

Institute for Hadronic Structure and Fundamental Symmetries  
School of Natural Sciences  
Technical University of Munich

# **Development of FPGA based frontend electronics for the scintillating fiber hodoscope of the AMBER experiment at CERN**

**Tim Maehrholz**

Bachelor's Thesis

Supervisor:

**Prof. Dr.**

Chair of

Second Examiner:

PD Dr.

January 2025



---

## Abstract

---

The AMBER experiment at CERN aims to measure the proton radius by scattering muons on protons. The scintillating fiber hodoscope (SFH) is a key component of the PRM experiment, providing high precision time measurements of the incoming and scattered muons. The frontend electronics of the SFH, responsible for the detector readout, are FPGA-based and utilize the Citiroc1A ASIC for signal processing.

I developed the FPGA firmware for the control and readout of the Citiroc1A ASIC of the frontend electronics of the SFH.

I performed a threshold scan in order to characterize the frontend electronics and test the developed firmware.

The detailed structure of the firmware and the full characterization of the frontend electronics are presented in this thesis.

I conclude that the developed firmware is capable of controlling and reading out the signals from the Citiroc1A ASIC, but that the frontend electronics are currently not performing as desired. I propose further research and steps to investigate and understand the behaviour of the frontend electronics of the scintillating fiber hodoscope and complete it's development.



---

## Contents

---

<b>1. Introduction</b>	<b>1</b>
<b>2. The AMBER Experiment</b>	<b>3</b>
2.1. Measurment of the Charge Radius of the Proton (PRM) . . . . .	3
2.1.1. Previous Measurements of the Proton Radius . . . . .	3
2.1.2. Elastic Scattering of Muons on Protons . . . . .	3
2.2. General Setup for PRM at AMBER. . . . .	4
2.2.1. Detectors for PRM . . . . .	4
2.2.2. Scintillating Fiber Hodoscope(SFH) . . . . .	7
<b>3. Frontend Electronic of the Scintillating Fiber Hodoscope</b>	<b>9</b>
3.1. Field Programmable Gate Arrays (FPGA) . . . . .	9
3.2. Overview of the Frontend Electronics . . . . .	9
3.2.1. Proccesing of the SFH Signal . . . . .	9
3.3. The Citiroc1A ASIC . . . . .	12
3.3.1. Signal Proccesing of the Citiroc1A . . . . .	12
3.3.2. Configuration of the Citiroc1A . . . . .	13
3.3.3. The iFTDC . . . . .	17
<b>4. Development of the FPGA Firmware for the SFH</b>	<b>19</b>
4.1. Overview of the Firmware . . . . .	19
4.1.1. The IPBUS Protocol . . . . .	19
4.1.2. The Firmware Structure . . . . .	20
4.2. The Address Space . . . . .	21

## Contents

---

4.3.	Configuration of the Citroc1A ASIC . . . . .	22
4.3.1.	Status and Control Register . . . . .	22
4.3.2.	Configuration State Machine . . . . .	23
4.3.3.	Verification State Machine . . . . .	25
4.4.	The Provisional Readout . . . . .	26
4.5.	Testing the Firmware . . . . .	26
4.5.1.	Setup for the Noise Measurement . . . . .	26
4.5.2.	Theoretical Backround for the Noise Measurement . . . . .	28
<b>5.</b>	<b>Measurement Results</b>	<b>31</b>
5.1.	Noise Measurements . . . . .	31
5.2.	Frontend Electronics Characterization . . . . .	36
5.2.1.	Firmware Evaluation . . . . .	36
5.2.2.	S-curve Analysis . . . . .	37
5.2.3.	Comparison with Evaluation Board . . . . .	41
5.2.4.	Behavior at different Gain Setting . . . . .	42
5.2.5.	Channel Behavior Analysis . . . . .	42
5.2.6.	Summary of the Characterisation . . . . .	42
<b>6.</b>	<b>Conclusion and Outlook</b>	<b>45</b>
6.1.	Personal Contribution . . . . .	45
6.2.	Conclusion . . . . .	45
6.3.	Outlook . . . . .	45
<b>Appendix A.</b>	<b>Configurable Registers of the Citroc1A ASIC</b>	<b>47</b>
<b>Appendix B.</b>	<b>Fit Parameters of the S-Curve Analysis</b>	<b>55</b>

# CHAPTER 1

---

## Introduction

---

"What we observe is not nature itself, but nature exposed to our method of questioning." [Werner Heisenberg]<sup>[9]</sup>

Progress in particle physics has always been driven by the desire to understand the fundamental building blocks of our universe.

Our current best theory for the innerworkings of our world, the standard model of particle physics tells us, that the matter we see around us is mostly made up of down and up quarks and electrons. Combinations of these quarks, held together by the strong nuclear force form the proton and neutron, the nuclei of the atoms that make up the matter of the everyday world.<sup>[15]</sup> Even though the proton was discovered over a century ago by Ernest Rutherford<sup>[13]</sup>, it still holds several mysteries that continue to puzzle physicists. One of these mysteries is the size of the proton.

The semantic meaning of size in the realm of particle physics is not as straight forward as in the macroscopic world. An answer to the question, what is the size of the proton can be given by looking at the charge radius of the proton, first measured in 1956 by E. E. Chambers and R. Hofstadter.<sup>[4]</sup>

The radius of the proton has been measured several times more since then, with different methods, such as electron-proton scattering experiments and the lamb shift in muonic and ordinary hydrogen. The results of these measurements differ by five standard deviations, giving rise to the so called proton radius puzzle.<sup>[1]</sup>

The proton radius measurement (PRM) experiment at AMBER at CERN aims to help resolve this puzzle by measuring the proton radius with a new method, the elastic scattering of muons on protons.

To achieve this, the PRM experiment will measure the cross section of this scattering process and from this extract the form factors of the proton, which in turn allows for the calculation of the proton radius.

The scintillating fiber hodoscope (SFH) is a key component of the PRM experiment, as it provides precise time measurements of the incoming and scattered muons, needed for particle distinction in pile-ups in the Pixel detectors .[\[11\]](#)

This thesis will focus on the development of the FPGA driven frontend electronics of the scintillating fiber hodoscope (SFH), designed at the Technical University of Munich by the Physics Department E18 for the proton radius measurement at AMBER at CERN. Especially on the development of the FPGA firmware required for the control and readout of the Citroc1A ASIC, a crucial part of the readout electronics.

The framework of this thesis is structured in order to provide the necessary background information for the development of the frontend electronics of the SFH.

The next chapter gives a general overview of the PRM experiment and the theoretical background of the proton radius measurement.

In chapter 3 the frontend electronics of the SFH, with a focus on the functionality and behaviour of Citroc1A ASIC are described.

Chapter 4 explains the developed FPGA firmware for the control and readout of the Citroc1A ASIC, as well as give a short overview of the setup that was used for testing the developed firmware and characterizing the frontend electronics.

Chapter 5 presents the results of the testing of the developed firmware and the performance of the frontend electronics of the SFH.

The final chapter of this thesis gives a short summary of the results and an outlook on the future development of the frontend electronics of the SFH, necessary for it to successfully contribute to the resolution of the proton radius puzzle.

# CHAPTER 2

---

## The AMBER Experiment

---

### 2.1. Measurement of the Charge Radius of the Proton (PRM)

The proton is a baryon, a composite particle made up of one down and two up quarks. From this follows that the proton is not a point particle, but has an internal structure.[15] The internal structure can be described by the structure functions of the proton, the electric and magnetic form factors  $G_E$  and  $G_M$ .[1]

#### 2.1.1. Previous Measurements of the Proton Radius

The charge radius of the proton has been measured several times before with different methods. The two premier methods are electron proton scattering experiments and the Lamb shift in muonic and ordinary hydrogen. The results of these measurements differ by five standard deviations as shown in Figure 2.1, this has given rise to the so called proton radius puzzle.[1]

#### 2.1.2. Elastic Scattering of Muons on Protons

The AMBER PRM experiment at CERN aims to help resolve the puzzle, by measuring the elastic scattering of muons on protons. The first order cross section, taking into account

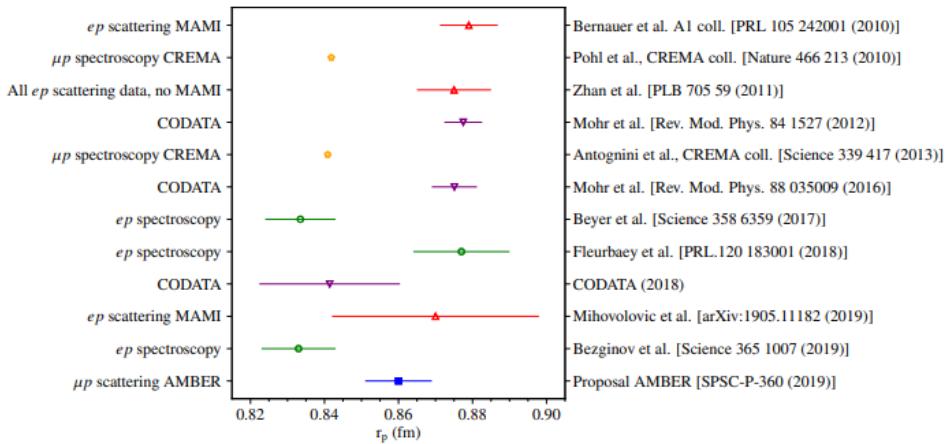


Figure 2.1.: Previous measurements of the proton radius from electron proton scattering experiments and the Lamb shift in muonic and ordinary hydrogen, the measurements differ from each other by five standard deviations. An arbitrary result for the proton radius by the PRM experiment, with the expected uncertainty is included.[1]

only interactions where one virtual photon was exchanged, for the elastic scattering of muons on a proton target is

$$\frac{d\sigma}{dQ^2} = \frac{\pi\alpha^2}{Q^4 m_p^2 p_\mu^2} \left[ (G_E^2 + \tau G_M^2) \frac{4E_\mu^2 m_p^2 - Q^2(s - m_\mu^2)}{1 + \tau} - G_M^2 \frac{2m_\mu^2 Q^2 - Q^4}{2} \right] \quad (2.1)$$

with  $Q^2 = -q^2$  the squared transferred four-momentum,  $\tau = Q^2/4m_p^2$ ,  $s = (p_\mu + p_p)^2$ ,  $G_E$  the electric form factor of the proton,  $G_M$  the magnetic form factor of the proton and  $\alpha$  the fine structure constant.[2] Through determining the form factor  $G_E$  for small  $Q^2$ , the charge radius of the Proton can be claculated with the following equation[2]

$$r_p^2 = -6 \frac{dG_E}{dQ^2} \Big|_{Q^2=0} \quad (2.2)$$

Since the cross section can not be measured at  $Q^2 = 0$ , the form factors  $G_E$  and  $G_M$  have to be extrapolated to  $Q^2 = 0$ . This is done by fitting the form factors to the experimental data and then extrapolating to  $Q^2 = 0$ .[2]

## 2.2. General Setup for PRM at AMBER.

### 2.2.1. Detectors for PRM

To determine the magentic  $G_M$  and electric form  $G_E$  factors of the proton and thus the charge radius of the Proton, the experimental cross section of the elastic scattering of

## 2.2. General Setup for PRM at AMBER.

muons on protons has to be measured. The general setup of the PRM experiment, with focus on the new detectors needed for the proton radius measurement, is shown in Figure 2.2.

