

THE GEORGE WASHINGTON UNIVERSITY

DEPARTMENT OF PHYSICS

UNDERGRADUATE THESIS

Detector Integration for MUSE

Author:

Jack Hirschman

Supervisor:

Professor Evangeline Downie

Co-advisor:

Ievgen Lavrukhin

November 8, 2022

Abstract

Until recently, it was thought that the proton radius was known with an uncertainty of 1%. However, in 2010, a measurement of the proton radius using muonic hydrogen yielded a radius value an order of magnitude more precise but 4% smaller than and completely inconsistent with the previous results. Resolving this discrepancy, known as the Proton Radius Puzzle, requires new and different measurements. The MUon proton Scattering Experiment (MUSE), carried-out at the Paul Scherrer Institute (PSI), Switzerland, will thus be the first to measure elastic muon proton scattering with sufficient precision to address the Proton Radius Puzzle.

This project involves working on several aspects of the MUSE detector and readout system. It incorporates the assembly of silicon photomultiplier (SiPM) detectors and improving the timing resolution and efficiency of MUSE beamline detectors with the development of high voltage (HV) calibration software based on Charge-to-Digital Converter (QDC) spectra. Additionally, in order to perform gas mixture tests for optimization of straw tube tracking (STT) detector performance, an adjustable gas mixing system was assembled at the Hebrew University of Jerusalem, Israel. Moreover, to test the performance of the Time-to-Digital Converters (TDC), a hardware test stand and supporting software were developed, which were used during the assembly of the MUSE Trigger and Data Acquisition (TDAQ) system in order to test the performance of new TDC modules. Lastly, the project covers the design and development of an ECL-to-LVDS logic level converter. The converter is currently used as a part of the MUSE TDAQ system, interfacing between hardware with different logic standards.

This material is based upon work supported by the National Science Foundation under Grant Numbers PHY-1714833, PHY-1614850, and OISE-1358175 and by a GW Luther Rice Undergraduate Research Fellowship.

Declaration

The developments presented in this thesis are for the MUon proton Scattering Experiment (MUSE) based at the Paul Scherrer Institute, Switzerland. I participated fully in the projects presented in this thesis. This thesis was composed by myself with generous guidance by Professor Evangeline Downie and Ievgen Lavrukhin.

Jack Eden Hirschman

Acknowledgements

The opportunity to be a part of the MUSE collaboration has been a unique research opportunity, especially for an undergraduate student. So many people on the collaboration provided guidance and instruction that I will be able to carry with me into the future. Thank you to the entire MUSE collaboration.

A few people warrant special gratitude for their advising roles, dedication, and endless support. Thank you Professor Evangeline Downie, Ievgen Lavrukhin, Professor Guy Ron, Dr. Tigran Rostomyan, and Dan Cohen.

Outside of MUSE, many others at both GW and PSI dedicated time and effort to helping me understand tasks and perfect designs. I am extremely grateful.

In my personal life, I would like to give special thanks to my family for being extremely supportive and willing to accept my long communication absences while I am working. Special thanks to my Mom and to my Dad for supporting me and for providing a great environment to cultivate learning and research. A special thanks also goes to my girlfriend who has completed countless edits of my research presentations and documents, has listened to my practice talks, and has put-up with my somewhat ridiculous late night (or early morning) study hours at Corcoran Hall (my home away from home away from home).

This work at GW, HUJI, and PSI would not have been possible without funding. This material is based upon work supported by the National Science Foundation under Grant Numbers PHY-1714833, PHY-1614850, and OISE-1358175 and by a GW Luther Rice Undergraduate Research Fellowship.

Contents

List of Figures	i
List of Tables	v
List of Symbols	vi
1 Introduction	1
1.1 The Proton Radius Puzzle	1
1.2 Measurements of the Proton Radius	3
1.2.1 Spectroscopy of hydrogen	3
1.2.2 Scattering of electrons	4
1.2.3 Muonic hydrogen spectroscopy	5
1.2.4 Muon Scattering	5
1.3 MUSE and this project	6
2 MUon Scattering Experiment (MUSE) Overview	7
2.1 PSI Accelerator Complex	7
2.2 MUSE Kinematics	8
2.3 MUSE Detector Setup	10
2.3.1 Beam Line Detectors	11
2.3.2 Scattered Particle Detectors	12
2.4 Trigger and Data Acquisition (TDAQ) System	13
2.4.1 General Concept of TDAQ	13
2.4.2 MUSE TDAQ	15
2.5 Data Analysis Package	16
3 Current Work	18
3.1 SiPM Assembly & HV Calibration Software	18
3.1.1 Detecting Principle	18
3.1.2 Assembly	18
3.1.3 HV Calibration Software Development	19
3.1.4 Conclusions	26
3.2 STT Gas Mixture Optimization Studies	28
3.2.1 Introduction	28
3.2.2 Work	29
3.2.3 Conclusions	29

3.3 PaDiWa FPGA Design Modification for TDC Test Setup	32
3.3.1 Introduction	32
3.3.2 Work	33
3.3.3 Conclusions	37
3.4 ECL-to-LVDS Converter PCB	37
3.4.1 Introduction	37
3.4.2 Design	38
3.4.2.1 Prototype	38
3.4.2.2 Final Design	38
3.4.3 Testing	41
3.4.4 Conclusions	43
4 Conclusions	49
References	50
Appendices	54
A PaDiWa FPGA Design	54
B SiPM HV Calibration	57
C ECL-to-LVDS Design Schematic and PCB Layers	79
D Additional ECL-to-LVDS Board Design Information	85
E Vulom4b Information	87
F Additional Collaboration Service Work	89
G Simplified Scattering Theory	92

List of Figures

1.1.1	Summary of recent proton radius results taken from [1].	2
2.1.1	The three stages of proton acceleration at PSI: (a) Cockcroft-Walton Accelerator, first stage of accelerator at PSI; (b) Injector 2 small ring cyclotron, second stage of accelerator at PSI; (c) Large ring cyclotron, third stage of accelerator at PSI [2].	8
2.1.2	Magnetic elements in $\pi M1$ beamline [2].	9
2.2.1	Time separation of particles at different momenta [3].	9
2.3.1	Sketch of the detector setup in the $\pi M1$ area, taken from [3].	11
2.4.1	General overview of elements needed in TDAQ system [4]. The trigger starts a “busy” signal, which persists as long as any of the data acquisition system elements are still processing data. Once all of the DAQ systems are ready, this busy signal is cleared. While the busy signal remains in place, the formation of new Trigger signals is blocked.	14
2.4.2	Two level trigger. The Level 1 trigger contains several triggers that determine PID and scattered event. The Level 2 trigger collects information from Level 1 trigger in order to form a decision [5].	16
3.1.1	(a) Hamamatsu SiPM [6]; (b) SiPM photon detection efficiency [7].	19
3.1.2	Steps for full assembly of SiPM hodoscope planes used for Beam Hodoscope and Beam Monitor [6].	20
3.1.3	SiPM time resolution. Dividing the standard deviation by two yields the timing characteristics for individual SiPMs. Here $\frac{113.6 \text{ ps}}{2} = 56.8 \text{ ps} < 100 \text{ ps}$ [6]. . .	21

3.1.4	QDC spectra for Beam Hodoscope. Higher QDC channels means more charge collected. As voltage increases, the gain increases, and the peaks shift to the right [5].	23
3.1.5	Breakdown of QDC spectra consisting of pedestal and peak with background underneath.	24
3.1.6	Linear (left) and logarithmic (right) scales for QDC spectra with fits: (a) asymmetric Gaussian for peak and linear background; (b) Landau for peak and exponential for background.	26
3.1.7	Output from HV calibration software: (a) File output used for extraction of calibration factor from QDC data; (b) Histogram output of calibration factors.	27
3.2.1	(a) STT design [5]; (b) One wall of STT channels [8].	28
3.2.2	Schematic overview of lab test setup for Ar to CO ₂ gas proportion tests for STT detector performance optimization.	30
3.2.3	Lab test setup for Ar to CO ₂ gas proportion optimization for STT detectors: (a) Gas mixing system; (b) STT test setup with radioactive source; (c) Data readout setup with PaDiWa and TRB3.	31
3.2.4	Rate versus voltage for different gas proportion mixtures of Ar and CO ₂ . The source used was Pa-234 and was placed approximately 28 cm from STT detectors. Test setup is shown in figure 3.2.3	31
3.3.1	(a) TRB3 with four TDCs, one in each corner; (b) PaDiWa board with Lattice FPGA [5].	32
3.3.2	Simulated waveform of generated clock signal showing 11 KHz clock signal waveform and zoomed-in capture of 30 ns clock pulse width.	33
3.3.3	Block diagram of PaDiWa FPGA architecture.	34
3.3.4	TRB3 connected to PaDiWa and power supply [5].	35

3.3.5 Trigger channel (a) and TDC register (b) values from TRB3 with PaDiWa providing test signal input [5]	35
3.3.6 Timing vs edge plots: (a) good channel; (b) bad channel.	36
3.3.7 Cross-channel resolution plots: (a) good channel; (b) bad channel.	36
3.4.1 ECL-to-LVDS converter's placement in MUSE's TDAQ system.	39
3.4.2 Prototype PCB with soldered components.	39
3.4.3 Single channel design overview. The board layout of the single channel is shown in figure 3.4.5	44
3.4.4 The schematic of different components for a single channel of logic level converter: (a) TTL Input, (b) ECL Input, (c) TTL-to-LVDS converter, (d) ECL-to-TTL converter, (e) LVDS output, (f) TTL output, (g) Switch1 to switch between ECL and TTL inputs, (h) Switch2 to switch between TTL and LVDS outputs.	45
3.4.5 Labeled board of single channel.	46
3.4.6 Power circuitry for positive and negative supplies along with red and yellow LED indicators.	46
3.4.7 Test setup for ECL-to-LVDS and TTL-to-LVDS (red dashed line). (a) Function generator (b) Power supply (c) Oscilloscope (d) Design under test.	47
3.4.8 Jitter analysis histograms for (a) TTL-to-LVDS conversion and for (b) ECL-to-LVDS conversion, showing test setup issues at lower frequencies.	47
3.4.9 Final board placement for MUSE at PSI, Switzerland.	48
C.0.1 Top layer of PCB design (signal layer).	80
C.0.2 Second layer of PCB design (ground layer).	80
C.0.3 Third layer of PCB design (+5V layer).	81
C.0.4 Fourth layer of PCB design (-5V layer).	81
C.0.5 Fifth layer of PCB design (ground layer).	82

C.0.6	Bottom layer of PCB design (signal layer).	82
C.0.7	Top layer silkscreen.	83
C.0.8	Bottom layer silkscreen.	83
C.0.9	Schematic for final board design.	84
D.0.1	Stacked board layers.	85
D.0.2	Example of impedance calculation tool from KiCAD.	86
E.0.1	Position of adapter board (on left) with Vulom4b board (on right) [5].	88
E.0.2	Assembled Vulom4b adapter board for interface to TRB3 [6].	88
E.0.3	Vulom4b's function in MUSE's TDAQ system [5].	88
F.0.1	(Me) Drilling mounting holes in frame to hold SPS walls.	91
F.0.2	(a) Mounting plate adapter attached to SPS wall; (b) View from top of SPS wall; (c) SPS still attached to crane while being place on carriage rails.	91
G.0.1	Setup for Rutherford scattering with incident electron elastically scattering off of a target [9].	92

List of Tables

3.4.1 Maximum current consumption on the positive and negative supply lines for one channel is presented here.	40
3.4.2 Total current consumption for the design is presented here.	41
3.4.3 Statistics for frequency, rise time, fall time, positive duty cycle, and negative duty cycle measurements for TTL-to-LVDS conversion from 100 KHz to 75 MHz.	43
3.4.4 Statistics for frequency, rise time, fall time, positive duty cycle, and negative duty cycle measurements for ECL-to-LVDS conversion from 100 KHz to 75 MHz.	43

List of Symbols

ADC	Analog-to-Digital Converter
BH	Beam Hodoscope
BM	Beam Monitor
CF	Calibration Factor
CODATA	The Committee on Data for Science and Technology
DAQ	Data AcQuisition (System)
ECL	Emitter-Coupled Logic
FPGA	Field Programmable Gate Array
GEM	Gas Electron Multiplier
GW	George Washington University
HUJI	Hebrew University of Jerusalem, Israel
LVDS	Low Voltage Differential Signaling
PID	particle identification
PSI	Paul Scherrer Institute
QDC	Charge-to-Digital Converter
RF	radio frequency (signal from HIPA accelerator used for PID)
SiPM	Silicon Photomultiplier
SPS	Scattered Particle Scintillator
STT	Straw Tube Tracker
TDC	Time-to-Digital Converter
TDAQ	Trigger and Data AcQuisition (System)
TOF	Time-of-flight
TTL	Transistor-Transistor Logic

Chapter 1

Introduction

1.1 The Proton Radius Puzzle

The proton radius, r_p , seems like something that should be well-known; however, the question of “what is *a* proton radius” immediately leads to the distinction of a charge and a magnetic radius for the proton, and the question “what is *the* proton radius” begs for a value set in stone. Historically there have been two main methods for measuring the proton radius: electron-proton scattering, which involves scattering a beam of electrons at a proton target and measuring how they scatter, and electronic hydrogen spectroscopy, which involves exciting electrons in an atom and measuring energy level transitions (Both of these methods will be explained in more detail later).

Over time, the uncertainty in the results from electron scattering and electronic hydrogen spectroscopy has reduced, with both methods yielding radii measurements that are in agreement, as reflected in the CODATA¹ value of 0.8775(51) fm [10]. Nonetheless, the uncertainty of the accepted value of r_p was still the limit on precision in many atomic physics measurements. This need for increased precision motivated the development of a new method of

¹CODATA, or the Committee on Data for Science and Technology, provide recommended values for physical constants and conversion factors, based on review of current literature encompassing recent experimental results and meta-analyses of all available data.

radius measurement via spectroscopy of muonic hydrogen, a more technically challenging experiment than electronic hydrogen spectroscopy.

The muonic hydrogen spectroscopy results, published in 2010 [11, 12], report a proton radius of $0.84184(67)$ fm and $0.84087(39)$ fm, respectively, and with a precision an order of magnitude better than the accepted radius value. However, these new results are incompatible with measurements obtained from ep scattering by JLab [13] and Mainz [14], electronic hydrogen spectroscopy, and the resulting CODATA recommended value of the proton radius [10]. The results from these scattering and spectroscopy experiments are presented in figure 1.1.1. This discrepancy became known as the "Proton Radius Puzzle" (PRP).

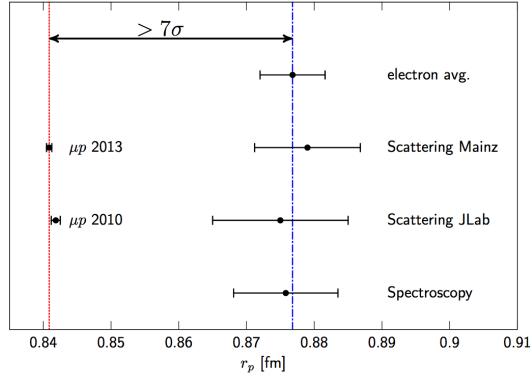


Figure 1.1.1: Summary of recent proton radius results taken from [1].

The proton radius is used frequently in physics, anywhere from nuclear and particle physics experiments, where r_p can become a limiting factor in precision, to emission of electromagnetic radiation where r_p affects the Rydberg constant [15]. Furthermore, while the value of the proton radius is interesting in and of itself, even more interesting are some of the postulated causes of the PRP. One of the proposed solutions, known as hadronic or two-photon effects, is that the presence of the muon deforms the proton, and these measurements are actually of a radius of a deformed proton rather than of the undisturbed r_p . Another proposed solution postulates that new particles exist which react only with muons and not electrons. If such particles do exist, they would present a departure from the Standard Model

by breaking the notion of lepton² universality, which prescribes that the only difference among leptons (in this case, electrons and muons) are their masses [15, 17].

1.2 Measurements of the Proton Radius

The various methods of scattering and spectroscopy are explained here.

1.2.1 Spectroscopy of hydrogen

Inside the hydrogen atom, electrons populate allowed energy levels, or orbitals. Excited electrons will move or transition to higher energy levels if provided the correct input excitation energy. These electrons will then transition back into their ground states with the release of photons from this de-excitation.

Therefore, hydrogen spectroscopy experiments use finely-tuned lasers³ to excite atomically bound electrons to higher energy levels and then directly measure the subsequent de-excitation. The differences between the energy levels can be calculated precisely using quantum mechanics and knowledge of the electromagnetic potential caused by the attraction between the positively charged proton and the negatively charged electron. However, the finite size of the proton results in a correction to the electromagnetic potential, which, in turn, affects the magnitude of the energy differences between orbitals. This perturbation to the electromagnetic potential manifests as the electron can pass through the proton. While the electron is inside the proton, the charge of the proton is partially screened, reducing the average attractive potential felt by the electrons [11].

²A lepton is an intrinsic particle of half-integer spin that does not interact via the strong force. Electrons, muons, and tauons, and their accompanying neutrinos, are leptons [16].

³The precision for these experiments comes from the ability to precisely tune the lasers. The released photons from de-excitation serve as a tag that the orbital transition of interest took place, but the precise energy determination comes from the precisely determined frequency of the laser which excites the transition [11].

1.2.2 Scattering of electrons

Electron scattering experiments scatter a beam of electrons off a proton target and measure, via systems of detectors, the scattering cross section. The cross section expresses the probability that, given a single electron and proton in the initial conditions, the electron would scatter at the particular scattering angle θ and is a function of the incident electron's momentum and outgoing electron's scattering angle.

In the analysis of the distribution of scattered particles, the measured cross section is equal to a precisely calculable differential cross section, known as the Mott cross section,⁴ multiplied by the Form Factor, which accounts for the non-point-like structure of the proton. Said another way, the Form Factor, or this parameter of proportionality, is a value used in nuclear physics to relate the scattering cross section to a charge distribution. In this case, the Form Factor parameterizes the modification of the Mott cross section.

The Form Factor is dependant upon the transfer of momentum between the incident electron and the proton target. The parameter representing this exchange is known as four-momentum transfer, q^2 , which accounts for energy and momentum transfer. Using scattering theory (see Appendix G), the radius can be determined as

$$r_p^2 \equiv -6 \frac{dG_E}{dQ^2} \Big|_{Q^2=0} \quad (1.2.1)$$

where Q^2 is the negative of the square of four-momentum transfer and where G_E is a Form Factor that accounts for charge and magnetic Form Factors [15, 18]. With cross sections measured, one can calculate the associated Form Factors. From equation 1.2.1, the radius can be extracted from the Form Factor data by plotting Form Factor as a function of momentum transfer, fitting the data, and measuring the slope as the momentum transfer approaches zero.⁵

⁴The Mott cross section can be thought of as the Rutherford cross section adjusted for recoil of the target proton [18].

⁵There are limitations to this method, experimentally. To capture the whole picture, one must account

1.2.3 Muonic hydrogen spectroscopy

The discussion in the section on electronic hydrogen spectroscopy (section 1.2.1) is generalizable to muonic hydrogen⁶ as well. In fact, the probability of a lepton being inside the proton is proportional to the ratio of sizes of the atom to the size of the proton. Since muons are roughly 207 times more massive than electrons, the atoms are more compact, and the situation results in a higher probability of the muon spending time inside the proton. As such the perturbations of the orbitals are more pronounced and the component of the energy transition due to the finite size of the proton becomes more significant. This means that muonic hydrogen spectroscopy can lead to a far more precise radius determination than electronic hydrogen spectroscopy [11]. However, this more precise measurement entirely disagrees with the other two forms of measurement, thus establishing the PRP and motivating the need for a muon scattering experiment.

1.2.4 Muon Scattering

While muonic hydrogen spectroscopy, electronic hydrogen spectroscopy, and electron proton scattering have been used for measuring the proton radius, there are no muon-proton scattering data thus far with the necessary precision to address the proton radius puzzle. Muon scattering is more difficult to measure in a precise manner as, unlike in the case of electron or proton beams, the muons must be generated before they can be formed into a beam. This means that muon beamlines are always secondary beamlines, leading to issues with low flux, mixed species, decaying, and divergent beams with larger spot-sizes.

Using the secondary beamline at the Paul Scherrer Institute (PSI) which supplies a

for all scattering angles; however, at large scattering angle, the momentum transfer is no longer close to zero. Furthermore, it is impractical to have complete coverage of all the scattering angles. A more detailed discussion of this phenomenon is beyond the scope of this thesis.

⁶It should be noted that muonic hydrogen is difficult to make, as the process involves replacing electrons with muons in a hydrogen atom. This process requires a negative muon beam with the correct energy to have a high probability of binding to a hydrogen target. Additionally, the laser required for these experiments was one of the most challenging design aspects, requiring high power and fast turn on times [11].

mixture of muons, pions, and electrons, at a set momentum, the MUon proton Scattering Experiment (MUSE) will thus fill in the missing measurement technique by using muon proton scattering with simultaneous measurements of elastic $e^\pm p$ and $\mu^\pm p$ scattering cross sections in order to extract the proton radius from the slope of the Form Factor. Since hadronic effects⁷ lead to changes in the cross section with charge of scattered particle, MUSE can directly measure whether or not hadronic effects, or two-photon effects, are the source of the PRP by studying the ratios of e^+ to e^- cross sections and μ^+ to μ^- cross sections. MUSE can also directly compare simultaneous e and μ measurements, minimizing systematic errors in the extraction of the radius difference between electron and muon measurements.

1.3 MUSE and this project

In order to extract a proton radius to high enough precision to constrain the PRP, MUSE must have a precise detector setup and readout system. As such, the contributions presented in this thesis cover projects related to detector development, trigger and data acquisition (TDAQ) development, and analysis, which all aid MUSE’s performance.

This thesis will begin with an overview of MUSE, including sections on the PSI accelerator complex, MUSE kinematics, MUSE detector setup, general TDAQ concepts as well as MUSE’s TDAQ system, and the data analysis package. The projects covered here include silicon photomultiplier assembly and high voltage calibration software development, studies on gas mixture optimization for straw tube tracking detectors, time-to-digital converter test setup development, and design and testing of a logic level converter board for MUSE’s TDAQ.

⁷Hadrons are strongly interacting particles and include baryons, such as the proton, and mesons, such as the pion [16].

Chapter 2

MUon Scattering Experiment (MUSE) Overview

2.1 PSI Accelerator Complex

PSI has maintained its High Intensity Proton Accelerator (HIPA) since the early 1970s. This three-stage accelerator first uses a Cockcroft-Walton accelerator, which feeds the Injector 2 small ring accelerator. This accelerator brings the protons to 72 MeV. They are then fed into the large ring accelerator, bringing the protons to 590 MeV or 80% of the speed of light. The three stages of the PSI accelerator complex can be seen in figure 2.1.1 [2].

The primary proton beam hits a ^{12}C target, producing mainly π^+ , π^- , and π^0 . The π^\pm decay to μ^\pm or e^\pm along with the associated muonic or electronic neutrinos. The π^0 decays to two photons which undergo pair production inside the carbon ring, resulting in e^\pm that then pass into the beam. This mixture of particles formed by the interaction of the proton beam and carbon target is known as a secondary beam. The particles produced in the target have a large range in momenta and directions [2].

