performed on the tracker boards was the strip check. We needed to know that each strip is functioning properly before moving on to other tests that assume a well-functioning set of GTFEs. Therefore, the purpose of this test was to check the integrity of the GTFE channels. If each channel is pulsed with the same charge value, and the threshold is

constant for all strips, the resulting response values for each strip should be the same. So the procedure was simple: run the FPGA code on one layer at a time on calibration mode (which enables pulsing and varying thresholds) with constant pulse and threshold values for each channel individually. Specific DAC values in the FPGA code correspond to the different voltage values for the pulse and threshold. For this test, we pulsed the strips with a DAC value of 63, which corresponds to a voltage of 97.96 mV across a 46 fF capacitor in the GTFE. Using $Q = CV$, this converts to a charge of 4.51 fC. For the strip check, a threshold DAC value of 40 is set, corresponding to a voltage of 184.5 mV, which discriminates charge above about 2 fC.

The next test is the calibration. When the detector receives a charge, we need to know what output to expect from the electronics. In other words, we need to know what the gain and response will be from the electronics when injecting a known charge. When testing with the actual proton beam, we will know (from this calibration test) the gain and response behavior so that the initial charge (of the protons) can be confirmed. To get a sizeable pool of data, we used multiple pulse and threshold values for each of the 384 strips. Much like the FPGA code settings for the strip check, this test is run in calibration mode. We want to know the gain and response for each layer individually, so we test one layer at a time. This means using the layer select feature of the FPGA code to select the appropriate layer. Then we use pulse DAC values starting at 5, corresponding to 9.22 mV, up to a final value of 63, or 97.96 mV. The pulse values are incremented in steps of 1 unit, or 1.53 mV. Using $Q = CV$ formula, we can know the value of the charge being pulsed into the tracker board. This converts to a starting charge of 0.42 fC, increasing by steps of 0.07 fC up to a final charge value of 4.51 fC. This pulse range is used for each of a number of threshold values for each channel. These DAC threshold values started at 22 or 103.5 mV, increased by steps of 1 or 4.5 mV, and ended at a final value of 40 or 184.5 mV.

This means the charge threshold started at about 1.2 fC, and went up to about 2 fC. So ultimately, for each threshold we have a range of pulse values to give us a curve by which to determine the gain and response of the electronics. This will be discussed more in the analysis section.

The final and most significant test dealt with taking measurements using a radioactive source. This test is the most important because it most closely simulates what the tracker boards will be doing in the proton beam tests. The physical configuration of the equipment is exactly the same as with the other tests in terms of how it is all connected, but there are a few additions. The silicon strip detectors of each board are extremely sensitive to light, so it was absolutely necessary to cover the board in a dark shroud. This ultimately prevents the current in the board from running high. Ideally we only want charge from particles we inject ourselves, not the charge from background or ambient light particles. Once covered, we place a Strontium-90 radioactive source over the detector to simulate a controlled "beam" of particles. The FPGA code will also have slight alterations during this test. We change the mode from calibration to measurement, which will disable pulsing, and allow particles hitting the detectors to trigger either the independent mode, or the coincidence mode for data collection. While there is no pulsing (incoming particles deposit the charge, no artificial charge is needed), the threshold DAC value is set to a constant 22, or 103.5 mV. This is an ideal value for discriminating out noise from the electronics. We want event data from particles, not noise. There will inevitably be background particles adding unwanted events to our data, so it was also useful to run this test without the Strontium-90 source as a bit of a "control". This would give us an idea of how many background particles to expect in the rest of our data. Each source measurement and background

measurement test was set to collect for 15 minutes. This value is long enough to gather an ideal amount of events without saturating the data with background hits.

	Test Mode	Layer(s) Selected	DAC Pulse Value (Voltages)	Pulse Charge Value	DAC Pulse Step size (Charge)
Strip Check	Calibration	One at a time	63 (97.96 mV)	4.51 fC	None
Calibration	Calibration	One at a time	5 – 63 (9.22 – 97.96 mV)	0.42 – 4.51 fC	1 (0.07 fC)
Source Measurements	Measurement	Both	None	None	None

Table 1: Pulse Settings in the FPGA Code

	Test Mode	DAC Threshold Value (Voltages)	Threshold Value (Charge)	DAC Threshold Step Size
Strip Check	Calibration	40 (184.5 mV)	≈ 2 fC	None
Calibration	Calibration	22 – 40 (103.5 – 184.5 mV)	≈ 1.2 – 2 fC	1
Source Measurements	Measurement	22 (103.5 mV)	≈ 1.2 fC	None

Table 2: Threshold Settings in the FGPA Code

For each test, coded scripts were used to construct graphs and images that allowed us to view the data in a meaningful way. These scripts were written in C++ using a set of libraries

called ROOT. ROOT was developed by employees at CERN for the purpose of data analysis; it is a useful way to use code to output detailed graphs, histograms, and many other data analysis tools. Former employees of SCIPP wrote the ROOT scripts (in addition to the FPGA code) which would be run on the data output files created by the FPGA code. The graphs and histograms generated by these scripts will be used to analyze the data further in the Results and Analysis sections.

4. Results

I. Strip Check

The following graphs show the two kinds of typical data for the strip check. They come from the strip check run on board 8, and show the outcome of the test for both layers (0 profile, and 1 profile).