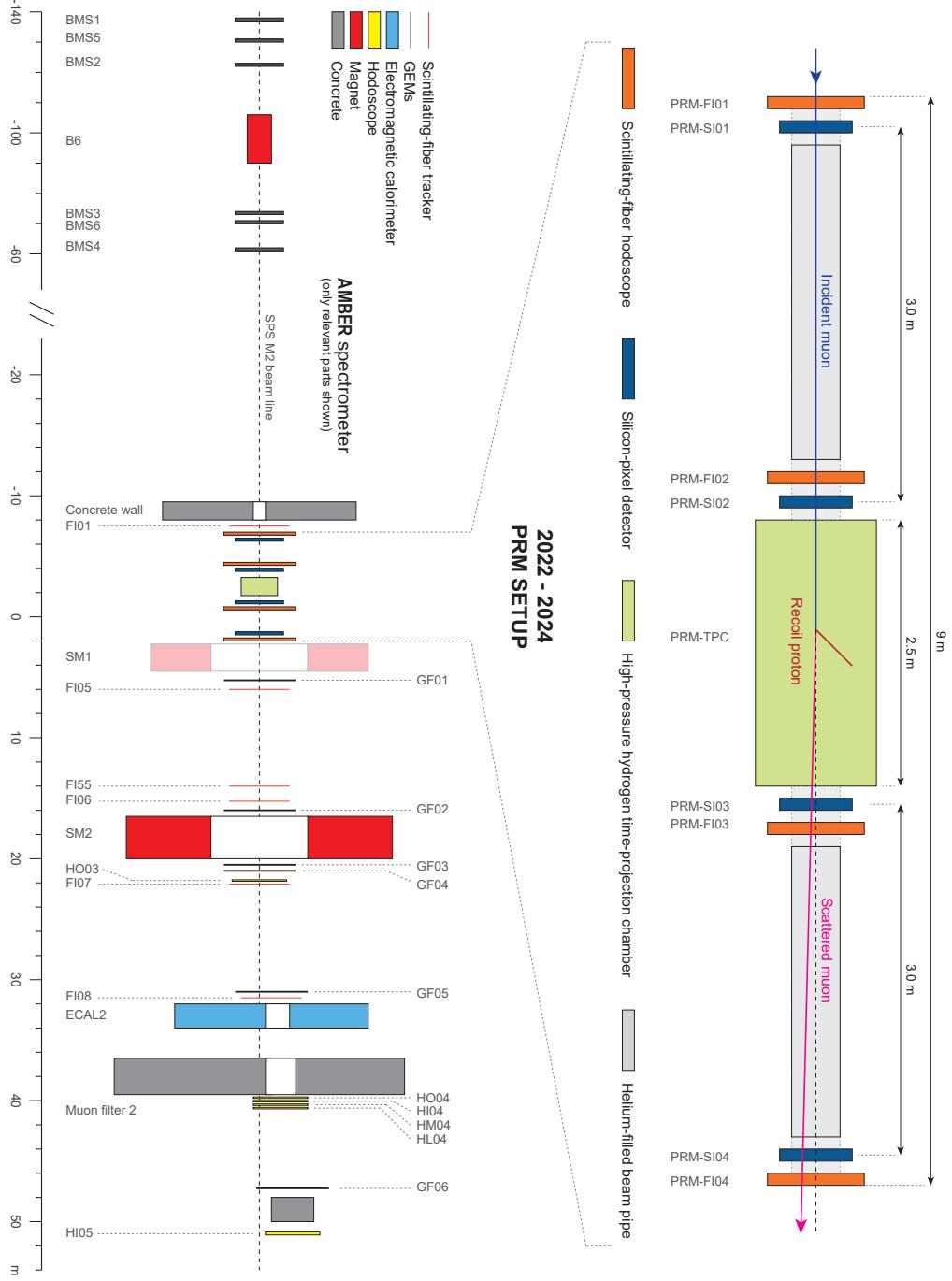


Figure 2.2.: General setup of the Amber experiment with new detectors for PRM.[7]

The incoming muon beam with an energy of 100 GeV[1] and an beam rate of  $2 \times 10^6$ [3] particles per second is scattered on a pressurized hydrogen gas target, located in the

Time Projection Chamber (TPC), which also acts as the detector for the recoil path of the scattered proton.

The reconstruction of the path of the muon is achieved through the usage of two detector types, combined into one unified tracking station (UTS) as shown in Figure 2.3.

Each UTS consists of three layers of pixilated silicon detectors (ALPIDEs), for precise positional measurements (spacial resolution of about  $8 \mu\text{m}$ [8]) of the incoming and scattered muons, but they lack, with their time uncertainty of  $5 \mu\text{s}$ [8] the required time resolution for the PRM experiment. For this reason each UTS includes a scintillating fiber hodoscope (SFH), the detector of interest for this thesis, which provides the time precision( $300 \text{ ps}$ [8]) for the measurement.

Four of these unified tracking stations, two before and two after the active target, are placed in the beamline as shown in Figure 2.2. The measurement of the momentum of the scattered muon is done by existing COMPASS detectors located after the, for the PRM newly included, detectors[1].

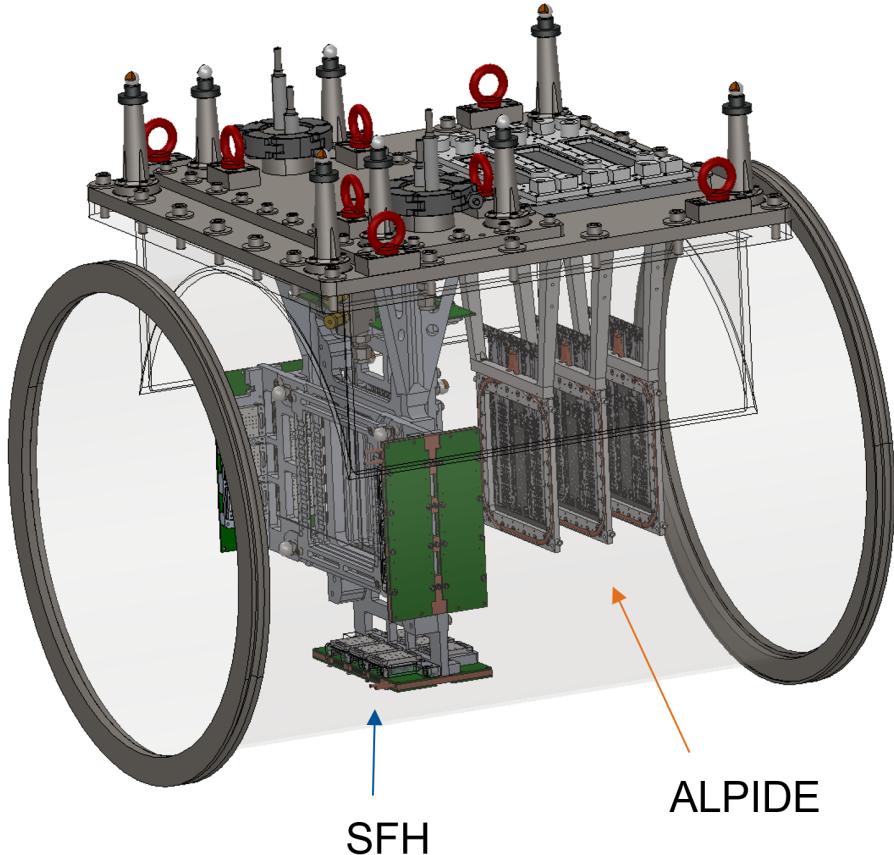


Figure 2.3.: Unified tracking station (UTS) with three layers of pixilated silicon detectors (ALPIDEs) and the scintillating fiber hodoscope (SFH).[7]

---

### 2.2.2. Scintillating Fiber Hodoscope(SFH)

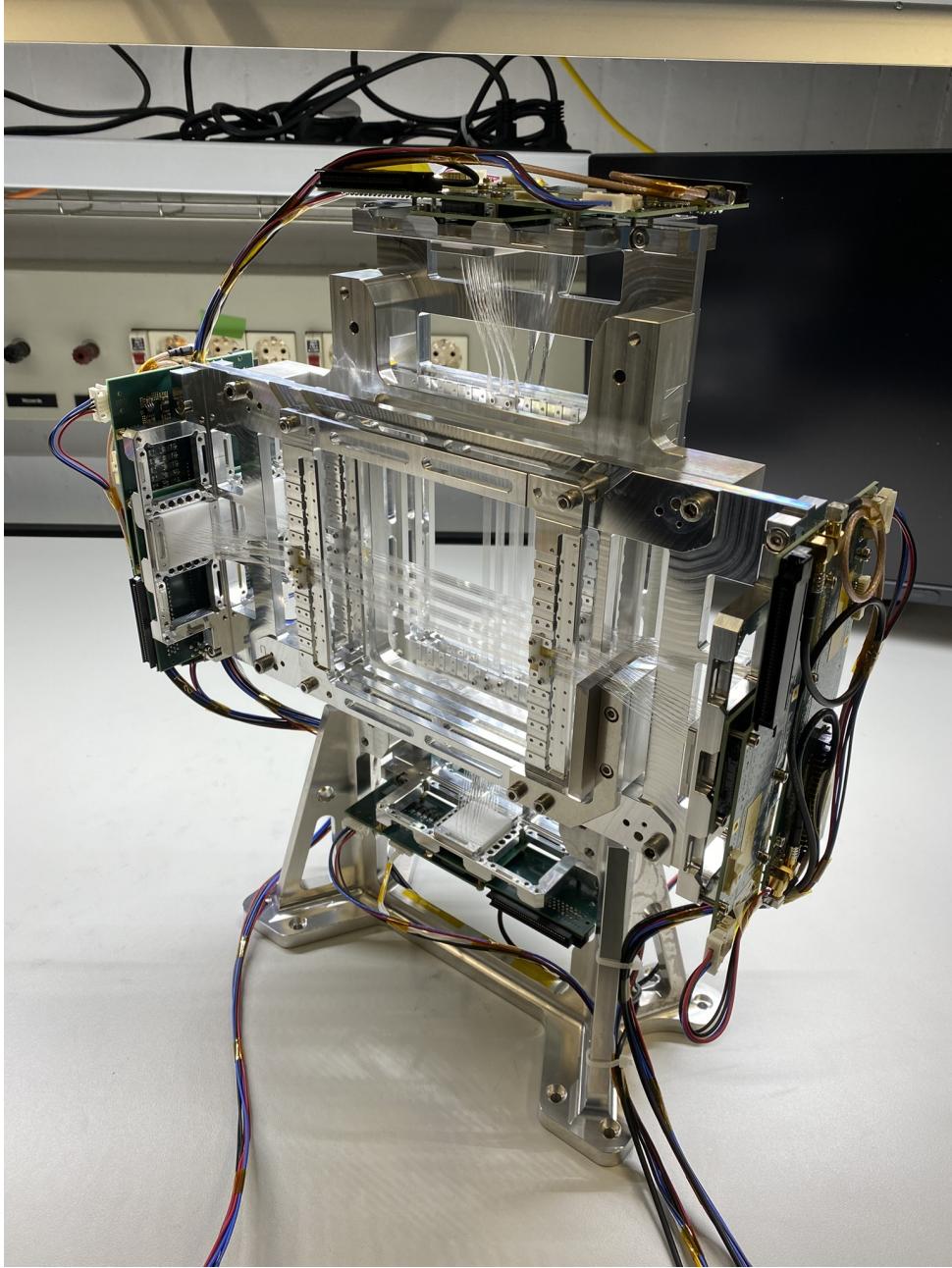


Figure 2.4.: Scintillating fiber hodoscope (SFH) with two of the four layers of scintillating fibers installed. The frontend electronics are not attached.[7]

