Department of Physics and Astronomy
Heidelberg University

Bachelor Thesis in Physics
submitted by

Yanick Prianon

born in Ludwigshafen am Rhein (Germany)

2024

Development and Validation of the Mu3e Tile Detector Front-End Board Firmware

This Bachelor Thesis has been carried out by Yanick Prianon at the
Kirchhoff Institute for Physics in Heidelberg
under the supervision of
Prof. Hans-Christian Schultz-Coulon

Abstract

The Mu3e experiment aims to search for the charged lepton-flavor violating decay $\mu^+ \rightarrow e^+e^-e^+$ at a sensitivity of 10^{-16} at Paul Scherrer Institute, Switzerland. The observation of this decay would be a clear indication of physics beyond the Standard Model. Therefore, the detector is optimized to find this decay by precisely measuring vertex, momentum and timing information of the decay products at the required high rates. The Mu3e data acquisition system relies on a triggerless readout scheme, that streams data to a GPU based filter farm. This work discusses the validation of the front-end firmware for the Scintillating Tile Detector, which is one of the timing sub-systems of the Mu3e experiment. In order to test the functionality of the firmware, each block was validated separately and as a full chain by simulation and generation of hits at different places in the data processing chain. As FPGA memory is limited, only a limited search window is available and data taken during a previous testbeam campaign was used to validate requirements of the time sorter.

Zusammenfassung

Das Mu3e-Experiment zielt darauf ab, den Leptonenzahl verletzenden Zerfall $\mu^+ \rightarrow e^+e^-e^+$ mit einer Empfindlichkeit von 10^{-16} am Paul Scherrer Institut in der Schweiz zu suchen. Die Beobachtung dieses Zerfalls wäre ein eindeutiger Hinweis auf Physik jenseits des Standardmodells. Daher ist der Detektor darauf optimiert, diesen Zerfall zu finden, indem er die Informationen über Vertex, Impuls und Zeit der Zerfallsprodukte bei den erforderlichen hohen Raten präzise misst. Das Mu3e Datenerfassungssystem basiert auf einem triggerlosen Ausleseschema, welches die Daten an eine GPU-basierte Filterfarm überträgt. Diese Arbeit behandelt die Validierung der Front-End-Firmware für den Scintillating Tile Detector, der eines der Zeiterfassungs-Subsysteme des Mu3e-Experiments darstellt. Um die Funktionalität der Firmware zu testen, wurde jeder Block einzeln und als vollständige Kette durch Simulation und Erzeugung von Treffern an verschiedenen Stellen in der Datenverarbeitungskette validiert. Da der FPGA Speicher begrenzt ist, steht nur ein begrenztes Suchfenster zur Verfügung und Daten, die während einer früheren Testbeam Kampagne aufgenommen wurden, wurden verwendet, um die Anforderungen des Zeitsortierers zu validieren.

Contents

Acronyms

ASIC Application Specific Integrated Circuit

CLB configurable logic block

DAB Detector Adapter Board

DAQ Data Acquisition

FEB Front-End Board

FPGA Field Programmable Gate Array

HV-MAPS High-Voltage Monolithic Active Pixel Sensors

LUT lookup table

LVDS Low-Voltage Differential Signaling

MIDAS Maximum Integrated Data Acquisition System

MuTRiG Muon Timing Resolver including Gigabit-link

ODB Online Data Base

PRBS Pseudorandom binary sequence

PSI Paul Scherrer Institute

RTL register transfer level

SciFi Scintillating Fibre

SciTile Scintillating Tile

SiPM Silicon Photomultiplier

SPAD Single Photon Avalanche Diode

SPI Serial Peripheral Interface

SWB Switching Board

TDC Time-to-Digital Converter

TMB Tile Module Board

ToT Time-over-Threshold

VHDL Very High Speed Integrated Circuit Hardware Description Language

1 Introduction

This thesis presents the development and validation of the Data Acquisition (DAQ) firmware used by the Mu3e tile detector. It begins with the introduction to the Mu3e experiment highlighting the tile detector as well as its DAQ components. In the second chapter this will be followed by a study of the data latencies introduced by the different parts of the signal processing chain. The third chapter provides an in-depth overview of the firmware solution and its components, with a particular emphasis on the newly introduced elements and their implementation. Chapter four focuses on the validation of the implemented components with various approaches.

1.1 Mu3e Experiment

The Standard Model of Particle Physics represents the current theoretical framework for understanding fundamental particles and interactions governing the universe [1]. Despite its successes, the Standard Model has limitations, notably its exclusion of neutrino masses and the gravitational force, motivating ongoing experimental efforts to search for physics beyond the Standard Model.

One such experiment is the Mu3e experiment that will be conducted at Paul Scherrer Institute (PSI) in Villigen, Switzerland [2]. It is designed to investigate the charged lepton-flavor violating decay $\mu^+ \to e^+e^-e^+$, which, if successful, could signal new physics beyond the Standard Model. One possible decay shown in figure 1.

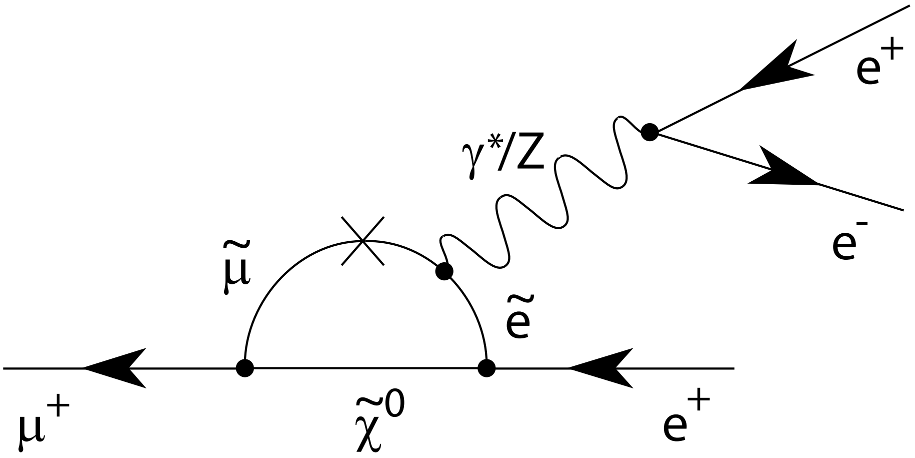


Figure 1: Diagram for lepton flavour violation involving supersymmetric particles (taken from [3])

This process has an extremely suppressed branching ratio BR $= \frac{p(\mu\to eee)}{p(\mu\to X)} \ll 10^{-50}$ [4] in the Standard Model, incorporating neutrino oscillations. The Mu3e experiment aims to improve upon the limits set by its predecessor, the SINDRUM experiment, which established an upper branching ratio limit of 10^{-12} [5]. Mu3e targets a sensitivity of 10^{-16} [2].

The experimental setup needs a highly precise detector design that is optimised to find the rare $\mu^+ \to e^+e^-e^+$ decay, while distinguishing background processes that would mimic the signal process, like internal conversion $\mu^+ \to e^+e^-e^+\nu_e\overline{\nu}_\mu$ and combinatorial backgrounds such as Michel decays $\mu^+ \to e^+\nu_e\overline{\nu}_\mu$, radiative muon decays and Bhabha scattering. Hence excellent momentum, timing and vertex resolution is needed to suppress aforementioned backgrounds.

In order to conduct such an experiment a source of muons is needed. The current muon beamline at PSI, used for the first phase of the experiment and producing a rate of 10^8 muons per second, will be upgraded to more then 10^9 muons per second in later phases to achieve the desired sensitivity. Therefore the experiment must be capable to operate at such rates, which presents significant challenges for the detector systems, data acquisition, and readout processes [3].

1.2 Technical Design

The Mu3e Experiment is a fixed-target experiment housed within a superconducting magnet generating a 1 T solenoidal magnetic field, with a cylindrical detector geometry coaxial to the muon beam. The beam will be focused onto a hollow double-cone mylar target, where the muons decay at rest. The detector setup consists of a combination of pixel detectors, scintillating fibre detectors, and scintillating tile detectors [2].

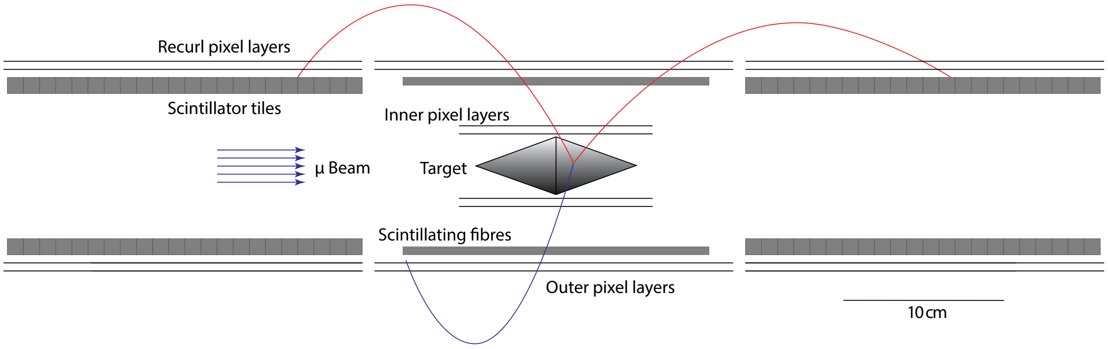


Figure 2: Mu3e Detector Setup (Phase I) (taken from [2])

The detector is grouped into three stations as shown in figure 2. The central station comprises the inner vertex pixel detector, a layer of scintillating fibres and two outer layers of of pixel detectors. After the decay products pass the layers of this station they recurl into the upstream and downstream stations include two pixel layers and one layer of scintillating tiles. Each layer ensures complete azimuthal coverage.

The Pixel Detector is responsible for tracking the particles and reconstructing their trajectories. It consists of a vertex detector with two inner pixel layers surrounding the target and additional tracking detectors situated in both the central and recurl stations. These pixel layers are responsible for good spatial resolution which is needed for precise vertex information and track reconstruction. The pixels require spatial resolution of ≤ 30 µm and a timing resolution of ≤ 20 ns [2]. This is achieved with High-Voltage Monolithic Active Pixel Sensors (HV-MAPS) [6].

The Scintillating Fibre (SciFi) Detector, located close to the interaction point, consists of multiple 300 mm long fibres arranged in staggered layers along the z-axis. This subsystem achieves good spatial resolution of ≈ 100 µm and very good timing resolution of ≈ 250 ps. This detector is needed to suppress all forms of combinatorial background from tracks with different timings [2].

The Scintillating Tile (SciTile) Detector is placed at the end of the electron tracks in the recurl stations. This detector, which is not constrained by material budget limitations, aims to achieve the best timing resolution of ≤ 100 ps. With its efficiency close to 100% the Tile Detector identifies coincident signals of electron triplets and suppresses accidental background [2].

1.3 Mu3e Tile Detector

The Tile Detector is composed of scintillating plastic tiles. As electrons traverse through the material they deposit energy and produce scintillating light, which can then be detected by Silicon Photomultipliers (SiPMs).