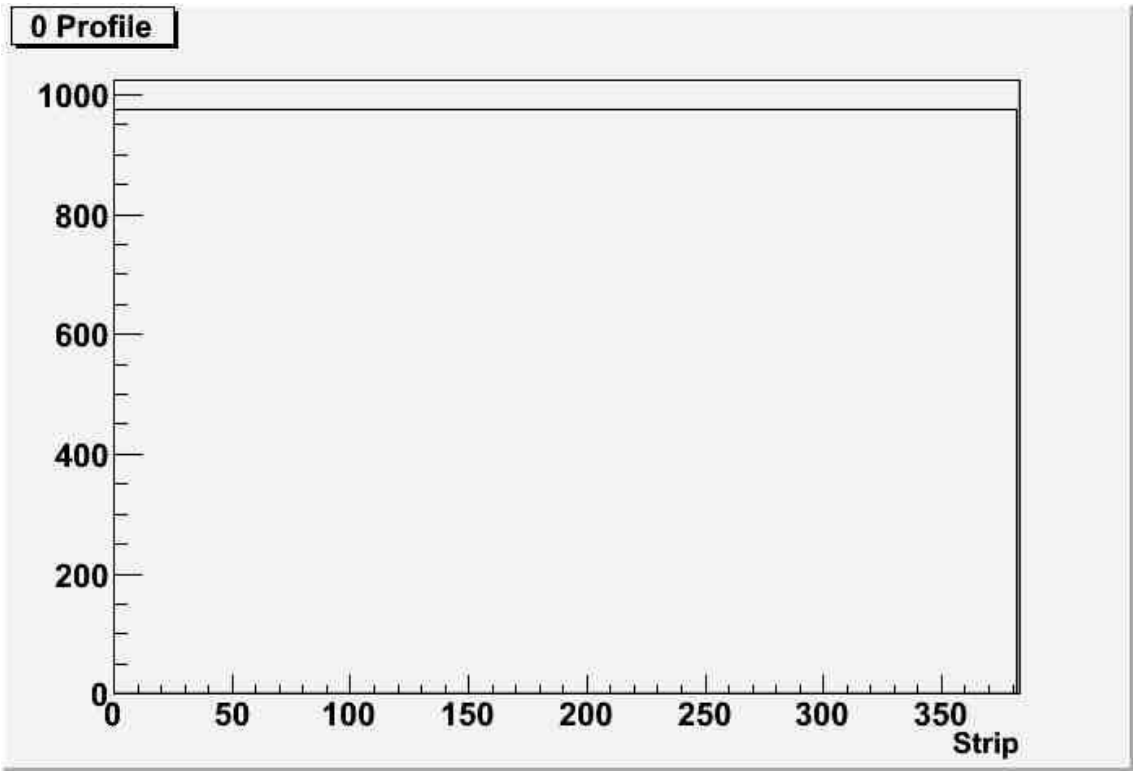


Figure 3: Expected Outcome for Strip Check

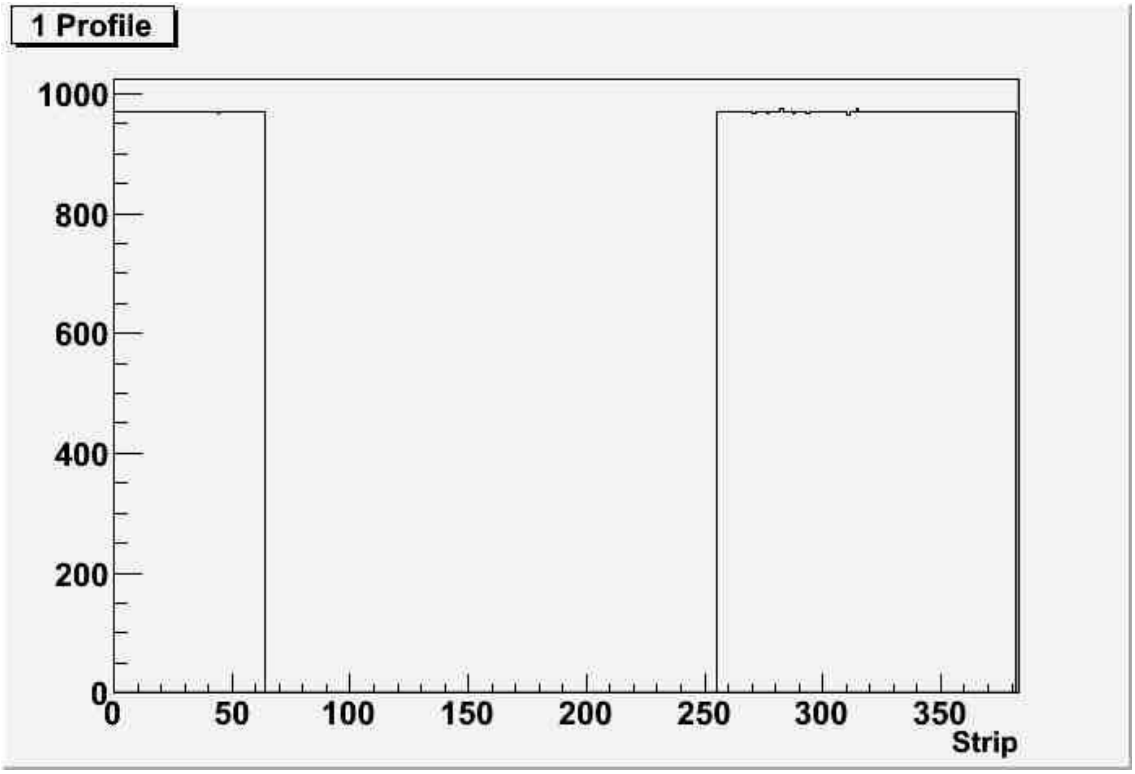


Figure 4: Unexpected Outcome for Strip Check.

Figure 3 displays the strip check results on layer 0 of board 8. This graph is an example of a layer with all of its GTFEs working properly. Figure 4 is a graph showing the results of the strip check on layer 1 of board 8. The GTFEs on this layer obviously have problems. This will be discussed in more detail in the analysis section. These histograms show why the strip check is so important to the rest of the procedure; a look at the plot shows where problems may lie, and prevents us from continuing construction with faulty equipment.

II. Calibration

Figure 5 displays the calibration results for board 4, layer 0. The data for this layer are typical of a board whose electronics are working properly. Figure 6 displays the calibration data for typical layer with a few problems. These data comes from the calibration data for board 10, layer 1.