The scintillating fiber hodoscope shown in Figure 2.4, the detector for which the FPGA driven frontend electronics are developed in this thesis, is used to measure the precise timing(300 ps[8]) of the incoming and scattered muons. Every SFH contains four layers of scintillating fibers, two in x and two in y direction. Each layer is made up of 192[8], 500  $\mu m$  thick[5] fibers, in total 768[8] fibers per SFH. When charged particles, muons in

this case, pass through a scintillating fiber they excite the scintillating material, which then emits photons. One end of every fiber are connected to a silicon photomultiplier (SiPM) which converts the photons into an electrical signal, that is then proccesed by the frontend electronics.[\[7\]](#)

# CHAPTER 3

---

## Frontend Electronic of the Scintillating Fiber Hodoscope

---

### 3.1. Field Programmable Gate Arrays (FPGA)

The frontend electronics for the scintillating fiber hodoscope (SFH) are based on Field Programmable Gate Arrays (FPGAs). Field Programmable Gate Arrays (FPGAs) are integrated circuits that can be programmed after manufacturing. FPGA are made up of a grid of programmable logic blocks (PLBs) and programmable interconnects. They posses the advantage of being flexible, reconfigurable and being able to process large amounts of data in parallel. The frontend electronics for the SFH require large amounts of data to be processed in real time, which makes FPGAs with their large data throughput and parallel processing capabilities ideal for this task.[\[6\]](#)

The FPGA used in the frontend electronics for the scintillating fiber hodoscope is part of the Xilinx Artix-7 family.[\[11\]](#)

### 3.2. Overview of the Frontend Electronics

#### 3.2.1. Proccesing of the SFH Signal

The frontend electronics of the scintillating fiber hodoscope process the signals from the scintillating fibers. They are attached on all four sides of the SFH, as shown in Figure 2.4.

Two types of readout are considered for the SFH, a mirrored readout and a non-mirrored readout. In the mirrored scenario only one end of the fibers is connected to the SiPM

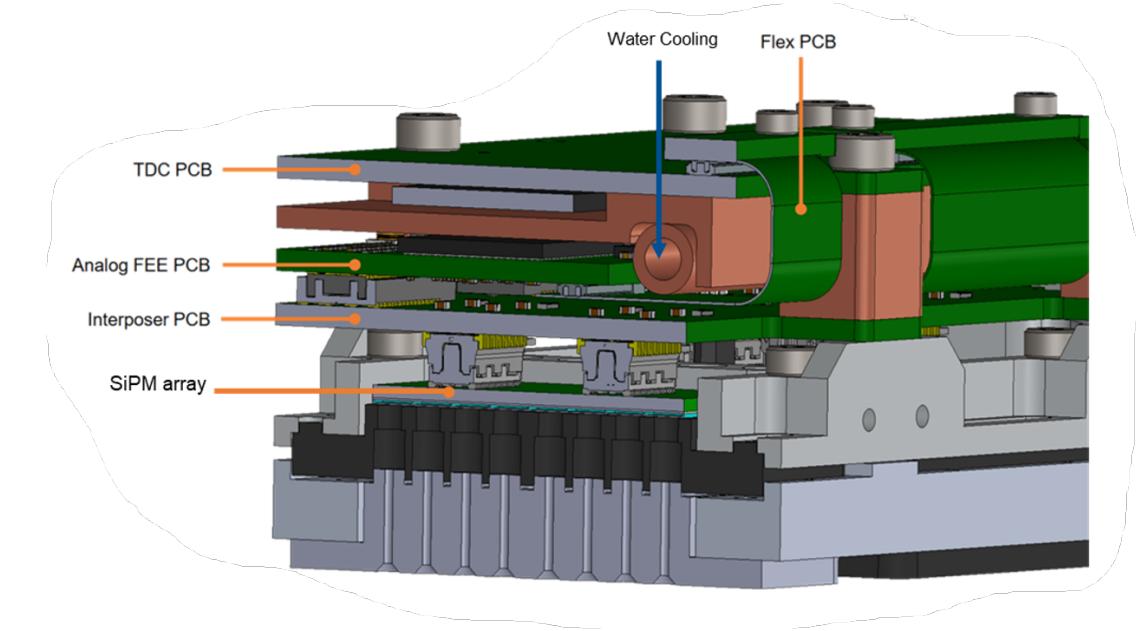


Figure 3.1.: Sideview of the frontend electronics that will be attached on the sides of the SFH, the fiber holders will be attached to the fibers.[7]

array and the other end is mirrored, while in the nom mirrored readout both ends are connected.[7]

The mirrored scenario has the advantage of only requiring 4 instead of 8 frontend electronics units, halving the amount of channels and thus the amount of data that has to be processed. Furthermore, test beam measurements show that the mirrored readout has a higher efficiency than the non-mirrored readout.[11]

Due to these advantages currently the mirrored configuration is planed for the SFH.[7]

Each SFH posses 768 fibers, for the mirrored configuration this results in 768 signals that have to be processed, spread out over 4 frontend electronics units each handling 192 signals.

The fibers are connected to the fiber holders as shown in Figure 3.1. The incoming photons are converted into electric signals by the SiPM arrays.

These are then transmitted to the heart of the frontend electronics via the interposer PCB also shown in Figure 3.1.

The heart of the frontend electronics is made up of two components, the FEE PCB and iTDTC stacked on top of each other, connected by three flex PCBs and plug connectors.

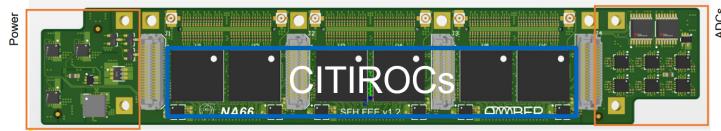


Figure 3.2.: The analog frontend electronics (FEE) PCB with the six Citiroc1A ASICs. The output of the Citiroc1A ASICs is transmitted to the iFTDC over three flex PCBs.[7]

The analog frontend electronics (FEE), illustrated in Figure 3.2, incorporates six Citiroc1A ASICs, each handling 32 channels. The SiPM signals are amplified, shaped and discriminated by the Citiroc1A ASICs and then transmitted to the iFTDC over the three flex PCBs.

The FPGA housed on the iFTDC, described in Section 3.3.3, reads these signals out and processes them as well as controlling the Citiroc1A ASICs with the signals shown in Figure 3.3.[11]

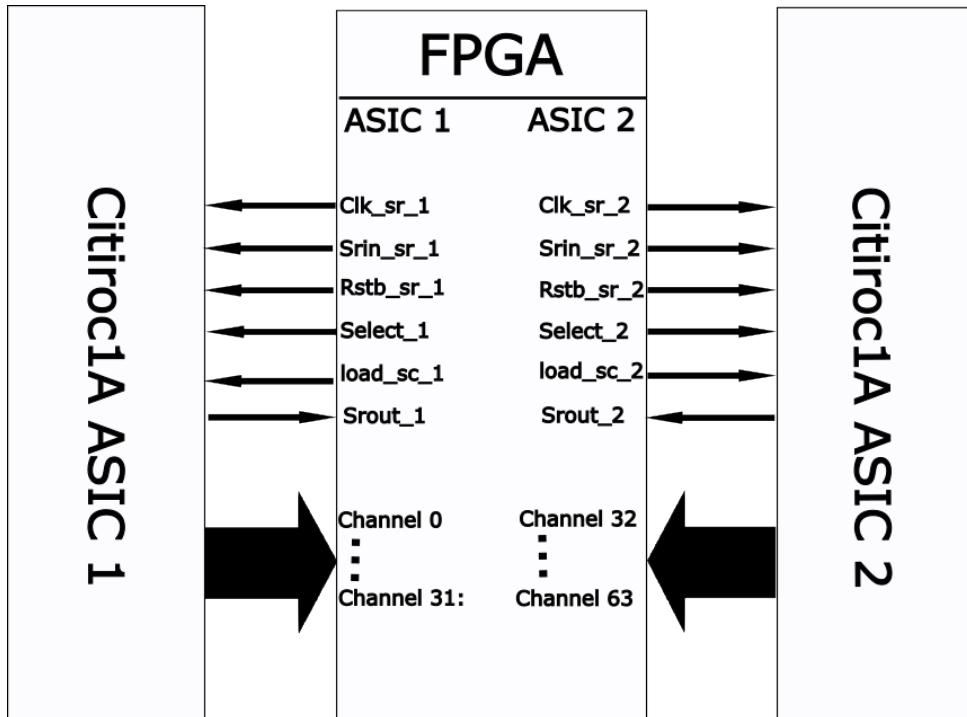


Figure 3.3.: The signals used by the FPGA to control and readout the two Citiroc1A ASICs.[14]

### 3.3. The Citiroc1A ASIC

The Citiroc1A ASIC is a Application-Specific Integrated Circuit developed by Weeroc company for the readout of SiPM detectors. It allows for the readout of 32 channels and is sensitive to  $\frac{1}{3}$  of a photoelectron.[14]

#### 3.3.1. Signal Processing of the Citiroc1A

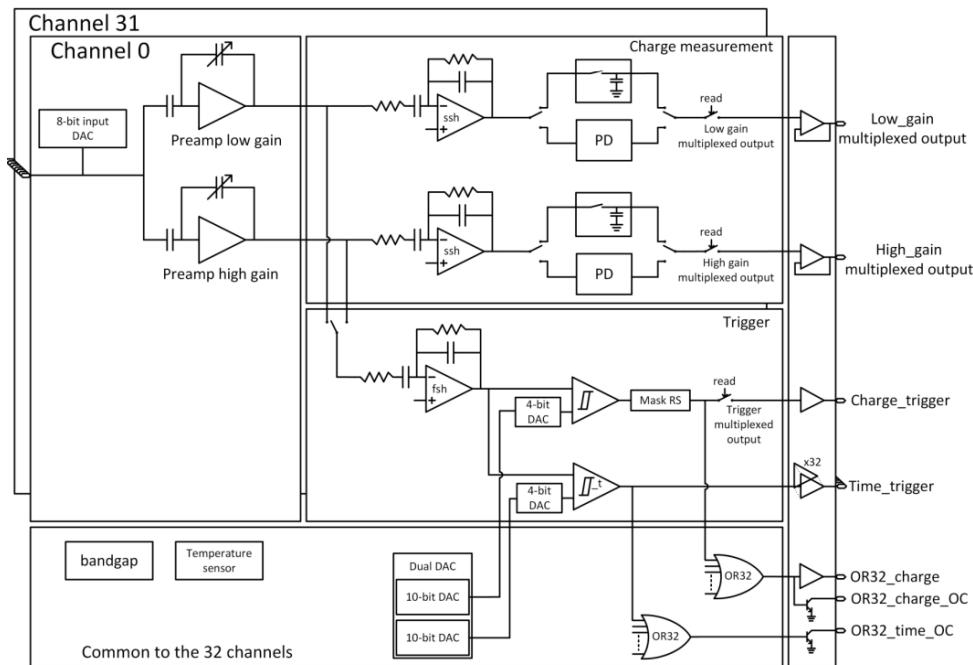


Figure 3.4.: General ASIC block scheme of the Citiroc1A.[14]

The general block scheme of the Citiroc1A is shown in Figure 3.4.