1.3.1 Silicon Photomultiplier

A SiPM is made of Single Photon Avalanche Diodes (SPADs). In a SPAD bringing p- and n-doped semiconductors together initially causes electrondrift towards the p-doped side and vice versa holes drift towards the n-doped side. This creates an electrical field that counteracts this drift. Hence a depleted region forms, which can be expanded by applying an external reverse bias voltage.

When a photon hits the depletion region, it can generate one electron-hole pair. When operated in Geiger-mode regime, that means the applied reverse voltage is above the so-called breakdown voltage, electrons and holes gain enough velocity to create secondary electrons and secondary holes as shown in figure 3. Therefore a single generated carrier can trigger an avalanche process where the discharge current increases rapidly and allows for precise timing of the incoming photon. In order to quickly stop the avalanche a quenching resistor is used [7].

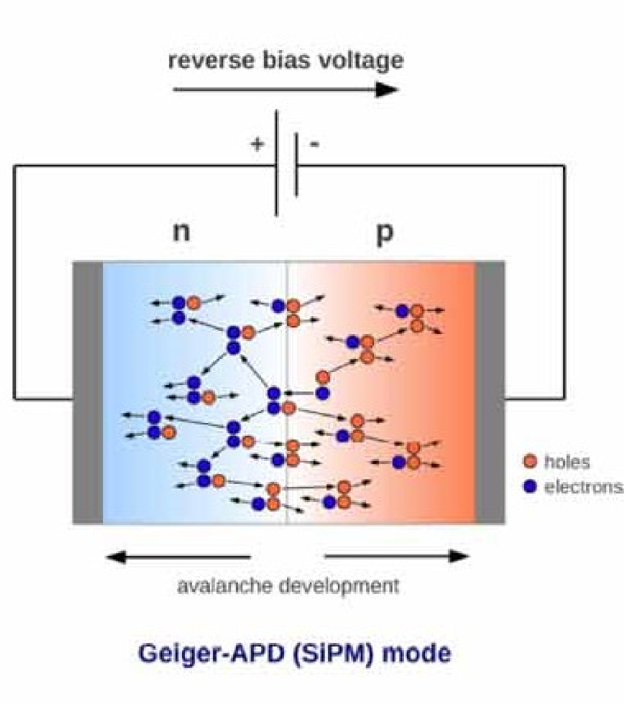


Figure 3: Avalanche Creation in Geiger Mode (taken from [7])

Since these avalanches are independent of the primary ionization energy, multiple SPADs get combined to one SiPM in order to get a signal proportional to the incoming number of photons. This allows for a detector with single-photon sensitivity and very good timing information [7].

1.3.2 Muon Timing Resolver including Gigabit-link

Charge pulses of the SiPMs are routed to the Muon Timing Resolver including Gigabit-link (MuTRiG) chip which is an Application Specific Integrated Circuit (ASIC) developed for the readout of both the SciFi and the SciTile Detector. Each MuTRiG chip accepts 32 input channels. For each SiPM, the incoming signal is processed by a timing and an energy branch thus allowing to set different thresholds for the discriminators in each branch. This is needed in order to apply a low timing threshold to get precise time of arrival information, and a higher energy threshold. The latter is used to validate signals in order to reduce noise and calculate energies with a linearized Time-over-Threshold (ToT) method [8].
Figure 4 illustrates how these two discriminator outputs get combined with a XOR-logic in a hit logic such that one Time-to-Digital Converter (TDC) can than be used to digitize both signals. The first rising edge (rising edge of T-Trigger) represents the timing measurement and via the second rising edge (falling edge of E-Trigger) the energy information can be gained by taking the difference between both rising edges of the combined signal. The TDC generates timestamps in bins with a 1.6 ns coarse counter and 50 ps fine counter [9, 10].

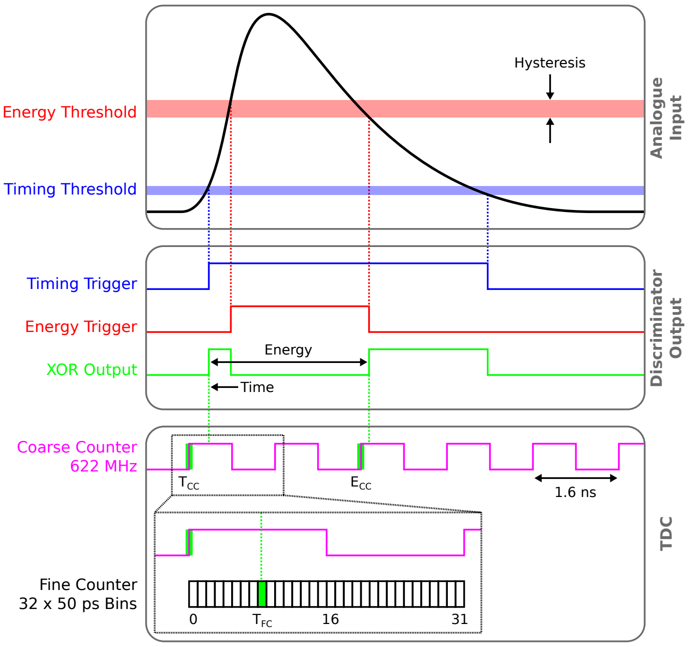


Figure 4: MuTRiG Trigger Principle (taken from [2])

As presented in figure 5 the resulting timestamps of each channel are converted into events and then buffered in a common memory before being send out. Via a Serial Peripheral Interface (SPI) the submodules of the chip can be configured e.g. E and T threshold can be modified.

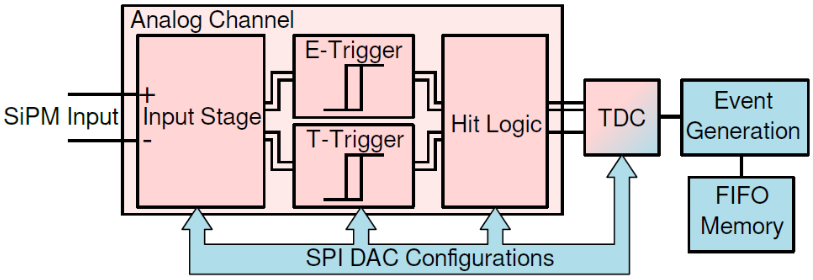
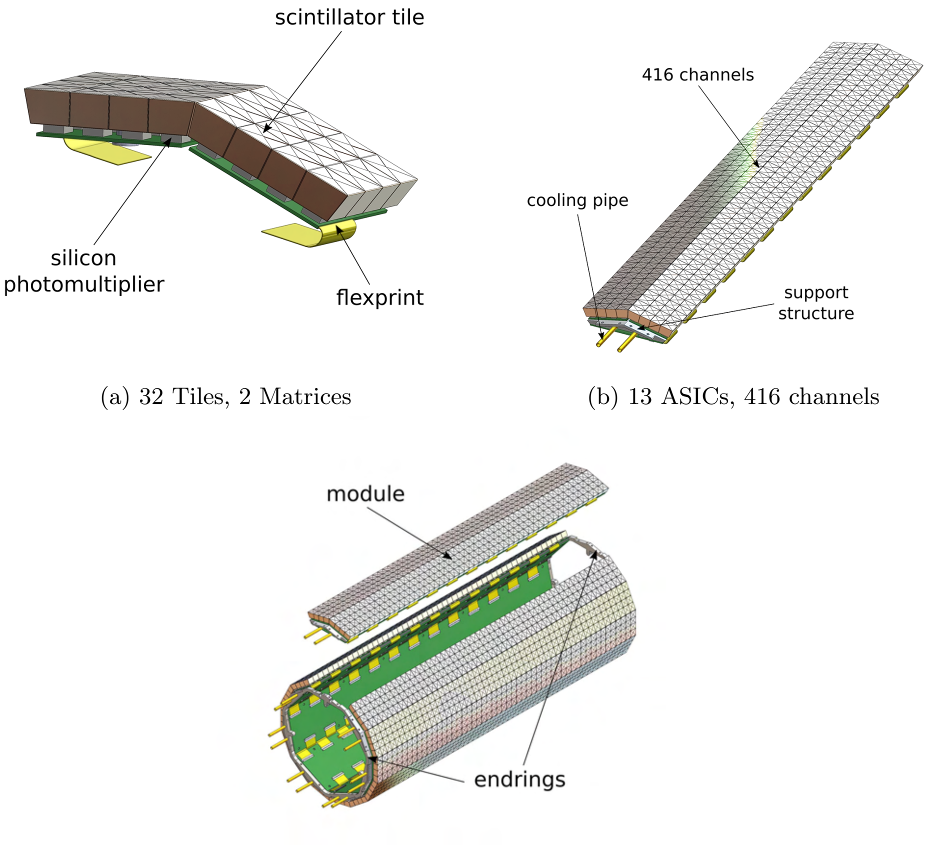


Figure 5: Diagram of one MuTRiG Channel (taken from [2])

When operated in the SciTile Detector, the MuTRiG ASIC constructs 48 bit events that contain the channel number and timing informations of both rising edges of the XOR Output representing a timing (T) timestamp and energy (E) timestamp. In the configuration the SciTile detector will use, the MuTRiG will send out up to 255 hits in frames every 12.4 µs.

1.3.3 Parts of the Tile Detector

To construct the SciTile Detector 16 SiPMs get combined onto a single matrix. Two of these matrices are then connected via a flexprint connection to one MuTRiG chip and 13 MuTRiG chips are furthermore combined onto a single Tile Module Board (TMB). Seven of those then form one Tile Station as depicted in figure 6. One Station will then be installed in each of the Up- and Downstream Stations.
Each TMB is a circuit board is designed to connect 13 ASICs through a 40-pin micro coaxial cable via Low-Voltage Differential Signaling (LVDS) to a Front-End Board (FEB), providing data lines and lines for the control protocol. Additionally, the TMB provides power for the ASICs and SiPMs and is responsible to distribute synchronously clock and reset signals towards the MuTRiGs.



(a) 32 Tiles, 2 Matrices

(b) 13 ASICs, 416 channels

(c) 7 TMBs, 1 Tile Station

Figure 6: Parts of the Tile Detector (taken from [2])

1.4 Mu3e Data Acquisition System

For the Mu3e experiment high statistics and thus a high rate experiment is necessary. To achieve high detection and readout efficiency triggered detector systems that preselect incoming events directly in the hardware and software have become essential in many experiments. The Mu3e system can not implement such a triggered readout system due to the low momentum of decay products, resulting in strongly curved tracks, creating hits in physically distant parts of the detector, that necessitate complete event reconstruction across the entire detector. Therefore complete readout without trigger systems will be implemented.

In order to accomplish this the Data Acquisition (DAQ) System needs to bundle signals in multiple stages as presented in figure 7. At first data from multiple ASICs get sent to one FEB. A total of 114 FEBs equipped with Intel Arria V Field Programmable Gate Arrays (FPGAs), collect and bundel data and forward the data out of the magnet via 6.25 Gbit/s optical links to one of four Switching Boards (SWBs). Those forward further merged data to the Server Farm.