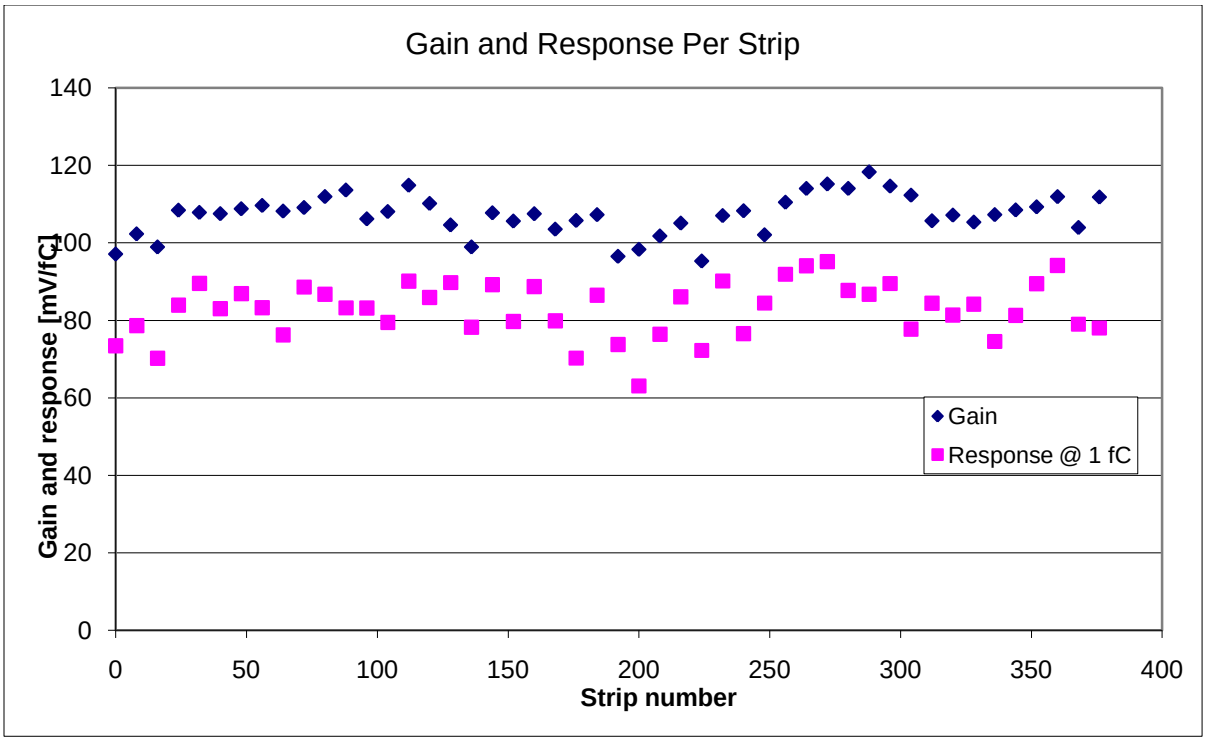


Figure 5: Calibration Data for a Properly Functioning Layer. Board 4, Layer 0

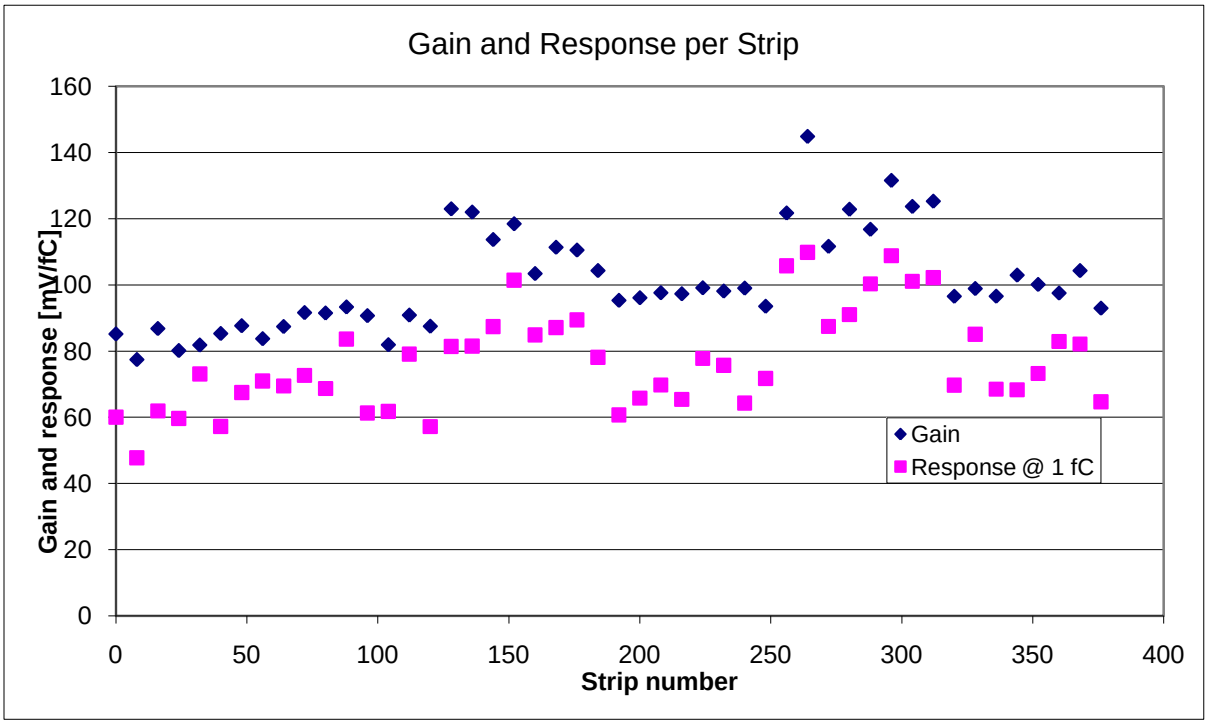


Figure 6: Calibration data for a layer with a few problem areas. Board 10, Layer 1

These graphs only display the gain and response on every 8 channels. The gain values are in units of mV/fC, while the response values are in units of mV. Though in different units, the two are still comparable on the same graph because it shows the value of the response at 1 fC. Dividing this through again to put the gain and response into the same units doesn't change the value. A response of 5 mV at 1 fC translates into 5 mV/fC.

III. Source Measurements

Figures 7 through 10 display the typical results in each trigger mode (independent and coincidence) for functioning and malfunctioning boards. Included on these x-y position plots are the respective one-dimensional profiles for each layer of the board. These profiles show the number of events registered per strip on each individual layer (x and y).

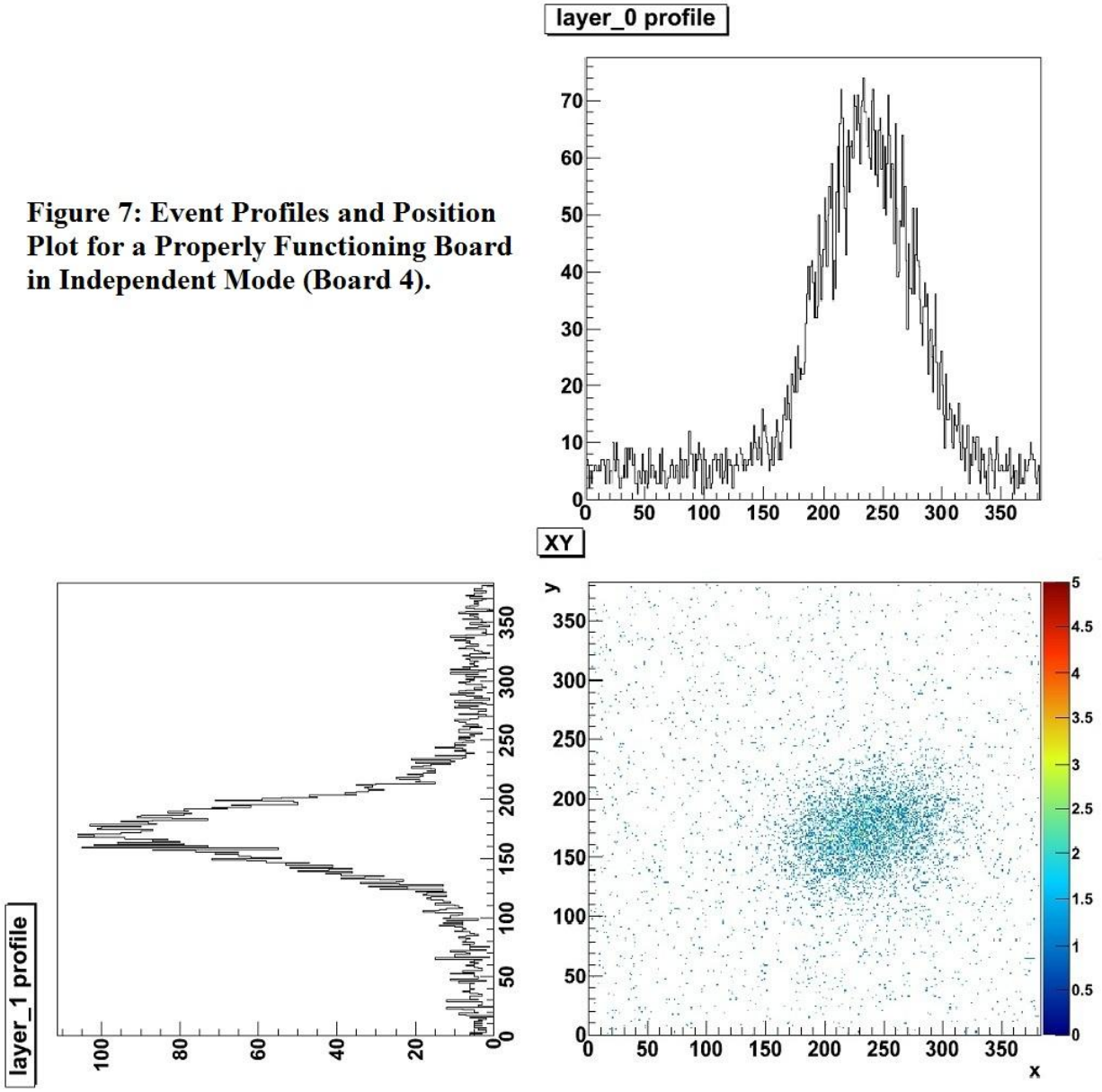


Figure 7: Event Profiles and Position Plot for a Properly Functioning Board in Independent Mode (Board 4).

Figure 8: Event Profiles and Position Plot for a Properly Functioning Board in Coincidence Mode (Board 9).