The Citiroc1A allows fine tuning of the SiPM bias voltage for each channel via the 8-bit input DAC.

The input signals are amplified with a for every channel configurable high or low gain by an integrator as depicted in Figure 3.5. The high gain is adjustable from 0 to 1575 ff in 25 ff steps, via a 6 bit dac.[14] The PRM experiment requires the maximal high gain of 62.[11]

The amplified signals are then shaped by either the slow (ssh) or fast shaper (fsh) as shown in Figure 3.4. The fast shaper is used for the PRM experiment, since it has a better time resolution, which is important for the SFH performance.[14]

Each channel of the ASIC has two discriminators, namely the charge discriminator and the time discriminator. In this thesis i discuss only the time discriminator, since it provides the time information. The time discriminator threshold is adjustable via a 10 bit dac for

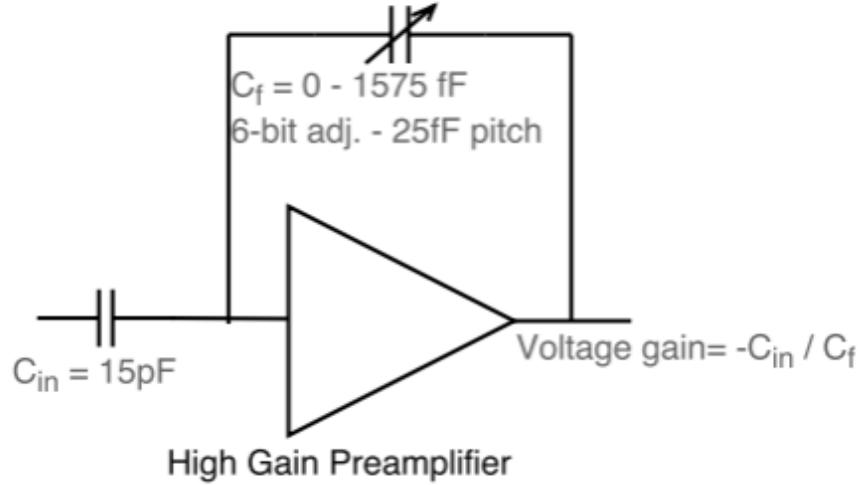


Figure 3.5.: High gain amplification of the Citiroc1A. The gain is adjustable from 0 to 1575 fF in 25 fF steps.[14]

all channels and an additional 4 bit dac for every channel individually, as shown in Figure 3.4[14].

### 3.3.2. Configuration of the Citiroc1A

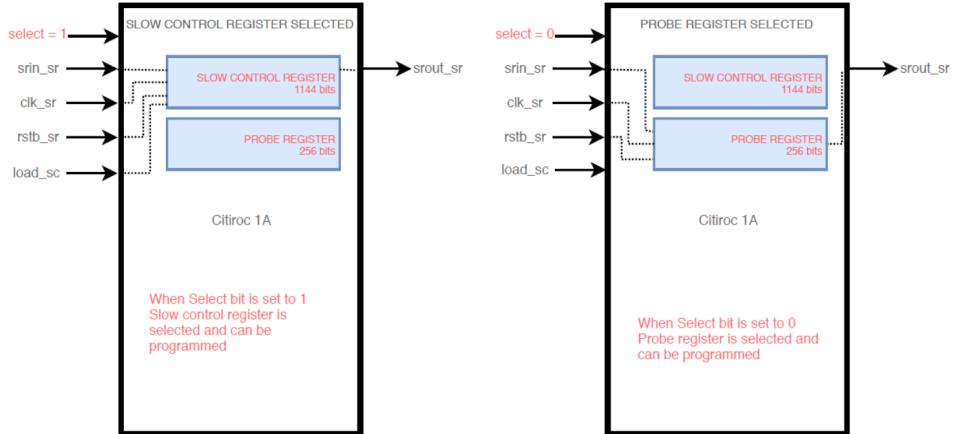


Figure 3.6.: The two configurable registers of the Citiroc1A are selected using the **Select** signal. The FPGA communicates with the Citiroc1A through the signals **Clk\_sr**, **Rstb\_sr**, **Srin\_sr**, and **Load\_sr**, while the **Srount** signal is sent back from the Citiroc1A to the FPGA for verification.[14]

The configuration of the Citiroc1A is achieved by the FPGA via the five signals shown in Figure 3.6. The **Select** signal allows the choice between configuring the slow control, for **Select = 1** or the probe register, for **Select = 0**.[14]

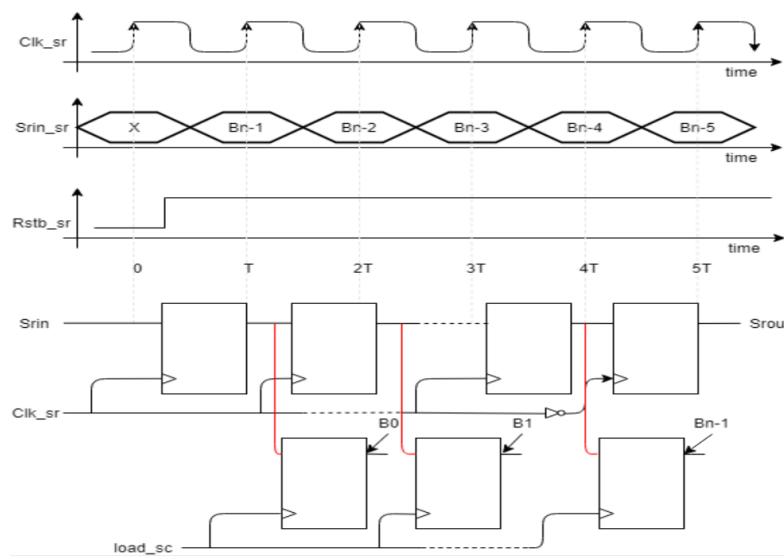


Figure 3.7.: The slow control chronogram, depicting the bitstream writing process controlled by `Clk_sr` the clock signal and `Srin_sr` the data signal. A rising edge of `Load_sr` is required, after successful verification with the `Srouut` signal to load the slow control register.[14]

### The Slow Control Register

The slow control register is used to set values for internal variables like the high gain for a channel or the time discriminator threshold. It also allows for the FPGA to turn off specific stages of the Citiroc1A, such as the slow shaper or the time discriminator. The register is 1144 bits long. A full list of all the register that can be set is shown in Table A.1 in Appendix A.

The process of writing the bitstream into the slow control register by the FPGA is illustrated in Figure 3.7.

The `Rstb_sr` signal is an asynchronous active-high reset for the serial register, applying to both the slow control and probe register.

The FPGA processes the bitstream sequentially, starting with the least significant bit (LSB). The first bit of the bitstream to enter the serial register will be the last bit of the slow control register. Each bit is sent on the `Srin_sr` signal in coordination with a rising edge of the `Clk_sr` clock signal.

The `Load_sr` signal is used to load the bitstream into the slow control register. After all bits have been sent to the Citiroc1A, a rising edge on `Load_sr` is required to load the slow control register.

The `Srouut` signal is sent back from the Citiroc1A to the FPGA for bitstream verification. Only after the FPGA has sent the full bitstream twice, does the `Srouut` signal take on the

value of the bitstream, since the **Srout** signal is shifted by the length of the bitstream.[14] One should only set the rising edge of the **Load\_sr** signal after verifying that the **Srout** signal takes on the correct values.

### The Probe Register

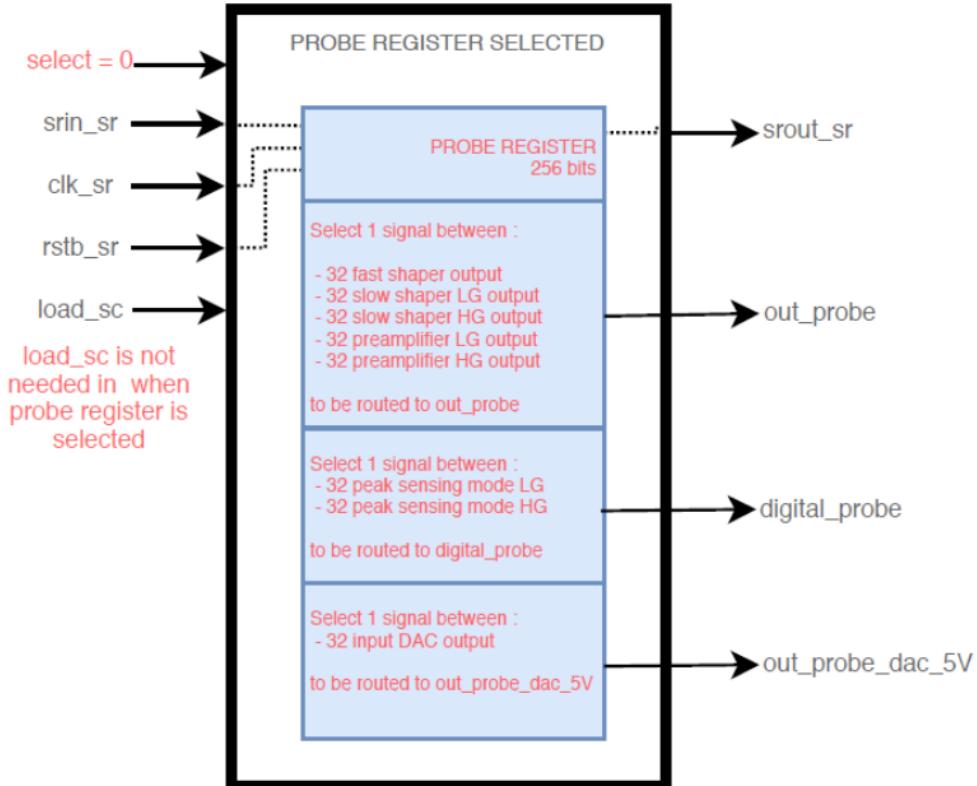


Figure 3.8.: Scheme block of internal probing system, allowing the routing of internal signals to probe pins for debugging purposes. It is configured via the probe register.[14]

The probe register is used for routing internal signals to several output pins for debugging purposes. Its functionality is illustrated in Figure 3.8. The register consists of 256 bits and is written the same way as the slow control register, with the difference that the bits are directly written into the Citiroc1A without requiring a rising edge on **Load\_sc**.[14] The full list of all the register that can be set in the probe register is shown in Table A.2 in Appendix A.

The internal signals for each channel that can be routed to the output pins are shown in Table 3.1.