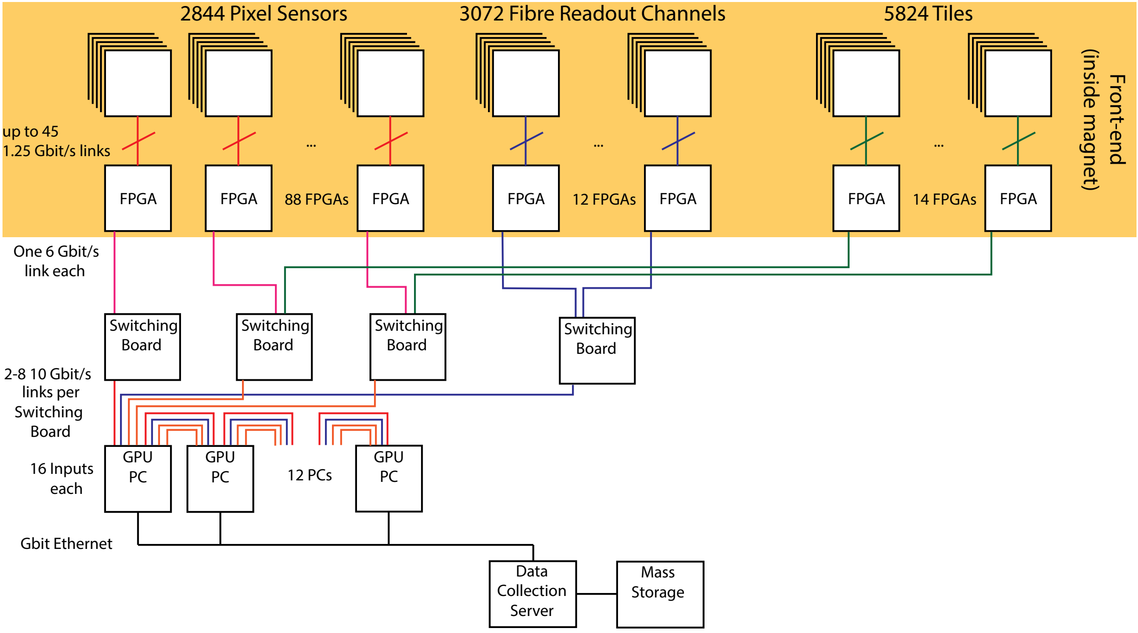


Figure 7: Mu3e Readout Scheme (taken from [2])

As data collection rates are expected to rise to as much as 100 Gbit/s across all detector components, the data from the central pixel station is processed by GPUs, which are capable of performing the necessary geometrical analysis tasks in order to decide what data to be permanently stored. Those GPUs can handle overlapping frames of 8 ns time stamp bin size. Currently the expected total size is 64 ns but might be adapted with respect to the exact specifications of the pixel sensors. A base clock domain at 125 MHz is distributed to all parts of the experiment and serves as reference clock for all sub-systems. Each sub-detector then needs to send data at this rate in packets covering 8 ns of detection [11].

1.5 Slow control and MIDAS framework

The Maximum Integrated Data Acquisition System (MIDAS) is a software package, that provides functionality for the data acquisition, monitoring and control of the Mu3e experiment. MIDAS provides a webserver which in turn allows to communicate with sub-systems through programmable front-ends. Parameters can be stored in an Online Data Base (ODB).
Communication between each part of the system from MIDAS down to each TMB and their respective MuTRiG chips and vice versa is possible via slow control. It allows to send commands, as for example MuTRiG configurations. Furthermore, data such as from power or temperature sensors can be transferred.

1.6 Field Programmable Gate Arrays

To handle such high rates the use of Field Programmable Gate Arrays (FPGAs) is advantageous as FPGAs combine the advantages of higher speed compared to software solutions and the possibility of modifying the system.

FPGAs are a type of semiconductor device that can be programmed after manufacturing to perform specific digital logic tasks. The core architecture are configurable logic blocks (CLBs), a programmable interconnect network and input/output blocks. Each CLB consists of lookup tables (LUTs), flip-flops, and multiplexers. The LUTs can be used to implement any binary logic function. The flip-flops store the results of these logic operations, enabling sequential logic operations within the FPGA. These blocks can then be interconnected via the programming of the FPGA, allowing for more complex circuits. As illustrated in figure 8 at each clock cycle data of one register stage is processed through the programmed logic and saved in the subsequent register stage at the following clock cycle [12].

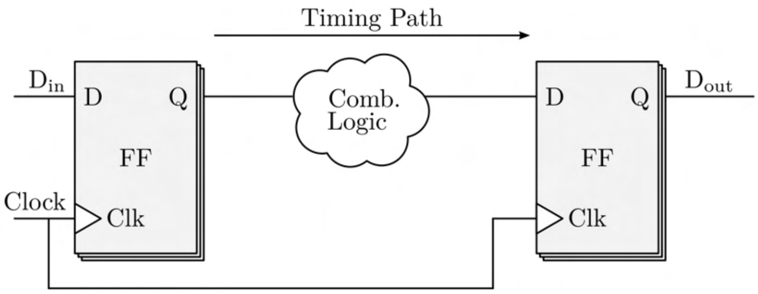


Figure 8: Synchronous Digital Design at RTL (taken from [12])

The input/output blocks enable interaction between the FPGA and external signals. Most logic is a synchronous circuit, requiring clock and reset signals. Additionally, FPGAs often include dedicated resources such as block RAM (BRAM) for memory storage or various other components.

The FPGA design can be described at the register transfer level (RTL) using hardware description languages like VHDL or Verilog. This code is then synthesized into a netlist, a low-level representation that maps the abstract design onto the concrete arrangement of logic gates and interconnects of the FPGA.

Simulation of the design can be done by simply calculating logic signal states at the following clock cycle. Besides the logic functionality, the signal timing needs to be checked for new implementations. Figure 9 shows that for proper data transfer the signals need to be defined during the clock edge including times t_{setup} before and t_{hold} after the clock edge. If data changes during the clock edge this leads to undefined values at the output [12].

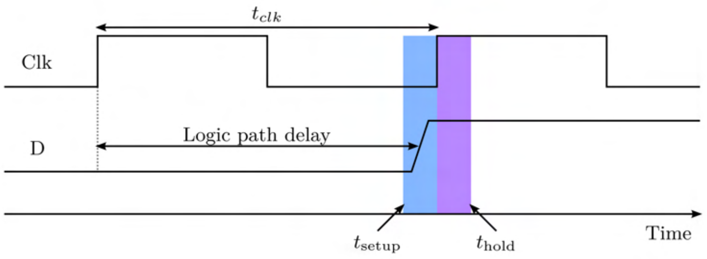


Figure 9: Setup and Hold Time Windows (taken from [12])

Such timing constraints and limitations are mostly checked automatically but need to be considered when implementing new functions in the firmware. For example, only a limited number of calculations may be done in a single clock cycle to ensure timing is held.

2 Latency studies

One key problem for the front-end firmware that needs to be taken account of is the occurrence of latency through the system. There are several sources that introduce fixed and variable latency. It is required that hits arriving on the SWB should be ordered by their timestamps. A fixed latency offsets the point at which a hit will arrive in the DAQ in reference to the global timestamp of the system. Variation of the latency leads to wrong ordering of hits, thus requiring a sorting mechanism. The variance of latency determines the size of a time span over which this mechanism needs to sort. Therefore, fixed and variable latency of hits arriving at the sorter needs to be known.

The first source of variable latency occurs during the event building on the MuTRiG as described in 1.3.2. As hits are only generated after the TDC generates an E-timestamp hits with higher energy are expected to be constructed later. This means that two hits with different energies but arriving at the same time, will be send out into the DAQ at different times. An example of this is shown in figure 10 that demonstrates that this can lead to wrong ordering.

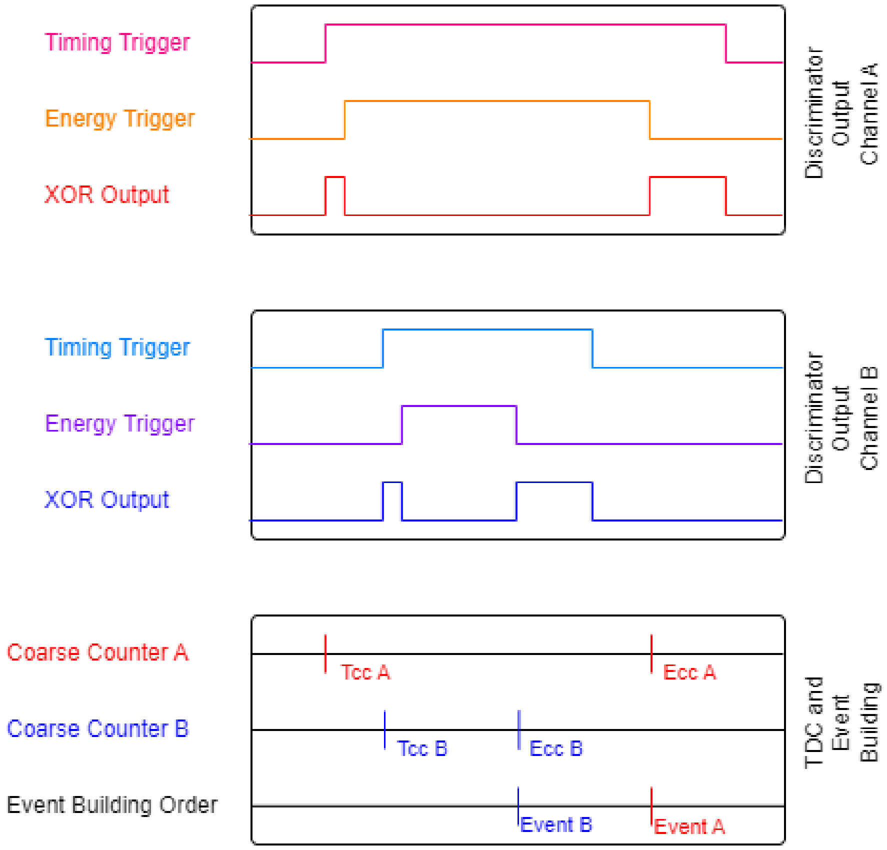


Figure 10: Hit A occurs before hit B. Since B has a lower energy, event B will constructed and send to the DAQ before hit A

This effect causes latency in the order of several 100 ns, depending on the ToT. Furthermore, the ≈ 12 µs frame rate of the MuTRiG can lead to more latency that is not fixed. In a worst-case scenario, if two events are generated in consecutive clock cycles, the first event is sent to the DAQ in one frame, while the second event is only sent in the subsequent frame. This leads to an additional latency of $0 - 12$ µs of variable latency.

In order to investigate the expected latency data of an earlier Testbeam campaign was analyzed, where no hit sorting was implemented in the firmware. The incoming hits were stored in frames as they were sent by the ASICs.