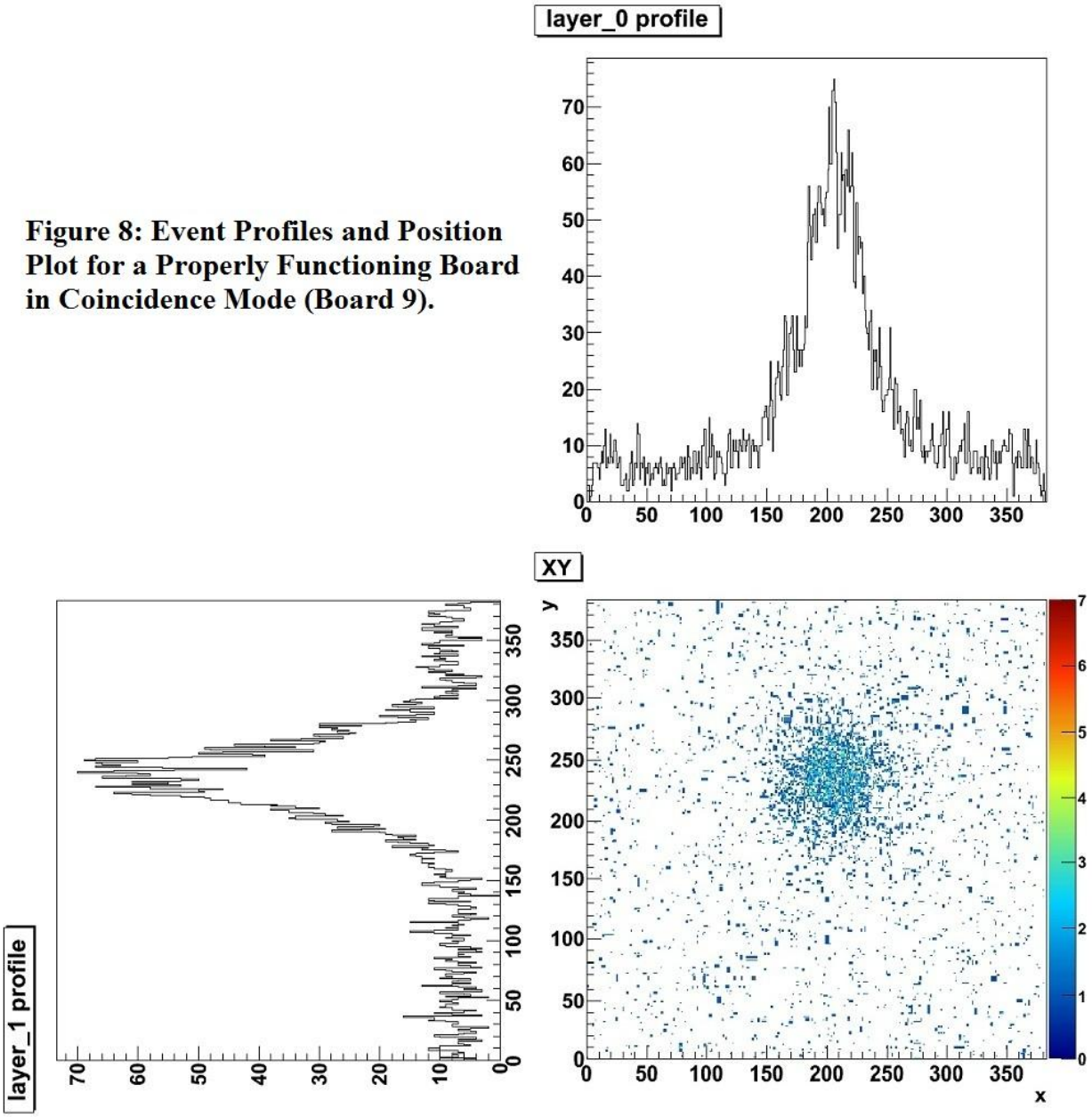


Figure 9: Event Profiles and Position Plot for an Improperly Functioning Board in Independent Mode (Board 3).

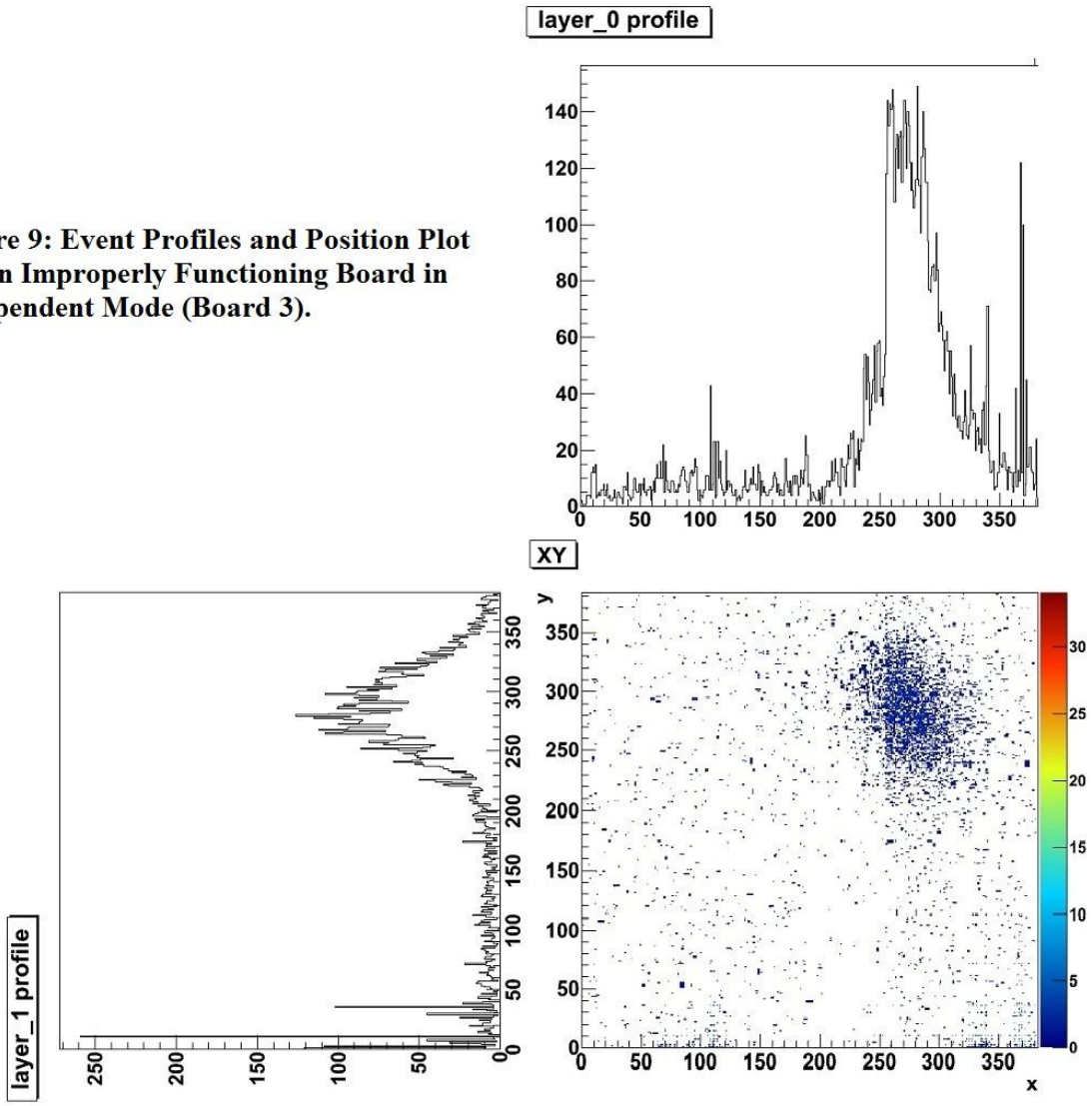
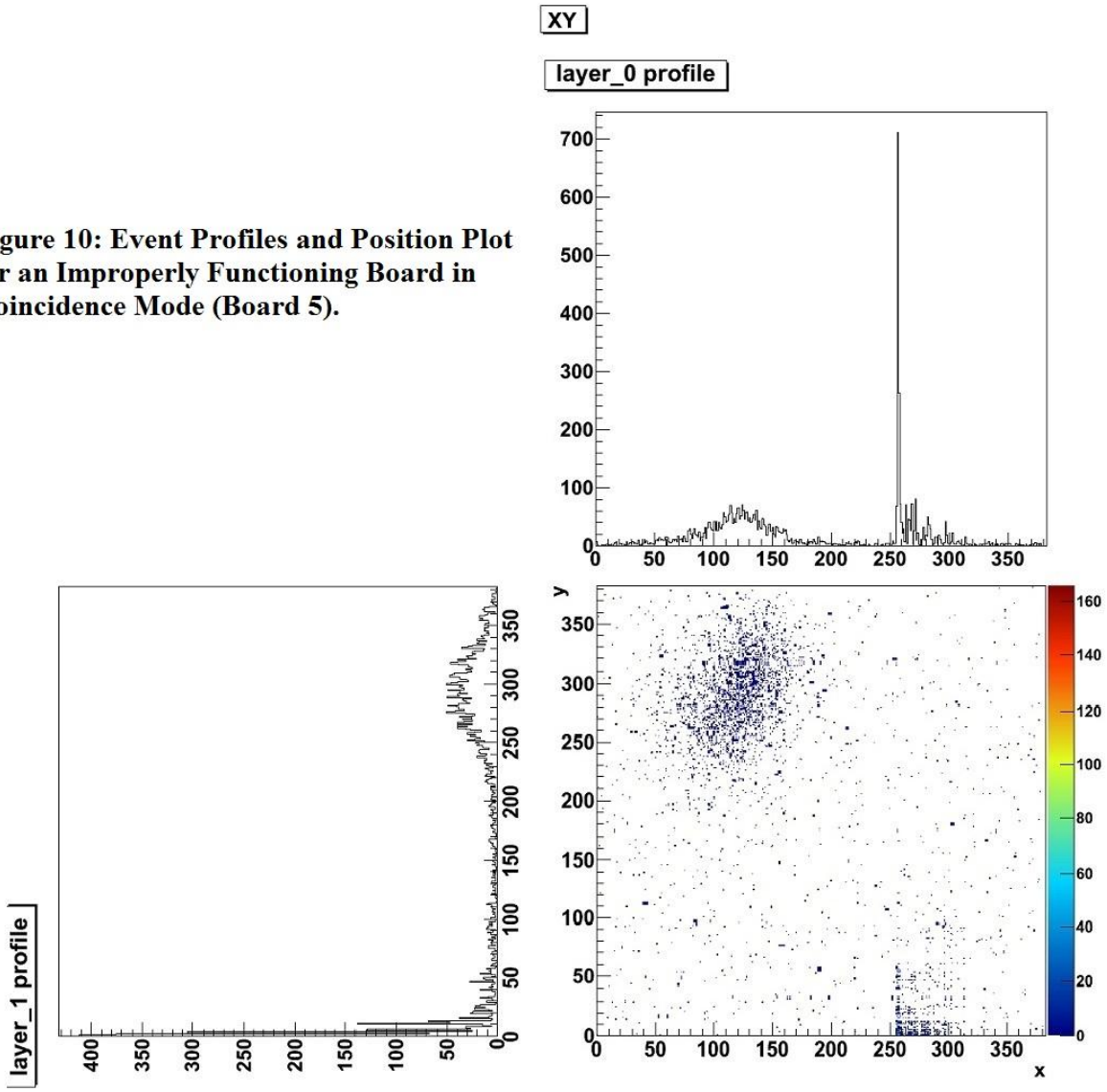