<b>Signal Source</b>	<b>Description</b>	<b>Output Pin</b>
High and low gain preamplifier, slow and fast shapers	Outputs of preamplifiers and shapers	<code>out_probe</code>
<code>PeakSensing_modeb_LG</code>	Internal peak-sensing signal for low gain	<code>digital_probe</code>
<code>PeakSensing_modeb_HG</code>	Internal peak-sensing signal for high gain	-
Output of input DAC	DAC output voltage (5 V)	<code>out_probe_dac_5_V</code>

Table 3.1.: Internal signal routing to output pins for each channel.

Only one signal can be routed to each output pin at a time, without potentially causing a short circuit.[[14](#)]

### 3.3.3. The iFTDC



Figure 3.9.: The iFTDC with three Artix-7 FPGA, the three flex PCBs that connect the iFTDC with the FEE PCB and the connected power suply.

The iFTDC, depicted in Figure 3.9 is a FPGA based time-to-digital converter. It houses three Artix-7 FPGA. Each FPGA handles the readout as well as the control of two of the Citiroc1A ASICs.[\[11\]](#)

The FPGA firmware is loaded via jtag. The control and readout of the FPGAs is performed via Ethernet.



# CHAPTER 4

---

## Development of the FPGA Firmware for the SFH

---

### 4.1. Overview of the Firmware

The firmware of the different FPGAs differs only by their port assignment, but otherwise are identical.

The firmware must perform several functionalities.

The main tasks of the firmware are to configure the Citiroc1A ASICs, explained in Section 3.3.2, communication with the controlling computer via Ethernet and the IPBUS protocol and a provisional readout of the time triggerd data from the Citiroc1A ASICs.

The firmware is written in VHSIC Hardware Description Language (VHDL) and is synthesized and implemented using Xilinx Vivado. Several IP-cores, provided by Xilinx, are used to simplfy the development of the firmware for several tasks like the implementation of the configuration memory.

#### 4.1.1. The IPBUS Protocol

The IPBUS protocol used for the communication between the FPGA and the controlling computer is a simple protocol for controlling IP-aware hardware devices with a 32 bit read and write bus using UDP as the transport protocol.[\[12\]](#)

The IPBUS protocol defines a read and write command enabling successful write and read operations of a 32 bit register, with a 32 bit address in the FPGA.

The commands can be issued on the controlling computer with the  $\mu$ HAL library, which allows the user to issue read and write commands using a python script and an XML file

defining the addresses of the registers.[12]

The address space inside the FPGA is defined in the firmware.

#### 4.1.2. The Firmware Structure

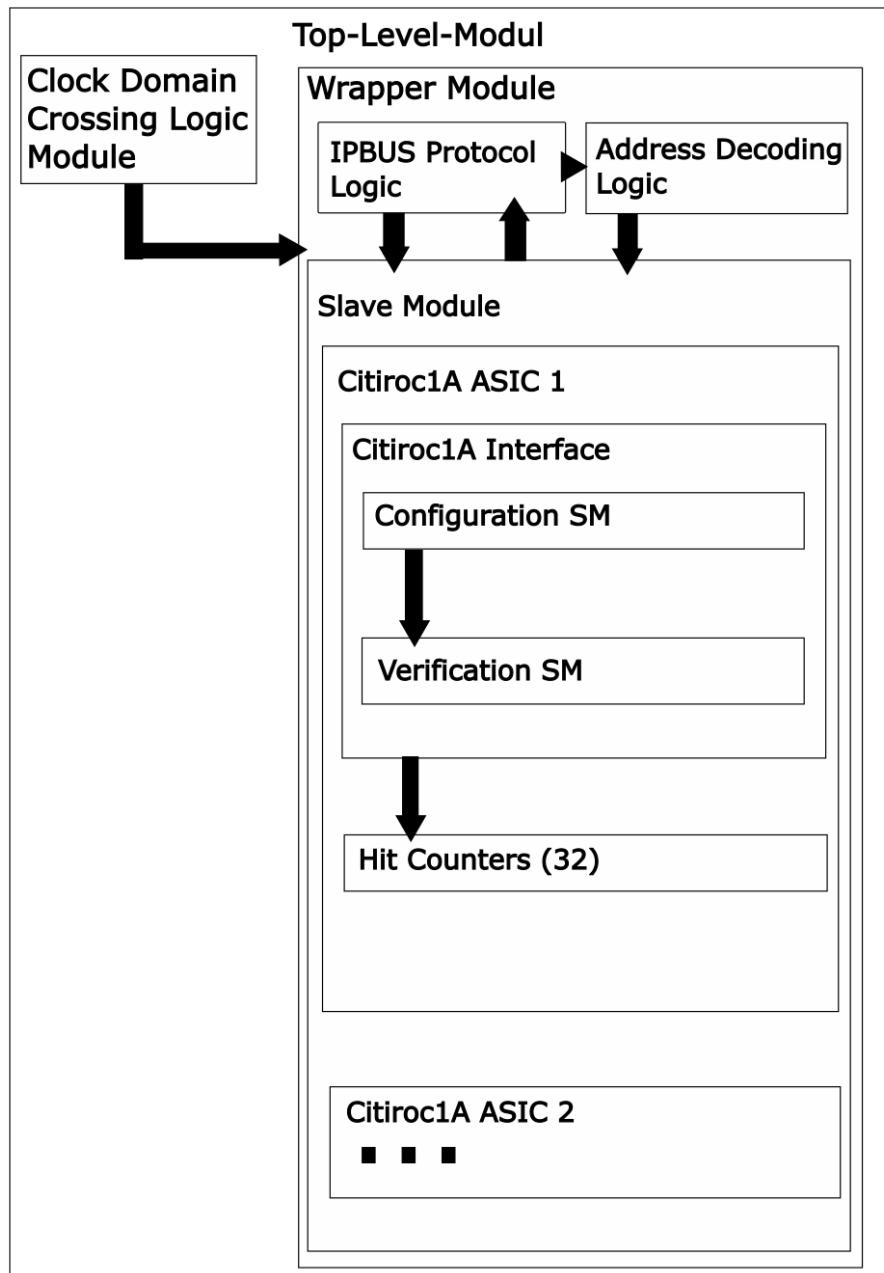


Figure 4.1.: Hierarchical structure of the firmware, where each module contains its sub-modules. The black arrows indicate connections that are semantically correct but are routed through higher-level modules.

The firmware has a hierarchical structure illustrated in Figure 4.1.

The top level modul of the firmware defines the interconnections between internal FPGA

logic and external components such as the Ethernet transceiver, clock oscillators and Citiroc1A ASICs.

The FPGA logic includes an IPBUS IP core and the slave entities in addition to a Clock Crossing Domain logic modul which synchronizes the time triggered Citiroc1A signals.

The slave modul contains two identical Citiroc1A modules one for each ASIC. The Citiroc1A interface is instantiated for the configuration and verification of the ASIC. The hitcounters allow to perform hit rate measurements for the 32 channels of the Citiroc1A ASIC.

## 4.2. The Address Space

The 32 bit address is divided into sub addresses by the address decoding logic of the firmware.

The address space is first divided between the two Citiroc1A ASICs:

<b>Citiroc1A Interface ASIC 1:</b>	0000_0000_0000_0000_0 <b>001</b> _0000_0000
<b>Citiroc1A Interface ASIC 2:</b>	0000_0000_0000_0000_0 <b>010</b> _0000_0000
<b>Hitcounters of ASIC 1:</b>	0000_0000_0000_0000_0 <b>011</b> _0000_0000
<b>Hitcounters of ASIC 2:</b>	0000_0000_0000_0000_0 <b>100</b> _0000_0000

The address space of the Citiroc1A Interface is further divided into the following sub addresses:

<b>Status and Control Register:</b>	0000_0000_0000_0000_0 <b>000</b> _0000
<b>Configuration RAM:</b>	0000_0000_0000_0000_0 <b>0100</b> _0000
<b>Verification RAM:</b>	0000_0000_0000_0000_0 <b>1000</b> _0000

In order to read the second address of the configuration RAM of the first Citiroc1A ASIC, the address would be:

0000\_0000\_0000\_0000\_0001\_0100\_0001

The address space of the hitcounters is also further divided into sub addresses:

<b>Hitcounter 0:</b>	0000_0000_0000_0000_0000_0 <b>0000</b> _0001
<b>Timecounter 0:</b>	0000_0000_0000_0000_0000_0 <b>0100</b> _0001
<b>Hitcounter 1:</b>	0000_0000_0000_0000_0000_0 <b>0000</b> _0010
<b>Timecounter 1:</b>	0000_0000_0000_0000_0000_0 <b>0100</b> _0010
...	...
<b>Hitcounter 31:</b>	0000_0000_0000_0000_0000_0 <b>0001</b> _1111
<b>Timecounter 31:</b>	0000_0000_0000_0000_0000_0 <b>0101</b> _1111

The address to read the timecounter of the fifth channel of the second Citroc1A ASIC would be:

0000\_0000\_0000\_0000\_0100\_0100\_0101

### 4.3. Configuration of the Citroc1A ASIC

The configuration of the slow control and probe registers of the Citroc1A ASIC, along with the verification of this configuration, is handled by two finite state machines.

Each state machine controls a random access memory (RAM) with a depth of 64 addresses, with each address storing 32 bits of data.

The state machiens in turn are controlled by the status and control register of the corresponding Citroc1A Interface, which can be written by the controlling computer via the IPBUS protocol.

#### 4.3.1. Status and Control Register

Each Citroc1A interface has a 32 bit status and control register, but only the first 7 bits are used. It is responsible for the control of the configuration and verification state machines as well as the hitcounters. The bits of the status and control register, with the exception of bit 1 do not reset themselves after being set to 1 and have to be deaserted by the controlling computer.

Bit 1 is set to 0 by the FPGA after the configuration is loaded into the serial register. Its structure is shown in Table 4.2.

Bit	Description
bit 0	Selects between slow control (1) and probe register (0)
bit 1	Loads the the configuration into the serial register
bit 2	Resets the Configuration and Verification SM and the Configuration RAM
bit 3	Resets the Citroc1A serial register
bit 4	Loads the configuration into the slow control register
bit 5	Resets the hitcounters associated with the Citroc1A
bit 6	Enables the hitcounters associated with the Citroc1A

Figure 4.2.: Structure of the status and control register of the Citroc1A interface. The specified commands are executed when the bits are set to 1.