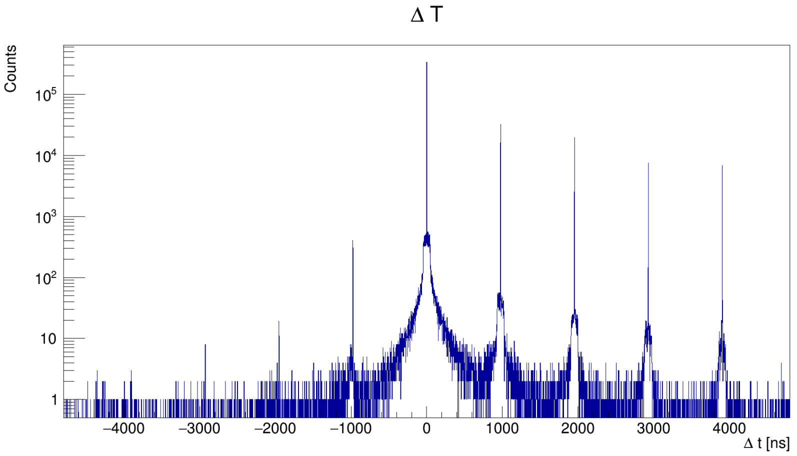


Figure 11: Testbeam Data

Hits arriving after each other in the DAQ from the same chip are compared in figure 11. For each pair of hits the difference of their T-timestamps is calculated. If hits would arrive in the DAQ sorted by their T-timestamps only positive differences would be expected. Hits however are not sorted by their timing timestamp and negative time differences can be seen, showing the above described scenario occurs commonly. Peaks at multiple of ≈ 1 µs correlate to the DESY II bunch cycle period [13]. These peaks arise because of hits generated by two distinct particles, originating from different bunches.

3 Mu3e Front-End Board Firmware

The firmware for both SciFi and SciTile detector is similar to a certain degree since both use the MuTRiG and thus allows common development of many blocks. However, there are multiple differences such as the event sizes and number of ASICs per FEB. In the following the focus will be on the SciTile firmware.

The FEB Firmware needs to accomplish several tasks. Serial data from the MuTRiGs first needs to be received on the FEB and then decoded. The goal is then to condense data from 13 ASICs to one common output. Meanwhile redundant information need to be compressed (e.g. the most significant bits of timestamps) in order to reduce the amount of transferred bits. Furthermore, the incoming data must be sorted and grouped into accurate 8 ns frames to enable processing by the filter farm. Although a concept already existed, testing showed the need for modifications and additional processing and debugging blocks. The latest concept is shown in figure 12. Received hits first get multiplexed into three groups. For each group pre-processing of the data is performed. Then all hits enter the sorter that outputs sorted hits to the SWB.

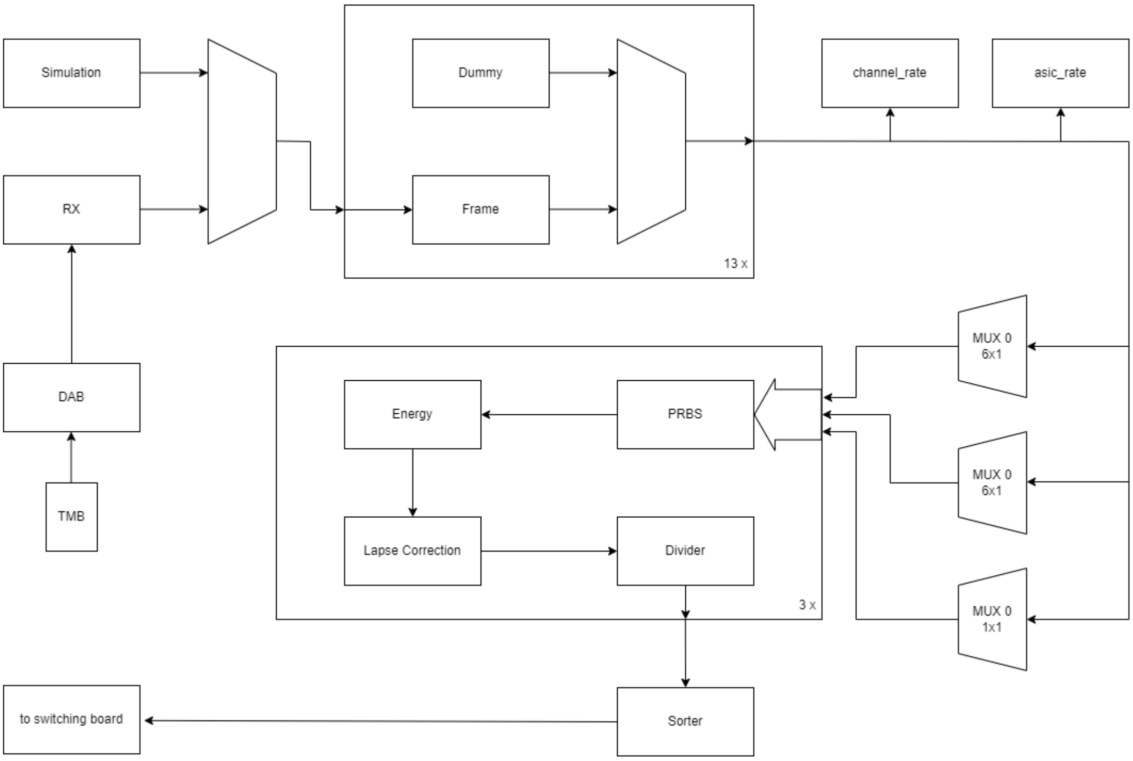


Figure 12: Tile Datapath

3.1 Frame Building

Each TMB is connected to one Detector Adapter Board (DAB) which is responsible for adapting cable to FEB pinout and cleaning of the signal. On the FEB the first module are 13 LVDS Receivers (RX) that deserialize the incoming 8b10b-encoded bitstreams into 9 bit words. The data from the ASICs is packed in frames. Those need to be unpacked by 13 frame receivers. One hit recorded by the MuTRiG has a size of 48 bits. Therefore, it takes 6 clock cycles to construct one hit. After the frame receivers, each hit contains information about the channel number, the T-timestamp and E-timestamp. The outputs of the LVDS or frame receivers can alternatively be simulated. Furthermore, each frame receiver output is tracked such that the e.g. frame rate, 8b10b error rate and the incoming rate of hits for each channel and each ASIC can be checked on the MIDAS front-end.

3.2 Multiplexing

Since only every 6 clock cycles one hit will be generated by the frame receivers, up to 6 frame receivers outputs can be grouped together and be read out sequentially without the need of additional buffering. This saves up resources as the following modules do not need to be implemented 13 times. In order to multiplex the incoming hits, for each hit information about the recording ASIC is added. Furthermore, a binary signal is added in order to flag a valid hit. Therefore, one hit contains information according to table 1.

Description	**Size [bits]**	**Additional Info**
Valid	1	
ASIC Number	4	range 0 to 12
Channel Number	5	range 0 to 31
T - Coarse Counter	15	PRBS encoded
T - Fine Counter	5	range 0 to 31 in bins of 50 ps
E - Coarse Counter	15	PRBS encoded

Table 1: Hit event after Frame Receivers

3.3 Pseudorandom binary sequence decoder

On the MuTRiG site a coarse counter which provides a 1.6 ns bin time information with ≈ 50 µs dynamic range is realized with a linear-feedback shift register that runs through $2^{15} - 1$ states. In the Pseudorandom binary sequence (PRBS) decoder block in the FEB firmware is mapped to their corresponding counter value. This mapping is applied to both T and E, hence afterwards both coarse counters are 15 bit integers ranging from 1 to 32768 representing 1.6 ns bins (note that 0 is excluded due to how a linear-feedback shift register works).

3.4 Energy module

After decoding the coarse counters the T - timestamp can be subtracted from the E - timestamp in order to calculate the ToT and hence the energy. This is the first step in the energy module. Afterwards, the 15 bit E coarse counter is no longer needed and can be disregarded. Since expected ToTs will be smaller than $512 \cdot 1.6$ ns, only 9 bits will be needed to store information about energies. Additionally, functionality, controllable via MIDAS, needed to be implemented for offsetting and rescaling the result, which is helpful for testing and calibration purposes because of the limited dynamic range. Finally any overflow will be put into the last bin. After the few clock cycles these processes take the result is merged with the hit data again according to table 2.

Description	Size [bits]	range	nominal
Valid	1		
ASIC Number	4	0-12	
Channel Number	5	0-31	
T - Coarse Counter	15	1-32767	1.6 ns
T - Fine Counter	5	0-31	50 ps
Energy (ToT)	9	0-511	1.6 ns

Table 2: Hit event after Energy Module

3.5 Lapse Correction

In order to sort the data, timestamps will be needed in a units of 8 ns. In order to achieve this a division by five will be necessary. Hence it is the goal to map the incoming coarse counter with a dynamic range of $2^{15} - 1$ to a counter with a dynamic range of $2^{12} \cdot 5 = 20480$. After the division block a counter with a dynamic range of $2^{12} \cdot 8$ ns remains. This was newly implemented in the course of this work. In order to accomplish this, the lapse correction module has an internal counter that increments in steps of 5 at a 125 MHz rate to replicate the MuTRiG counter that increments at a 625 MHz with its $2^{15} - 1$ states. Each time this counter overflows a correction term is increased by $2^{15} - 1 = 32767$. Adding this term to the coarse counter removes any overflowing behavior.

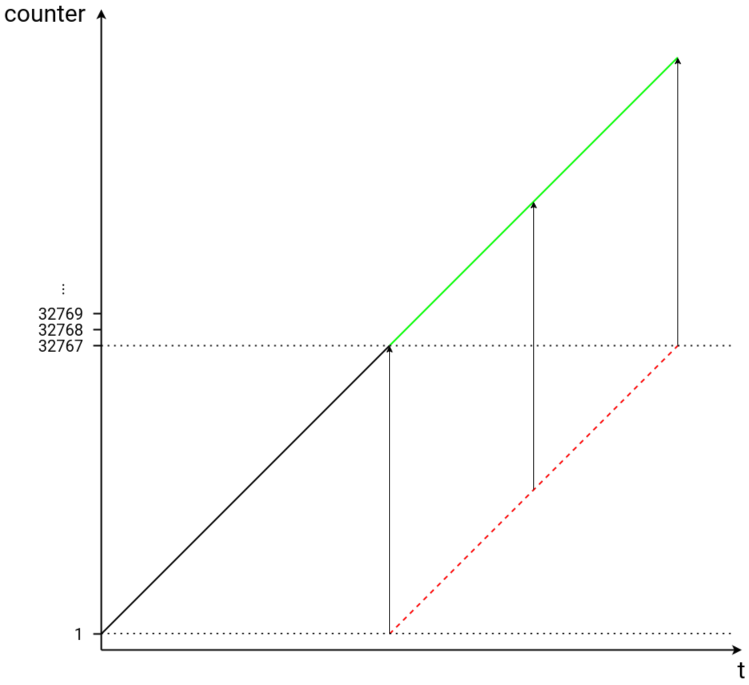


Figure 13: Base Concept of the Lapse Correction Block

This method leads to increasingly bigger values, which would also need more resources. As a first measure results ≥ 40960 can be subtracted by 40960. The similar approach can be applied to the correction term itself. As only a dynamic range of a 20480 is required, each counter ≥ 20480 can be subtracted by 20480 leading to the required counter.

However before this can be done latency arises as further complication. Since hits arrive with an uncertain delay an edge case arises when the internal counter already overflowed but the incoming hit should only be corrected by the previous correction term. This case can be catched by its characteristic of the incoming coarse counter being above an upper bound and the internal counter being below a lower bound.