Figure 10: Event Profiles and Position Plot for an Improperly Functioning Board in Coincidence Mode (Board 5).



5. Analysis

I. Strip Check

When doing the strip check, each channel is pulsed 1000 times with a charge we know exceeds the threshold. Thus, each graph should show a constant readout of 1000 events on each channel. However, looking at Fig. 3, we can see that this value falls short of 1000. I suspect this discrepancy occurs if two events trigger so close together that the discriminator counts them as a single event. If this were to happen in the system, we would expect to see a few less events than we expected. On the other hand, the number of events missing is extremely consistent; it seems as though each channel is missing the same number of events. It is important to consider the fact that the number of events read out is consistent across all functioning channels. So while I am not sure of the reason for missing events, we can see that each working channel is still working in a consistent fashion.

Looking at Fig. 4, we see a big gap in event data for channels 64 through 256. As each GTFE governs 64 strips, it would be reasonable to conclude that GTFEs 2, 3 and 4 are not functioning properly. Another explanation would be that the GTRC is not communicating properly with those GTFEs. To test this theory, we could examine the GTRC and make sure the wire bonds connecting it to the GTFEs are attached properly. Beyond that, we could use a pico probe on the wire bond of the GTRC sending a signal to those GTFEs to determine if that signal is even being sent. In this case, the proper functioning of the other GTFEs is a good indicator that the problem lies with the GTFEs, and not the GTRC. If the GTRC is working properly, it will be sending commands to the GTFEs to put forth the data stored in their buffers. The functioning GTFEs are sending back data like we would expect; so the problem with the bad GTFEs could be that they are not receiving data, not buffering data, or just not sending the data back. In any

case, the GTFE must be replaced by one that works. When the new one has been set and bonded onto the board, the whole strip check process must be repeated until each of the GTFEs are determined to be functioning properly, as in Fig. 3.

II. Calibration

The important factors to consider in the calibration process come from the gain and the response. Examining Fig. 5 and 6, we can see that each GTFE has its own values for gain and response. Ideally, each GTFE will have gain and response values close to the same level. However, in cases like Fig. 6, we can see there are a few hyper-sensitive GTFEs. Once calculated, the charge value of the threshold (which we can see from the gain and response graphs) can be used to compare to the expected charge distribution. Figure 11 shows a gain curve created by the calibration data analysis script.

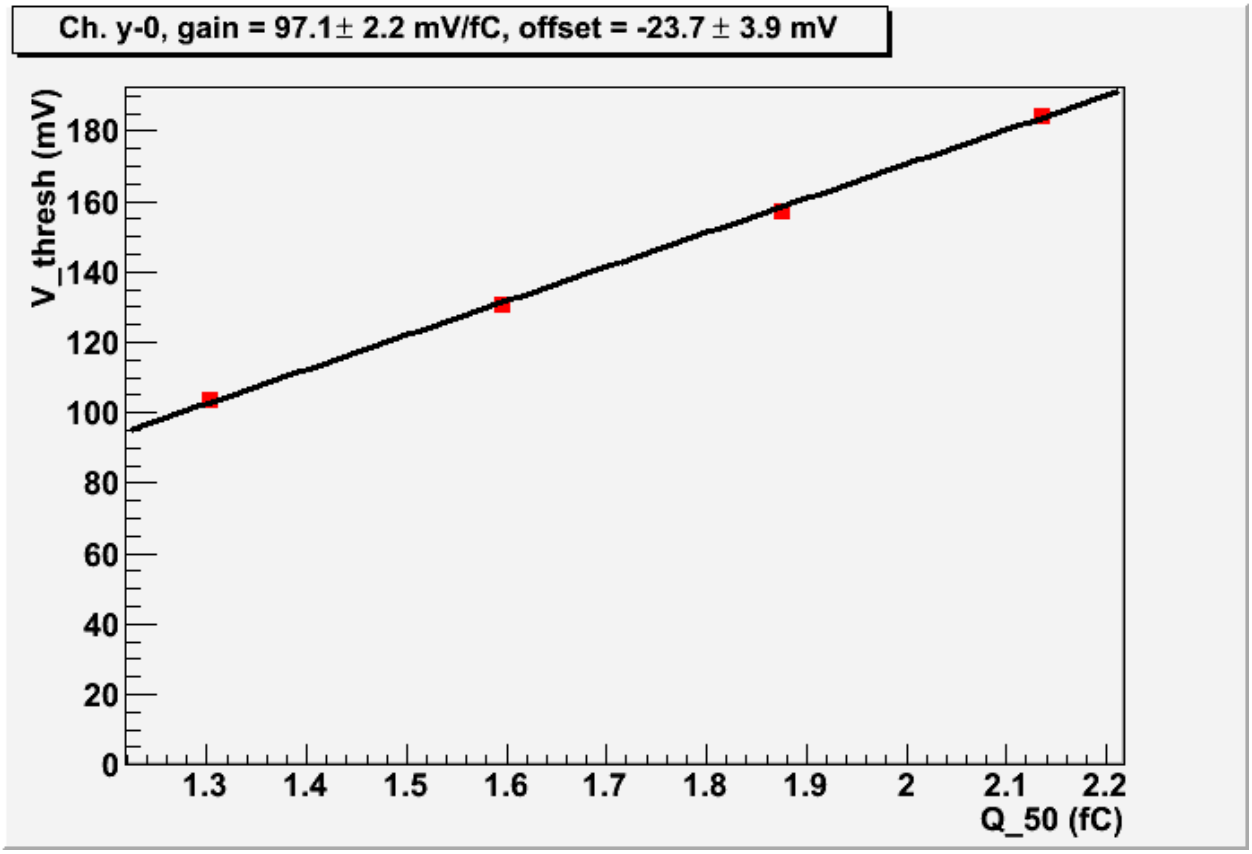


Figure 11: $V_{threshold}$ vs. Charge for Channel 0 on Board 4, Layer 0

This curve shows the charge value associated with the different threshold values on the channel. Each red point is a threshold value. The first point is at V_{thresh} = 103.5 mV. This threshold is especially important because it is the threshold set for source measurements. The best-fit line on the data describes the gain, which is the derivative of the electronics' response, and follows the form:

$$V_{thresh} = QG + C_{off}$$

In this equation, Q is the charge, G is the gain, and C_{off} is a constant offset value. To find the charge, the equation can be rearranged and solved using the known values for the threshold