### 4.3.2. Configuration State Machine

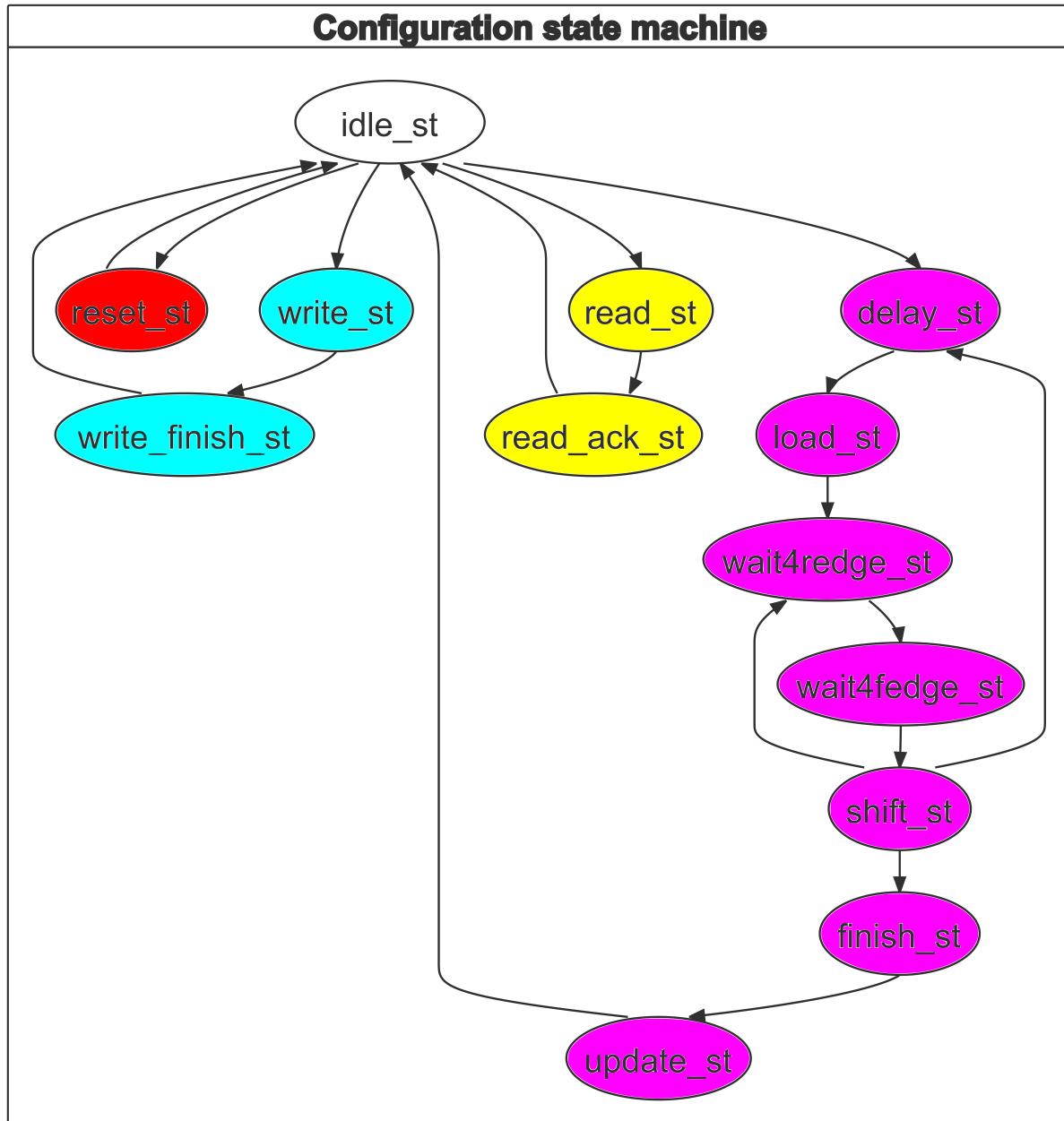


Figure 4.3.: Finite state machine for configuring the Citiroc1A ASIC. States associated with the same processes are highlighted using the same color.

The Configuration state machine, embedded in the firmware as shown in Figure 4.1, is responsible for configuring the slow control and probe registers of the Citiroc1A ASIC. The state machine is controlled by the status and control register, which is written by the controlling computer via the IPBUS protocol. It has four processes, whose states are shown in different colors in Figure 4.3.

The **write** process is responsible for writing the configuration data to the configuration

RAM. 32 bits can be written to a specified address in the configuration ram at a time by the controlling computer.

The **read** process allows the controlling computer to read the content of the configuration RAM at the specified address.

The **reset** process resets the configuration RAM and the Configuration state machine and is initiated by setting bit 2 of the status and control register to 1.

The **load** process loads the configuration data from the configuration RAM into the serial register of the Citiroc1A ASIC. The first 1144/256 bits of the configuration RAM are shifted into the serial register of the Citiroc1A ASIC, depending on the selected register. Each bit is shifted in by setting the data line **Srin\_sr** and clock **clk** as described in Section 3.3.2. Since as described in Section 3.3.2 the first bit written to the serial register is the last bit of the configuration, the controlling computer has to write the bitstream in reverse order to the configuration RAM.

The **clk** clock signal for the ASIC's serial register is generated by the FPGA with a frequency of 1 MHz. The **load** process modulates this clock signal to control the operation of the serial register.

The process is repeated until all the data is shifted in and the configuration is loaded. It can be initiated by the controlling computer by setting bit 1 of the status and control register to 1 and is only interruptible by the **reset** process.

In order to load the configuration into the slow control register from the serial register, bit 4 of the status and control register must be set to 1, this is not necessary for the probe register.

### 4.3.3. Verification State Machine

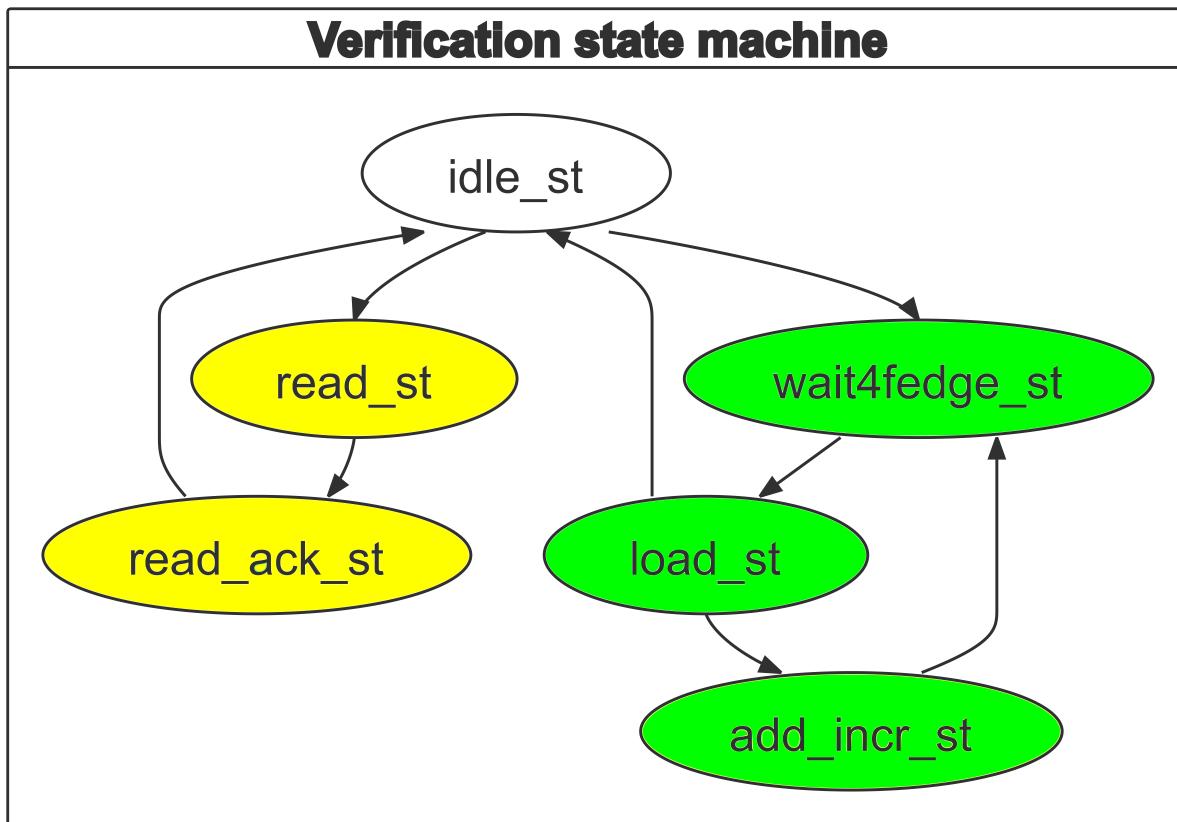


Figure 4.4.: Finite state machine for verifying the configuration of the Citiroc1A ASIC.  
States belonging to the same processes are represented with the same color.

The Verification state machine, illustrated in Figure 4.4 is responsible for verifying the configuration of the Citiroc1A ASIC. It is incorporated into the firmware, as shown in Figure 4.1.

It can be reset by the controlling computer by setting bit 2 of the status and control register to 1.

If bits are shifted into the serial register of the Citiroc1A ASIC, the **verification** process is automatically started. The **srout** signal coming back from the ASIC, explained in Section 3.3.2, is used to read back the shifted bits from the serial register of the Citiroc1A ASIC. The bits arrive at the FPGA in the same order as they were loaded in, but shifted by the length of the bitstream(1144/256). To compare the read back bits with the the loaded bits the bitstream has to be loaded twice.

The readback bits are stored in the verification RAM by the **verification** process and can be read by the controlling computer via the **read** process. This allows the controlling computer to compare the loaded bits with the read back bits and ensure that the configuration was loaded correctly into the serial register of the Citiroc1A ASIC.

## 4.4. The Provisional Readout

In addition to the configuration and verification of the configuration of Citroc1A ASIC, the firmware also provides a provisional readout in form of hitcounters and timecounters. The hitcounters are used to count the number of hits on each channel of the Citroc1A ASIC.

They are limited to a minimal time between hits of two periods of the synchronization clock. In this case a 325 MHz clock is used, which corresponds to a minimal time between hits of 6.16 ns.

Each Citroc1A has 32 hitcounters, one for each channel. The 32 bit hitcounters are enabled by setting bit 6 of the status and control register of the corresponding Citroc1A Interface to 1. This also enables the timecounter, which counts the number of clock cycles since it was last reset.

The hitcounters and timecounters are reset by setting bit 5 of the status and control register of the corresponding Citroc1A Interface to 1.

To read the hit and time counters, the controlling computer first has to disable the hitcounters by setting bit 6 of the status and control register to 0, in order to avoid metastability issues arising from the clock domain crossing. The hitcounters can then be read by the controlling computer via the IPBUS protocol.

The combination of the hitcounters and the timecounter allows the controlling computer to determine the number of hits per second on each channel of the Citroc1A ASIC.

## 4.5. Testing the Firmware

### 4.5.1. Setup for the Noise Measurement

The firmware will be evaluated through a noise measurement of the frontend electronics of the scintillating fiber hodoscope. The measurement will be performed with the SiPMs disconnected from the frontend electronics. The experimental setup is illustrated in Figure 4.5.