MUSE uses one of the secondary beamlines, called $\pi M1$ whose particles (electrons,

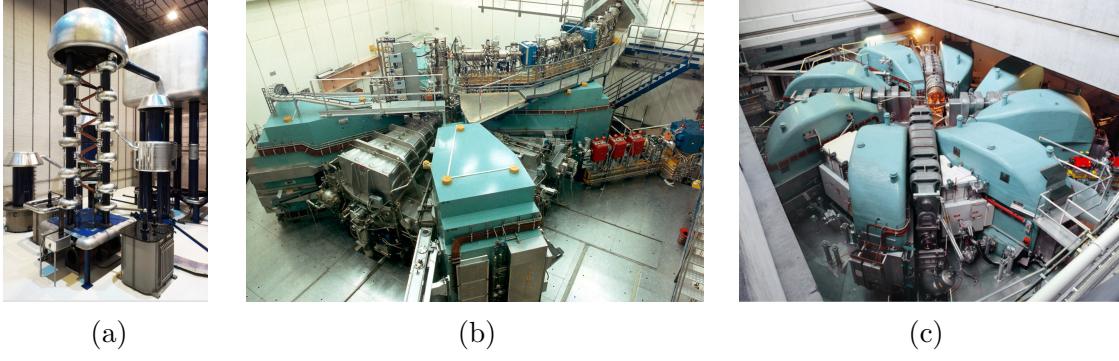


Figure 2.1.1: The three stages of proton acceleration at PSI: (a) Cockcroft-Walton Accelerator, first stage of accelerator at PSI; (b) Injector 2 small ring cyclotron, second stage of accelerator at PSI; (c) Large ring cyclotron, third stage of accelerator at PSI [2].

muons, and pions of both polarities) can be selected to have a momentum in the range of $100 \text{ MeV}/c$ to $500 \text{ MeV}/c$. The schematic diagram of the $\pi M1$ secondary beamline is shown in figure 2.1.2. Specific momenta are first selected through use of dipole magnets ASM11 and ASM12 and collimators. Beam tuning is accomplished with quadrupole magnets QSL11-QSL16 set along the beamline. The intensity of the beam is controlled using FS11 and FS12 jaws, and the collimator set in the intermediate focus can be used to more finely select the momentum range under study.

2.2 MUSE Kinematics

Due to the mass differences, secondary beam particles arriving at the $\pi M1$ target area are separated in time. Figure 2.2.1 shows the temporal separation of μ , π , and e given particular momenta used by MUSE. As electrons with momenta above $100 \text{ MeV}/c$ are highly relativistic, the electron time-of-flight in all stages of the experiment remains constant regardless of beam momentum. Thus the time difference between the accelerator radio frequency (RF) and the detection of the electrons in the BH remains constant for all beam momenta. Muons with mass $105.67 \text{ MeV}/c^2$ and pions with mass $139.57 \text{ MeV}/c^2$ and momentum in the range $100\text{-}300 \text{ MeV}/c$ are not relativistic and are detected at different times relative to the accel-

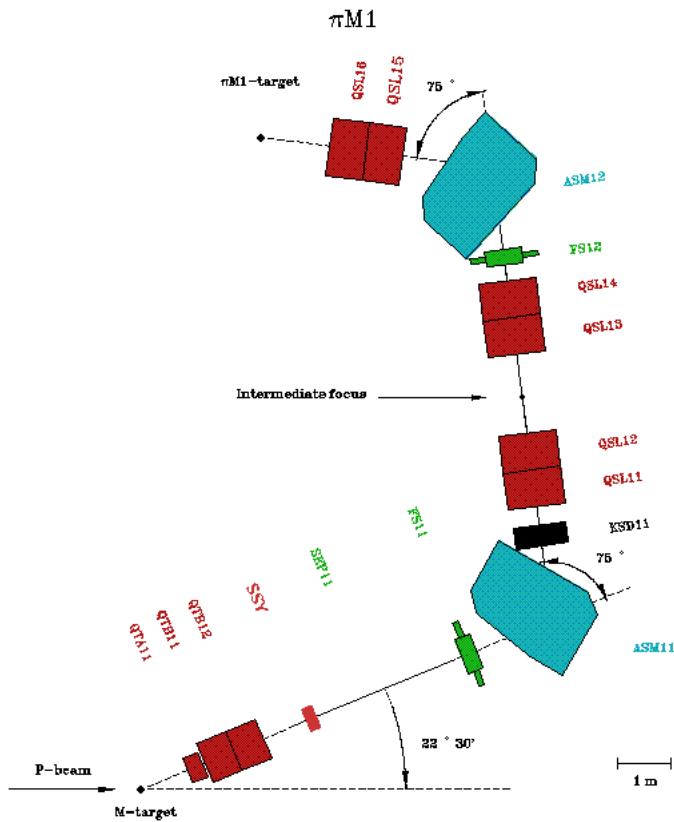


Figure 2.1.2: Magnetic elements in $\pi M1$ beamline [2].

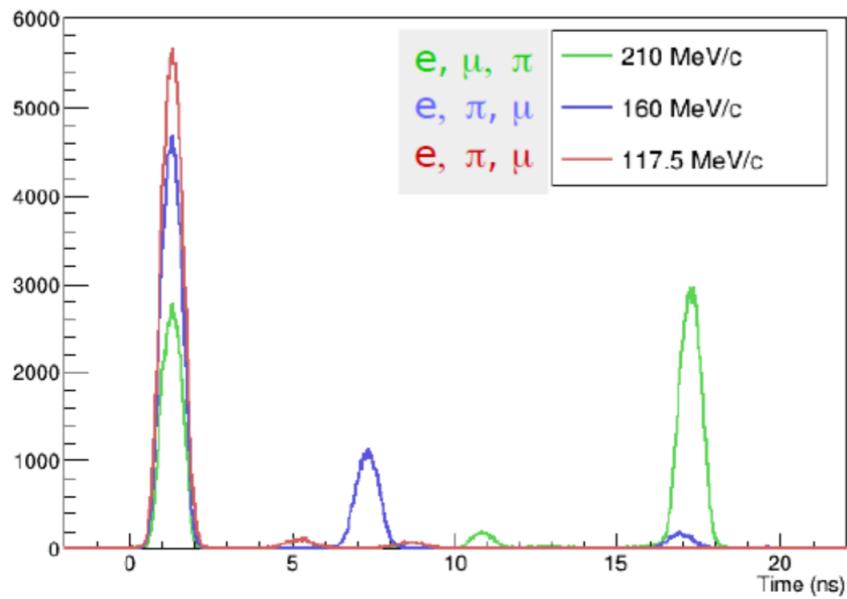


Figure 2.2.1: Time separation of particles at different momenta [3].

erator RF. Momenta are chosen to optimize temporal separation of the particle species in the $\pi M1$ target area.

MUSE will cover scattering angles from 20° to 100° with beam momenta of 115, 153, and 210 MeV/ c . The range of Q^2 for electrons is 0.0016 GeV 2 to 0.0820 GeV 2 , and the range of Q^2 for muons is 0.0016 GeV 2 to 0.0799 GeV 2 [3].

The distribution of the beam momentum must therefore be determined, and separate time-of-flight beam measurements provide this information. The scattering angle is measured event-by-event. Additionally, the beam flux is 3 MHz, so most events will have only one beam particle.

2.3 MUSE Detector Setup

To determine the differential cross section one needs to know the scattering angle from the trajectories of the beam particles (determined using beam line detectors) and scattering particles (measured by scattered particle detectors). Since the beam contains multiple particle types, precise timing of incident and scattered particles is necessary in order to distinguish among different beam particle types and thereby suppress pion scattering and decay events. These requirements lead to the need for various detector types optimized for specific tasks.

The detector setup for MUSE is shown in figure 2.3.1. This setup consists of beam line detectors, which are detectors positioned along the beamline of the $\pi M1$ secondary beamline, and scattered particle detectors, which are positioned to detect particles that scatter off the target in the target chamber. Different detectors are optimized for particular roles: either particle identification (PID) or trajectory reconstruction. The beam line detectors include the Beam Hodoscope (BH); three sets of gas electron multiplier (GEM) detectors; a Veto Scintillator; and the Beam Monitor (BM). The scattered particle detectors include the Straw-Tube Tracker (STT) and the Scattered Particle Scintillator (SPS). The following sections

provide additional details about the various detectors.

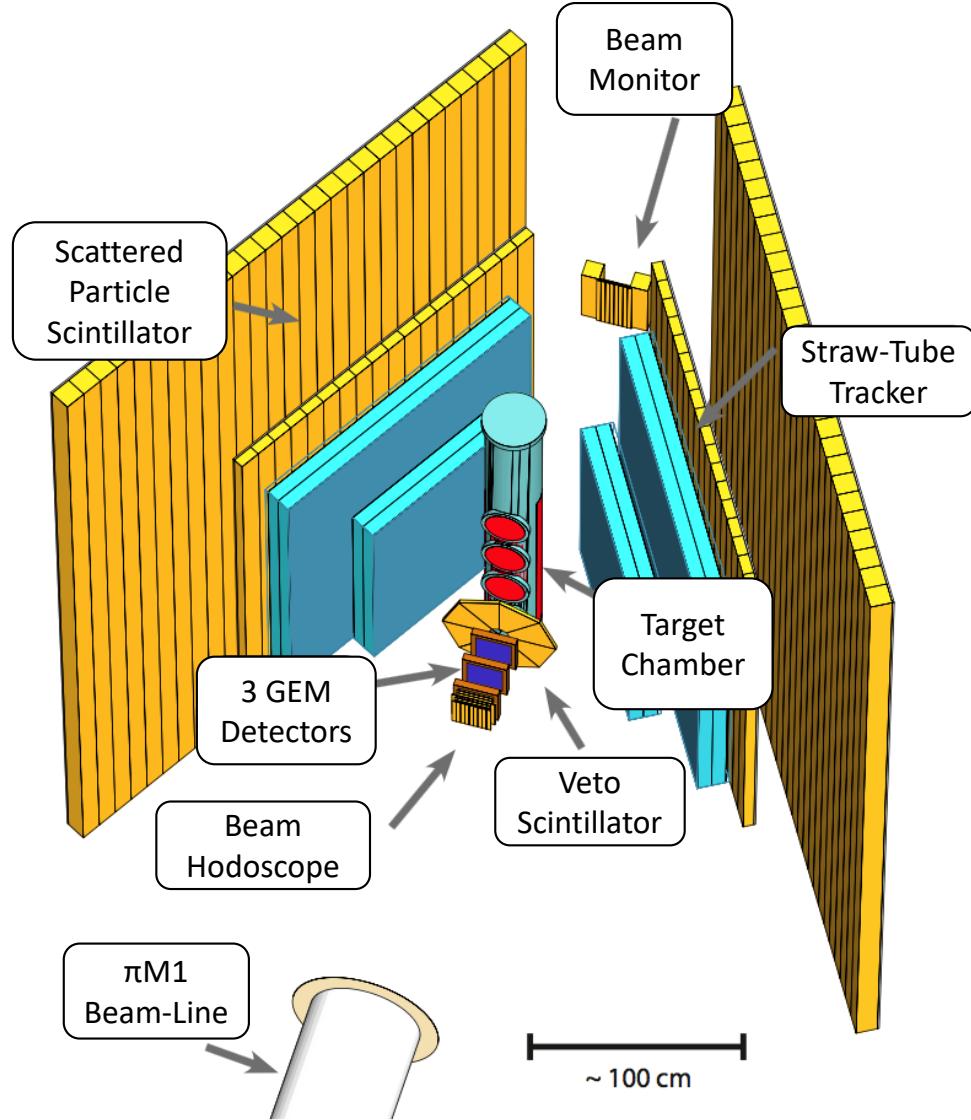


Figure 2.3.1: Sketch of the detector setup in the $\pi M1$ area, taken from [3].

2.3.1 Beam Line Detectors

Due to the mixed particle and diffuse nature of the $\pi M1$ beam, a series of beam line detectors is needed before the target to determine the incoming beam particle species and trajectories, and to allow time-of-flight measurements in combination with the scattering

detectors for background suppression. In addition, a beam monitor detector is required to monitor the beam position and flux during normal running, and for high precision time-of-flight measurements in separate beam momentum calibration runs.

The **Beam Hodoscope** is mainly used to give a precise timing signal, which, combined with the accelerator RF timing, identifies the beam particle species. Thus the BH requires good time resolution (below 100 ps) and high efficiency (above 98%). The most recent prototype has 55-70 ps resolution along with 99.6-99.9% efficiency.

A stack of three **GEM** chambers provide a high resolution (can determine particle position with $< 100 \mu m$ resolution) tracking detector. They are used to determine trajectories of the beam particles on an event-by-event basis.

Veto scintillators are used to reject, at the trigger level, particles that scatter directly into the scintillators from upstream beamline elements and particles that decay before reaching the target.

The **Beam Monitor** is used for measuring beam current and relative beam position during normal running. In special calibration measurements, it will be used for identifying beam momenta by measuring time-of-flight of particles from BH to BM.

2.3.2 Scattered Particle Detectors

Scattered particles are detected by two identical spectrometers. Each with two STT chambers and two planes of scattered particle scintillators.

The **STT detectors** are used to determine scattering particle track with a precision $\approx 150 \mu m$. Each chamber contains 5 horizontal and 5 vertical planes of straws.

The **SPS walls** are used for triggering and to suppress background events via time-of-flight measurements. This detector has time resolution of the order of 50-70 ps depending on particle type and momenta. Its detection efficiency is better than 99 %.

2.4 Trigger and Data Acquisition (TDAQ) System

2.4.1 General Concept of TDAQ

The trigger and data acquisition system of a nuclear experiment is the essential connection between the detectors, described in section 2.3, and the data analysis, discussed in section 2.5. Following the general flow up to this point, the beam contains particles which interact with the various detectors in specific ways. These interactions cause the detectors to generate electronics signals. The TDAQ system allows researchers to select and record these signals for later analysis. The general overview of such a system is shown in figure 2.4.1.

The first element is the trigger which identifies events of interest. The trigger provides a decision to the DAQ about whether to digitize and record or discard particular signals. The DAQ then reads-out signals from the detectors, forms complete events, allows for run controls and monitoring, and stores raw data for later analysis.

As shown in figure 2.4.1, the initial output from the chosen detectors will go both to the trigger logic and to a delay. In order for the DAQ to function correctly and for the produced data to be interpretable, it is essential that while the DAQ is processing an event trigger, no other event trigger can be generated, and that we precisely know for which fraction of time the system was processing triggers and therefore in a “dead” state. This is known as the “dead time.” To do this we generate a “busy” signal. The trigger feeds into a subsystem which generates and distributes this “busy” signal.

As the figure 2.4.1 shows, a signal arrives, is split, and sets the output of an AND gate to TRUE assuming the output of the busy logic latch is FALSE, i.e. the system is not busy. The output of the AND gate then starts an analog-to-digital converter (ADC) in this case. The delayed analog signal then arrives at the ADC and can be converted to a digital signal that can go through further processing and be stored.

An ADC is required because the signals from the detectors and the frontend electronics

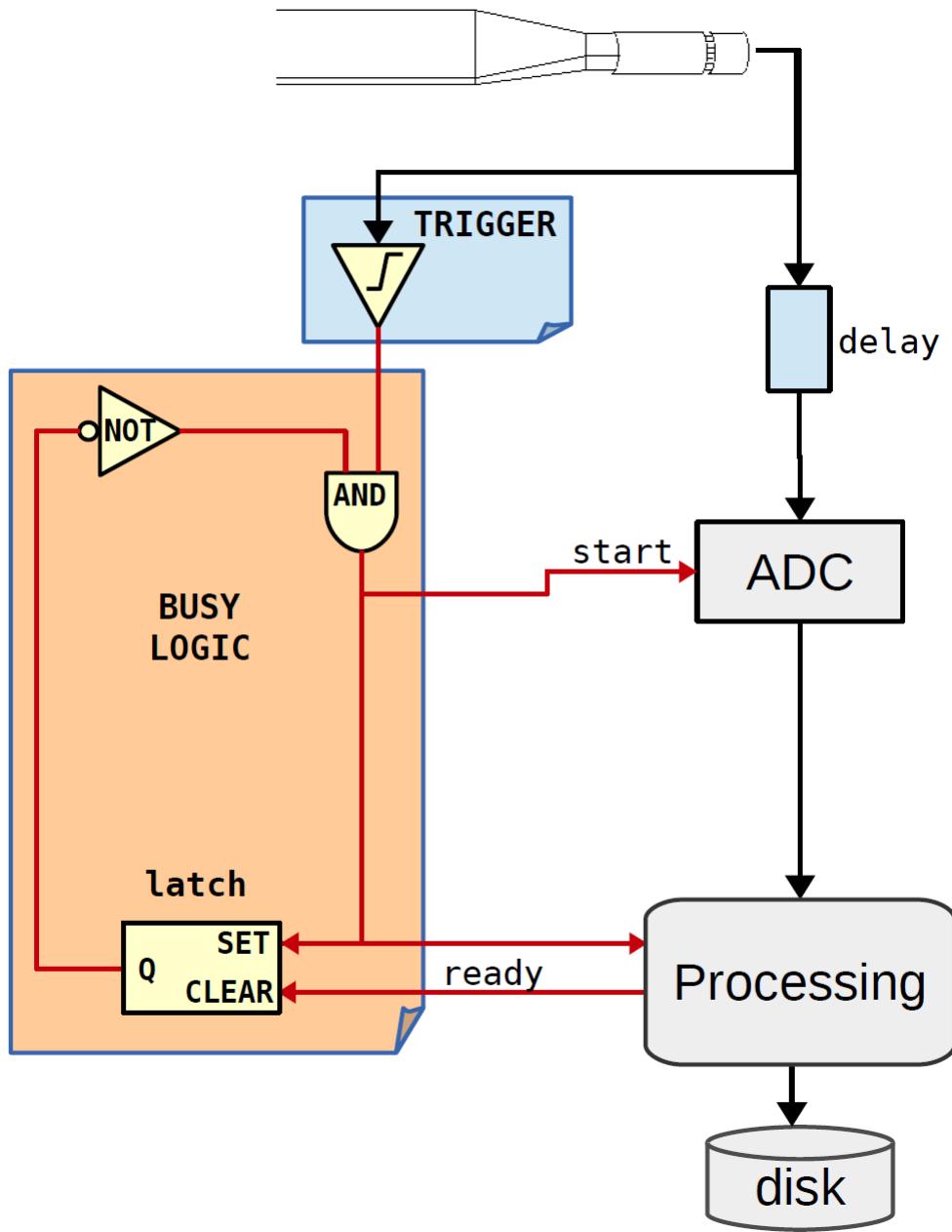


Figure 2.4.1: General overview of elements needed in TDAQ system [4]. The trigger starts a “busy” signal, which persists as long as any of the data acquisition system elements are still processing data. Once all of the DAQ systems are ready, this busy signal is cleared. While the busy signal remains in place, the formation of new Trigger signals is blocked.

connected to the detectors are analog. All processing must be done in the digital domain. An ADC converts an analog signal to a digital signal by assigning voltage levels to a digital value encoded in a series of binary bits (the larger the voltage, the larger the binary number). While figure 2.4.1 only depicts an ADC, most TDAQ systems contain charge-to-digital converters (QDCs) and time-to-digital converters (TDCs). QDCs integrate the analog pulses over a specified time range in order to extract charge (given a known resistance that a measurement is taken across, the voltage of a pulse can be interpreted as current whose integral is charge). QDCs provide valuable energy-related information about the analog pulses. TDCs provide the time between the edges of an input signal or signals, often by using a counter with a start and stop time defined by these edges or by using a signal arrival edge compared to a reference signal arrival edge [19]. These provide valuable timing information required by many experiments.

2.4.2 MUSE TDAQ

The trigger logic is experiment-dependent, and the logic for MUSE is shown in figure 2.4.2. MUSE uses a two level trigger. Level 1 uses four input signals: RF¹, Beam Hodoscope, Scattered Particle Scintillators (SPS), and VETO. The signals combine as shown in the figure to form PID logic, pre-target scattering logic, and scattered event logic. The outputs of Level 1 trigger logic along with VME² busy signals from the beam line detectors, Gas Electron Multipliers (GEM), and SPS feed into the final trigger logic which forms the final trigger decision. An event is recorded only if all of the following conditions are met:

1. The beam particle is identified by timing in the BH detector as an electron or a muon and no particle is identified as a pion.

¹The RF signal refers to the frequency of the radio frequency generators (Klystrons) which drive the accelerator cavities. Particles come in bunches at a specific point in the cycle. Here, RF is 50.6 MHz.

²The VMEbus is one of several standard systems used in nuclear and particle physics experiments for signal interfacing between modules in the TDAQ system. The VME standard sets up a protocol for module addressing and inter-module communication along with power distribution via a backplane in the VME crates [20].

2. The SPS reports a signal consistent with a scattered particle originating in the target.
3. The veto detector does not fire.
4. The system busy signal is not active.

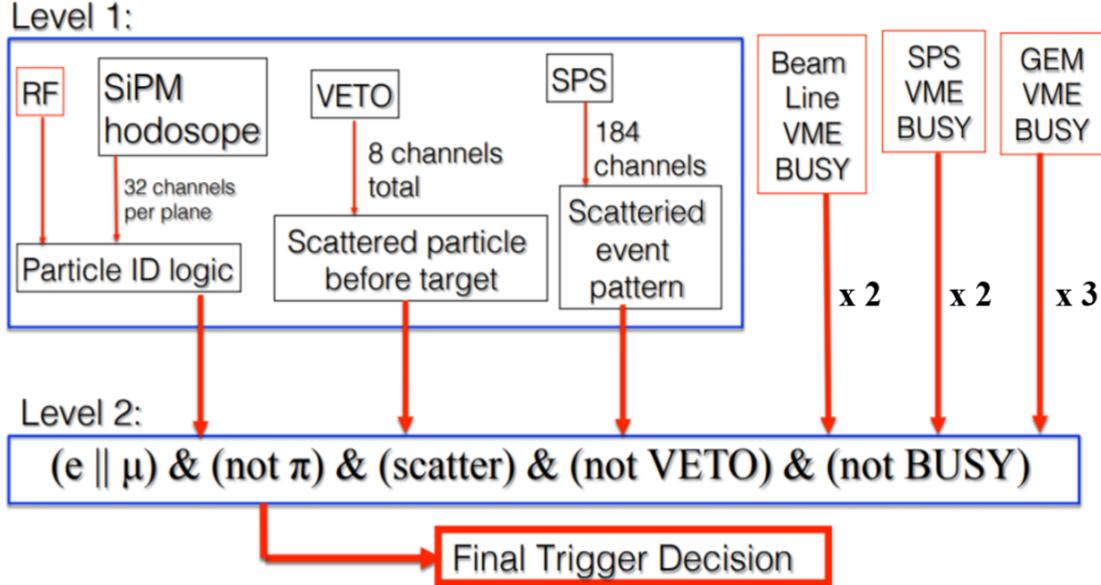


Figure 2.4.2: Two level trigger. The Level 1 trigger contains several triggers that determine PID and scattered event. The Level 2 trigger collects information from Level 1 trigger in order to form a decision [5].

2.5 Data Analysis Package

Data analysis is performed once data has been extracted via the TDAQ system and stored on local hard-drives in Midas format.³ MUSE uses the open source package called “cooker,” which handles the analysis with various plugins, which are in turn called using “recipes.” The first step in analysis is to use one of the recipes to convert the Midas data file

³Midas formatting is one of the popular data formats used in frontends for nuclear and particle physics experiments. The format includes an event identification and time stamp among other items [21].

to a ROOT⁴ file. Once in ROOT format, one can choose the appropriate recipe to perform the desired analysis [23].

This system for analysis is modular in design and allows for easy addition of new plugins and recipes to extend analysis capabilities.

⁴ROOT is a software toolkit maintained by CERN and used for data analysis [22].