(which we set), the gain and offset (which are determined from the best-fit equation). For Fig. 1 (channel 0), the charge is 1.31 fC.

$$Q = \frac{V_{thresh} - C_{off}}{G}$$

Using the graphs created for each channel analyzed (every 8th), we calculated the charge threshold for each channel. We then organized all of these charges into a frequency histogram with 0.05 fC bin size to show their distribution. This will show us the most probable values for the charge threshold.

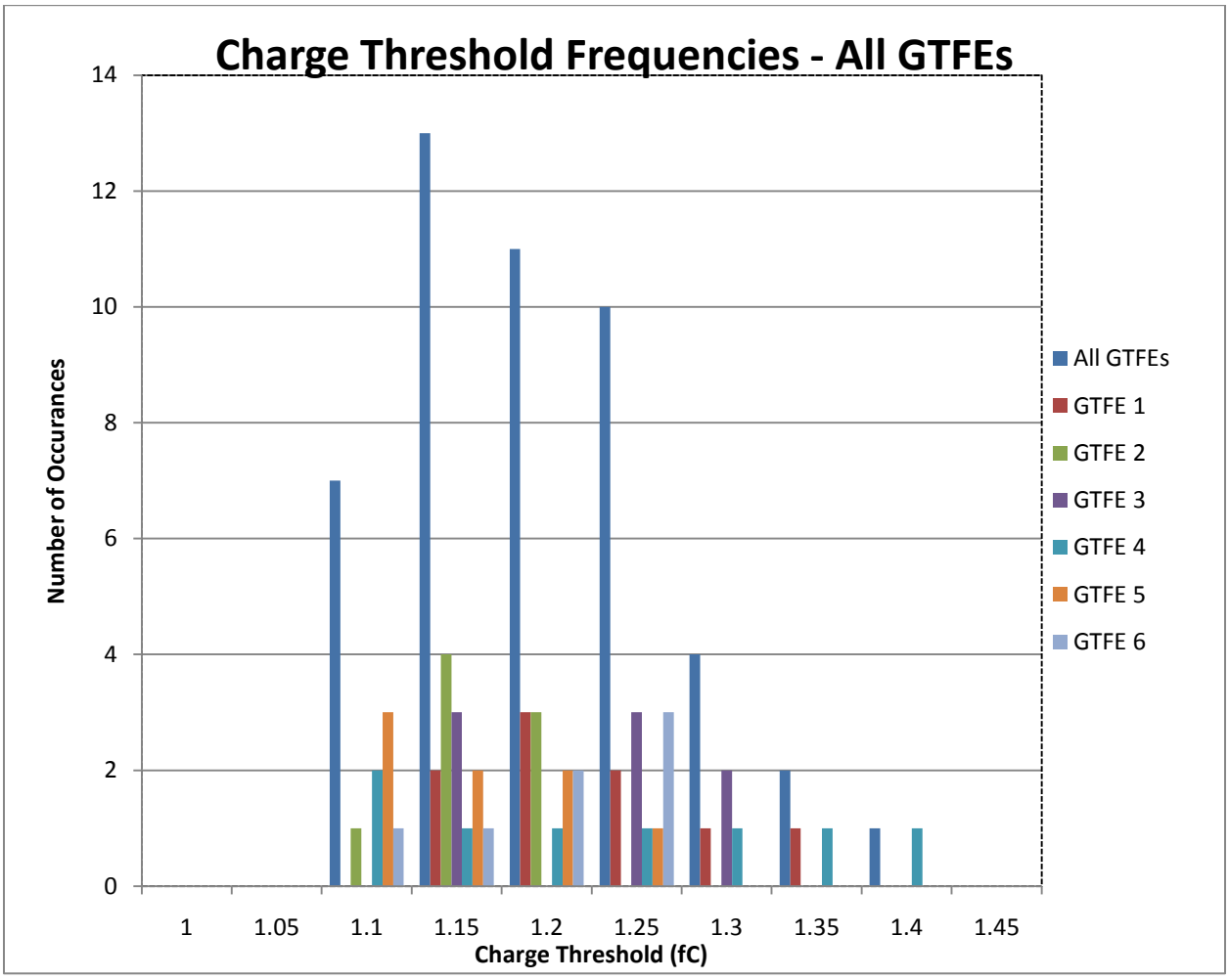


Figure 12: Charge Frequency Distribution for All Channels of Board 4, Layer 0

Figure 12 shows the distributions of charge for the channels on each individual GTFE, as well as across all GTFEs. The average, or most probable value (MPV), for this distribution is 1.2 fC.

For a 400 micron SSD with 250 MeV protons, the expected energy loss is 0.295 MeV [6]. Dividing this by the energy per electron-hole pair (3.6×10^{-6} MeV), we can get the number of electron hole pairs for this energy loss. Then we can multiply by the charge of an electron to get the energy loss in terms of charge. This gives us a value of 13.3 fC. This is the expected theoretical value we expect to see with the 250 MeV protons. Figure 13 displays the theoretical charge distribution on the same graph with the charge threshold.