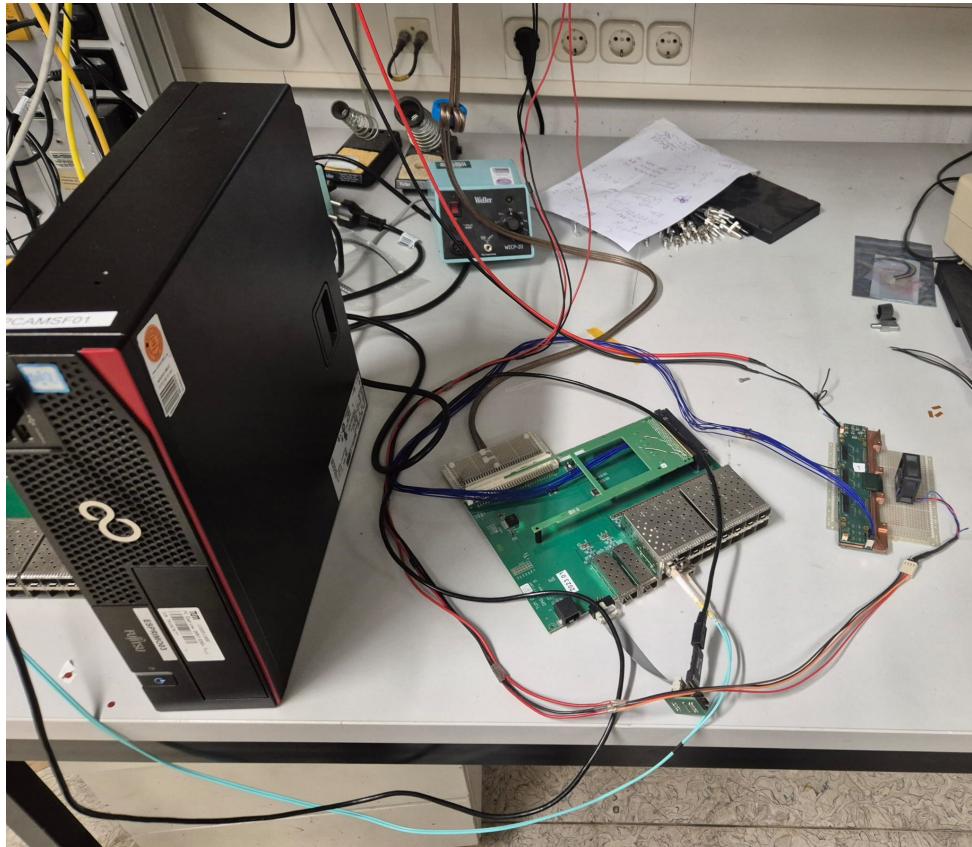


Figure 4.5.: Experimental setup for the noise measurement of the frontend electronics of the scintillating fiber hodoscope.

The measurement will be performed by the controlling computer, which will scan through a range of thresholds of the time discriminator of the Citiroc1A ASIC and record the number of hits per second on each channel with a measurement time of 400 ms. This scan will be performed for several different high gains near the maximum value of 62.

For comparison, the threshold scan will also be performed with the same configuration of the Citiroc1A ASICs on an Evaluation Board provided by Weeroc company, with the setup shown in Figure 4.6.

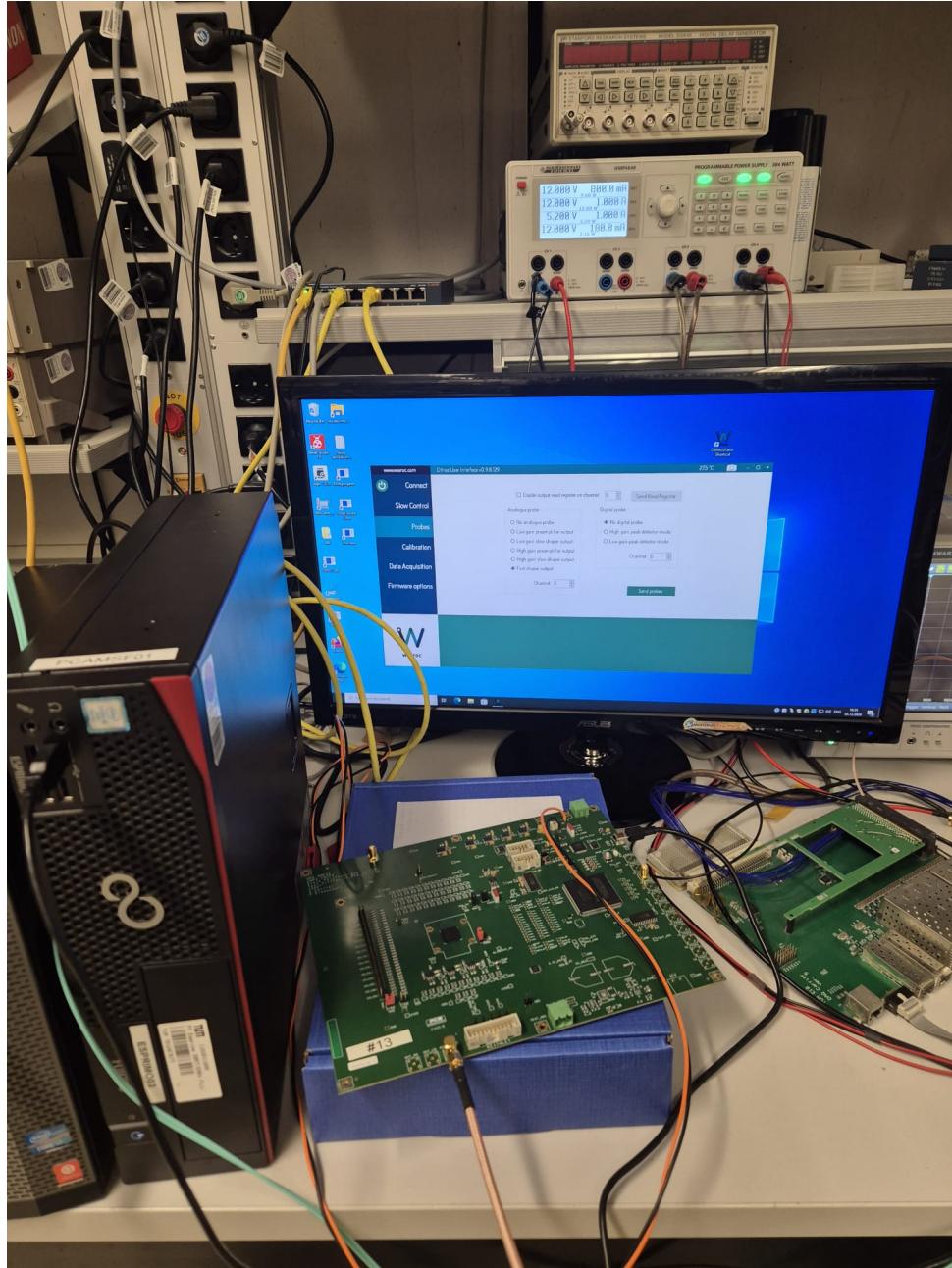


Figure 4.6.: Experimental setup for the noise measurement with an Evaluation Board provided by Weeroc company.

#### 4.5.2. Theoretical Background for the Noise Measurement

The noise on the unconnected frontend electronics is assumed to be of gaussian distribution,

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4.1)$$

where  $\mu$  is the mean and  $\sigma$  the standard deviation of the noise.[10]

The number of hits per second on each channel of the ASIC is proportional to the number of noise events above the threshold of the time discriminator, but limited by the bandwidth of the fast shaper.[10]

The number of hits per second vs the threshold of the time discriminator is expected to follow an S-curve, which can be described by a modified error function,

$$N = \frac{A}{2} \left( 1 - \operatorname{erf} \left( \frac{x - \mu}{\sqrt{2}\sigma} \right) \right) \quad (4.2)$$

where  $N$  is the number of hits per second,  $x$  the threshold of the time discriminator and  $A$  an renormalization factor.[10]



# CHAPTER 5

---

## Measurment Results

---

### 5.1. Noise Measurements

For the noise measurement, the threshold was scanned in steps of 1 for a range of high gains, starting with gain 58 and ending with the maximial gain of 62.

The number of hits was measured for each of the 32 channels of the ASICs within a time interval of 400 ms. The threshold scan was performed for both the ASICs of FPGA 1 and 2.

The results of the threshold scan are shown for both ASICs of FPGA 1 and 2 for gains 58 to 62 in Figures 5.1, 5.2, 5.3, 5.4 and 5.5 respectively.

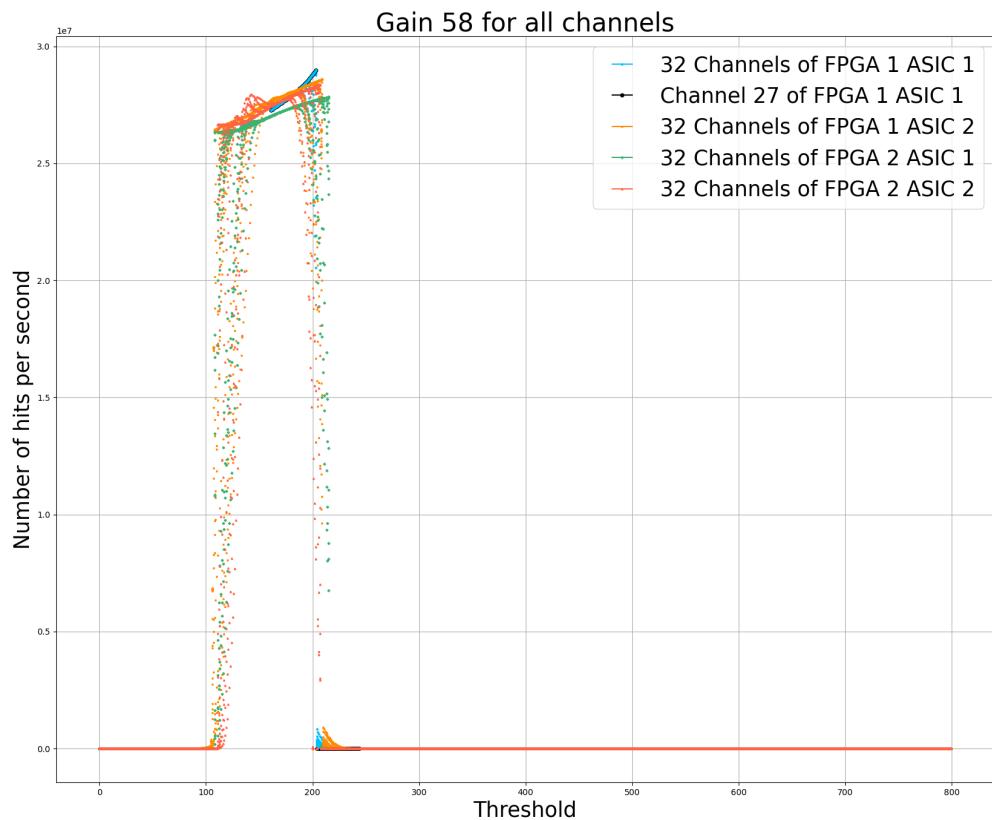


Figure 5.1.: Threshold scan of the Citiroc1A ASICs of FPGA 1 and 2 for gain 58 from threshold 0 to 800.