Chapter 3

Current Work

3.1 SiPM Assembly & HV Calibration Software

3.1.1 Detecting Principle

BH and BM beam line detectors consist of silicon photomultipliers (SiPM) (seen in figure 3.1.1a) attached to plastic scintillator (seen in figure 3.1.2b). When charged particles pass through plastic scintillators, they deposit energy which excites the complex molecules of the scintillator, resulting in the production of photons of visible or UV light. In MUSE's BH and BM scintillators, the light has a wavelength of approximately 404 nm. These SiPMs have high efficiency for collection of light around 430 nm (as seen in figure 3.1.1b) which closely matches the most frequently produced wavelength of light from the scintillator.

3.1.2 Assembly

Both the BH and BM detectors consist of planes of scintillators with SiPMs attached to both ends. After the scintillator has been cut, the ends must be lightly sanded in order to provide a surface for optical glue to adhere to. Then, SiPMs can be glued, one at a time, to the ends of the scintillator while the scintillator sits in an assembly frame, as can be seen

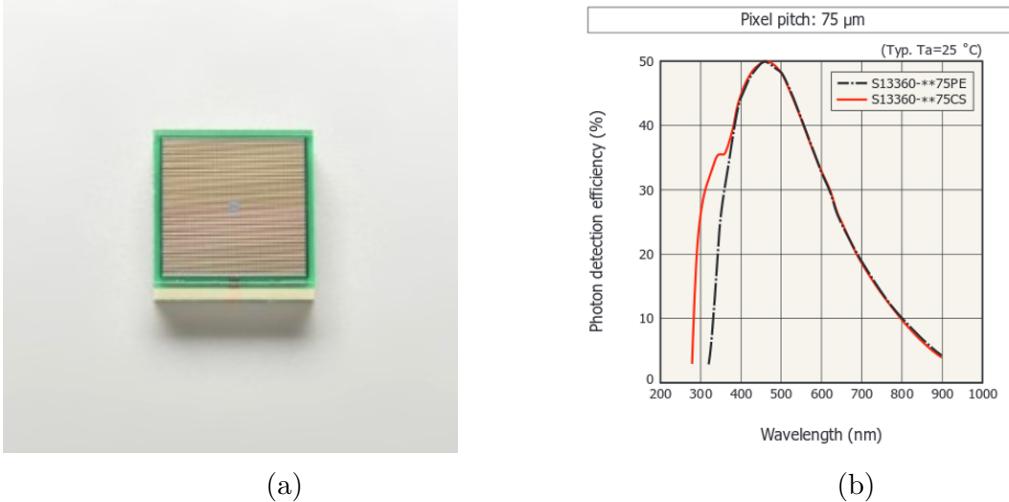


Figure 3.1.1: (a) Hamamatsu SiPM [6]; (b) SiPM photon detection efficiency [7].

in figure 3.1.2a. Once both sides of the scintillator have SiPMs, the SiPMs are soldered to small printed circuit boards (PCBs) with LEMO connector outputs, as seen in figure 3.1.2b. Once each channel is constructed, it is removed from the assembly frame and placed in the main detector frame for either the BH or BM detectors, as seen in figures 3.1.2c and 3.1.2d respectively.

The final timing resolution of the detectors is the most important feature as the BH and BM detectors are used mainly for time-of-flight based particle identification (PID). An example of time resolution, which needs to be below 100 ps, is shown in figure 3.1.3. This plot shows the time difference between when the signal arrives at the SiPM on one side of the scintillator and the SiPM on the other side of the scintillator while tracks are constrained to a path through the point-like area in the center of the scintillator.

3.1.3 HV Calibration Software Development

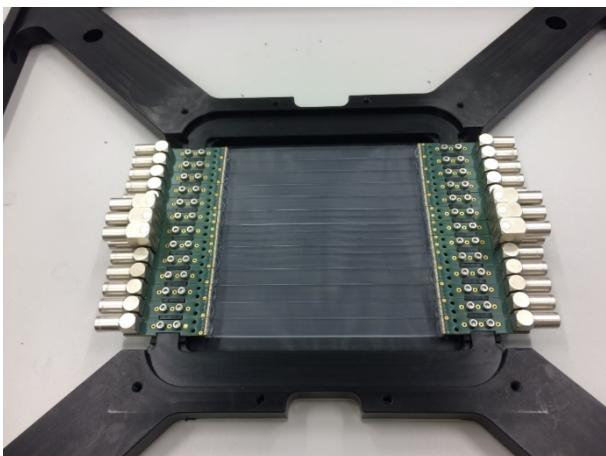
The SiPMs in the BH and BM detectors can degrade over time, both naturally and because of heavy exposure to beam. In order to keep the performance of the detectors matched among MUSE run times and in order to obtain good channel-to-channel time resolution on



(a) Step 1: two SiPMs glued to ends of thin scintillator plastic.



(b) Step 2: four SiPMs on two scintillator pieces soldered to electrical readout structure.



(c) Final: single plane of Beam Hodoscope detector fully assembled.



(d) Final: single plane of Beam Monitor detector fully assembled.

Figure 3.1.2: Steps for full assembly of SiPM hodoscope planes used for Beam Hodoscope and Beam Monitor [6].

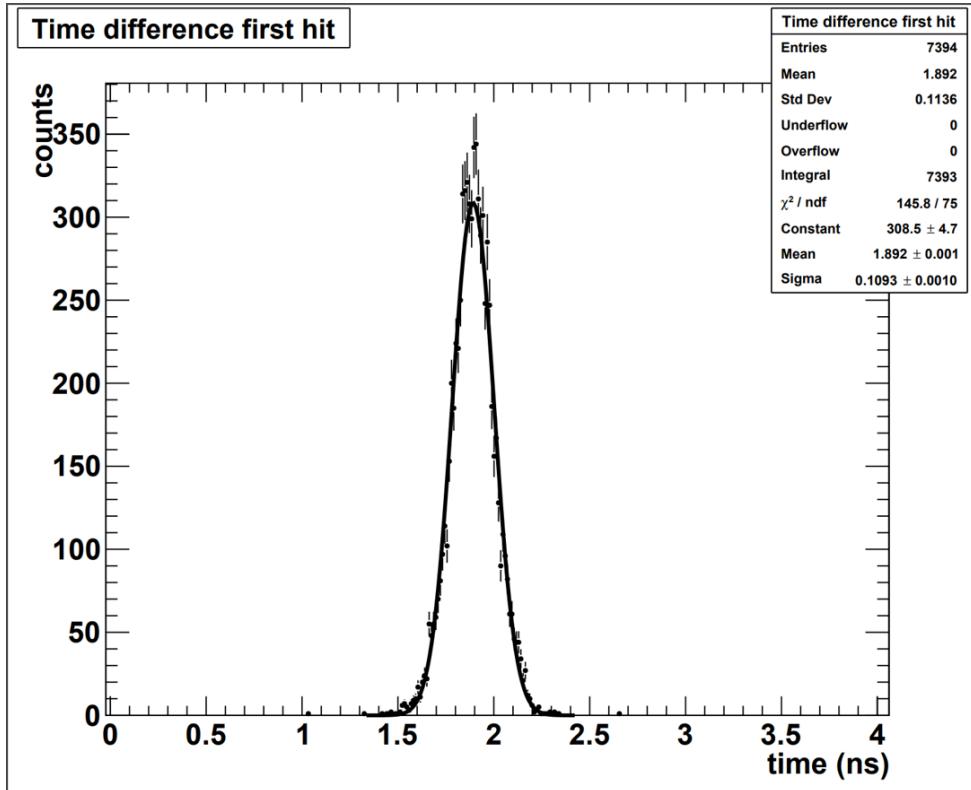


Figure 3.1.3: SiPM time resolution. Dividing the standard deviation by two yields the timing characteristics for individual SiPMs. Here $\frac{113.6 \text{ ps}}{2} = 56.8 \text{ ps} < 100 \text{ ps}$ [6].

the BH and BM detectors, it is imperative to keep the signal size of the detectors similar, which means adjusting the high voltage to compensate for any losses in gain. To do this, a hardware calibration must be performed to measure the pulse size and, if necessary, adjust the high voltages so that the calibration factors (CF), the constant of proportionality in the relationship between the ADC channel and energy deposit, remains even across the detector. This is especially important as different detector elements are exposed to different beam conditions, with more central elements receiving a higher beam dose.

This calibration factor, related to the inverse of the gain, is used to inspect the performance of these detectors and is provided by the manufacturer of the detectors. CF is defined as the energy deposited per charge-to-digital converter (QDC) channel and directly relates to the difference between where the peak of a signal is relative to the minimum charge required to power the electronics (pedestal), as seen in equation 3.1.1. The CF has energy normalized to one because the energy deposited per detector channel is the same given the same source, same source location on each channel, and same geometry of scintillator.

With CF, the performance of the detectors can be evaluated across runs. In order to perform calibration, one must account for detector response based on the high voltage (HV) supplied to the detectors. Figure 3.1.4 shows how QDC spectra shift to higher values (increase in magnitude) with an increased HV supply. This relation depicts how, with the same energy deposited in the detector, one can affect detector response with a change in HV, which translates to shifting the peak of the signal in or out, changing the value of CF.

Therefore, the CF for the detectors can be checked at the beginning of a run, and, to keep this factor matched across runs, the HV for each detector channel can be adjusted accordingly.

$$CF = \frac{Energy(=1)}{Peak - Pedestal} \quad (3.1.1)$$

In order to extract CF from QDC spectra and thus motivate how to alter HV supplied to

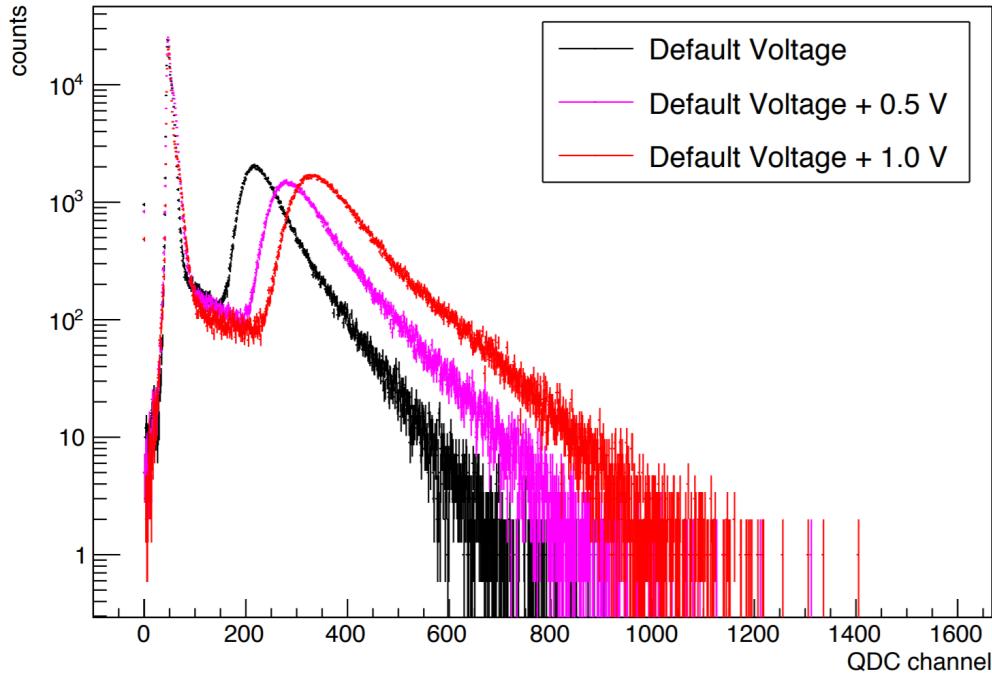


Figure 3.1.4: QDC spectra for Beam Hodoscope. Higher QDC channels means more charge collected. As voltage increases, the gain increases, and the peaks shift to the right [5].

the detectors, a software package was developed. The software package plots histograms of QDC values and applies fits in order to determine peak position of the signal and pedestal position from the electronics.

Each QDC spectra consists of a combination of peak, pedestal, and background as can be seen in figure 3.1.5. To fit the QDC spectra, a fit was chosen for each constituent part as shown in equation 3.1.2.

The general technique for choosing fits consisted of trial-and-error with various fitting functions and parameters, all motivated by a combination of spectra shape and underlying physics. The fitting functions are applied to the spectra with an open source software package called ROOT [22]. For a given function, initial parameters were used along with a five to ten percent allowance for ROOT to optimize these parameters. These initial parameters are derived from the spectra itself (such as mean in a particular range) such that the fitting

procedure would be generalized to all QDC spectra. A particular fit and accompanying parameters was deemed acceptable via a visual test of the fit. A more rigorous test of fits using a Chi-squared test was not possible for these spectra as the fitting procedure needed to be automated, the QDC spectra varied widely in shape and magnitude from channel to channel, and because (as will be seen) the fitting functions were not, strictly, simple.

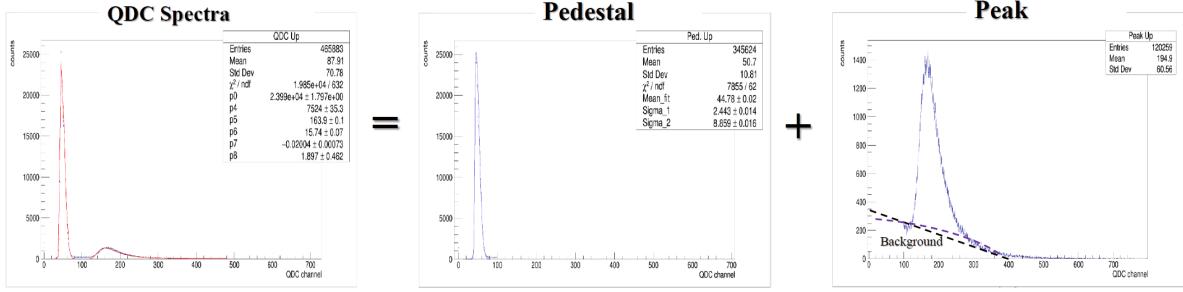


Figure 3.1.5: Breakdown of QDC spectra consisting of pedestal and peak with background underneath.

$$Fit = Background + Pedestal + Peak \quad (3.1.2)$$

The pedestal is fit with an asymmetric Gaussian ($AG(x)$) distribution. Equation 3.1.3 shows the general equation for a Gaussian function ($G(x)$) as implemented in ROOT where $p0$ is the scaling factor, $p1$ is the mean, and $p2$ is the standard deviation [24]. The asymmetric Gaussian function (shown in equation 3.1.4) combines two Gaussian functions with the same amplitude scaling factor and mean but with different standard deviations, where one standard deviation is used for values below the mean and another standard deviation is used for values above the mean.

$$G(x) = p0 * e^{\frac{-(x-p1)^2}{2*p2^2}} \quad (3.1.3)$$

$$AG(x) = p0 * \begin{cases} e^{\frac{-(x-p1)^2}{2*p2}} & x \leq p1 \\ e^{\frac{-(x-p1)^2}{2*p3}} & x > p1 \end{cases} \quad (3.1.4)$$

For the peak, the fits that best provide location of the most probable value of the peak are a Gaussian and a Landau function. The same Gaussian function as shown in equation 3.1.3 was used but with initial parameters chosen for the peak. A Landau function ($L(x)$) was also tried as it is a function well known for modeling deposition of energy in matter by charged particles. The Landau function is implemented in ROOT but takes the form as shown in equation 3.1.5 where $p0$ is a scaling factor, $p1$ is the most probable value, and $p2$ is a measure of the width of the distribution [24–26].

$$L(x) = \frac{1}{p0 * 2\pi i} \int_{c-i\infty}^{c+i\infty} e^{\frac{s(x-p1)}{p2} + s*\log s*p2} ds \quad (3.1.5)$$

Backgrounds are notoriously difficult to fit nicely as the exact shape and distribution can vary widely across channels. No in-depth background studies were carried-out for this project. For the background, an exponential function ($\text{Exp}(x)$), a linear fit ($P1(x)$, polynomial degree one), and a quadratic fit ($P2(x)$, polynomial degree two) were tried (as seen in equations 3.1.6, 3.1.7, and 3.1.8, respectively) [24]. These fits, while not perfect, were good enough for the purposes of extracting peak and pedestal positions.

$$\text{Exp}(x) = p0 * e^{p1*x}; \quad (3.1.6)$$

$$P1(x) = p0 + p1 * x; \quad (3.1.7)$$

$$P2(x) = p0 + p1 * x + p2 * x^2. \quad (3.1.8)$$

Examples of two fits are shown in figure 3.1.6. In the figure, both fits use an asymmetric Gaussian for the pedestal, but one fit uses an asymmetric Gaussian for peak and linear for background while the other fit uses Landau for the peak and exponential for the background. These fits meet the visual qualification for extracting peak and pedestal positions to a precision high enough to calculate CF.

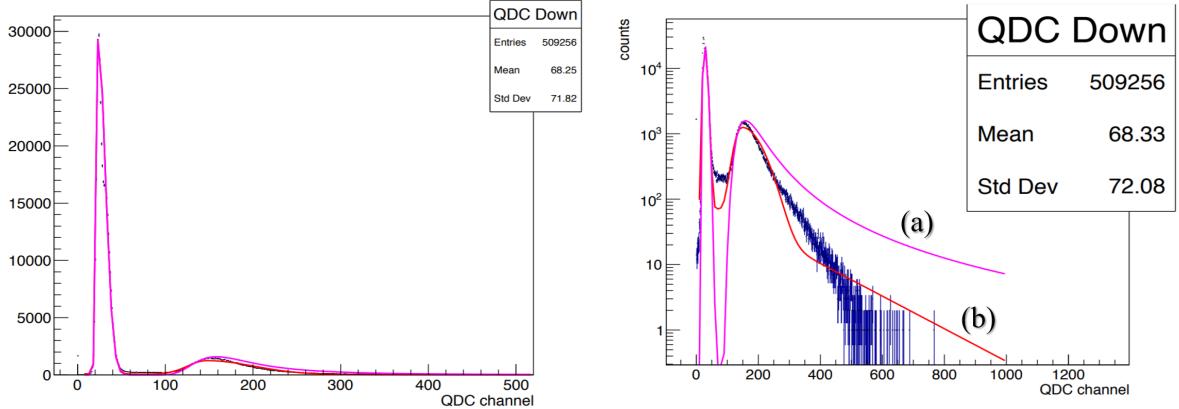
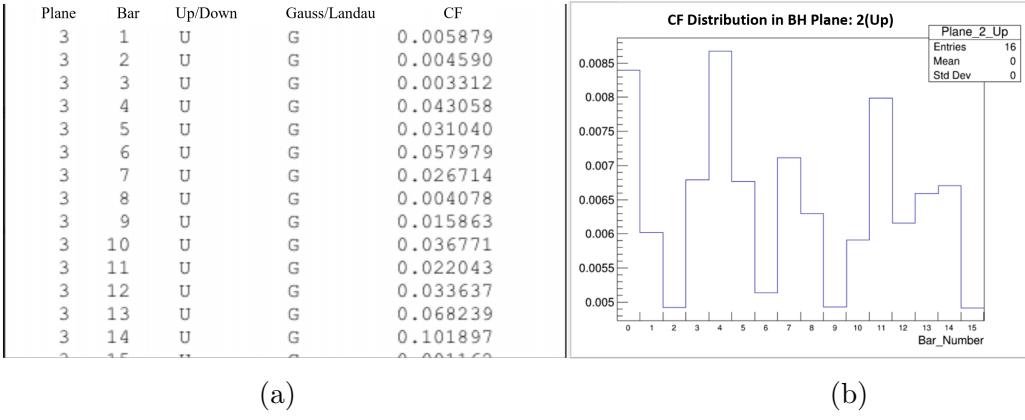


Figure 3.1.6: Linear (left) and logarithmic (right) scales for QDC spectra with fits: (a) asymmetric Gaussian for peak and linear background; (b) Landau for peak and exponential for background.

For convenience, the software package outputs a table and plots containing information about all the SiPMs in the detectors and their CFs. This allows for easily accessible information for how to adjust the voltages to match CFs. Figures 3.1.7a and 3.1.7b show the CFs for several channels.

3.1.4 Conclusions

The SiPM plane construction and HV calibration software portion of the contributions to MUSE involved assembling detector channels for the BH and BM detectors and developing an analysis recipe for the cooker used for matching calibration factors across all channels in these detectors. The successfully developed software determines CF by extracting peak and pedestal positions from QDC spectra and can thus indicate how to adjust voltage to keep CFs matched.



(a)

(b)

Figure 3.1.7: Output from HV calibration software: (a) File output used for extraction of calibration factor from QDC data; (b) Histogram output of calibration factors.

Through this work, knowledge pertaining to the parts and techniques involved in SiPM-based detector assembly was gained. These techniques included how to prepare scintillator, how to properly glue scintillator to a detector, how to assemble detector channels in a hodoscope plane, and how to control and operate the BH and BM detectors. Through the software package development and data analysis, better understanding of typical fitting procedures and information extraction from data was acquired (The code for the software analysis is presented in Appendix B.).

The results from this work have been presented at a MUSE collaboration meeting.

3.2 STT Gas Mixture Optimization Studies

3.2.1 Introduction

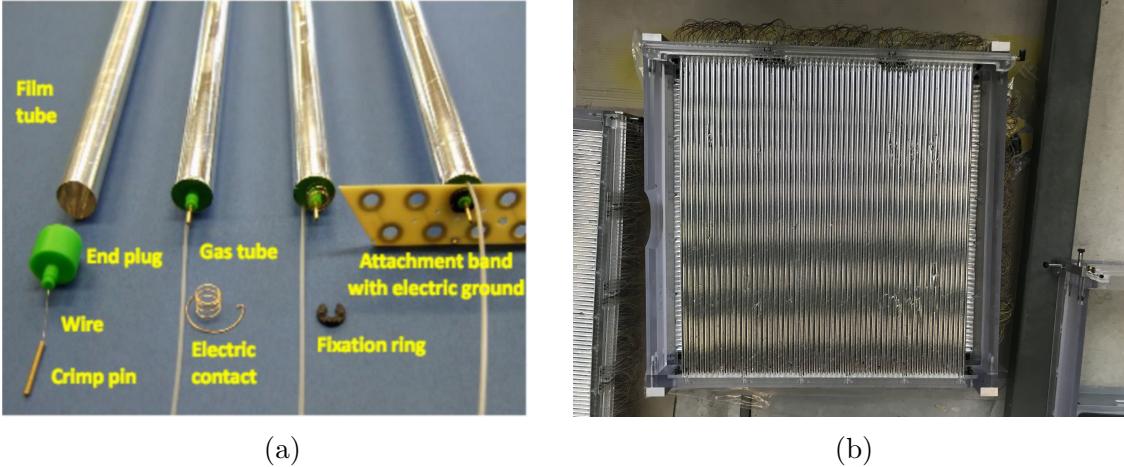


Figure 3.2.1: (a) STT design [5]; (b) One wall of STT channels [8].

STT detectors are scattered particle detectors used for positional measurements of the trajectory of charged particles with a precision less than $150 \mu m$. As seen in figure 2.3.1, several walls of STT detectors (see figure 3.2.1b) in vertical and horizontal arrangements are used. Particles are located using a combination of vertical and horizontal straws firing.

As seen in figure 3.2.1a, the STTs are made of a long, thin wire surrounded by aluminized-mylar tubing inflated to 1 bar over atmospheric pressure. The wire is kept at a voltage between 1500 and 2000V. The inflation gas is a mix of Argon and Carbon Dioxide. An incoming charged particle ionizes the Ar gas, releasing electrons that move toward that high voltage central wire, creating a signal. The CO₂ uptakes electrons to prevent an avalanche, where the number of electrons produced by the ionizing particle is no longer proportional to the signal produced.

Both the voltage on the central wire and the proportion of the Ar to CO₂ affect the signal produced by the detector and to what extent avalanching occurs. Typically, a ratio of 87/13 of Ar to CO₂ is used to optimize the performance of STTs. The purpose of this work was to

test various other ratios of gases to find the mixture to best optimize the detection efficiency of the detector.