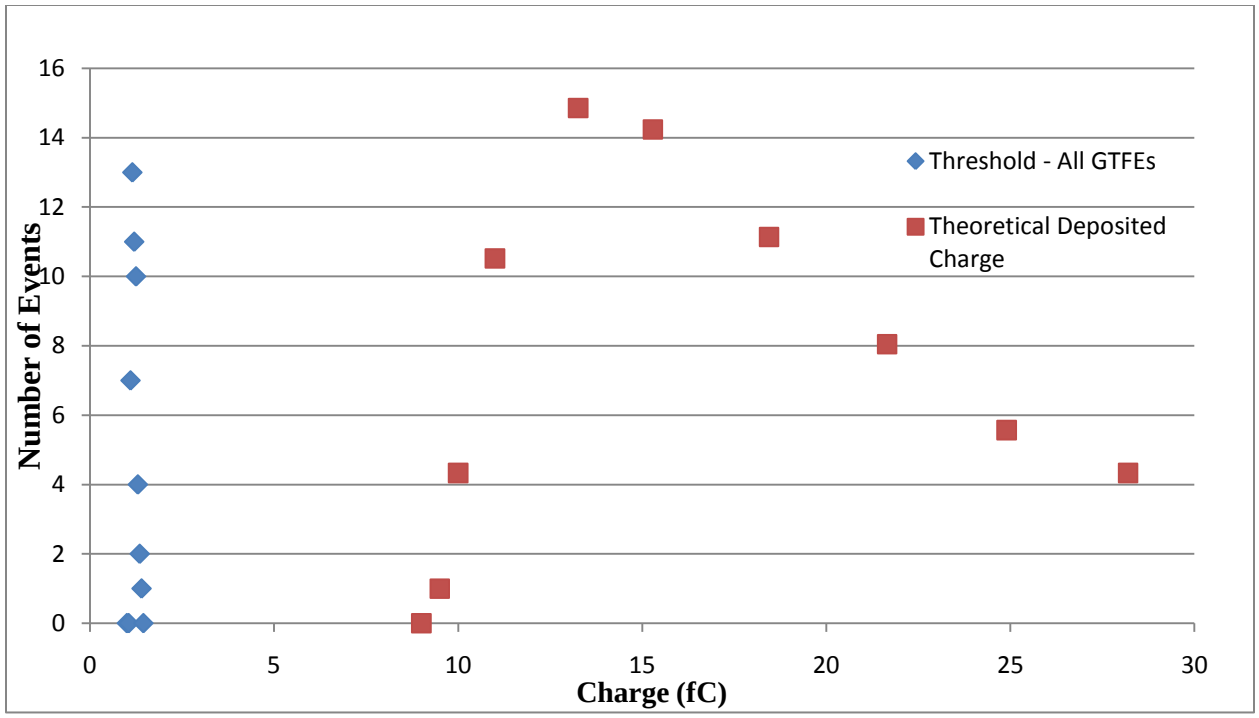


Figure 13: Charge Threshold Compared to Generated Charge from 250 MeV Protons

This shows that most particles hitting the boards will be well enough over the threshold we calculated to be counted as events. If the boards were built and calibrated with too high a charge threshold, we may be excluding legitimate events from our data, thus being an inaccurate representation of the true number and energy of particles passing through the pCT system. However, the magnitude of noise from the electronics will be small enough to be excluded from the event data.

III. Source Measurements

Figures 7 through 10 show the typical results from the source measurement tests. The profiles themselves show the event data for each of the individual layers, which reflect the activity on the X and Y axes. These profiles can be put together to create the 2-D position plot shown in the middle of each figure. The color displays the density of particles hitting that specific coordinate. The way the profiles are put together to form the 2-D graph depends on the

trigger mode used during the source measurement test. For the coincidence mode, when a detector registers an event on one layer, there is a short time allowed by the FGPA code for that event to register on the other layer. If it isn't registered within that time, the whole event is thrown out. In the independent mode, if a layer detects an event the trigger is left open (like in coincidence mode) for the other layer to be hit. The difference between the two modes lies here. If the event is not detected within the given time, it is still registered as an event. Thus, the coincidence mode is more discriminatory, and should yield a fewer number of events than the independent mode. Unfortunately, we see the opposite effect happening in most of the tests. For instance, the tests on board 5 yield 6889 events for coincidence mode and 7058 events recorded in independent mode. While the gap is not always large (sometimes the event count only differs by 5 events), there is a clear efficiency difference. The question is: from what? Here are the layer profiles for board 5:

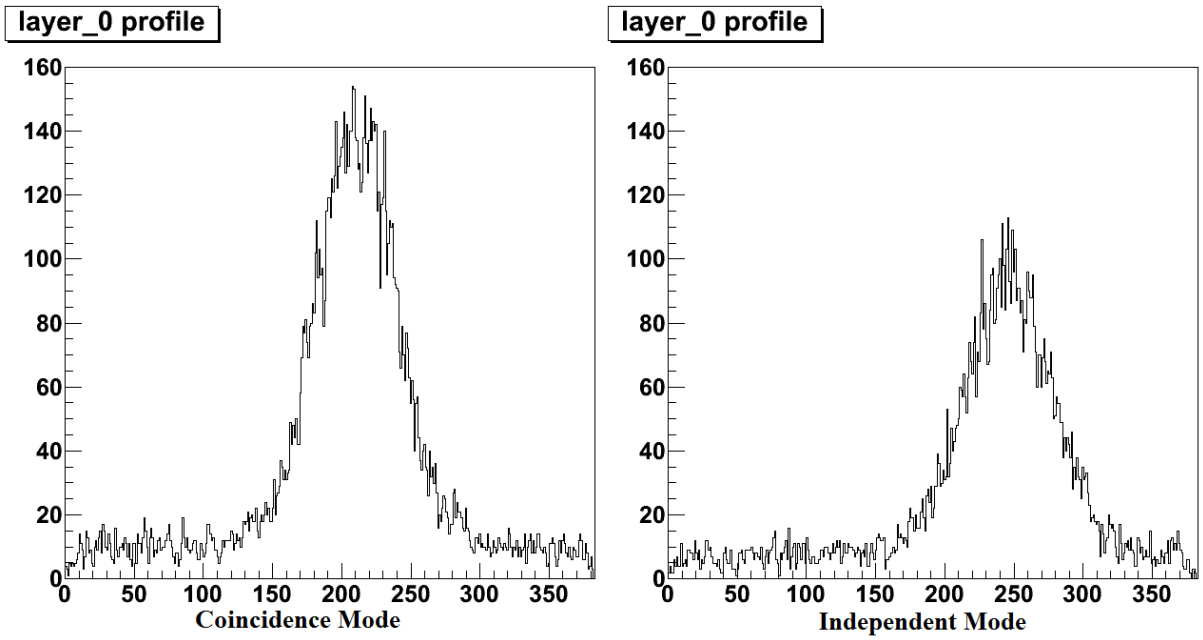


Figure 14: Layer 0 Profiles for Board 5 in Both Trigger Modes

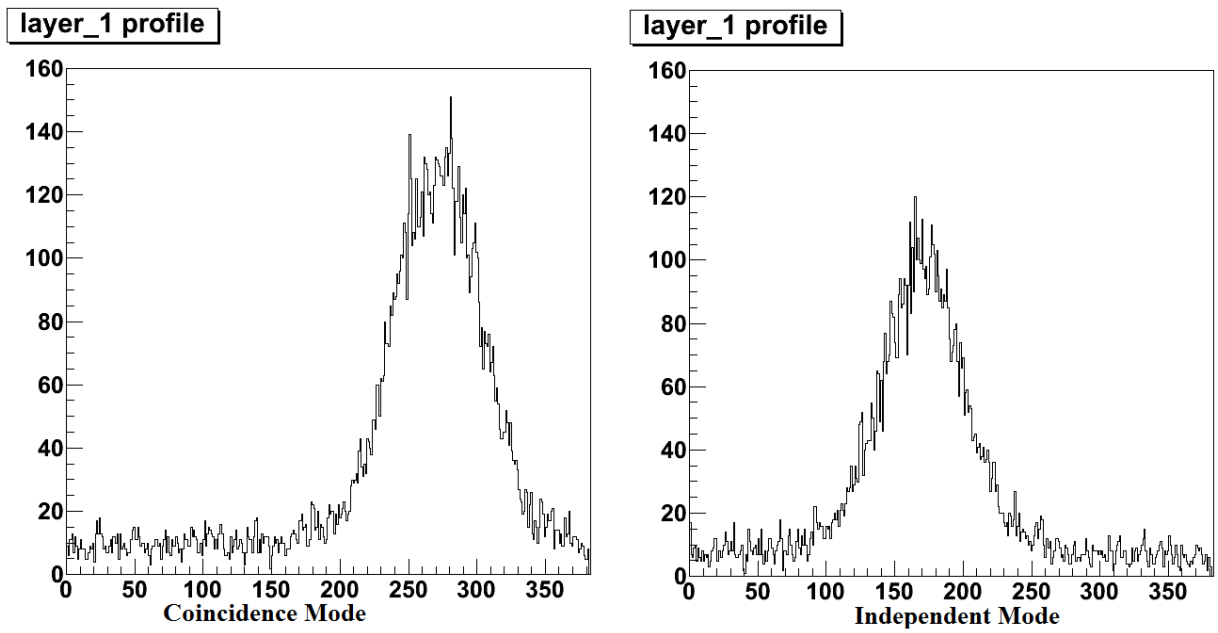


Figure 15: Layer 1 Profiles for Board 5 in Both Trigger Modes

On both layers the coincidence mode showed a larger number of events than the independent mode. It has been suggested that the way the trigger modes were coded may be the culprit. When we ran source measurements in the independent mode masking an entire layer, there were some strange occurrences. As I stated before, when an event triggers on one layer, that event will be counted regardless of if it registers on another layer. So, if I run the test in independent mode and mask an entire layer, I should have a profile for the unmasked layer and a blank graph for the masked layer. A test on board 9 in independent mode, masking layer 1, shows over 8000 events on one channel! There are several tests on different boards showing different anomalies when a layer mask is used. Noise counts also seem to be higher when a mask is used. Figure 16 shows the test with both layers active.