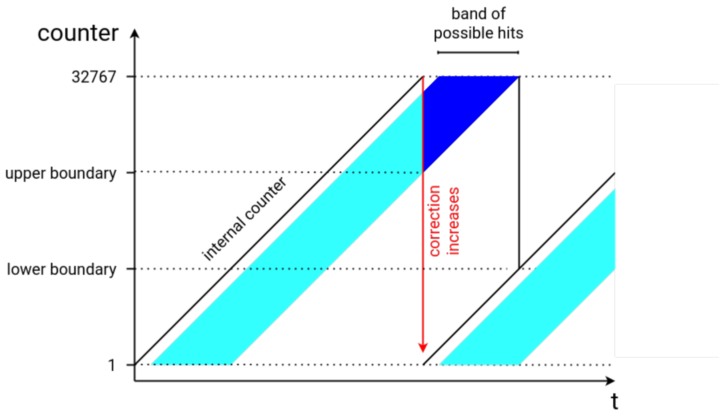


Figure 14: Lapse Correction for Edge Cases

To make sure this works properly the internal counter needs to be shifted such that it always runs ahead of the incoming hit timestamps. Additionally, with the assumption that the incoming variable latency is always less then 16 µs, upper and lower boundaries can be set accordingly. For validation purposes, the latency, that means in this specific case the time difference between the incoming hit and the internal counter can be plotted.

A previous solutions, implemented in the course of this work, included a first correction before the dividing block that accounted for the missing 2^{15}th state of the MuTRiG counter and a second correction after the divider in order to get the desired counter with 2^{12} values. Using a single solution reduces needed settings and error sources.

3.6 Timestamp Divider

The divider block takes incoming hits and divides the the 1.6 ns counter by 5 resulting in a 8 ns counter with a reminder resulting in the format presented in table 3. As described in the previous section the dynamic range of the input is specifically chosen such that no lapse issue arises after division.

Description	Size [bits]	range	nominal
Valid	1		
ASIC Number	4	0-12	
Channel Number	5	0-31	
T - Quotient	12	0-4097	8 ns
T - Remainder	3	0-4	1.6 ns
T - Fine Counter	5	0-31	50 ps
Energy (ToT)	9	0-511	1.6 ns

Table 3: Hit event after Divider

3.7 Sorter

The sorter aims to sort incoming hits according to their 8 ns timestamps. This is the most complicated block in the datapath and developed for all sub-systems. It brings output data in the form of events, containing a global timestamp as header and time sorted hits.

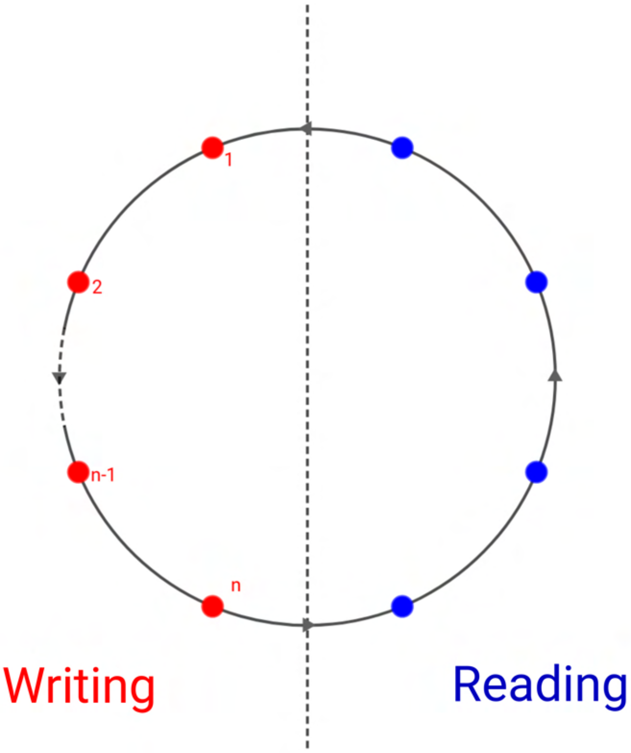


Figure 15: Sorter Schematic (inspired by [14])

The sorter works with internal bins, where each bin gets assigned a time stamp [14]. Incoming hits are filled into a bin according to their time stamp. As depicted in figure 15 every 8 ns the bin with the oldest time stamp enters the reading side where it can no longer be filled with hits. There all hits inside the bin get read out and afterwards the bin gets reset such that it can serve as a new bin with the then newest time stamp. However, the amount of bins is limited by the amount of available memory on the FPGA. The number of bins is determined by the variable latency present in the system.

Based on the ASIC frame size of around 12 µs and ToT of $(1-2)$ µs, a sorter window ≥ 14

µs is needed. Smaller additions to the variable latency e.g. through the multiplexers are only a few 125 MHz clock cycles corresponding to only O(10) ns. Those are within a safety margin with the chosen sorter window of $2048 \cdot 8$ ns $\approx$ 16 µs. This is confirmed by the data taken at the testbeam as discussed in section 2.

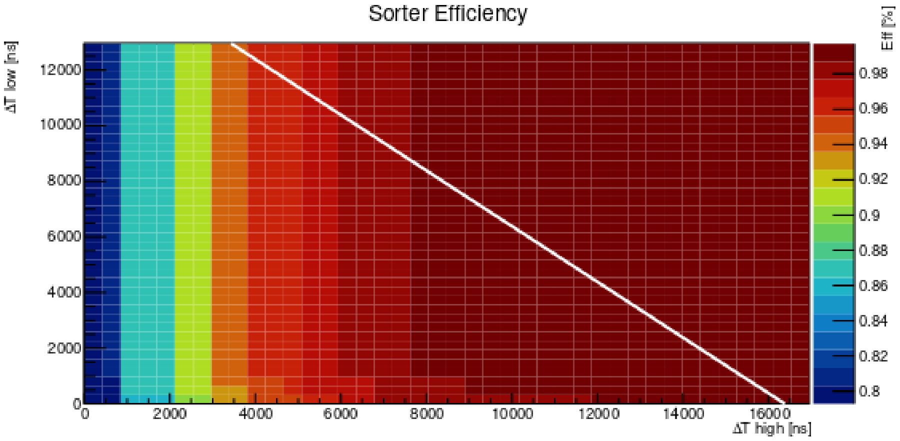


Figure 16: Sorter Efficiency, white line represents possible windows

Figure 16 show the theoretical sorter efficiencies. Different sorter windows are tested and the ratio of kept hits to incoming hits is plotted in the z dimension. The Δt high-axis represents the time by which the window extends into the future, while the Δt low-axis shows the time by which the window extends into the past. Additionally, the implemented window size is plotted. In a wide range of possible windows the sorter should be able to keep almost all hits. However, the window needs to be shifted in time, during the commissioning of the system, due to the fixed latency through the system such that it can fully cover all hits.

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Type	-	FPGA ID	K28.5	preamble
ts (47:16)				Data Header
ts (15:0)	package counter			
-	Subheader Counter	Hit Counter		Debug Word
-	Send TS Counter (30:0)			
ts(11:4)	overflow	K23.7		sub-Header
MuPix / Tile / Scifi				hit
...				hit
ts(11:4)	overflow	K23.7		sub-Header
MuPix / Tile / Scifi				hit
...				hit
overflow	-	K28.4		trailer

Figure 17: Payload (taken from [15])

Figure 17 provides a visual representation of the output packets with 32 bit words produced by the sorter. The 2 word header contains the 48 bit timestamp. Two additional debug words are sent with the header. This is followed by a sub-header containing the 4th to 11th bit of the timestamp, the overflow of the preceding sub-header block and a comma symbol. Then a list of hits follows until the lowest 4 bits of the timestamp overflow at which point a new sub-header is inserted. This repeats until the sub-header timestamps overflows and a trailer ends the packet. Those packets are then sent via optical transmission to the SWB. Therefore, a preamble with the packet type, the identification of the FPGA and a comma symbol is added.

3.8 Switching Board

The SWB receives data from multiple FEBs and combines the packets into events covering the same time frame. The SWB is equipped with an Arria 10 FPGA connected to the switching PC via Peripheral Component Interconnect Express (PCIe). For the test setup, this PC takes also the role of the farm PC and is responsible for storing incoming data as well as for slow control. Each incoming frame get stored in a seperate MIDAS event, a predefined storage structure which can then be processed further in software. The SWB saves hits from the tile detector in a specified format with 64 bits per hit:

bit	**Additional Info**
00-04	Time in 50 ps bins
05-07	Time in 1.6 ns bins (remainder)
08-31	Time in 8 ns bins
32-40	ToT in 1.6 ns
41-47	-
48-52	Channel ID
53-60	ASIC ID
61-63	0
48-63	Channel ID in global addressing scheme

Table 4: MIDAS Hit Format of Scintillating Tile Hits (taken from [15])

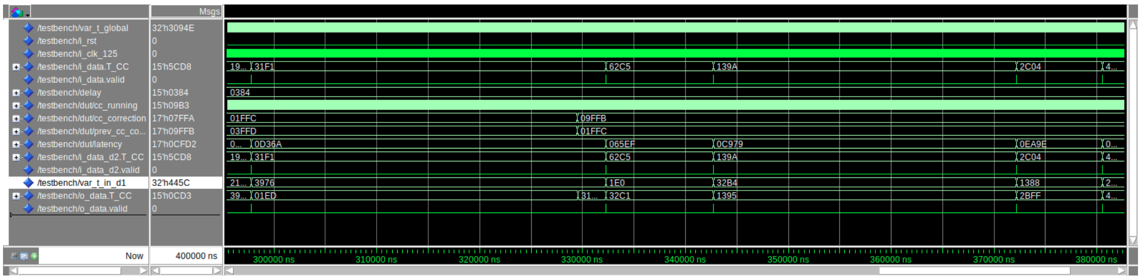
These hits can then be analyzed with software solutions.

4 Validation

Although some parts of the DAQ were already implemented, validation of the whole system needed to be done and showed opportunities for improvement.

4.1 Simulation

In order to ensure functionality of the newly implemented and modified modules they are simulated at RTL by calculating the states at each clock cycle. This allows for testing where edge cases can deliberately be created. The simulation computes each signal behavior for each simulated clock cycle as illustrated in figure 18. Simulating allows to inspect as many signals as desired but requires significant consideration to achieve good test case coverage, as each case needs to be implemented manually. In the course of this work, simulation was used to test multiplexing, energy and lapse correction block wise.