3.2.2 Work

Testing gas ratios required setting up a gas mixing system and inspecting detector response at a range of voltages. The gas mixing system was constructed using flow rate controllers attached to two separate tanks of Ar and CO₂. These flow rate controllers fed into a gas-combining device that took two gas inputs and produced one thoroughly mixed gas output. By adjusting the flow rate for each gas, one could choose the desired ratio. Pressure was adjusted using a pressure valve to keep the straws inflated to 2 atm.

The complete testing setup is shown in figures 3.2.2 and 3.2.3. Figure 3.2.2 shows the schematic overview of the test setup with gas mixing and gas input coming from the left and with HV input and signal output on the right. The signal output wire is connected to a frontend discriminator and to a TRB3 for data collection. Figures 3.2.3a, 3.2.3b, and 3.2.3c show the physical setup in the lab.

With limited data taking time available, three Ar to CO₂ proportions were tested: 87/13, 90/10, 93/7. The results of count rate versus voltage for these gas proportions are shown in figure 3.2.4. Ideally, a graph of rate versus voltage would increase approximately linearly up to a stable region and then would increase linearly until a breakdown region, followed by a rapid increase in rate due to the avalanche effect.

3.2.3 Conclusions

The results showed that the 93/7 ratio of Ar to CO₂ is too much Ar as the avalanching occurred too easily since there was too much Ar available for ionization. In the voltage region of interest, both 87/13 and 90/10 ratios were reasonable. Further testing was needed to make a decision on the optimum ratio.

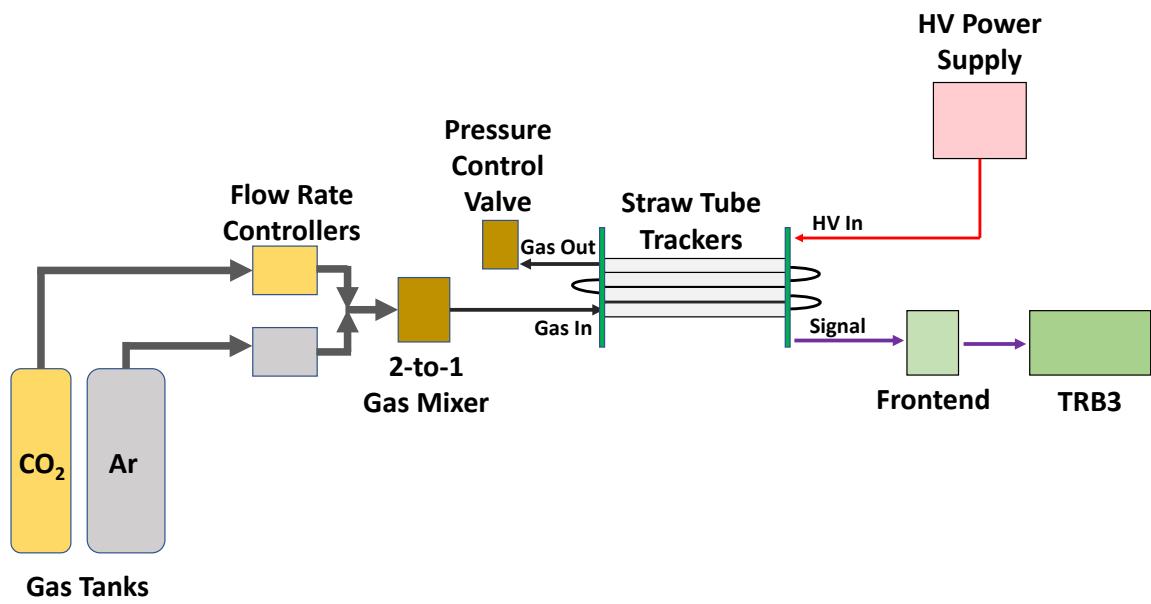


Figure 3.2.2: Schematic overview of lab test setup for Ar to CO_2 gas proportion tests for STT detector performance optimization.

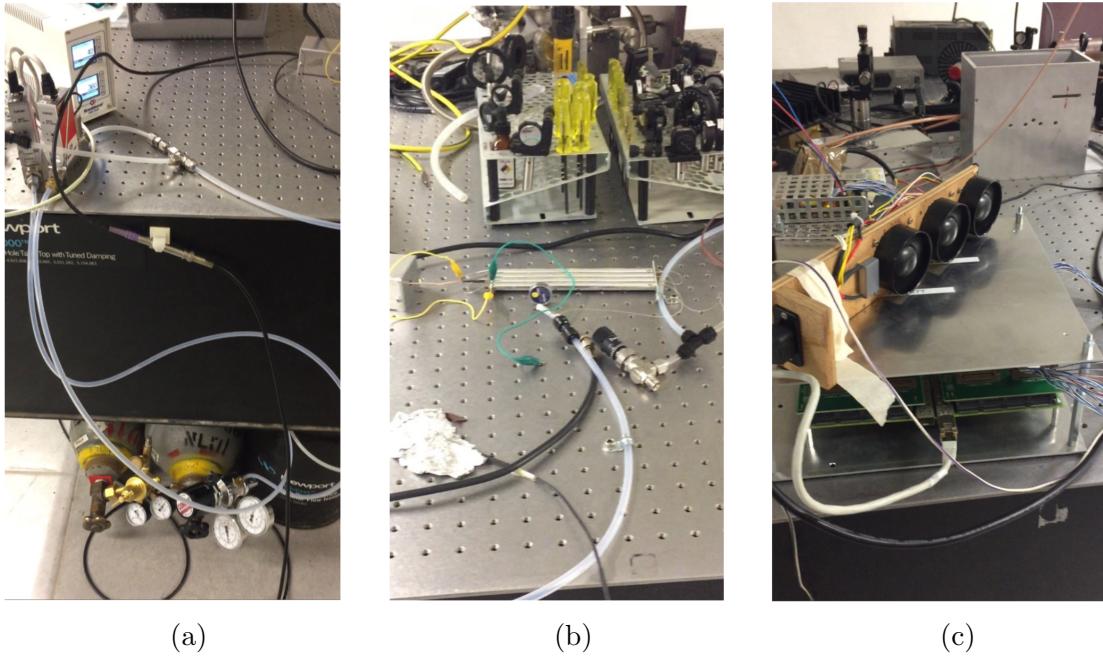


Figure 3.2.3: Lab test setup for Ar to CO₂ gas proportion optimization for STT detectors: (a) Gas mixing system; (b) STT test setup with radioactive source; (c) Data readout setup with PaDiWa and TRB3.

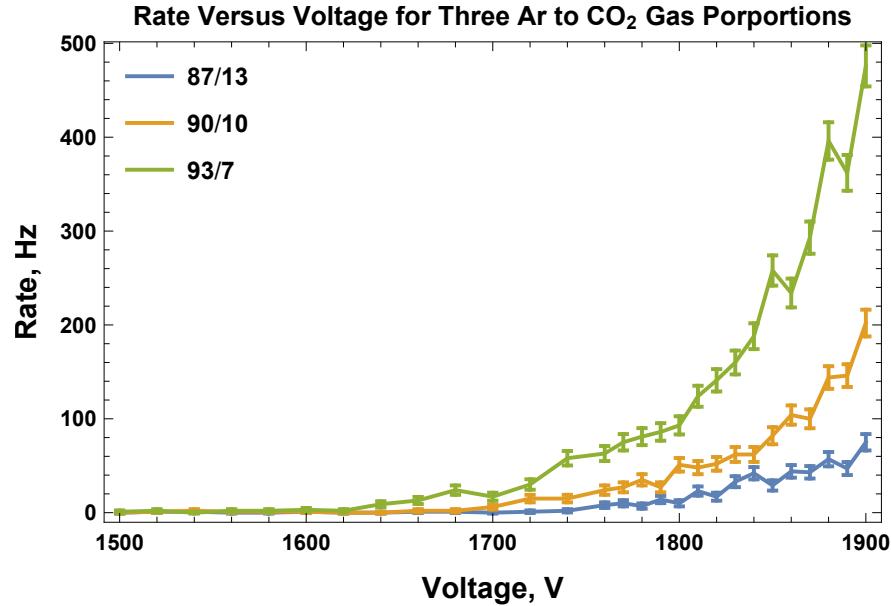


Figure 3.2.4: Rate versus voltage for different gas proportion mixtures of Ar and CO₂. The source used was Pa-234 and was placed approximately 28 cm from STT detectors. Test setup is shown in figure 3.2.3

3.3 PaDiWa FPGA Design Modification for TDC Test Setup

3.3.1 Introduction

For the TDAQ system, MUSE uses the TRB3 universal readout boards, which contain five field programmable gate arrays (FPGAs) as seen in figure 3.3.1a. The four FPGAs near the corners of the TRB3 are programmed as time-to-digital converters (TDC) and collect timing data from all the detectors in the system.

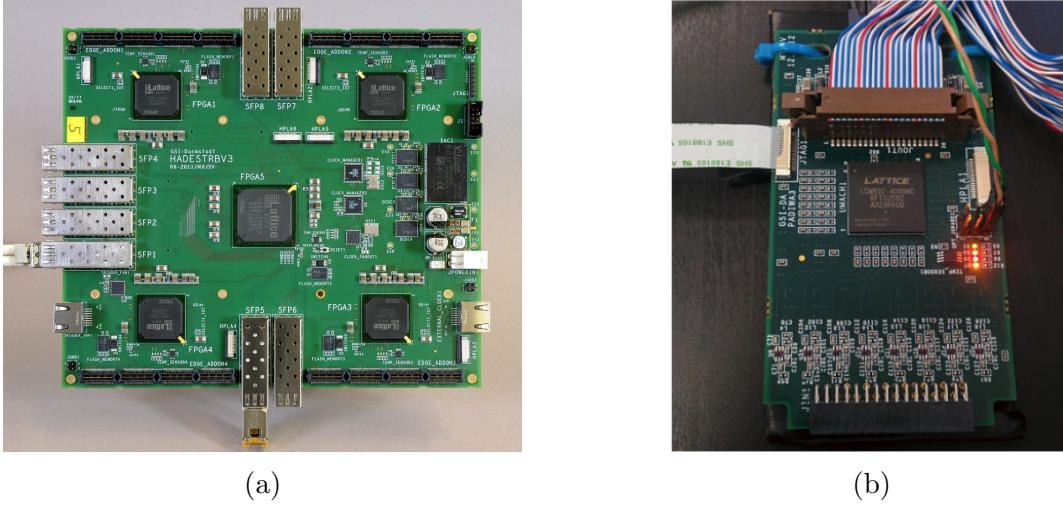


Figure 3.3.1: (a) TRB3 with four TDCs, one in each corner; (b) PaDiWa board with Lattice FPGA [5].

In order to send signals into the TDCs, the signals must be discriminated first by either a MCFD-16 or a PaDiWa discriminator (see figure 3.3.1b), which contain 16 channels for converting analog signals to digital low voltage differential signals (LVDS) that go to the TRB3.

The purpose of this project was to modify a FPGA design on the PaDiWa to mimic the working detector system. This FPGA design would produce identical signals on all 16 channels of the PaDiWa going to the TRB3 such that one can check that the TDC registers on the TRB3 are receiving the correct values. In addition, accompanying software was developed in order to analyze data from the tested TDCs to check their performance.

3.3.2 Work

The FPGA design produces an 11 KHz signal with a pulse width of 30 ns¹ using a 133 MHz clock signal that the TRB3 sends to the PaDiWa via a flat-ribbon cable (see figure 3.3.4). In addition, the new FPGA testing sends an identical copy of the signal to the test pin output of the PaDiWa to play the role of the trigger for the testing. See figure 3.3.2 for the simulation results of the waveforms used to generate this signal.



Figure 3.3.2: Simulated waveform of generated clock signal showing 11 KHz clock signal waveform and zoomed-in capture of 30 ns clock pulse width.

Figure 3.3.3 shows the block diagram of the PaDiWa FPGA architecture. The architecture has standard included logic blocks and space for user defined logic. This project, called signal_gen_x16, is implemented in the user defined logic section.

The TDC register values and trigger channel value are shown in figure 3.3.5. Both the trigger and the 16 TDC channels show the correct 11 KHz signal generated by the FPGA on the PaDiWa.

Further analysis on the signals is performed using a recipe in the cooker that was developed for TDC debugging. Figure 3.3.6 shows a channel with good timing and a channel with bad timing. Each signal generated by the FPGA will have a rising and falling edge.

¹11 KHz is the maximum rate at which the TDCs can reliably collect data, and 30 ns is the average pulse width received from the detectors in the system.

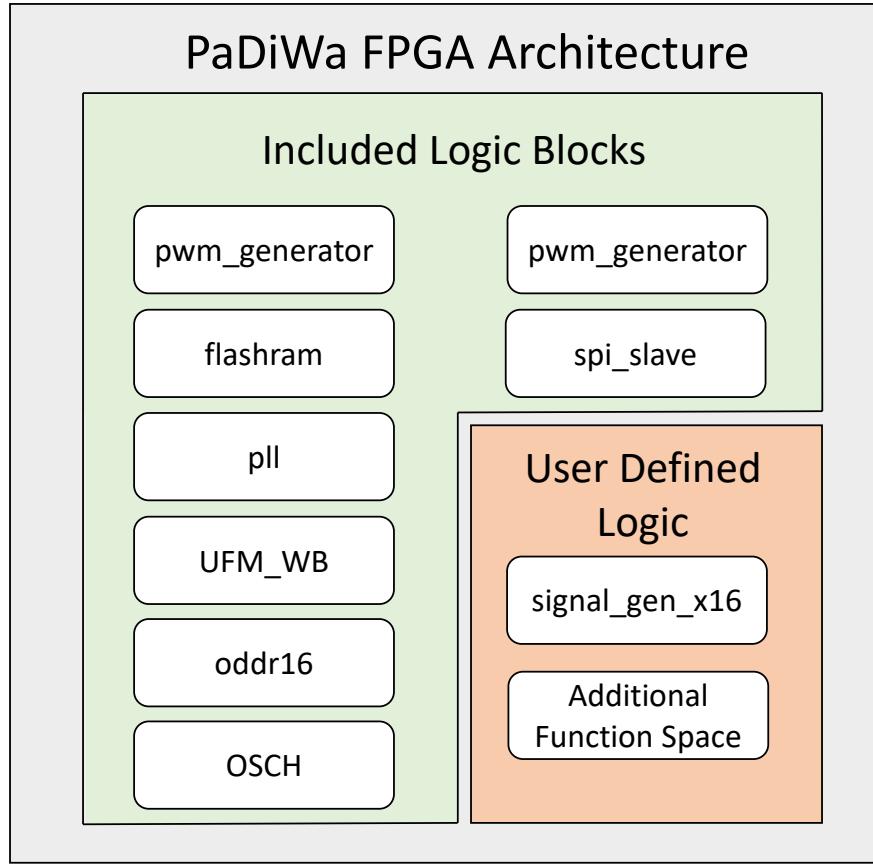


Figure 3.3.3: Block diagram of PaDiWa FPGA architecture.

If a TDC channel is working properly, there will be two distinct readings on the timing versus edge plot, indicating a rising edge followed by a falling edge. Figure 3.3.6a shows this distinctive edge characteristic of a good channel. Figure 3.3.6b shows a spread in timing that is characteristic of a faulty channel.

While timing versus edge plots show if individual channels are faulty, cross-channel resolution plots indicate if there is any cross-channel coupling or interference occurring or if one channel is handling a signal incorrectly (i.e., a TDC channel is faulty). The cross-channel resolution plot should be sharply peaked and have a small spread within tenths of a nanosec-

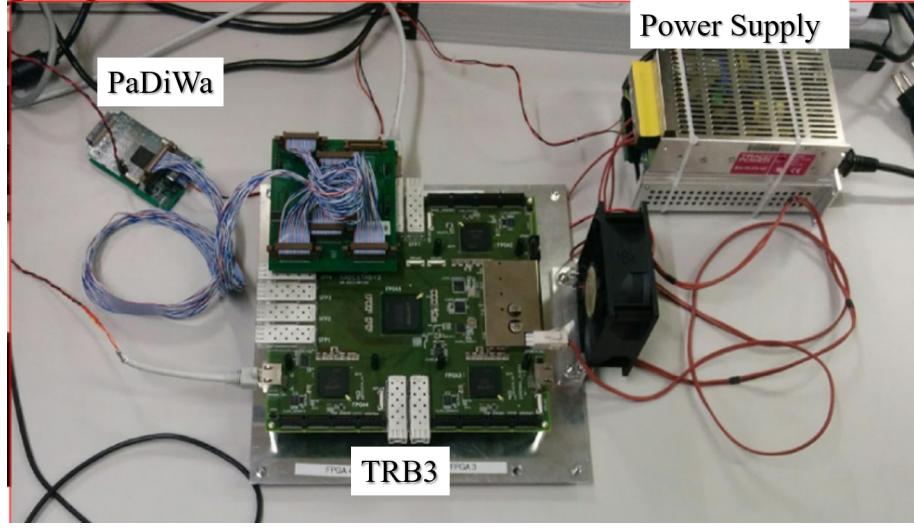


Figure 3.3.4: TRB3 connected to PaDiWa and power supply [5].

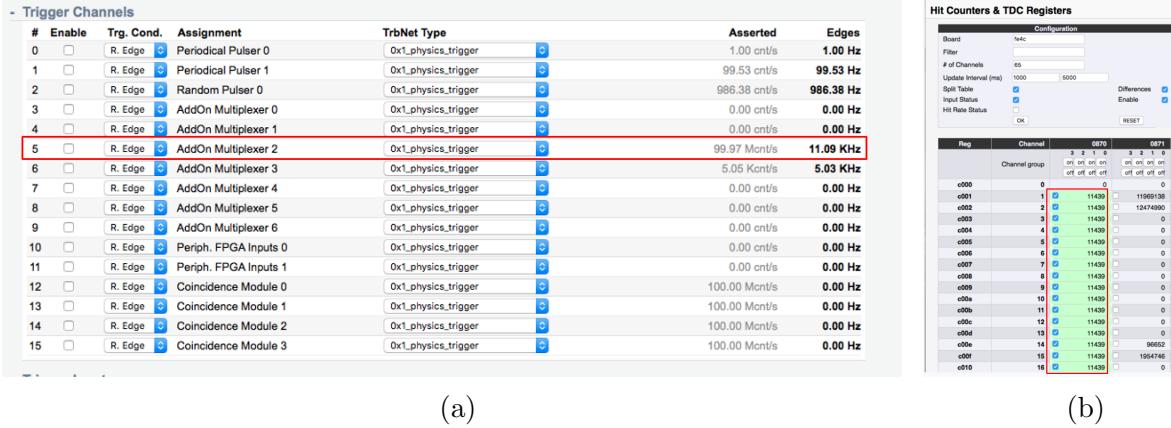


Figure 3.3.5: Trigger channel (a) and TDC register (b) values from TRB3 with PaDiWa providing test signal input [5].

ond, which indicates adjacent channels are receiving the same signal at the same time as they should be. Figure 3.3.7 shows a good channel and a bad channel. Figure 3.3.7a shows the narrow plot of a good channel, and figure 3.3.7b shows a plot of a bad channel with a lot of counts far from the peak.

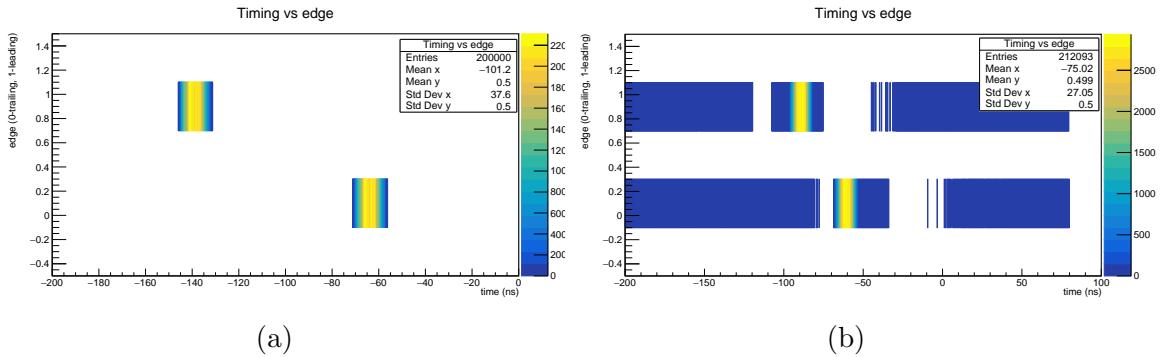


Figure 3.3.6: Timing vs edge plots: (a) good channel; (b) bad channel.

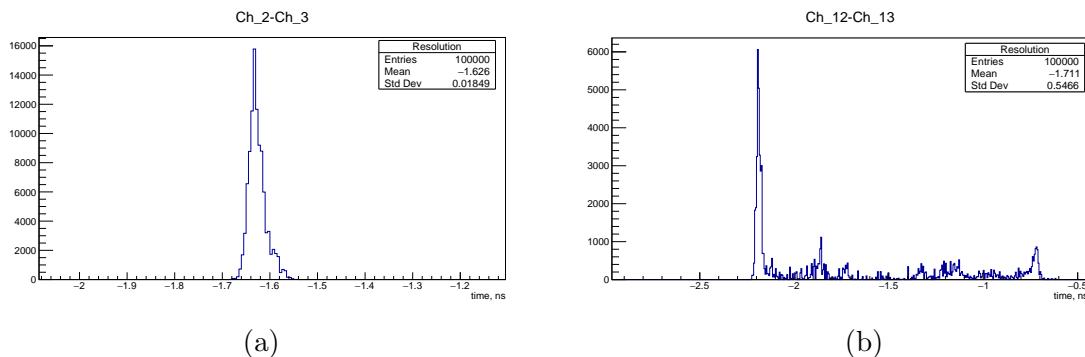


Figure 3.3.7: Cross-channel resolution plots: (a) good channel; (b) bad channel.

3.3.3 Conclusions

The successful implementation of the FPGA design allowed for more thorough testing of the TRB3s, which had been suffering from a manufacturing error on inputs used by the TDC registers. During beamtime in December 2018, 5376 TDC channels across 28 TRB3 boards were tested. Additionally, this project provided deeper exposure to MUSE’s TDAQ system and allowed for practice with hardware programming (The FPGA design code as well as a testbench are presented in Appendix [A](#)).

3.4 ECL-to-LVDS Converter PCB

3.4.1 Introduction

The Level 2 trigger logic for the MUSE DAQ, as seen in figure [2.4.2](#), involves BUSY logic which prevents new triggers being formed while a system is processing data. This additional data will be discarded to avoid further pile-up of signals and information in the pipeline. It is essential to handle this data explicitly by indicating that the system is busy and not accepting additional data; otherwise, there could be mismatches in dropped data. The Vulom4b board (see Appendix [E](#) for more information) produces the BUSY logic signal in emitter-coupled logic (ECL) format. However, the FPGA for the master trigger on the TRB3 only uses low-voltage differential signaling (LVDS). An ECL-to-LVDS converter is thus an essential hardware component for the TDAQ system, and this board must be able to handle BUSY signals up to 100 KHz rate. Figure [3.4.1](#) shows the top-level overview of placement of converter board in MUSE’s TDAQ system. The following section describes the design and design process of the ECL-to-LVDS converter board as well as testing and current usage.

3.4.2 Design

There were two stages of the design process for this project. The first was the design of a one-channel prototype board used for proof-of-concept. The final stage was an expanded and improved version of the prototype. The sections below will present the two versions and discuss the improvements from the first to the second. The prototype board discussion will be brief, and details about requirements and operations will be saved for the final design discussion.