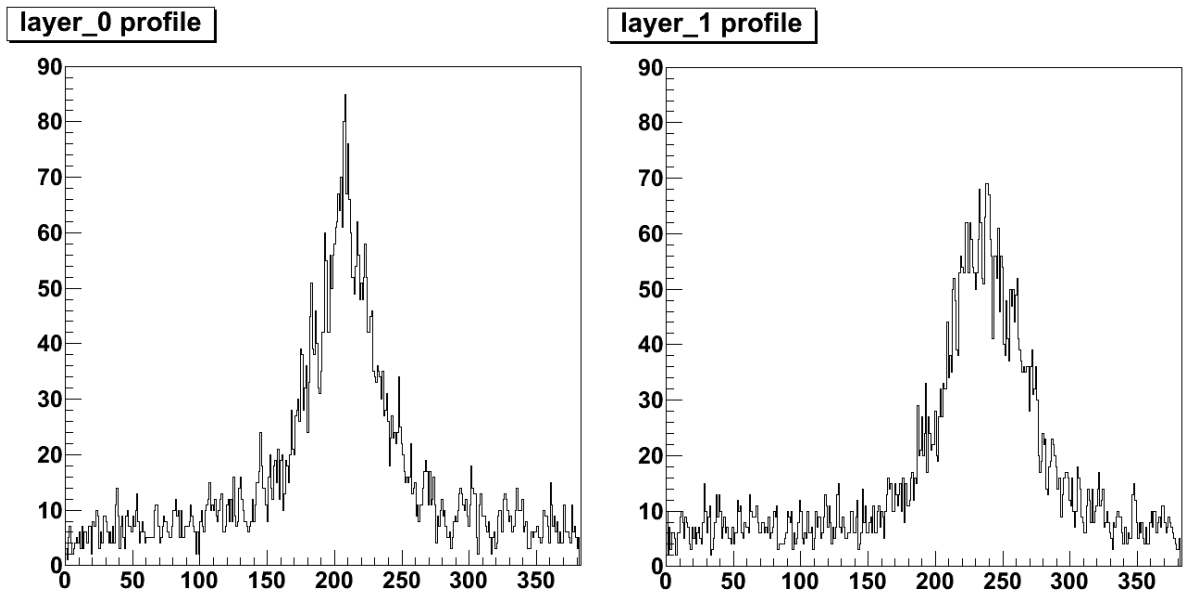


Figure 16: Board 9, Independent Mode, Both Layers Active

This test *seems* fine. But this is another board whose independent mode yielded more events than its coincidence mode. If there are particles that are stopped in one layer or deflected away, they would drive up the event count. So we ran some ROOT scripts on the data to pick out events that may have registered on one layer, but not the other. A great majority of the tests which were considered "good" showed blank histograms for event count on only a single layer. In theory, this means that all of the events that are counted involved particles passing through both layers. There was only one board that deviated from this; Board 5 had five events on layer 0 that didn't register on layer 1. Figure 17 shows this.

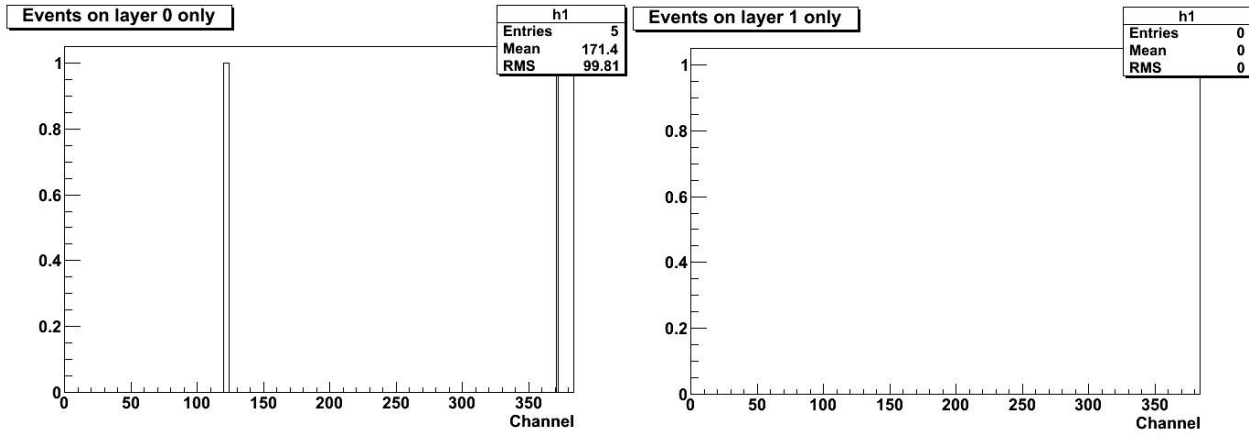


Figure 17: Single-Layer Events for Board 5 in Independent Mode

However, this same analysis done on a later test of Board 5 yielded another pair of blank (no events) histograms. It is suspicious that the independent mode showed so little of these kinds of events. I expected there to be a sizeable number of events that only registered on a single layer, since the independent mode trigger will record the hit as an event if a particle only hits a single layer.

Since there were a large number of events coming through on a layer that was supposedly masked by the code, but the profile seemed fine when untouched by the coded mask, it is reasonable to conclude that there is probably something in the code causing anomalies. The question is, where in the code? Since the events in the data are only considered events if they pass through the screening of the trigger, the ROOT scripts used to examine the event data will only be able to sort the events that the triggers allow. This, along with the fact that the independent mode yields fewer events than the coincidence mode leads me to believe that something in the "independent" trigger code may not be discriminating events the way we intend.

Since the coincidence mode is what will be used in all major tests, it is a relief that it seems to be working properly. If there was a major issue across all of the tests, it was usually the case that the equipment needed to be rebooted. Unfortunately this was another sign that equipment was playing a part in data anomalies. Performing the reboot and running the same test seemed to always resolve those issues, as was the case for tests like those shown in fig. 9 and 10.

6. Conclusion

The strip check proved to be a reliable and useful tool for confirming the integrity of the GTFE channels. If we saw large gaps missing from the event profile, it was usually the case that the GTFE itself was bad and had to be replaced. Constructing gain and response plots of the calibration data show us if we have any hyper-sensitive GTFEs. Examining the distributions of charge threshold frequency will give us an average charge threshold for that layer, enabling us to make sure the charge threshold is well below the average particle energy. This will ensure that we are reducing noise events and counting as many legitimate events as possible. Two-dimensional tracker profiles can be easily constructed using the x or y profiles of each plane. Analysis has shown that the coincidence triggering mode is reliable and consistent, while the independent trigger code may be responsible for producing some anomalies.

7. References

1. Petterson, M., et al. "Proton Radiography Studies for Proton CT." 2006. MS. Loma Linda University School of Medicine.

2. Schulte, R., et al. "Design of a Proton Computed Tomography System for Applications in Proton Radiation Therapy." 2003.

3. Mueller, K., et al. "Reconstruction for Proton Computed Tomography: A Practical Approach." 2003. MS.

4. Sadrozinski, Hartmut F.-W. "Imaging Tumors with Protons: Proton Computed Tomography (pCT)." Hiroshima. Aug. 2009. Lecture.

5. Sadrozinski, Hartmut F.-W. "Hardware Development for Proton Computed Tomography (pCT)." Santa Cruz. Aug. 2009. Lecture.

6. Fuerst, Carlin. "Energy Loss of Protons in Silicon Strip Detectors for Computed Tomography." Thesis. University of California, Santa Cruz, 2010.