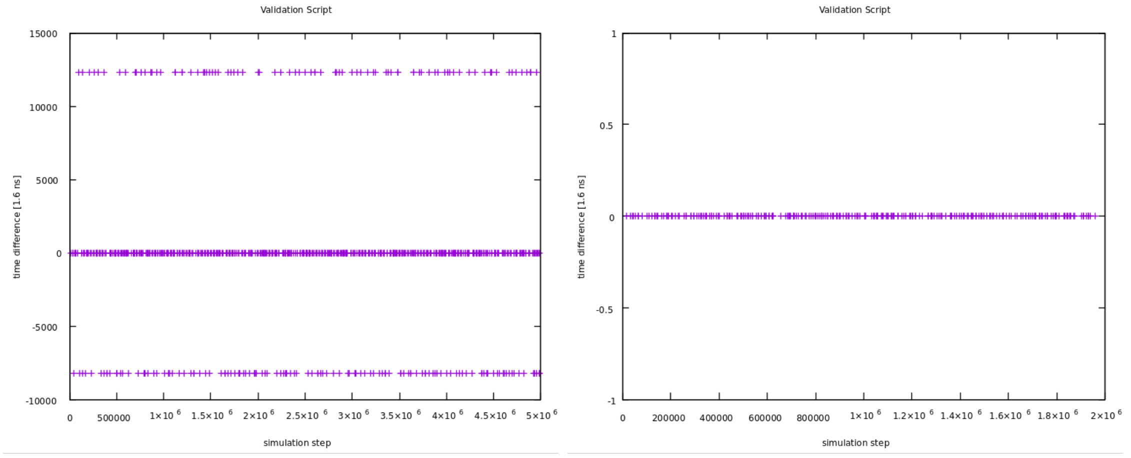


	Msgs
/testbench/var_t_global	32'h3094E
/testbench/i_rst	0
/testbench/i_clk_125	0
/testbench/i_data.T_CC	15'h5CD8
/testbench/i_data.valid	0
/testbench/delay	15'h0384
/testbench/dut/cc_running	15'h09B3
/testbench/dut/cc_correction	17'h07FFA
/testbench/dut/prev_cc_co...	17'h09FFB
/testbench/dut/latency	17'h0CFD2
/testbench/i_data_d2.T_CC	15'h5CD8
/testbench/i_data_d2.valid	0
/testbench/var_t_in_d1	32'h445C
/testbench/o_data.T_CC	15'h0CD3
/testbench/o_data.valid	0
Now	400000 ns

19... 31F1 62C5 139A 2C04 4...
0384
01FFC 09FFB
03FFD 01FFC
0... 0D36A 065EF 0C979 0EA9E 0...
19... 31F1 62C5 139A 2C04 4...
21... 3976 1E0 32B4 1388 2...
39... 01ED 31... 32C1 1395 2BFF 4...
300000 ns
310000 ns
320000 ns
330000 ns
340000 ns
350000 ns
360000 ns
370000 ns
380000 ns

Figure 18: VSIM Screenshot

The simulation also allows for additional output, such as files containing all outputs. These outputs can then be compared to the given inputs and discrepancies can be detected. The lapse correction module developed in this thesis underwent rigorous testing through simulation.



(a) wrong delay setting

(b) correct delay setting

Figure 19: Lapse Correction Validation

In figure 19 the time stamps of the outputs get subtracted from those of the inputs. With the wrong delay setting some timestamps dont match their inputs. With the correct setting no more outliers can be found.

4.2 Signal Tap logic analyzer

The Signal Tap logic analyzer is a tool that captures signal behavior in a FPGA. With the use of triggers specific signal data can be captured. While test case coverage can be easier on the real FPGA with methods described in the following, only a limited amount of signals and a limited amount of clock cycles can be captured. This was used to debug errors in all parts of the firmware and verify that incoming hits are correctly propagated through each each module of the FEB firmware.

4.3 Hit generation on MuTRiG

For further testing a setup as sketched in figure 20, was built to simulate real conditions. The goal was to generate hits on the MuTRiG. Therefore, a function generator is connected to a $C = 220\ pF$ capacitor such that the signal a SiPM produces can be simulated.

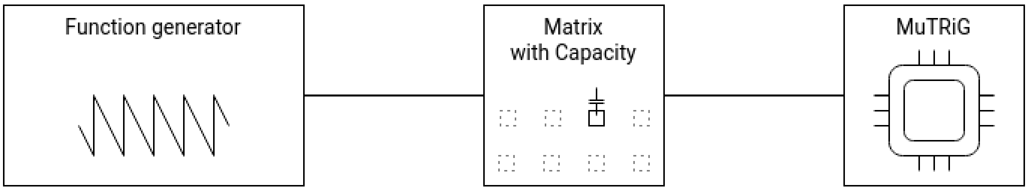
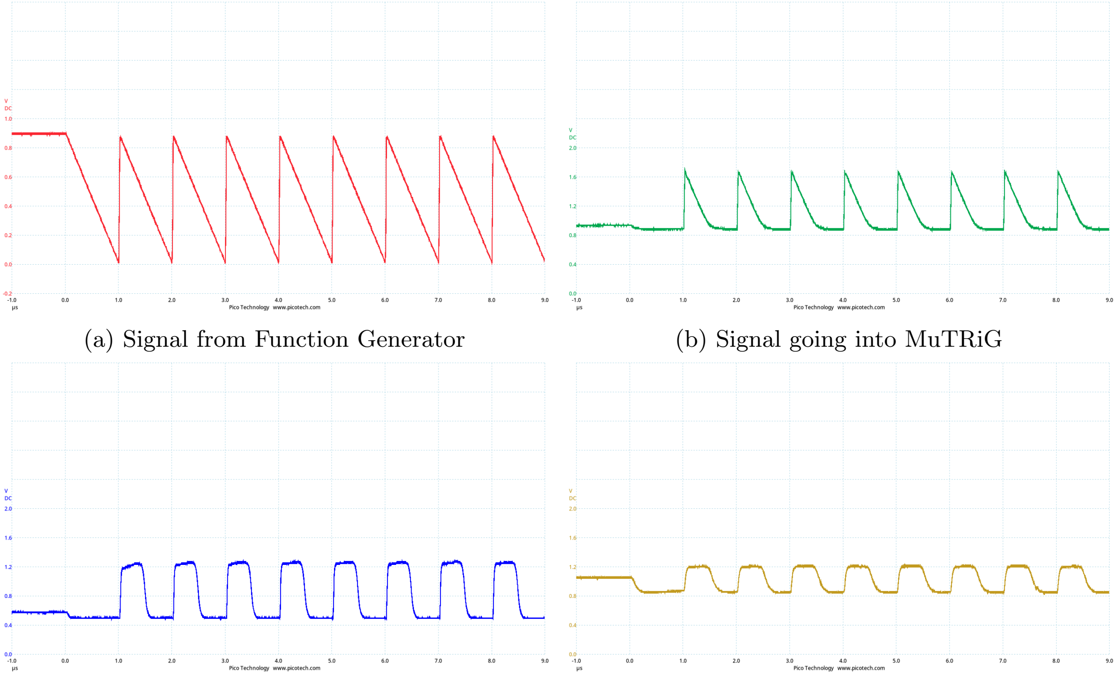


Figure 20: Schematic of Testing Circuit

Through testing ramp functions on the function generator were determined as the best solution for this simulation. The signal from the function generator and the signal at the MuTRiG are shown in figure 21. Additionally, the T- and E- Trigger signals that get injected into the XOR logic are captured.



(a) Signal from Function Generator

(b) Signal going into MuTRiG

(c) T-Trigger

(d) E-Trigger

Figure 21: Hit Generation on MuTRiG

Alternatively a pulse injection into the MuTRiG hitlogic is possible. The pulses get then turned into T- and E-timestamps from the TDC. These pulses are generated on the FEB at a constant rate of 100 kHz and can be distributed to each channel. In the following injection rates of 100 kHz always refer to this method.

4.4 Measurements

Since the Charge of the capacitor $Q = C \cdot U$ depends on the input Voltage U, the ToT measured by the MuTRiG can be varied by modifying the amplitude of the signal from the function generator.

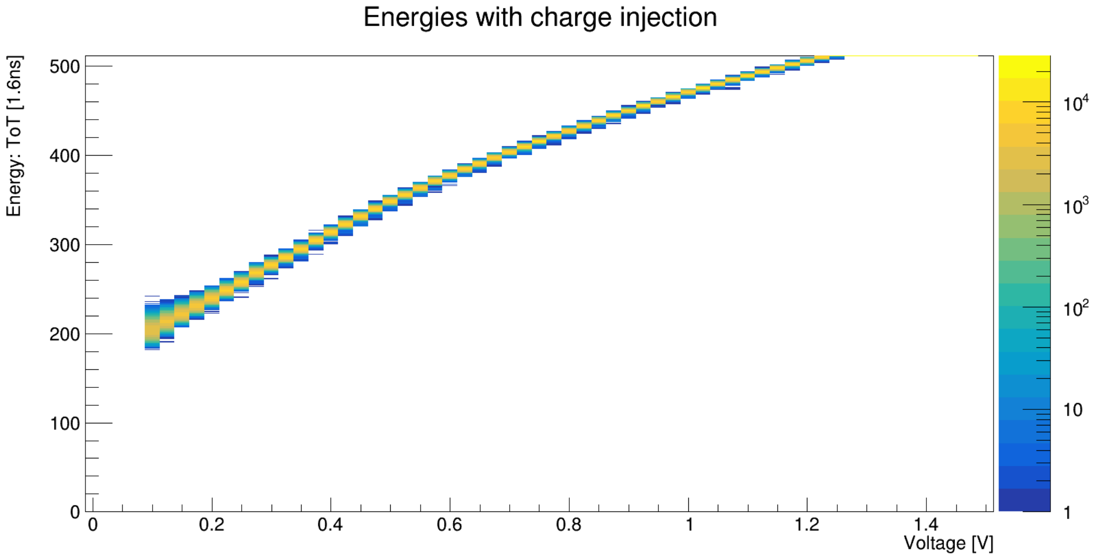


Figure 22: Measured Energies/Tots for different Amplitudes of the Function Generator

MIDAS provides the possibility to execute sequencer script. This was used to run different voltages and evaluate the corresponding ToT representing different energies of incoming hits as shown in figure 22.

Limited capability of the sorter can be seen when injecting hits at a 100 kHz. The monitored hit rates after the frame receivers corresponds to the input rate. No saturation effects can be seen up to this point. However saturation effects can be observed in figure 23.