3.4.2.1 Prototype

The first iteration of the board was designed in Eagle CAD schematic and printed circuit board design tool-set [27]. This prototype board was printed by GW's PCB print facility and is shown in figure 3.4.2. The prototype board met the needed criteria and allowed for a second iteration of the board to be developed with improved capabilities, including more channels, better signal isolation, and more care given to component placement and usability.

3.4.2.2 Final Design

The final board design consists of five channels along with power circuitry. Each channel is identical. This section will first provide a single channel description and then will describe full board design.

Single Channel Description

There are several input and output configuration options on the board depending on the particular desired use. A block-level overview of the single channel functionality is shown in figure 3.4.3. If using the ECL input option, the signal is first converted to transistor-transistor logic (TTL) through the ECL-to-TTL converter (MC100ELT25DG, [28]). The user then chooses via a switch (Switch 1 in figure 3.4.3) to convert the signal to LVDS or to output the TTL signal. If using the TTL input option, the user chooses via another switch

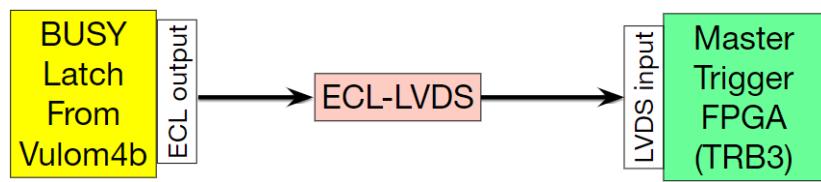


Figure 3.4.1: ECL-to-LVDS converter's placement in MUSE's TDAQ system.

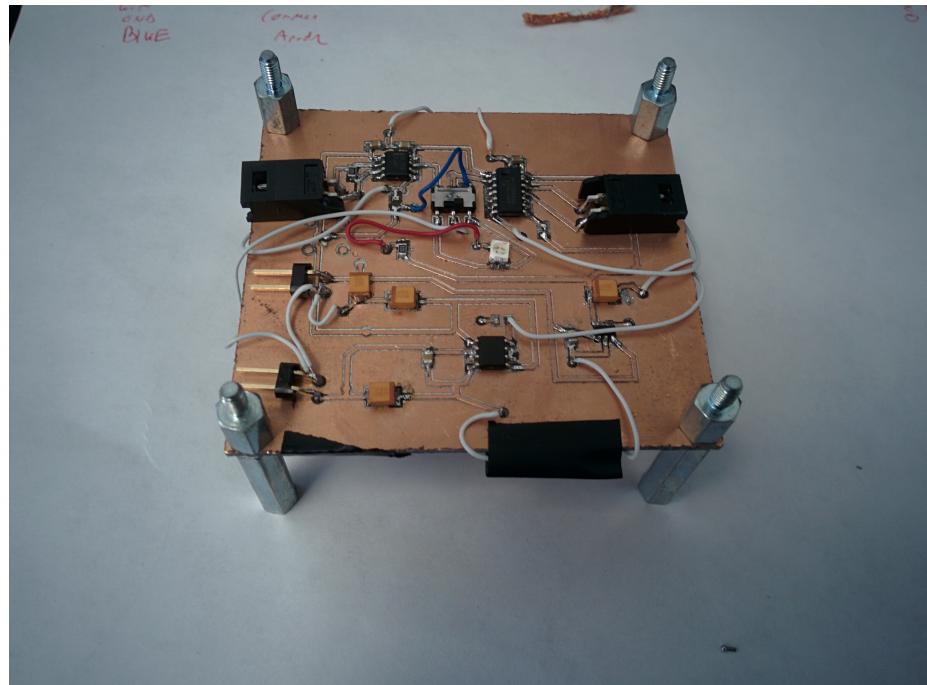


Figure 3.4.2: Prototype PCB with soldered components.

(Switch 2 in figure 3.4.3) to convert the signal to LVDS via the TTL-to-LVDS converter (DS90LV019, [29]) or to output the TTL signal. In this way, the user can convert ECL-to-LVDS, ECL-to-TTL, or TTL-to-LVDS.

A more detailed schematic-level overview of the single channel is shown in 3.4.4. The board-level overview of the design is presented in figure 3.4.5 and closely follows the logical flow of figure 3.4.3.

Each component in the single channel consumes power. The power requirements for each component are presented in table 3.4.1.

Table 3.4.1: Maximum current consumption on the positive and negative supply lines for one channel is presented here.

Component	Part Number	Current, mA			Num of Elements
		Pos	Neg		
ECL/TTL Converter	MC100ELT25DG	18	21		1
TTL/LVDS Converter	DS90LV019	48	0		1
Blue LED	APTR3216PBC/A [30]	20	0		2
Green LED	APTR3216MGC [31]	20	0		2
Total Current per Channel (mA)		106	21		

Full Design

In order to balance convenient channel spacing, board size, and board-usefulness, five channels are implemented. These five channels along with the power circuitry constitute the final board. The components require positive and negative 5V power supplies, and voltage regulators are used in order to stabilize the supply input voltages, retain a low-noise system, and allow for margin in the off-board power supplies.

The power circuitry is shown in figure 3.4.6, and the total current consumption of the board is presented in table 3.4.2.

Beyond the power requirements, signal quality was the most important design consideration. The final design consists of six layers, which allow for isolation of single-ended (TTL) and differential (ECL and LVDS) signals. This isolation is accomplished by placing these

Table 3.4.2: Total current consumption for the design is presented here.

Component	Part Number	Current, mA			Num of Elements
		Pos	Neg		
Red LED	APT3216LSECK [32]	2	0		1
Yellow LED	APT3216LSYCK [33]	0	2		1
Single Channel		106	21		5
Total Current (mA)		532	107		

single-ended signals on one plane and differential signals on a separate plane. Additionally, the four remaining planes were used for power and ground and further served to isolate the signal types (see Appendix D for additional details). Additionally, the impedance of all the traces is matched to 50Ω to reduce reflections. Moreover, an automated routing tool was used in the final design to route differential signals to ensure perfectly matched trace lengths (keeps differential signals in phase).

The final design also implements a small circuit on the ECL input side to supply a balancing bias to the input. This places the input voltage closer to the center when no signal is applied such that a true input signal can more easily swing the input in the correct direction.

The circuit also aids the pull-down resistor inside the chip that drives a low signal when no input is provided. This bias is necessary because a floating input could cause oscillations of the output signal from the ECL-TTL conversion chip which in turn would consume more power and potentially induce noise.

The schematic and PCB layouts were completed using KiCAD [34]. The boards were manufactured and assembled using Eurocircuits [35]. The complete schematic for the design is shown in Appendix C.

3.4.3 Testing

The design was tested for both the TTL-to-LVDS operational mode and the ECL-to-LVDS operational mode. Initial tests were conducted at PSI to ensure the design worked.

More detailed analyses were conducted at GW in order to inspect signal quality at frequencies above 100 KHz.

The test setup is shown in figure 3.4.7. Both testing configurations are shown. The TTL-to-LVDS test uses a single output from the function generator connected to a LEMO input connector on the board. The ECL-to-LVDS test uses two outputs from the function generator configured to match ECL logic standards. The LVDS output for both configurations measures a voltage across a resistor using two oscilloscope probes.

The tests included jitter analysis, frequency characterization, rise and fall time characterization, and positive and negative duty cycle characterization.

The jitter analysis was performed using the Tektronix MS05204 [36] Oscilloscope's DPO-JET Essentials jitter and eye pattern analysis software package [37]. This test looks for differences in arrival time for the signal, and, ideally, the jitter would have a value of zero. Histograms for jitter analysis are shown in figure 3.4.8 with the TTL input configuration shown in 3.4.8a and with the ECL input configuration shown in 3.4.8b. For each plot, seven histograms are overlaid, showing frequencies from 10 KHz to 75 MHz. The large spread in low frequency measurements for the ECL configuration is a product of the testing environment limits. Because two outputs from the function generator were configured to emulate ECL differential logic, instabilities and mismatches occurred at lower frequencies which broadened the spread and thus could motivate further testing. Fortunately, this large spread did not occur at higher frequencies, where the histograms show a narrow spread. Further testing with true ECL matched drivers should show operation at lower frequencies to be at least as good as higher frequencies.

The results from the remaining tests are shown in table 3.4.3 and table 3.4.4 for the TTL and ECL input configurations, respectively. These tests calculated statistics to check that the correct output frequency was measured for a given input frequency, to check if rising and falling times remained below 4 ns, and to see how duty cycle responds to frequency. The

results show that the design should be operated at or below 50 MHz.²

Table 3.4.3: Statistics for frequency, rise time, fall time, positive duty cycle, and negative duty cycle measurements for TTL-to-LVDS conversion from 100 KHz to 75 MHz.

Input Frequency	Frequency [MHz]	Rise Time [ns]	Fall Time [ns]	Duty Cycle (+) [%]	Duty Cycle (-) [%]
100 KHz	0.1000001(4)	1.46(4)	1.59(6)	49.9919(3)	50.0081(3)
1 MHz	1.00000(4)	1.51(4)	1.66(6)	49.917(3)	50.083(3)
10 MHz	10.000(4)	1.51(4)	1.72(9)	49.10(3)	50.90(3)
25 MHz	25.00(2)	1.36(3)	1.58(5)	48.08(7)	51.92(7)
50 MHz	50.0(1)	1.52(4)	1.51(5)	45.2(1)	54.8(1)
75 MHz	75.0(2)	1.83(4)	3.56(8)	40.8(2)	59.2(2)

Table 3.4.4: Statistics for frequency, rise time, fall time, positive duty cycle, and negative duty cycle measurements for ECL-to-LVDS conversion from 100 KHz to 75 MHz.

Input Frequency	Frequency [MHz]	Rise Time [ns]	Fall Time [ns]	Duty Cycle (+) [%]	Duty Cycle (-) [%]
100 KHz	0.1000001(3)	1.40(3)	1.74(5)	53.080(2)	46.920(2)
1 MHz	1.0000(4)	1.42(5)	1.57(4)	52.40(4)	47.60(4)
10 MHz	10.000(5)	1.41(3)	1.69(4)	51.20(4)	48.80(4)
25 MHz	25.0(3)	1.88(5)	1.76(8)	52.34(9)	47.66(9)
50 MHz	50.00(5)	2.28(7)	3.1(1)	57.9(2)	42.1(2)
75 MHz	75.0(3)	3.56(8)	3.6(1)	52.1(4)	47.9(4)

3.4.4 Conclusions

The final design of the board exceeded requirements with the ability to operate up to 50 MHz instead of 100 KHz. The product provides an economical and compact solution for converting the ECL-generated BUSY logic signals to LVDS for the TRB3 board to use. The boards are currently in use at PSI, and figure 3.4.9 shows the current setup.

²Higher frequencies beyond 75 MHz could have been tested in theory; however, the instrumentation available limited the frequency of square wave signals that could be generated.

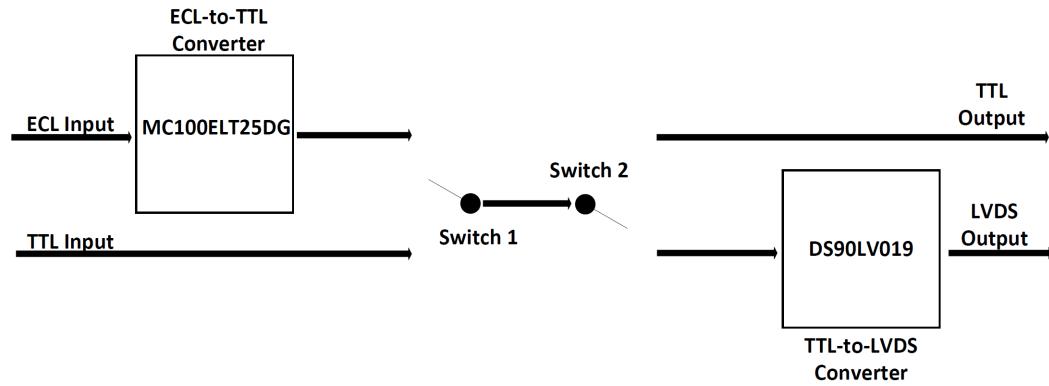


Figure 3.4.3: Single channel design overview. The board layout of the single channel is shown in figure 3.4.5.

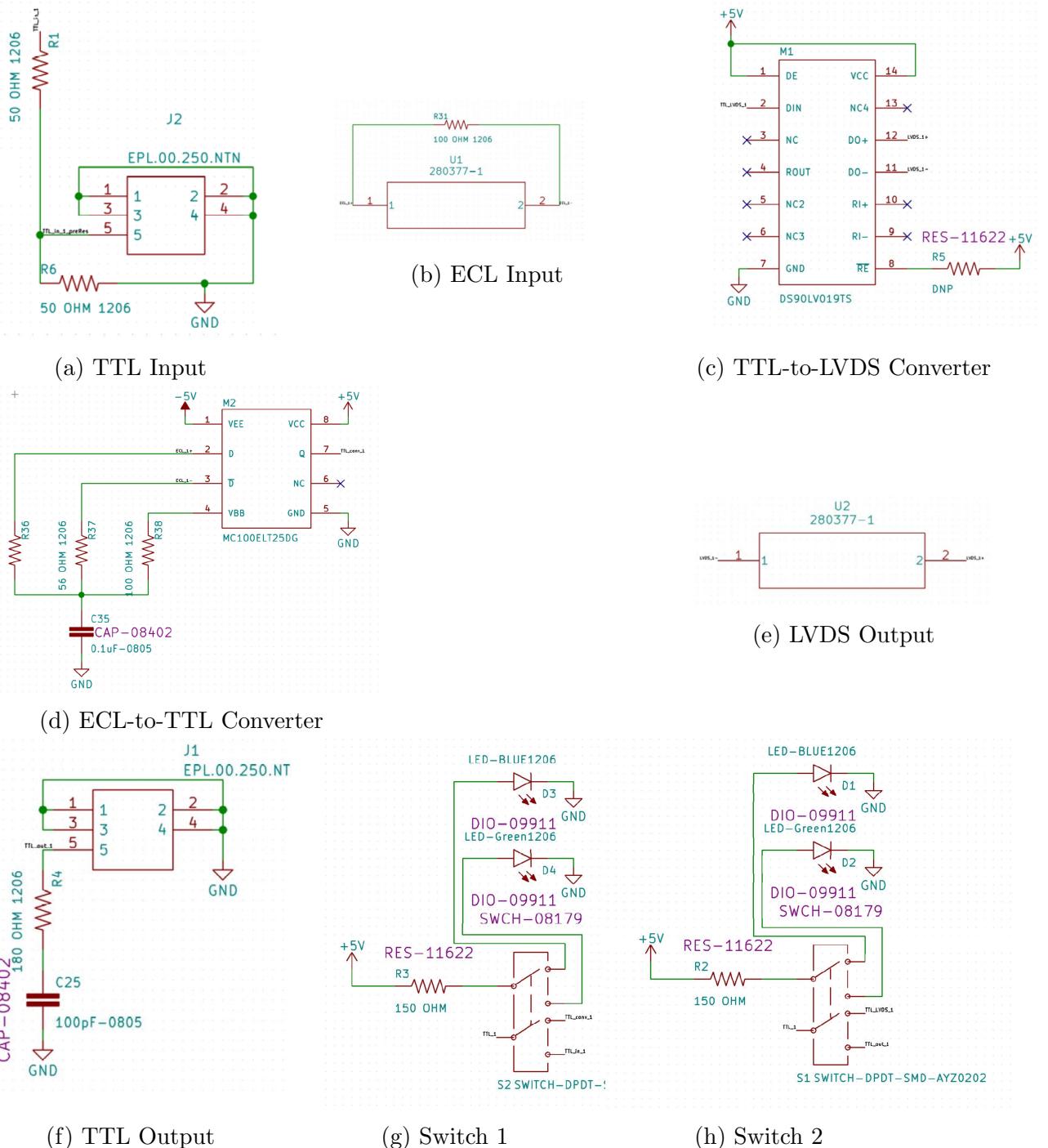


Figure 3.4.4: The schematic of different components for a single channel of logic level converter: (a) TTL Input, (b) ECL Input, (c) TTL-to-LVDS converter, (d) ECL-to-TTL converter, (e) LVDS output, (f) TTL output, (g) Switch1 to switch between ECL and TTL inputs, (h) Switch2 to switch between TTL and LVDS outputs.

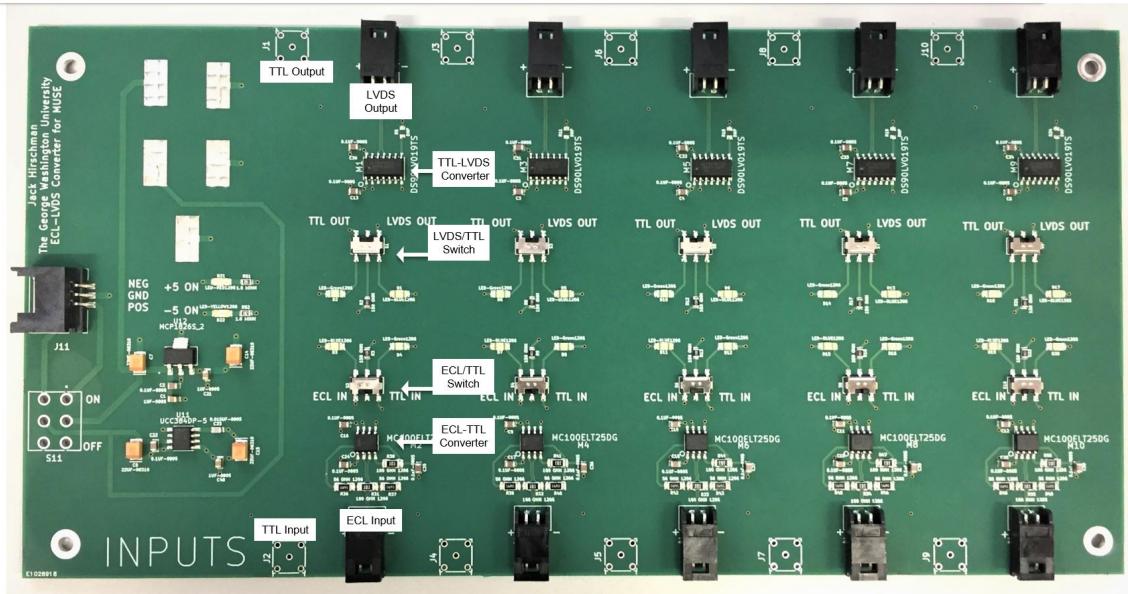
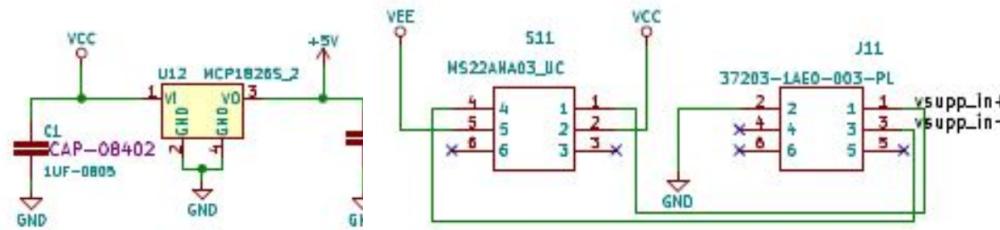
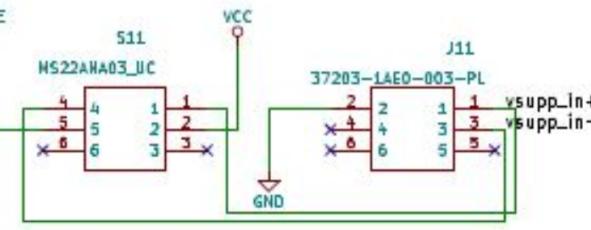


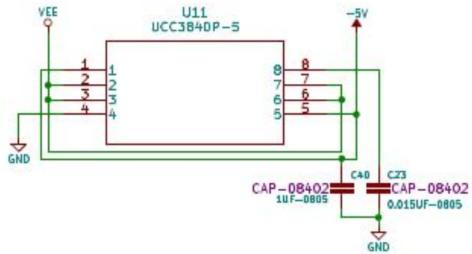
Figure 3.4.5: Labeled board of single channel.



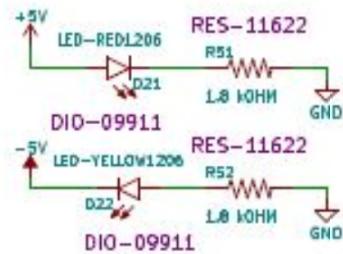
(a) Positive 5V Regulator



(b) Off-board Power Connector and Power Switch



(c) Negative 5V Regulator



(d) Red and Yellow LED Indicators

Figure 3.4.6: Power circuitry for positive and negative supplies along with red and yellow LED indicators.

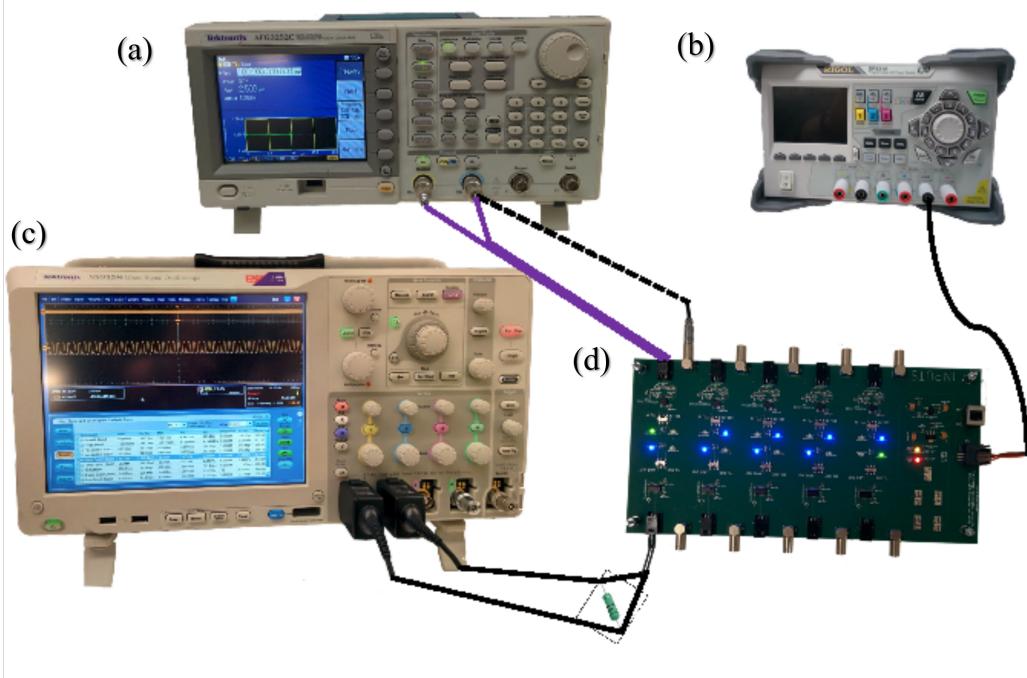


Figure 3.4.7: Test setup for ECL-to-LVDS and TTL-to-LVDS (red dashed line). (a) Function generator (b) Power supply (c) Oscilloscope (d) Design under test.

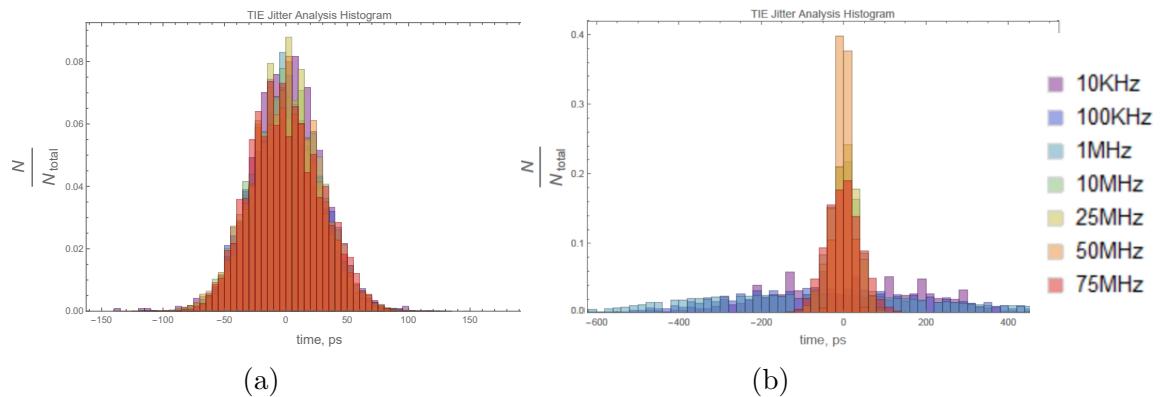


Figure 3.4.8: Jitter analysis histograms for (a) TTL-to-LVDS conversion and for (b) ECL-to-LVDS conversion, showing test setup issues at lower frequencies.

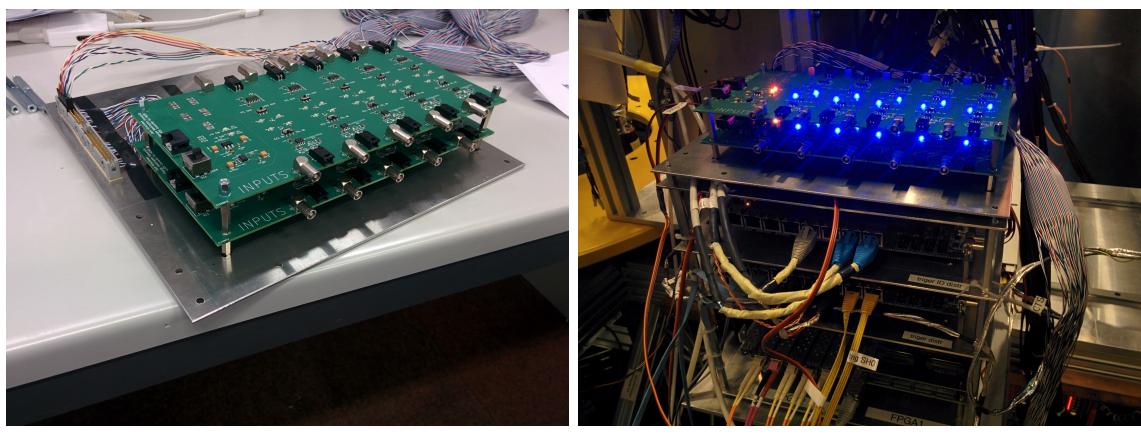


Figure 3.4.9: Final board placement for MUSE at PSI, Switzerland.

Chapter 4

Conclusions

The Proton Radius Puzzle emerged in 2010 when newer and more precise measurements of the proton radius extracted from muonic hydrogen spectroscopy were shown to be incompatible with previously established measurements of the proton radius from electronic hydrogen spectroscopy and electron proton scattering. MUSE will use muon and electron proton scattering to help shed light on the puzzle, potentially corroborating one of the previous measurements or introducing a path to new physics.

In order to impact the proton radius puzzle, MUSE must make a highly precise measurement. This project involved working with the MUSE TDAQ and detector systems, alongside collaborators at GW, HUJI, and PSI, to contribute to MUSE’s precision measurements. The work covered several sub-projects, all successfully completed with the assistance of graduate students, post-docs, research advisers, and engineers. These projects, while largely independent of one another, all contribute to enhancing MUSE’s performance.

Contributions to the BH and BM detectors included work during summer 2017 on construction of SiPM planes using new assembly frames and, during summer and fall of 2018, included development of HV calibration software for matching performance of these detectors over multiple run times. The successfully developed software extracts a calibration factor by

fitting peak and pedestal positions from QDC spectra, thus indicating how to adjust voltage to keep the calibration factors matched.

Contributions to the STT detectors included gas mixture tests for optimization of detector performance via checking various proportions of Ar to CO₂. For these tests, a gas mixing system that can control the proportion of Ar and CO₂ was successfully constructed.

For the TDAQ system, a test setup for probing the TDCs on the TRB3 was successfully designed and implemented on a FPGA on a PaDiWa. These TDCs are used for reading out all the detectors in MUSE’s detector setup. This test setup was used during beamtime in December 2018 to test 5376 channels across 28 TRB3 boards.

Also for the TDAQ system, a logic level converter board for interfacing between modules that produce the busy signals in the DAQ and the TRB3, which must read these signals to perform Trigger logic, was successfully designed, manufactured, and tested. These boards are currently ready for use during MUSE’s next beamtime.

Bibliography

- [1] Jan C. Bernauer. Proton Charge Radius and Precision Tests of QED. *XXXIV Physics in Collision Symposium Bloomington, Indiana*, September 2014.
- [2] <https://www.psi.ch/media/the-psi-proton-accelerator>.
- [3] R. Gilman et. al. Technical design report for the paul scherrer institute experiment r-12-01.1: Studying the proton “radius” puzzle with μp elastic scattering. June 2017.
- [4] <https://indico.cern.ch/event/643308/contributions/2610520/attachments/1467722/2536718/isotdaq18.Negri.DaqIntro.pdf>.
- [5] Private communication with Ievgen Lavrukhan.
- [6] Private communication with Dr. Tigran Rostomyan.
- [7] https://www.hamamatsu.com/resources/pdf/ssd/s13360_series_kapd1052e.pdf.
- [8] Private communication with Dan Cohen.
- [9] http://inside.mines.edu/~kleach/PHGN422/supplemental/eScatteringFormFactors_Supplemental.pdf.
- [10] Peter J. Mohr, Barry N. Taylor , and David B. Newell. CODATA recommended values of the fundamental physical constants: 2010. *Rev. Mod. Phys.*, 84:1527–1605, Nov 2012.

- [11] R. Pohl, A. Antognini, F. Nez, F. D. Amaro, F. Biraben, et al. The size of the proton. *Nature*, 466:213–217, July 2010.
- [12] A. Antognini et al. Proton structure from the measurement of 2s-2p transition frequencies of muonic hydrogen. *Science*, 339:417–420, January 2013.
- [13] G. Ron et al. Low- Q^2 measurements of the proton form factor ratio $\mu_p G_E/G_M$. *Phys. Rev. C*, 84:055204, Nov 2011.
- [14] J. C. Bernauer, M. O. Distler, J. Friedrich, et. al. Electric and magnetic form factors of the proton. *Phys. Rev. C*, 90:015206, Jul 2014.
- [15] R. Pohl, R. Gilman, G. Miller, K. Pachucki. Muonic hydrogen and the proton radius puzzle. *Annual Review of Nuclear and Particle Science*, 63(1):175–204, • 2013.
- [16] D. Griffiths. *Introduction to Elementary Particles*. Wiley-VCH, 2 edition, 2008.
- [17] G. Ciezarek, M. Franco Sevilla, B. Hamilton, R. Kowalewski, T. Kuhr, et al. A challenge to lepton universality in b meson decays. *Nature*, 546:227–233, June 2017.
- [18] <https://www2.ph.ed.ac.uk/~playfer/PPlect6.pdf>.
- [19] G. Roberts and M. Ali-Bakhshian. A brief introduction to time-to-digital and digital-to-time converters. *IEEE Transactions on Circuits and Systems*, 57(3):153–157, March 2010.
- [20] <http://www.vita.com/page-1855175#A>.
- [21] https://midas.triumf.ca/MidasWiki/index.php/Main_Page.
- [22] <https://root.cern.ch/>.
- [23] <https://wikis.mit.edu/confluence/display/MUSE/Running+Analysis+Code>.

- [24] <https://root.cern.ch/root/html/doc/guides/users-guide/FittingHistograms.html>.
- [25] L. Landau. On the energy loss of fast particles by ionization. *J. Phys. (USSR)*, 8:201–205, June 1944.
- [26] <https://root.cern.ch/root/html524/TMath.html>.
- [27] <https://www.autodesk.com/products/eagle/overview>.
- [28] <https://www.onsemi.com/pub/Collateral/MC10ELT25-D.PDF>.
- [29] <http://www.ti.com/lit/ds/symlink/ds90lv019.pdf>.
- [30] <http://www.kingbrightusa.com/images/catalog/SPEC/APTR3216PBC-A.pdf>.
- [31] <http://www.kingbrightusa.com/images/catalog/SPEC/APTR3216MGC.pdf>.
- [32] <http://www.kingbrightusa.com/images/catalog/SPEC/APT3216LSECK-J3-PRV.pdf>.
- [33] <http://www.kingbrightusa.com/images/catalog/SPEC/APT3216LSYCK-J3-PRV.pdf>.
- [34] <http://kicad-pcb.org/>.
- [35] <https://www.eurocircuits.com/>.
- [36] <https://www.tek.com/oscilloscope/mso5000-dpo5000>.
- [37] <https://www.tek.com/oscilloscope/dpo7000-digital-phosphor-oscilloscope-manual/dpojet>.
- [38] https://www.gsi.de/fileadmin/EE/Module/VULOM/vulom4B_3.pdf.
- [39] D. Griffiths. *Introduction to Quantum Mechanics*. Cambridge University Press, 2 edition, 2017.

Appendix A

PaDiWa FPGA Design

The user logic of a new PaDiWa FPGA design is shown in listing A.1. To test the user logic, the testbench shown in listing A.2 was developed. The waveform produced by the testbench was presented in the chapter 3.

Listing A.1: Content of the file "signal_gen_x16.v".

```
'timescale 1ns / 1ps

module signal_gen_x16(
    pulse_out, // LVDS 16 identical output signals generated with clock
    spare_out, // LVDS spare output that goes to trigger input
    clk //clock signal , clk_i = 133MHz <= Clock on padiwa
);

// Want to have 30 ns wide output pulses with frequency around 11 kHz.
// 133 MHz = 7.5 ns => pulse should be 4 periods wide.
// Then it should be dead for 90000 ns = 12000 clock periods.
// Assignng I/O and registers;
```

```

input  clk;
output reg[15:0]  pulse_out;
output reg  spare_out;

parameter PULSE_WIDTH = 4; //width of pulse;
parameter PULSE_DEAD = 12000-PULSE_WIDTH; //dead time after pulse;
reg [13:0] counter; // 14 bit clock period counter to count 12000 clock
periods

always @ (posedge clk)
begin
    counter = counter + 1;
    if(counter == PULSE_DEAD)
        begin
            pulse_out[15:0] <= 16'hFFFF;
            spare_out <= 1;
        end
    else if (counter == (PULSE_DEAD + PULSE_WIDTH))
        begin
            pulse_out[15:0] <= 16'h0000;
            spare_out <= 0;
            counter <= 0;
        end
    end
end

endmodule

```

Listing A.2: Content of the file "signal_gen_x16_tb.v".

```

'timescale 1ns / 1ps
module signal_gen_x16_tb;

```

```

reg clk;
wire [15:0] pulse_out;
wire spare_out;

signal_gen_x16 U0 (
    .clk (clk),
    .pulse_out (pulse_out),
    .spare_out (spare_out)
); // Links testbench to module being tested

initial
begin
    clk = 0; //initializes clock
    #10000000 $finish; //runs simulation for 10 seconds

end

always
    #3.75 clk = ~clk; //7.5ns clock period (needs to switch twice in
    7.5ns)

endmodule

```

Appendix B

SiPM HV Calibration

The software for SiPM HV calibration developed for the SiPM detectors of the BH and BM detectors is presented here.

Listing B.1: Content of the file ”SiPM_HV”.

```
#include <SiPM_HV.h>

#include<iostream>
#include<cmath>

SiPM_HV::SiPM_HV(TTree *in, TTree *out, TFile *inf_, TFile * outf_ ,
TObject *p):Plugin(in,out,inf_,outf_,p)
{
};

SiPM_HV::~SiPM_HV()
{
};
```

```

template <typename T>
auto findSmallestTime(T &container)
{
    auto best=container.begin();
    for (auto hit=container.begin();hit!=container.end();hit++)
    {
        if (best->second.time > hit->second.time){ //@@ !(hit->trailing)
            best=hit;
        }
    }
    return best;
}

//----- Main routing
-----
//defining all histograms that we need:
Long_t SiPM_HV::defineHistograms()
{

    debug(0,"\\t\\t\\tDefining histogram for BH SiPM MQDCs \\n");
    for(int i=0; i<4; i++){
        //loop over all channels:
        for(int j=0; j<16; j++){
            MQDC_up.BH[i][j] = dH1(TString::Format("Beam Hodoscope/SiPM
                Plane_%d/Bar_%d/QDC Up",i,j),"QDC UP; QDC channel; counts"
                ,4197,-100,4096);
        }
    }
}

```

```

MQDC_down.BH[i][j] = dH1(TString::Format("Beam Hodoscope/SiPM
Plane_%d/Bar_%d/QDC Down",i,j),"QDC Down; QDC channel;
counts",4197,-100,4096);

Ped_up.BH[i][j] = dH1(TString::Format("Beam Hodoscope/SiPM
Plane_%d/Bar_%d/Ped. Up",i,j),"QDC Pedestal UP; QDC
channel; counts",4197,-100,4096);

Ped_down.BH[i][j] = dH1(TString::Format("Beam Hodoscope/SiPM
Plane_%d/Bar_%d/Ped. Down",i,j),"QDC Pedestal Down; QDC
channel; counts",4197,-100,4096);

Peak_up.BH[i][j] = dH1(TString::Format("Beam Hodoscope/SiPM
Plane_%d/Bar_%d/Peak Up",i,j),"QDC Peak UP; QDC channel;
counts",4197,-100,4096);

Peak_down.BH[i][j] = dH1(TString::Format("Beam Hodoscope/SiPM
Plane_%d/Bar_%d/Peak Down",i,j),"QDC Peak Down; QDC
channel; counts",4197,-100,4096);

// MQDC_mean.BH[i][j] = dH1(TString::Format("Beam Hodoscope/
SiPM Plane_%d/Bar_%d/G_Mean",i,j),"QDC GeoMean; QDC channel; counts
",4197,-100,4096);
}

}

debug(0,"\t\t\tDefining histogram BM Big Bars MQDCs \n");
for(int i=0; i<4; i++){
//loop over all channels:
MQDC_up.BM_big[i] = dH1(TString::Format("Beam Monitor/Big
Scintilators/Bar_%d/QDC Up",i),"QDC UP; QDC channel; counts"
,4197,-100,4096);

MQDC_down.BM_big[i] = dH1(TString::Format("Beam Monitor/Big

```

```

    Scintilators/Bar_%d/QDC Down",i),"QDC Down; QDC channel; counts
    ",4197,-100,4096);

Ped_up.BM_big[i] = dH1(TString::Format("Beam Monitor/Big
    Scintilators/Bar_%d/Ped Up",i),"QDC Pedestal UP; QDC channel;
    counts",4197,-100,4096);

Ped_down.BM_big[i] = dH1(TString::Format("Beam Monitor/Big
    Scintilators/Bar_%d/Ped Down",i),"QDC Pedestal Down; QDC
    channel; counts",4197,-100,4096);

Peak_up.BM_big[i] = dH1(TString::Format("Beam Monitor/Big
    Scintilators/Bar_%d/Peak Up",i),"QDC Peak UP; QDC channel;
    counts",4197,-100,4096);

Peak_down.BM_big[i] = dH1(TString::Format("Beam Monitor/Big
    Scintilators/Bar_%d/Peak Down",i),"QDC Peak Down; QDC channel;
    counts",4197,-100,4096);

//      MQDC_mean.BM_big[i] = dH1(TString::Format("Beam Monitor/Big
//      Scintilators/Bar_%d/G_Mean",i),"QDC GeoMean; QDC channel; counts
//      ",4197,-100,4096);
}

debug(0,"\\t\\t\\tDefining histogram BM SiPM MQDCs \\n");
for(int i=0; i<32; i++){
    //loop over all channels:
    MQDC_up.BM[i] = dH1(TString::Format("Beam Monitor/SiPM Plane/Bar_%
        d/QDC Up",i),"QDC UP; QDC channel; counts",4197,-100,4096);
    MQDC_down.BM[i] = dH1(TString::Format("Beam Monitor/SiPM Plane/
        Bar_%d/QDC Down",i),"QDC Down; QDC channel; counts"
        ,4197,-100,4096);
}

```

```

Ped_up.BM[i] = dH1(TString::Format("Beam Monitor/SiPM Plane/Bar_%d
/Ped Up",i),"QDC Pedestal UP; QDC channel; counts"
,4197,-100,4096);

Ped_down.BM[i] = dH1(TString::Format("Beam Monitor/SiPM Plane/Bar_%
%d/Ped Down",i),"QDC Pedestal Down; QDC channel; counts"
,4197,-100,4096);

Peak_up.BM[i] = dH1(TString::Format("Beam Monitor/SiPM Plane/Bar_%
d/Peak Up",i),"QDC Peak UP; QDC channel; counts"
,4197,-100,4096);

Peak_down.BM[i] = dH1(TString::Format("Beam Monitor/SiPM Plane/
Bar_%d/Peak Down",i),"QDC Peak Down; QDC channel; counts"
,4197,-100,4096);

//      MQDC_mean.BM[i] = dH1(TString::Format("Beam Monitor/SiPM Plane/
Bar_%d/G_Mean",i),"QDC GeoMean; QDC channel; counts",4197,-100,4096)
;

}

return ok;
}

// startup() routine is running at the beginning of analysis
Long_t SiPM_HV::startup()
{
printf("\t\t\tstartup() routine has started:\n");

//opening SiPM Hodoscope tree in root file

```

```

bhraw = NULL;
getBranchObject("BH", (TObject **) &bhraw);
if (!bhraw) {
    debug(0, "Could not find BH tree in file\n");
}

//opening BM tree: => this we will do later

bmraw = NULL;
getBranchObject("BM", (TObject **) &bmraw);
if (!bmraw) {
    debug(0, "Could not find BM tree in file\n");
}

printf("\t\t\t End of startup() routine!\n");
return ok;
}

//process() routine is running for every event

Long_t SiPM_HV::process()
{
    //Plug in RF:
    if(bhraw->trb_reftime.size()==0){
        debug(1,"NO REFERENCE TIME FOR TRB SiPM\n");
        return ok;
    }
}

```

```

//Trigger:
if(bhraw->trig.size()==0){
    debug(1,"NO TRIGGER TIME FOR SIPM\n");
    return ok;
}

}

/***************** Plots for BH histograms:
********************/
for(int i=0; i<4; i++){
    for(int j=0; j<16; j++){
        auto tdc_up = findSmallestTime(bhraw->plane[i][j].tdc_trb
            [1]);
        auto tdc_down = findSmallestTime(bhraw->plane[i][j].tdc_trb
            [0]);

        //filling hist for Up SiPM
        for(auto qdc_up: bhraw-> plane[i][j].adc_mqdc[1]){
            MQDC_up.BH[i][j] -> Fill(qdc_up);

            //selectring pedestal & peak events based on TDC
            information:
            if(tdc_up->second.rising && tdc_down->second.rising &&
                qdc_up >=100) {
                Peak_up.BH[i][j]-> Fill(qdc_up)
                ;
            } else{
                Ped_up.BH[i][j]->Fill(qdc_up);
            }//end if statement
        } //end loop over up QDCs
    }
}

```

```

//filling hist for Down SiPM
for(auto qdc_down: bhraw-> plane[i][j].adc_mqdc[0]){
    MQDC_down.BH[i][j]-> Fill(qdc_down);

    //selectring pedestal & peak events based on TDC
    //information:
    if(tdc_up->second.rising && tdc_down->second.rising &&
       qdc_down >=100) {
        Peak_down.BH[i][j]-> Fill(qdc_down);
    } else{
        Ped_down.BH[i][j]->Fill(qdc_down);
    } //end if statement

} //end loop over down QDCs

//filling hist for both QDC_UP * QDC_DOWN;
for(auto qdc_up: bhraw-> plane[i][j].adc_mqdc[1]){
    for(auto qdc_down: bhraw-> plane[i][j].adc_mqdc[0]){
        //MQDC_mean.BH[i][j]-> Fill(sqrt(qdc_up*qdc_down));
        H2(qdc_up,qdc_down,TString::Format("Beam Hodoscope/
            SiPM Plane_%d/Bar_%d/ QDC UP vs QDC DOWN", i,j),
            "QDC UP vs QDC DOWN",500,0,500,500,0,500);
    }
}

} //end loop over cahnnels
} //end loop over planes

```

```

***** Plots for BM histograms:
*****
*****For BM in a new mapping plane[0] -> SiPMs; plane[1] -> Big
Bars *****

for(int j=0; j<32; j++){
    auto tdc_up = findSmallestTime(bmraw->plane[0][j].tdc_trb[1]);
    auto tdc_down = findSmallestTime(bmraw->plane[0][j].tdc_trb[0])
    ;
}

//filling hist for Up SiPM
for(auto qdc_up: bmraw-> plane[0][j].adc_mqdc[1]){
    MQDC_up.BM[j] -> Fill(qdc_up);

//selectring pedestal & peak events based on TDC information:
if(tdc_up->second.rising && tdc_down->second.rising && qdc_up
>=100){
    Peak_up.BM[j] -> Fill(qdc_up);
} else{
    Ped_up.BM[j] -> Fill(qdc_up);
}//end if statement

} // end loop over up SiPMs in BM

//filling hist for Down SiPM
for(auto qdc_down: bmraw-> plane[0][j].adc_mqdc[0]){
    MQDC_down.BM[j]-> Fill(qdc_down);

//selectring pedestal & peak events based on TDC information:
if(tdc_up->second.rising && tdc_down->second.rising &&
qdc_down >=100){

```

```

        Peak_down.BM[j]-> Fill(qdc_down);
    } else{
        Ped_down.BM[j]->Fill(qdc_down);
    } //end if statement

} //end loop over down SiPM in BM

/*
//filling hist for Geometric Mean = Sqrt(QDC_UP * QDC_DOWN);
for(auto qdc_up: bmraw-> plane[0][j].adc_mqdc[1]){
    for(auto qdc_down: bmraw-> plane[0][j].adc_mqdc[0]){
        MQDC_mean.BM[j]-> Fill(sqrt(qdc_up*qdc_down));
    }
}
*/
}

***** Plots for BM Big Bars histograms:
*****
for(int j=0; j<4; j++){
    auto tdc_up = findSmallestTime(bmraw->plane[1][j].tdc_trb[1]);
    auto tdc_down = findSmallestTime(bmraw->plane[1][j].tdc_trb[0]);

    //filling hist for Up Big Bars
    for(auto qdc_up: bmraw-> plane[1][j].adc_mqdc[1]){
        MQDC_up.BM_big[j] -> Fill(qdc_up);

        //selectring pedestal & peak events based on TDC information:
        if(tdc_up->second.rising && tdc_down->second.rising && qdc_up

```

```

>=100) {
    Peak_up.BM_big[j]-> Fill(qdc_up);
} else{
    Ped_up.BM_big[j]->Fill(qdc_up);
}//end if statement

}//end loop over up Big Bars

//filling hist for Down Big Bars
for(auto qdc_down: bmraw-> plane[1][j].adc_mqdc[0]){
    MQDC_down.BM_big[j]-> Fill(qdc_down);

//selectring pedestal & peak events based on TDC information:
if(tdc_up->second.rising && tdc_down->second.rising &&
qdc_down >=100){
    Peak_down.BM_big[j]-> Fill(qdc_down);
} else{
    Ped_down.BM_big[j]->Fill(qdc_down);
}//end if statement

}//end loop over down Big Bars

/*
//filling hist for Geometric Mean = Sqrt(QDC_UP * QDC_DOWN);
for(auto qdc_up: bmraw-> plane[1][j].adc_mqdc[1]){
    for(auto qdc_down: bmraw-> plane[1][j].adc_mqdc[0]){
        MQDC_mean.BM_big[j]-> Fill(sqrt(qdc_up*qdc_down));
    }
}
*/
}