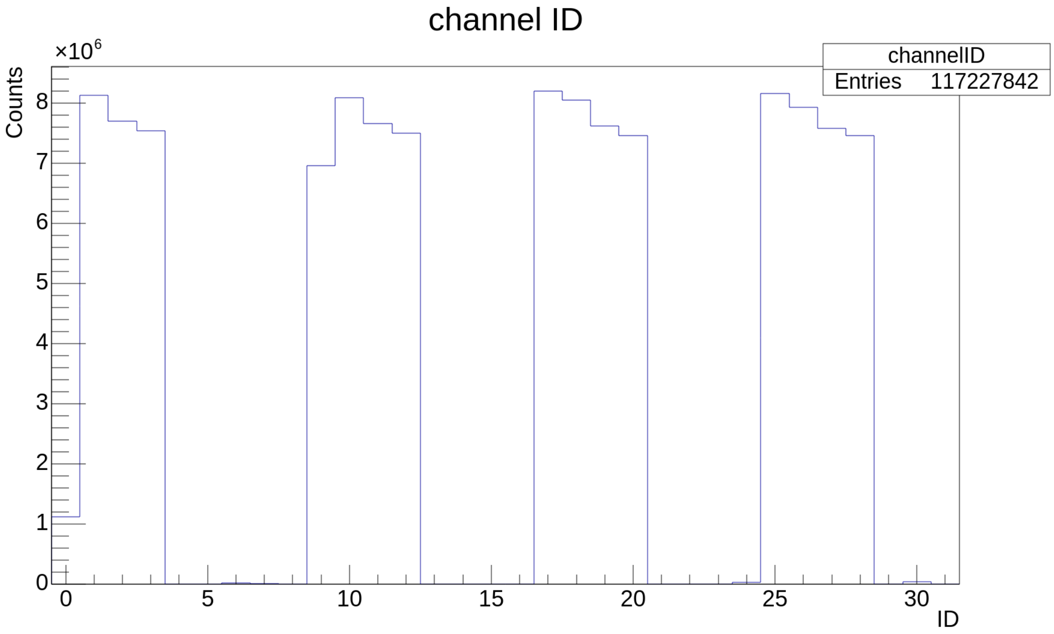
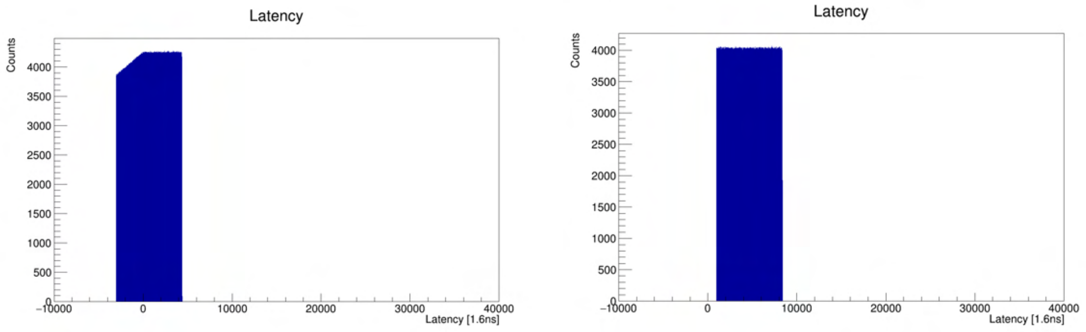


Figure 23: Channel Hits

Even though every channel produces the same amount of hits, after going through the sorter each channel seems to produce hits at different rates. For further sorter testing, counters were implemented to monitor incoming and outgoing hits. Since there are still problems in the sorter firmware only limited testing is possible at the moment.

In order to set the delay setting for the lapse correction according to the fixed latency of hits going into this block, functionality is implemented to plot latency of hits compared to the internal counter of the module as demonstrated in figure 24. Latency should always be positive but kept near zero to ensure that both the lowest and highest latency hits are handled correctly by the lapse correction. Increasing the delay setting lets the internal counter start later and thus latency is decreased for each hit resulting in latencies being shifted towards the negative direction.



(a) Wrong Setting with negative Latencies

(b) Correct Setting

Figure 24: Different Delay Settings

When latency is correctly set most output hits should have the same time difference as incoming hits, however the sorter produces a few amount of duplicate hits as presented in figure 25.

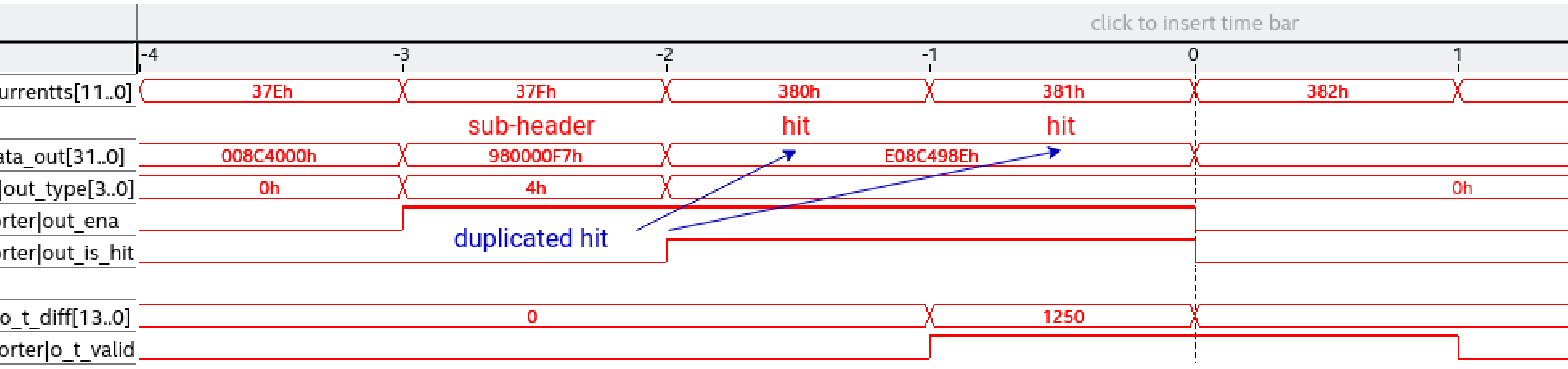
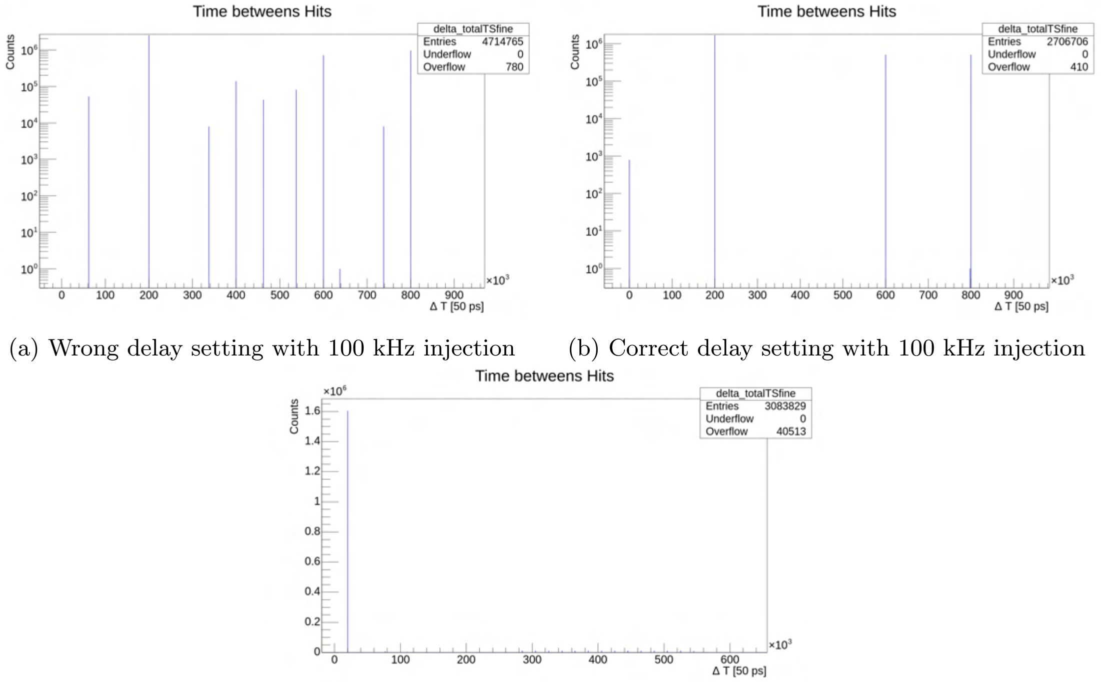


Figure 25: Duplicate Hit captured in Signaltap

This leads to time differences of 0 ns. Additionally, some hits are lost leading to time differences at the multiple of the incoming time difference. Figure 26 firstly shows measured time differences with wrong latency while injecting hits at a constant rate of 100 kHz. Secondly the delay setting was adjusted and only at time differences of zero or multiple of 10 µs can be seen, therefore matching to the input rate. For the third plot, hits with a rate of 1000 kHz were injected with the function generator and the expected time difference of 1 µs can be seen with almost no outliers. Why this method seems to perform better in regard to duplicate and lost hists, needs further investigation.



(a) Wrong delay setting with 100 kHz injection

(b) Correct delay setting with 100 kHz injection

(c) Correct delay setting with injection of hits with $\Delta t = 1$ µs through the function generator

Figure 26: Time Differences

A sequencer script is used to determine the ideal delay setting and thus makes it possible to find the fixed latency of hits going into the sorter . A constant input rate of hits is fed into the DAQ for a fixed amount of time and the number of outgoing hits is measured for different delay settings.

The results are plotted in figure 27.

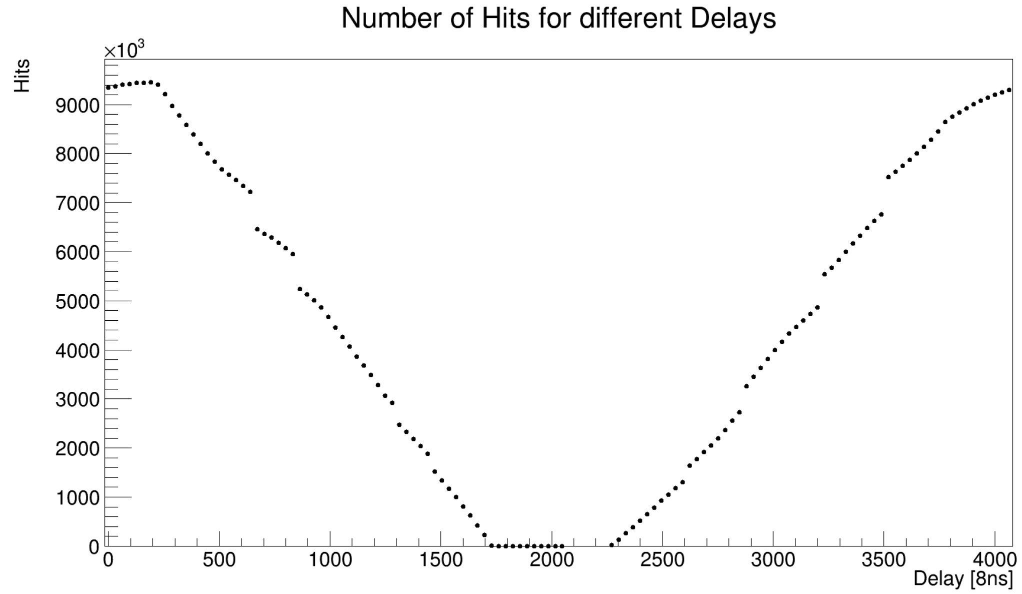


Figure 27: Output Rate for different Delay Settings

This test can be used to determine the optimal delay setting. However, even for the best setting, with the help of the implemented counters it can be determined that the sorter only achieves ≈ 60% efficiency. Since the sorter window covers 16 µs and variable latency is around 12 µs a plateau of $32 - 16 - 12 = 4$ µs with no recorded hits is expected and is seen in the data.

To confirm the assumption that latency is ≤ 16 µs another test is performed. The energy of incoming hits is expected to be correlated to the latency. Therefore, different energies were simulated according to the results depicted in figure 22 and compared to the maximal latency.