```

```

return ok;
}

***** Fitting Function Declaration *****

//QDC ped with piecewise gaussian: mean and peak height are the common
parameter for both of them.

Double_t fit_Gaus_Com(Double_t *x, Double_t *par) {
    return par[0]*((x[0]<=par[1])*TMath::Gaus(x[0], par[1], par[2]) + (x
        [0] > par[1])*TMath::Gaus(x[0], par[1], par[3]));
}

//QDC spectra has an exponential background:

Double_t fit_Background(Double_t *x, Double_t *par) {
    return par[0]*TMath::Exp(par[1]*x[0]);
}

Double_t fit_Background_lin(Double_t *x, Double_t *par) {
    return par[0]*x[0] + par[1];
}

Double_t fit_Landau(Double_t *x, Double_t *par) { //*
    return par[0]*TMath::Landau(x[0], par[1], par[2]);
}

```

```

//QDC spectra will be fitted with 2 assimetric gausians + exponential
background:

Double_t fit_Global1(Double_t *x, Double_t *par) {
    return fit_Gaus_Com(x, par) + fit_Gaus_Com(x, &par[4]) +
        fit_Background(x, &par[8]);
}

//QDC spectra will be fitted with 1 assymetric gaussian + landau +
linear background:

Double_t fit_Global2(Double_t *x, Double_t *par) {
    return fit_Gaus_Com(x, par) + fit_Landau(x, &par[4]) +
        fit_Background_lin(x, &par[7]); //changed to landau
}

Double_t fit_Global3(Double_t *x, Double_t *par) {
    return fit_Gaus_Com(x, par) + fit_Landau(x, &par[4]) + fit_Background
    (x, &par[7]); //changed to landau
}

Double_t fit_Global4(Double_t *x, Double_t *par) { //peak
    return fit_Landau(x, par) + fit_Background(x, &par[3]); //changed to
    landau
}

// Old Fitting Functions:

//QDC pedestal function
Double_t fit_Ped(Double_t *x, Double_t *par) {

```

```

// return par[0]*TMath::Landau(x[0], par[1], par[2]) ;
return par[0]*TMath::Gaus(x[0], par[1], par[2]);
}

// QDC Peak function
Double_t fit_Peak(Double_t *x, Double_t *par) {
    return par[0]*TMath::Gaus(x[0], par[1], par[2]);
}

// Sum of background and peak function
Double_t fit_Both(Double_t* x, Double_t *par) {
    return fit_Ped(x, par) + fit_Peak(x,&par[3]);
}

// This function that fits QDCs and returns the array of vaules in the
// following order:
// {pedestal, peak}
void SiPM_HV:: fit_QDC(double qdc_fit_range_max, double *data_output,
    TH1D *ped, TH1D *peak, TH1D *qdc ){ // plane -- detector plane (0-3
    for BH, 0-1 for BM); bar -- bar number, det_type = 0(BH)/1(BM)
//defining fitting functions for each bar:
    TF1 *fit_Ped = new TF1("fit_Ped",fit_Gaus_Com,0, 4000, 4);
    //setting names for fitting parameters:
    fit_Ped -> SetParNames("Const_fit", "Mean_fit", "Sigma_1","Sigma_2");
    //Setting Parameters limits for Fitting:
    double max_Ped = ped -> GetMaximum(); //When you are fitting with
    gausian, you should use this in "SetParameters"
    double mean_Ped = ped -> GetMean();
    double sigma_Ped = ped -> GetStdDev();

```

```

//Setting up constraints on fitting parameters:
fit_Ped -> FixParameter(0, max_Ped);
fit_Ped -> SetParLimits(1, 0.5*mean_Ped, 1.5*mean_Ped);
fit_Ped -> SetParLimits(2, 0.2*sigma_Ped, 4*sigma_Ped);
fit_Ped -> SetParLimits(3, 0.2*sigma_Ped, 4*sigma_Ped);
fit_Ped -> SetLineColor(9); //sets color to violet

//fitting histograms:
gStyle->SetOptFit(1);
ped -> Fit("fit_Ped", "Q", "", mean_Ped - 3*sigma_Ped, mean_Ped + 3*
sigma_Ped); //Fitting only in range from 1 to 100

***** Now we can fit the QDC spectra with fixed pedestal
parameters obtained from fitting pedestal separately
***** */

/*defining QDC fitting functions for each bar:
 [0]-[3] -> pedestal parameters
 [4]-[6] -> peak parameters
 [7]-[8] -> background
*/
TF1 *fit_QDC = new TF1("fit_QDC", fit_Global2, 0, 4000, 9); //

//Setting Parameters limits for Fitting:
//Up:
double max_QDC = peak -> GetMaximum();
double mean_QDC = peak -> GetMean();
double sigma_QDC = peak -> GetStdDev();

fit_QDC -> SetParLimits(0, 0.95*fit_Ped->GetParameter(0), fit_Ped->
GetParameter(0));
fit_QDC -> FixParameter(1, fit_Ped->GetParameter(1));

```

```

    fit_QDC -> FixParameter(2, fit_Ped->GetParameter(2));
    fit_QDC -> FixParameter(3, fit_Ped->GetParameter(3));

    fit_QDC -> SetParLimits(4, 0.9*5.8*max_QDC, 1.1* 5.8*max_QDC);
    fit_QDC -> SetParLimits(5, 0.8*.9*mean_QDC, 1.2 * .9*mean_QDC); // maybe do a range
    fit_QDC -> SetParLimits(6, 0.4*.4*sigma_QDC, 1.6*.4*sigma_QDC);
    fit_QDC -> SetParLimits(7, -100, 0); // the slope should be a negative
    fit_QDC-> SetParLimits(8, 0, 2*max_QDC); // the value at of line at x =0;
    fit_QDC -> SetLineColor(2); //sets color to red

    //fitting histograms:
    gStyle->SetOptFit(1);
    qdc -> Fit("fit_QDC", "Q", "", 5, qdc_fit_range_max);

    data_output[0] = fit_QDC->GetParameter(1); // pedestal position
    data_output[1] = fit_QDC->GetParameter(5); // peak position
    //printf("data[1] = %lf; data[0] = %lf \n", data_output[1],
    data_output[0]);
}

void SiPM_HV:: fill_gain_tree(TTree *tree, char const * name, int plane
, int bar, int up_down, double pedestal, double peak){
    TString gain_header ("plane/I:bar/I:up_down/I:pedestal/F:peak/F:qdc
/F:gain/F:shift_dV/F");
    tree_struct_t tree_struct;
    TBranch* branch = tree->GetBranch(name); //check if branch already
exist
}

```

```

if(!branch){
    printf("Gain branch was not defined\n");
    branch = tree->Branch(name, &tree_struct.plane, gain_header); // if branch doesn't exist, create it
}
else{
    printf("Gain branch was defined\n");
    branch->SetAddress(&tree_struct.plane); //if exist, reset the adress to the structure (just in case)
}

tree_struct.plane=plane;
tree_struct.bar = bar;
tree_struct.up_or_down = up_down;
tree_struct.pedestal = pedestal;
tree_struct.peak = peak;
tree_struct.qdc = peak - pedestal;
tree_struct.gain = 1.0/(peak - pedestal);
tree_struct.shift_dV = 0; //not implemented for now!
tree->Fill();
}

void SiPM_HV:: fill_gain_hist(int plane, double* data_array, int array_size, char const* D_name, char const *UpDown){
TH1D* hist = dH1(TString::Format("Gain_%s/Plane_%d_%s", D_name, plane , UpDown), TString::Format("Gain Distribution in %s Plane: %d (%s); Bar_Number", D_name, plane, UpDown), array_size, 0, array_size);
for (int j=0; j<array_size; j++){
    hist->SetBinContent(j+1, data_array[j]);
    hist->GetXaxis()->SetBinLabel(j+1, TString::Format("%d", j));
}
}

```

```

    hist->Draw();
}

// finalize() routine is running at the end of analysis
Long_t SiPM_HV::finalize()
{
    double PedPeak_BH_Up[4][16], PedPeak_BH_Down[4][16]; //array to
    store ped peak position differences
    printf("\t\t\tfinalize() routine has started:\n");
/****** Fitting QDC spectra for BH
*****/
// ----- BH detector
-----//  

    TTree *tree_BH = new TTree("BH_Gain","gain_table");

    for(int i=0; i<4; i++){
//loop over all bars:
        for(int j=0; j<16; j++){
            double qdc_pos_up[2];
            double qdc_pos_down[2];
//Fitting data:
            fit_QDC(1000, qdc_pos_up, Ped_up.BH[i][j], Peak_up.BH[i][j],
                    MQDC_up.BH[i][j]);
            fit_QDC(1000, qdc_pos_down, Ped_down.BH[i][j], Peak_down.BH[i]
                    ][j], MQDC_down.BH[i][j]);

            PedPeak_BH_Up[i][j] = 1.0/(qdc_pos_up[1] - qdc_pos_up[0]);
            PedPeak_BH_Down[i][j] = 1.0/(qdc_pos_down[1] - qdc_pos_down
                    [0]);
}
}

```

```

//Filling the tree with gain table:

    fill_gain_tree(tree_BH, "BH_Gain_Tree", i, j, 1, qdc_pos_up
[0], qdc_pos_up[1]);

    fill_gain_tree(tree_BH, "BH_Gain_Tree", i, j, 0,
qdc_pos_down[0], qdc_pos_down[1]);

} //end for loop over bars

//Plotting histograms of gain

    fill_gain_hist(i, PedPeak_BH_Up[i], 16, "BH", "Up");
    fill_gain_hist(i, PedPeak_BH_Down[i], 16, "BH", "Down");

}//end for loop over planes

//***** Fitting QDC spectra for BM
//----- BM detectors:
-----//

double PedPeak_BM_Up[32], PedPeak_BM_Down[32]; //array to store ped
peak position differences

TTree *tree_BM = new TTree("BM_Gain","gain_table");

for(int j=0; j<32; j++){
    double qdc_pos_up[2];
    double qdc_pos_down[2];

//Fitting data:
    fit_QDC(1500, qdc_pos_up, Ped_up.BM[j], Peak_up.BM[j], MQDC_up.BM
[j]);
    fit_QDC(1500, qdc_pos_down, Ped_down.BM[j], Peak_down.BM[j],
MQDC_down.BM[j]);

    PedPeak_BM_Up[j] = 1.0/(qdc_pos_up[1] - qdc_pos_up[0]);
    PedPeak_BM_Down[j] = 1.0/(qdc_pos_down[1] - qdc_pos_down[0]);

//Filling the tree with gain table:
    fill_gain_tree(tree_BM, "BM_Gain_Tree", 0, j, 1, qdc_pos_up[0],

```

```

        qdc_pos_up[1]);
    fill_gain_tree(tree_BM, "BM_Gain_Tree", 0, j, 0, qdc_pos_down
[0], qdc_pos_down[1]);
} //end for loop over bars
//Plotting histograms of gain
fill_gain_hist(0, PedPeak_BM_Up, 32, "BM", "Up");
fill_gain_hist(0, PedPeak_BM_Down, 32, "BM", "Down");
// ----- end fitting QDC BM -----//

// -----fitting for QDC BM Big Bars-----//
double PedPeak_BM_Big_Up[4], PedPeak_BM_Big_Down[4]; //array to
store ped peak position differences
TTree *tree_BM_Big = new TTree("BM_Big_Gain","gain_table");
for(int j=0; j<4; j++){
    double qdc_pos_up[2];
    double qdc_pos_down[2];
//Fitting data:
    fit_QDC(3500, qdc_pos_up, Ped_up.BM_big[j], Peak_up.BM_big[j],
MQDC_up.BM_big[j]);
    fit_QDC(3500, qdc_pos_down, Ped_down.BM_big[j], Peak_down.BM_big[
j], MQDC_down.BM_big[j]);

    PedPeak_BM_Big_Up[j] = 1.0/(qdc_pos_up[1] - qdc_pos_up[0]);
    PedPeak_BM_Big_Down[j] = 1.0/(qdc_pos_down[1] - qdc_pos_down[0]);
//Filling the tree with gain table:
    fill_gain_tree(tree_BM_Big, "BM_Big_Gain_Tree", 1, j, 1,
qdc_pos_up[0], qdc_pos_up[1]);
    fill_gain_tree(tree_BM_Big, "BM_Big_Gain_Tree", 1, j, 0,
qdc_pos_down[0], qdc_pos_down[1]);
} //end for loop over bars
//Plotting histograms of gain

```

```

    fill_gain_hist(1, PedPeak_BM_Big_Up, 4, "BM_Big", "Up");
    fill_gain_hist(1, PedPeak_BM_Big_Down, 4, "BM_Big", "Down");
/******End of BM_Big fitting******/
}

printf("\t\t\tend of finalize() routine!\n");

return ok;
}

Long_t SiPM_HV::cmdline(char *cmd)
{
//add cmdline hanling here

return 0; // 0 = all ok
};

extern "C"
{
Plugin *factory(TTree *in, TTree *out, TFile *inf_, TFile * outf_,
TObject *p)
{
    return (Plugin *) new SiPM_HV(in,out,inf_,outf_,p);
}
}

```

```
ClassImp(SiPM_HV);
```

Appendix C

ECL-to-LVDS Design Schematic and PCB Layers

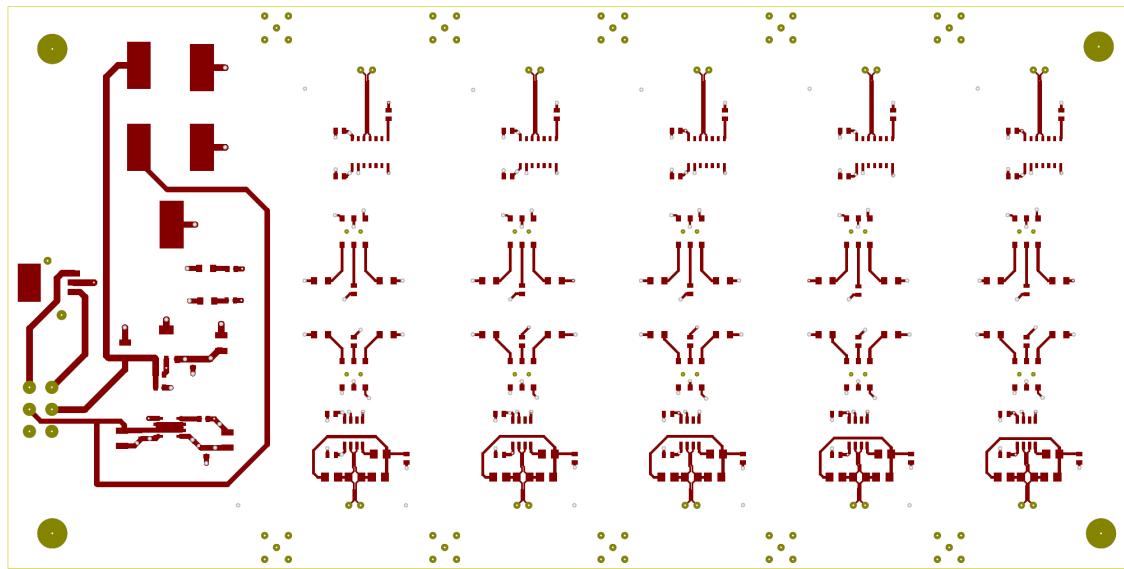


Figure C.0.1: Top layer of PCB design (signal layer).

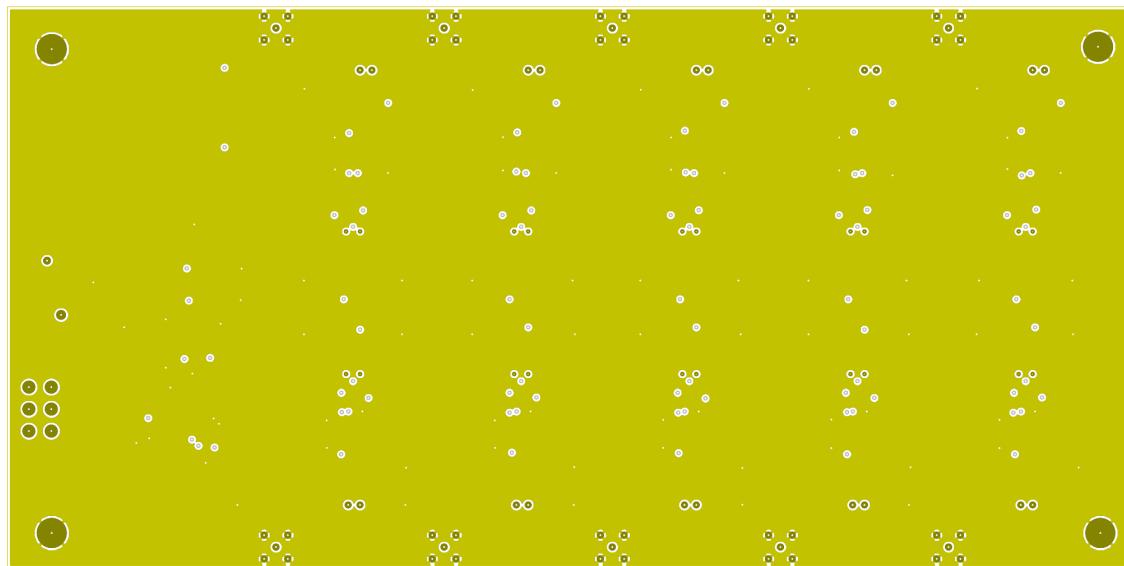


Figure C.0.2: Second layer of PCB design (ground layer).

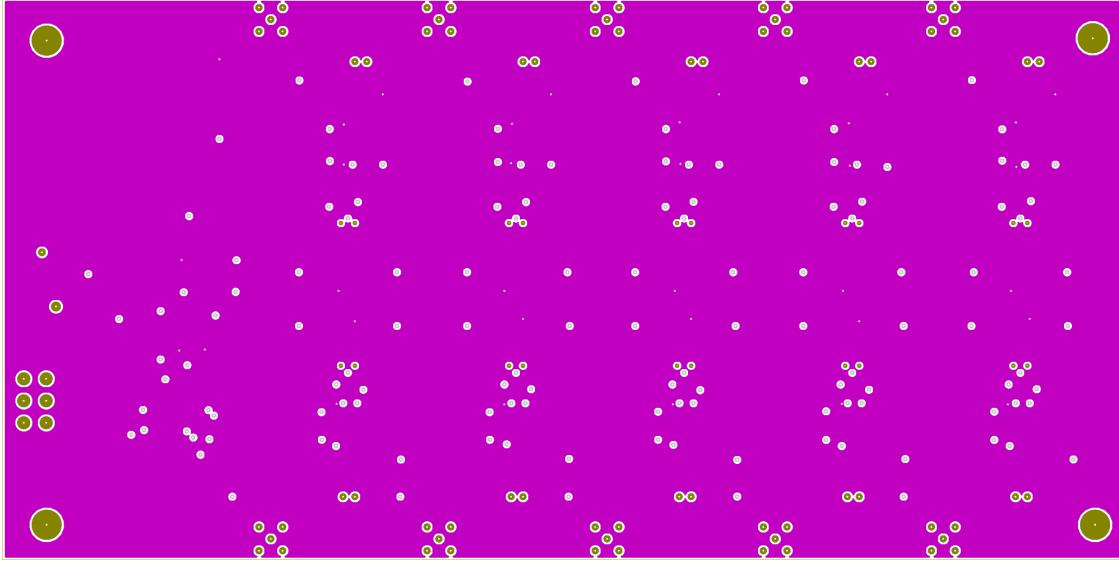


Figure C.0.3: Third layer of PCB design (+5V layer).

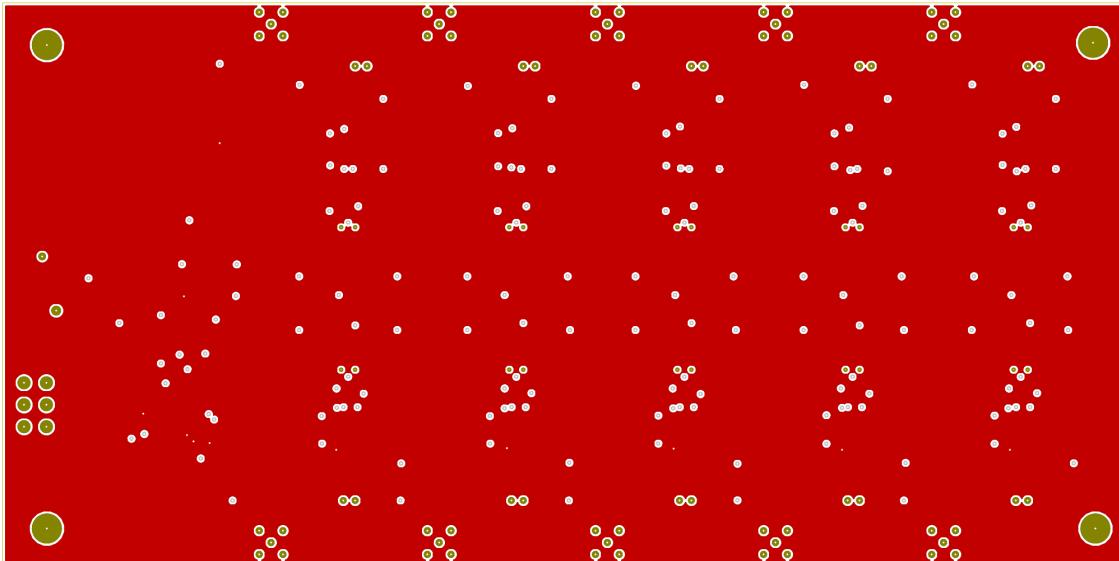


Figure C.0.4: Fourth layer of PCB design (-5V layer).

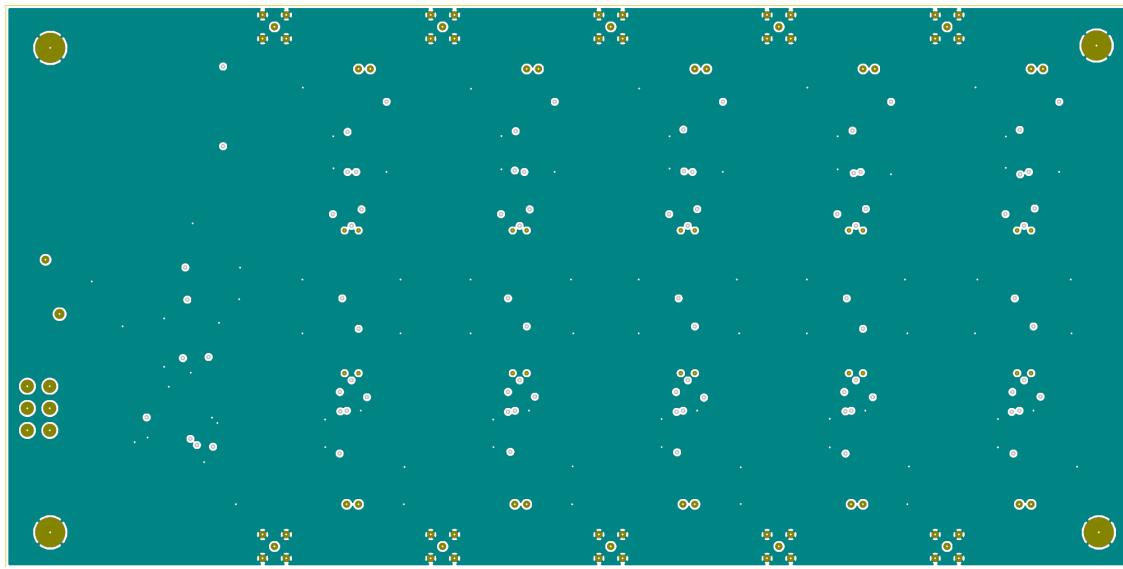


Figure C.0.5: Fifth layer of PCB design (ground layer).

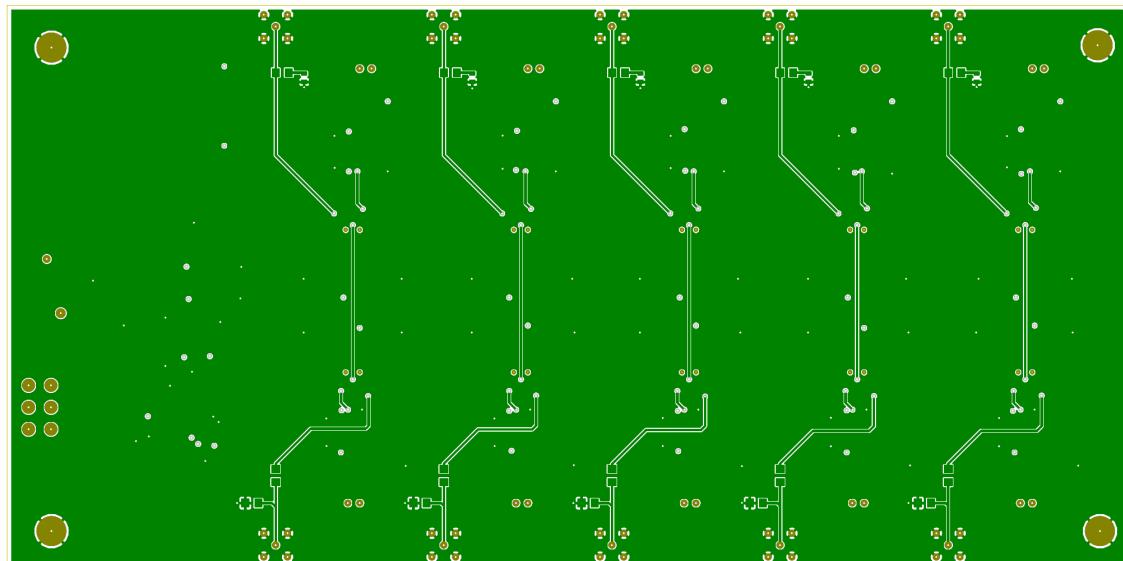


Figure C.0.6: Bottom layer of PCB design (signal layer).

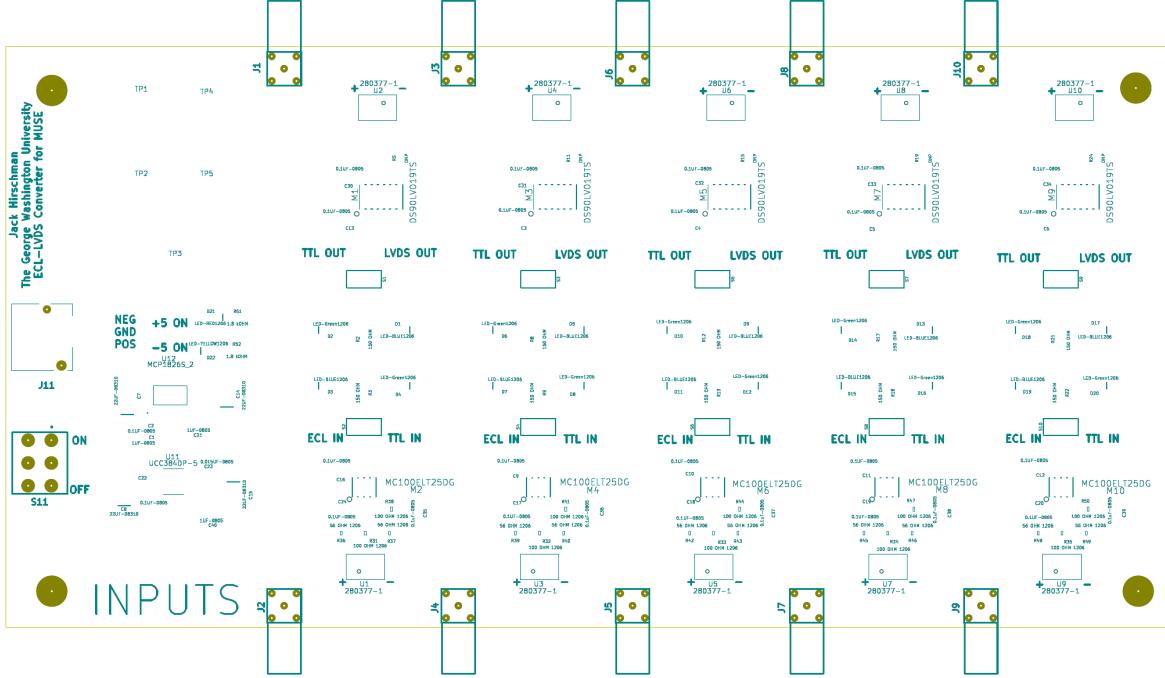


Figure C.0.7: Top layer silkscreen.



Figure C.0.8: Bottom layer silkscreen.

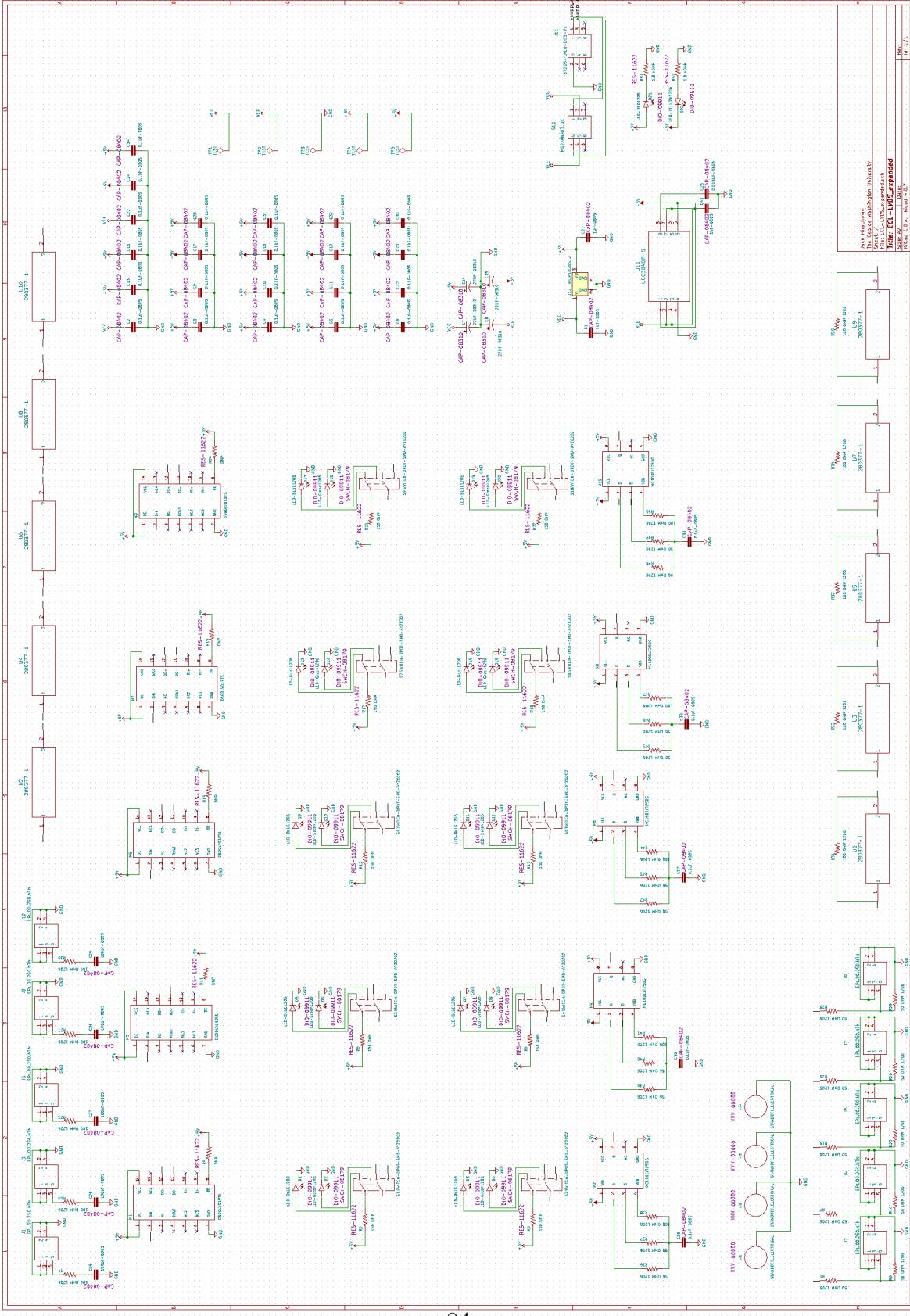


Figure C.0.9: Schematic for final board design.

Appendix D

Additional ECL-to-LVDS Board

Design Information

The board consists of six copper layers with five layers of FR4 as isolation. The top and bottom copper layers are 0.018 mm thick, and the inner copper layers are 0.035 mm thick. The top layer is used for differential signals, the second layer for ground, the third for +5V, the fourth for -5V, the fifth for ground, and the sixth for TTL signals embedded in a ground pour. A stack-up of the layers is shown in figure D.0.1. The copper layers are labeled and shown in orange. [35]

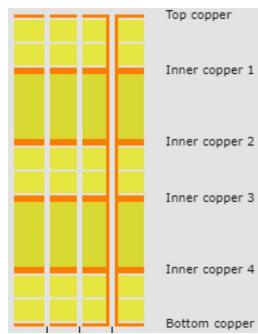


Figure D.0.1: Stacked board layers.

All signal traces utilized impedance matching at 50Ω , and differential signals utilized

trace length matching. The KiCAD toolset supplies a calculator for translating specified impedance to trace width [34]. Figure D.0.2 shows this tool where one can input values to arrive at desired impedance. Using this tool, it was determined that the TTL signals should have a trace width of 0.42 mm and that the differential signals should have trace widths of 0.31 mm.

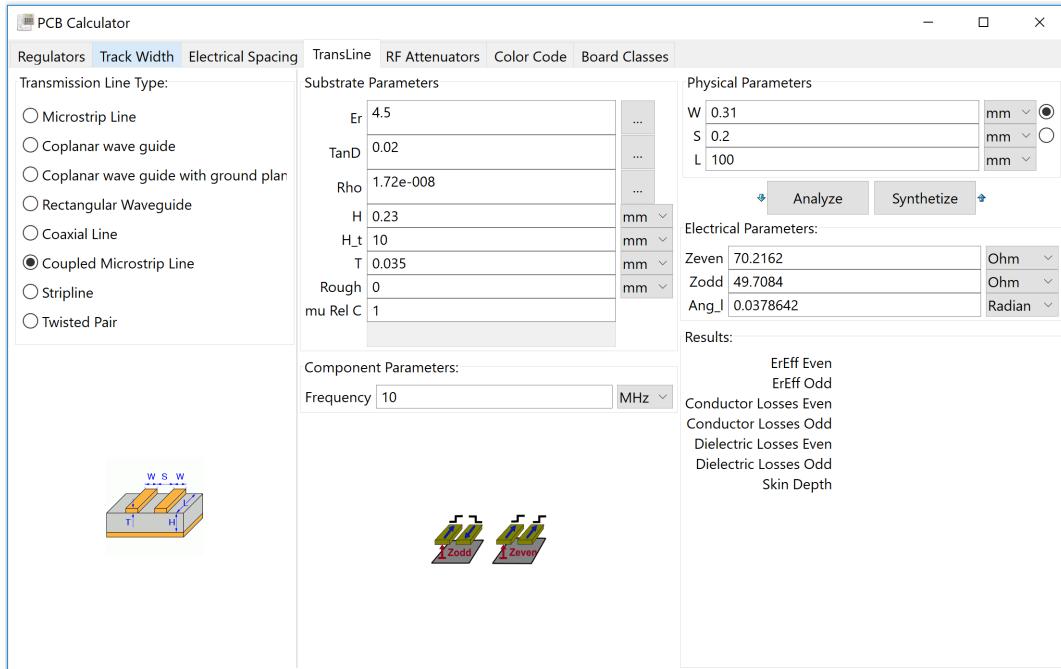


Figure D.0.2: Example of impedance calculation tool from KiCAD.

Appendix E

Vulom4b Information

A set of passive adapter boards was developed at the University of Basel. These are used for mapping from the RJ45 connector on the CAT7 cable coming from the TRB3 to LVDS differential signal pair input connectors on the Vulom4b [38] board as seen in figure E.0.1. There are three signals carried in via six wires. These signal pairs are trigger, triggerID, and clock which are used to form a trigger count packet of information within the Vulom4b board. This information is then used for trigger identification and associated count number within the DAQ setup. These boards came to PSI unassembled. The small project here consisted of assembling these boards. The finished product is shown in figure E.0.2.

Furthermore, these Vulom4b boards produce the BUSY signal used by the ECL-to-LVDS converter for converting the ECL output signal to the logic level the TRB3 can use.

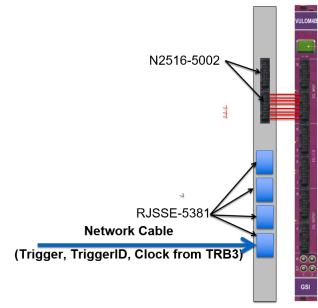


Figure E.0.1: Position of adapter board (on left) with Vulom4b board (on right) [5].



Figure E.0.2: Assembled Vulom4b adapter board for interface to TRB3 [6].

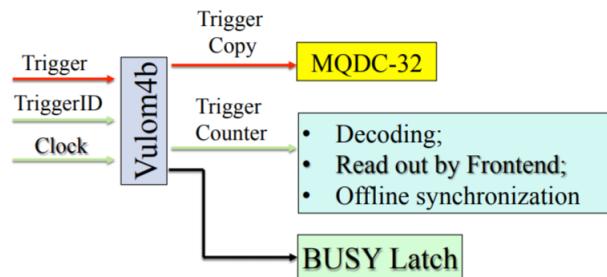


Figure E.0.3: Vulom4b's function in MUSE's TDAQ system [5].

Appendix F

Additional Collaboration Service Work

Various tasks and projects outside the scope of the main project were completed while at PSI. These tasks were essential in moving the experiment along in the development and construction process. Each of these tasks will be listed here along with a brief explanation of their purpose and importance.

SPS Wall and Beam Monitor Mounting Tasks

Drilling

Holes were drilled in the main experiment frame in order to properly mount the rails that hold the carriages for the SPS detector walls and the beam monitor (see figure F.0.1). There were roughly thirty holes, and this task took approximately four hours to complete.

Carriage Mounting and SPS detector wall preparation

Helped mount carriage and rails to the top of the frame where the holes were drilled. Properly oriented and attached metal adapter plates to the top rail on the SPS wall which allows for the wall to sit on and attach to the carriages mounted on the frame (see figure

[F.0.2a](#)). Also, ensured all bolts were properly tightened on the detector wall and taped all cables and connectors safely within the wall frame so that the wall could be craned into the main experiment frame for surveying (see figures [F.0.2b](#) and [F.0.2c](#)). Helped make adjustments to have the wall properly mounted within the frame on the carriages.

Beamtime

Day and night shifts were taken for data collection during beamtime during summer 2018.



Figure F.0.1: (Me) Drilling mounting holes in frame to hold SPS walls.

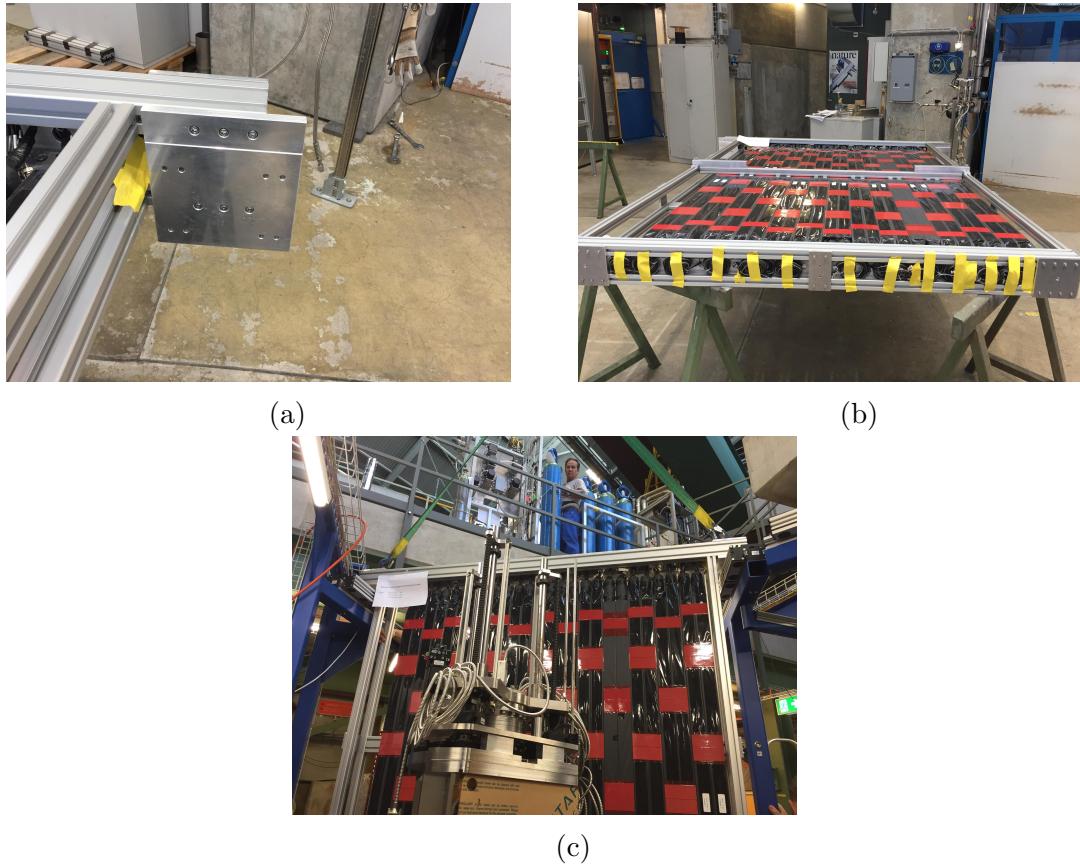


Figure F.0.2: (a) Mounting plate adapter attached to SPS wall; (b) View from top of SPS wall; (c) SPS still attached to crane while being place on carriage rails.

Appendix G

Simplified Scattering Theory

Figure G.0.1 shows the setup for scattering where an incident electron with initial momentum \vec{P}_i elastically scatters off of a target and is deflected an angle θ such that it has final momentum \vec{P}_f . A vector \vec{q} can be defined as the transfer of momentum as seen on the right in figure G.0.1 [9].

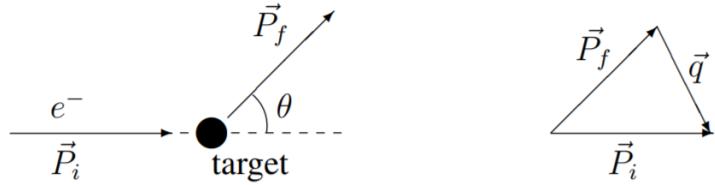


Figure G.0.1: Setup for Rutherford scattering with incident electron elastically scattering off of a target [9].

$$\frac{d\sigma}{d\Omega} = |f(\theta)|^2 \quad (\text{G.0.1})$$

$$f = \int e^{\frac{i\vec{q}\cdot\vec{r}}{\hbar}} V(\vec{r}) d^3(\vec{r}) \quad (\text{G.0.2})$$

For elastic scattering, the differential cross section ($\frac{d\sigma}{d\Omega}$) is given by the square of the

scattering amplitude ($f(\theta)$) as shown in equation G.0.1 [39]. One can then use the first order Born Approximation¹ and assume plane wave scattering to find the scattering amplitude. The first order Born Approximation is shown in equation G.0.2 where $V(\vec{r})$ is a potential [9].

$$V(\vec{r}) = \frac{-Ze^2}{4\pi\epsilon_0} \int \frac{\rho(\vec{r}')}{|\vec{r} - \vec{r}'|} d^3(\vec{r}') \quad (\text{G.0.3})$$

$$f = \frac{-Ze^2}{4\pi\epsilon_0} \int e^{\frac{i\vec{q}\cdot\vec{r}}{\hbar}} \int \frac{\rho(\vec{r}')}{|\vec{r} - \vec{r}'|} d^3(\vec{r}') d^3(\vec{r}) \quad (\text{G.0.4})$$

One can then use the Coulomb potential as shown in equation G.0.3 where Z is the number of protons in the target nucleus, e is fundamental charge, and ρ is the charge distribution [9]. Substituting the Coulomb potential for the generic potential yields a new equation for the scattering amplitude shown in equation G.0.4 [9].

$$f = \frac{-Ze^2}{4\pi\epsilon_0} \int \frac{e^{\frac{i\vec{q}\cdot\vec{r}}{\hbar}}}{|\vec{r} - \vec{r}'|} d^3(|\vec{r} - \vec{r}'|) \int e^{\frac{i\vec{q}\cdot\vec{r}'}{\hbar}} \rho(\vec{r}') d^3(\vec{r}') \quad (\text{G.0.5})$$

$$F(q) = \int e^{\frac{i\vec{q}\cdot\vec{r}'}{\hbar}} \rho(\vec{r}') d^3(\vec{r}') \quad (\text{G.0.6})$$

Rearranging this equation into yet another form (as seen in equation G.0.5) reveals where the Form Factor comes out of the second integral as shown in equation G.0.6 [9].

$$\frac{d\sigma}{d\Omega} = \alpha \frac{1}{\sin^4(\frac{\theta}{2})} |F(q)|^2 \quad (\text{G.0.7})$$

With the relation between differential cross section and Form Factor established (as seen in equation G.0.7), one can more closely inspect the Form Factor, noticing the Form Factor is the Fourier transform of the charge distribution [9].

¹The first order Born Approximation comes from using the first term in the expansion of the integral form of the Schrödinger equation, assuming that the incident wave is not altered too much by the potential [39].

$$a = \int_0^\infty r^2 4\pi r^2 dr \quad (\text{G.0.8})$$

A root-mean-square radius weighted by the charge distribution (a) can be defined as shown in equation G.0.8.

$$F(qa) = 1 - \frac{q^2 a^2}{6} \quad (\text{G.0.9})$$

$$a^2 = -6 \frac{dF(qa)}{dq^2} \quad (\text{G.0.10})$$

Using the root-mean-square radius along with a truncated Taylor expansion of the Form Factor, the relation between the two can be shown as in equation G.0.9 [9]. Multiplying both sides by -6 and taking the derivative relative to the square of the momentum transfer yields the relation shown in equation G.0.10.

$$r_p^2 \equiv -6 \frac{dG_E}{dQ^2} \Big|_{Q^2=0} \quad (\text{G.0.11})$$

This form of the equation closely relates to the widely accepted equation for the mean-square radius value as shown in equation G.0.11 where G_E is the Sachs electric Form Factor and where Q^2 is the negative of the square of the four-momentum transfer [15] as mentioned in chapter 1.