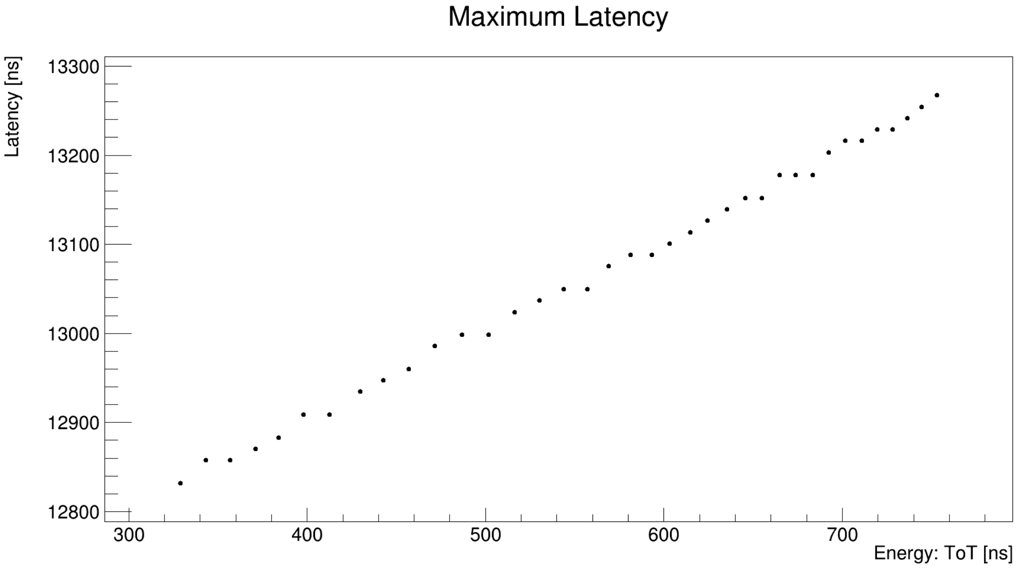


Figure 28: Latency dependence of Energy

Figure 28 demonstrates a linear correlation between the maximum latency and the measured ToT. This shows that even for high energies the assumption that latency does not get $\geq$ 16 µs stands and thus boundaries for the lapse correction do not need to be bigger than this size. Additionally, it confirms that a sorter window of 16 µs should be able to cover all incoming hits.

5 Summary

The Mu3e experiment conducted at the Paul Scherrer Institute in Switzerland searches the in the Standard Modell forbidden lepton-flavor violating decay $\mu^+ \to e^+e^-e^+$, which, if detected, would provide strong evidence for physics beyond the Standard Model. To detect this rare decay, a detector with excellent momentum, timing and vertex resolution is required. In order to reach the targeted sensitivity, of setting an upper limit to the branching ratio of 10^{-16}, high muon rates are required. Therefore, 10^8 muon decays per second will be detected. However, the high rates at which the experiment is running provide significant challenges to the Data Acquisition system.

The Mu3e data acquisition system is based on a triggerless readout scheme that sends all data packages to a GPU based filter farm. This thesis puts its focus on the validation of the front-end firmware for the Scintillating Tile Detector, the sub-system aiming to achieve the best timing resolution. The front-end firmware decodes data coming from the MuTRiG chips of the detector and then combines this data. Multiple processes reduce used bandwidth and transform the incoming data to allow sorting. Afterwards, incoming hits are sorted and finally transfered to the SWB, the next stage of the Mu3e DAQ system.

One aspect that needs to be taken account for, is latency introduced by different components in the signal processing chain. Both fixed and variable latencies must be accounted for in the Front-End Board firmware. Studies on the variable latency confirm that the aforementioned does not exceed 16 µs, which is crucial for proper working of the lapse correction and sorter.

As part of this work, the lapse correction was developed and other parts of the existing firmware were extended to obtain functionality for testing and calibration purposes. Those blocks were validated in a first step with the help of simulation software.

Additionally, real-world testing was performed for the entirety of the datapath using a setup to generate hits on the MuTRiG chips. A function generator was used to mimic the signals produced by SiPMs, allowing for controlled testing of the firmware's ability to process and sort hits accurately.

Although the sorter still needs improvements, other parts of firmware and notably the newly developed lapse correction block allow reliable performance in both simulations and real-world tests. Calibration methods for both the sorter and the lapse correction block could be realised. Those can be used in the commissioning of the detector system. The successful validation of the firmware is a key step toward the realization of the Mu3e experiment and the operation of the tile sub-system.

Appendix

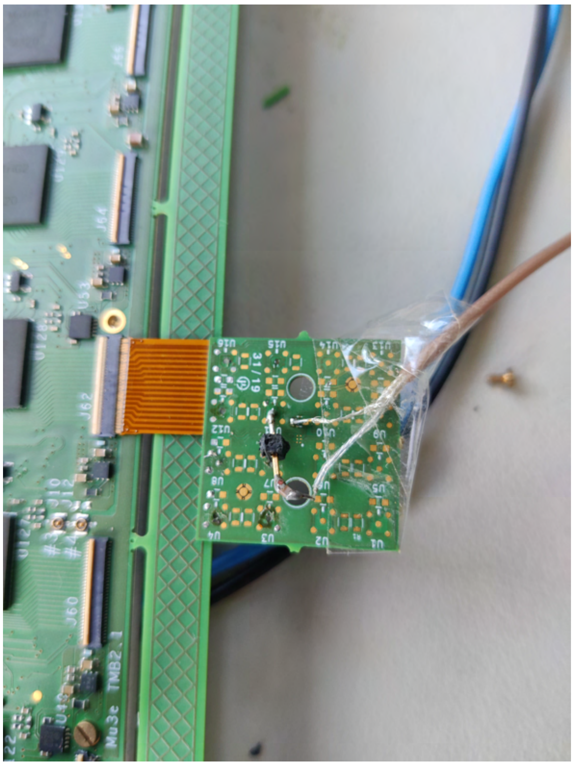


Figure 29: Capacitor attached to matrix

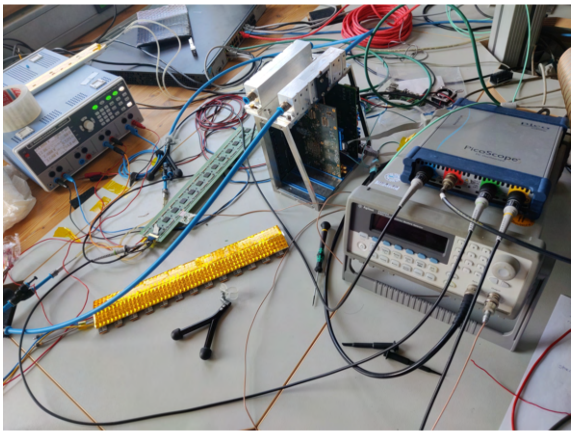


Figure 30: Test Setup

References

[1] Mark Thomson. *Modern Particle Physics.* Cambridge University Press, 2013.

[2] K. Arndt et al. Technical design of the phase i mu3e experiment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 1014:165679, October 2021.

[3] A. Blondel, A. Bravar, M. Pohl, S. Bachmann, N. Berger, M. Kiehn, A. Schöning, D. Wiedner, B. Windelband, P. Eckert, H. C. Schultz-Coulon, W. Shen, P. Fischer, I. Perić, M. Hildebrandt, P. R. Kettle, A. Papa, S. Ritt, A. Stoykov, G. Dissertori, C. Grab, R. Wallny, R. Gredig, P. Robmann, and U. Straumann. Research proposal for an experiment to search for the decay $\mu \to eee$, 2013.

[4] G. Hernández-Tomé, G. López Castro, and P. Roig. Flavor violating leptonic decays of τ and μ leptons in the Standard Model with massive neutrinos. *Eur. Phys. J. C*, 79(1):84, 2019. [Erratum: Eur.Phys.J.C 80, 438 (2020)].

[5] U. Bellgardt et al. Search for the decay$\mu \to e + e + e$. *Nuclear Physics B*, 299(1):1–6, 1988.

[6] Ivan Perić. A novel monolithic pixelated particle detector implemented in high-voltage cmos technology. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 582(3):876–885, 2007. VERTEX 2006.

[7] Stefan Gundacker and Arjan Heering. The silicon photomultiplier: fundamentals and applications of a modern solid-state photon detector. *Physics in Medicine & Biology*, 65(17):17TR01, aug 2020.

[8] Wei Shen. *Development of High Performance Readout ASICs for Silicon Photomultipliers (SiPMs).* PhD thesis, Universität Heidelberg, 2012.

[9] H. Chen, K. Briggl, P. Eckert, T. Harion, Y. Munwes, W. Shen, V. Stankova, and H.C. Schultz-Coulon. Mutrig: a mixed signal silicon photomultiplier readout asic with high timing resolution and gigabit data link. *Journal of Instrumentation*, 12(01):C01043, January 2017.

[10] Huangshan Chen. *A Silicon Photomultiplier Readout ASIC for the Mu3e Experiment.* PhD thesis, Universität Heidelberg, 2018.

[11] Heiko Augustin, Niklaus Berger, Alessandro Bravar, Konrad Briggl, Huangshan Chen, Simon Corrodi, Sebastian Dittmeier, Ben Gayther, Lukas Gerritzen, Dirk Gottschalk, Ueli Hartmann, Gavin Hesketh, Marius Köppel, Samer Kilani, Alexandr Kozlinskiy, Frank Meier Aeschbacher, Martin Müller, Yonathan Munwes, Ann-Kathrin Perrevoort, Stefan Ritt, André Schöning, Hans-Christian Schultz-Coulon, Wei Shen, Luigi Vigani, Dorothea vom Bruch, Frederik Wauters, Dirk Wiedner, and Tiancheng Zhong. The mu3e data acquisition. *IEEE Transactions on Nuclear Science*, 68(8):1833–1840, 2021.

[12] Tobias Harion. *The STiC ASIC High Precision Timing with Silicon Photomultipliers.* PhD thesis, Universität Heidelberg, 2015.

[13] R. Diener, J. Dreyling-Eschweiler, H. Ehrlichmann, I.M. Gregor, U. Kötz, U. Krämer, N. Meyners, N. Potylitsina-Kube, A. Schütz, P. Schütze, and M. Stanitzki. The desy ii test

beam facility. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 922:265–286, April 2019.

[14] Ann-Kathrin Perrevort. *Sensitivity Studies on New Physics in the Mu3e Experiment and Development of Firmware for the Front-End of the Mu3e Pixel Detector*. PhD thesis, Universität Heidelberg, 2018.

[15] Mu3e collaboration. The mu3e requirements specification book. unpublished.

Acknowledgments

At first, I would like to thank Prof. Dr. Schultz-Coulon for the opportunity to write my Bachelor Thesis in his group. Furthermore, I am also thankful that Prof. Dr. Herrmann agreed on being my second examiner.

Thank you to everyone in the group for welcoming me into your group and for giving me a great time. I especially want to thank Dr. Konrad Briggl for his support and guidance. I also want to thank Dr. Elizaveta Nazarova and Erik Steinkamp for the many helpful and informative discussions.

I would also like to thank all of my friends for consistently bringing joy into my life. Last but not least I want to thank my parents Isabelle and Frank for always supporting me.

Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 5. September 2024,

Y. Prianon