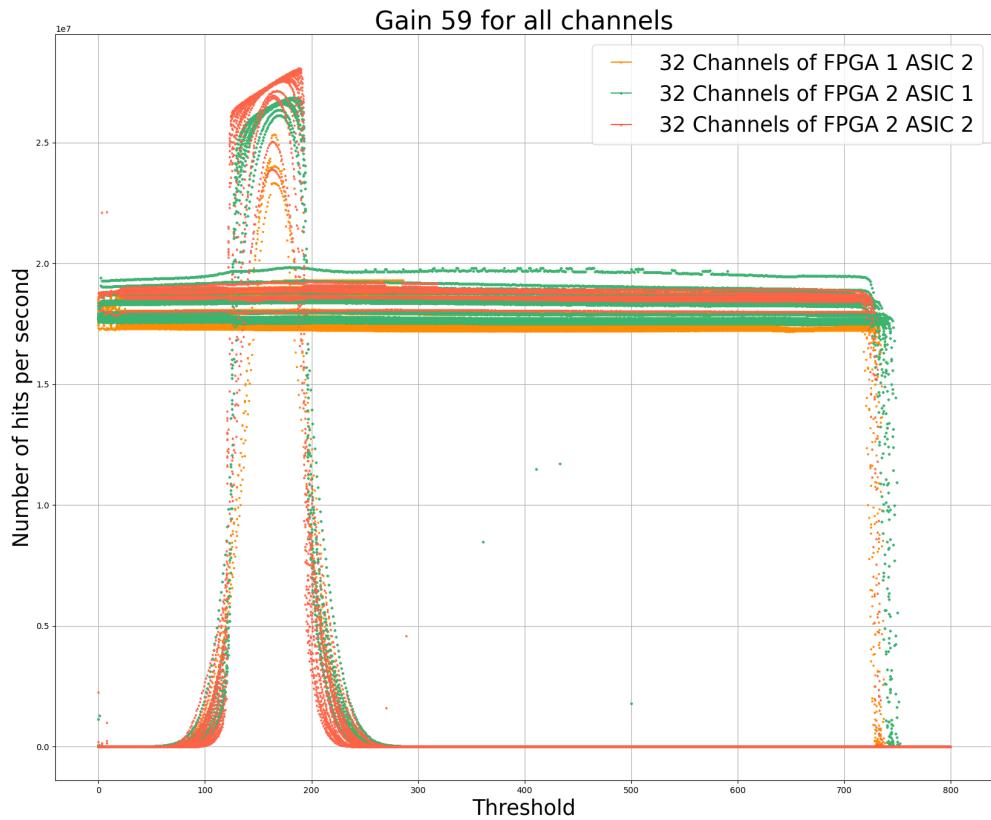


Figure 5.2.: Threshold scan of the Citiroc1A ASICs of FPGA 1 and 2 for gain 59 from threshold 0 to 800.

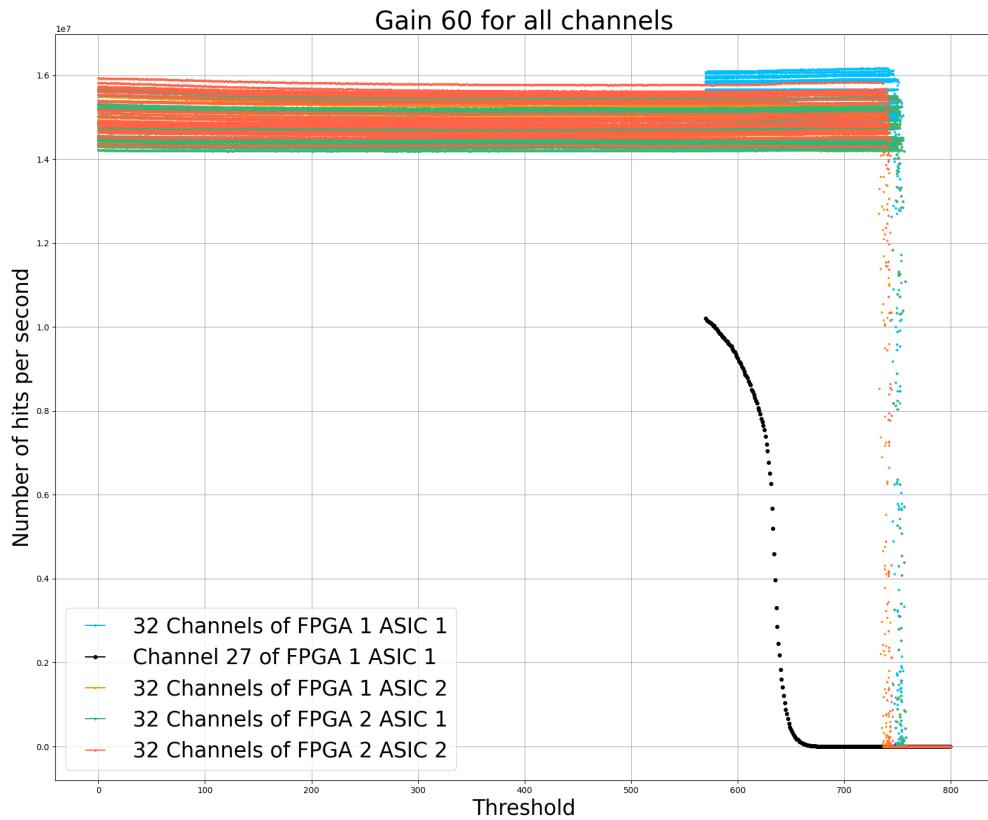


Figure 5.3.: Threshold scan of the Citiroc1A ASICs of FPGA 1 and 2 for gain 60 from threshold 0 to 800.

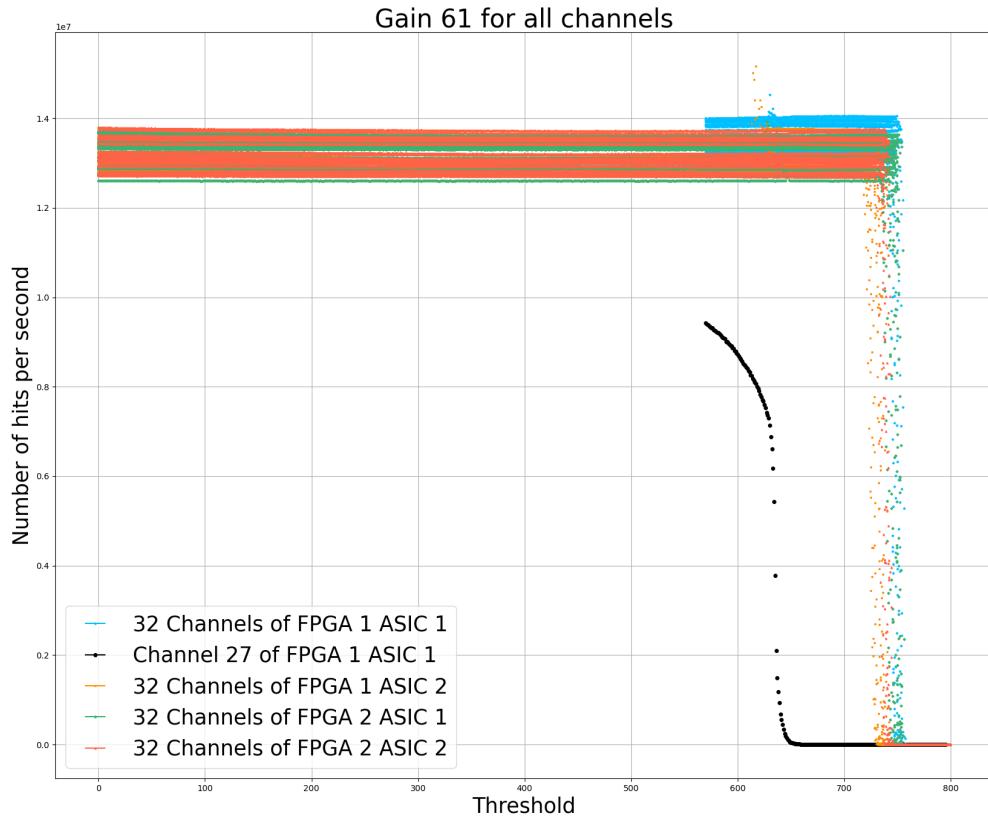


Figure 5.4.: Threshold scan of the Citiroc1A ASICs of FPGA 1 and 2 for gain 61 from threshold 0 to 800.

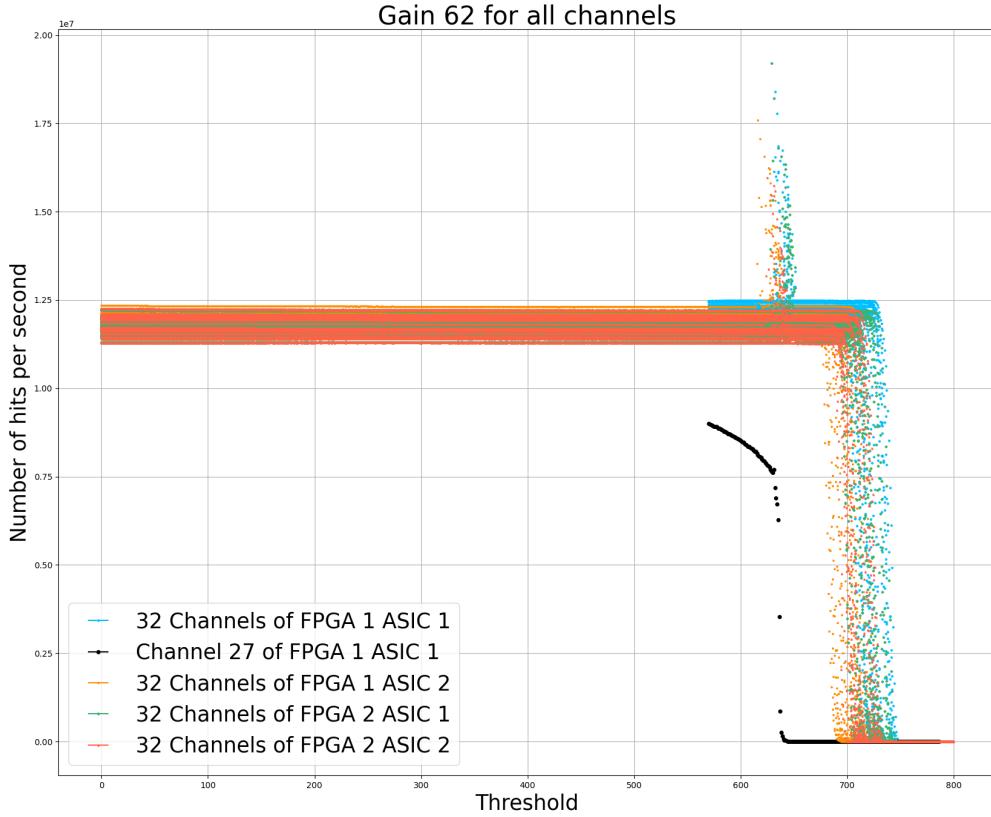


Figure 5.5.: Threshold scan of the Citiroc1A ASICs of FPGA 1 and 2 for gain 62 from threshold 0 to 800.

## 5.2. Frontend Electronics Characterization

### 5.2.1. Firmware Evaluation

Before performing the final noise measurments I thoroughly verified the firmware functionality by performing the following tests:

- Using the internal probing capabilities of the Citiroc1A to verify the configuration of the ASICs, described in Section 3.3.2.
- Turning of individual channels of the ASICs by turning of the amplification of the channels or setting it to a low gain.
- Disabling spesific stages of the ASIC, such as the fast shaper or the time discriminator.
- Using the output configuration of the Citiroc1A to turn of all triggers of the ASICs.

### 5.2.2. S-curve Analysis

In order to determine the optimal threshold and characterize the noise of the ASICs of FPGA 1 and 2, an S-curve analysis was performed.

The falling edge of the pedestal is fitted with the S-curve described in Section 4.5.2

The results of the S-curve analysis are shown exemplary for channel 0 of Citiroc1A 1 of FPGA 1 with gain 60,61 and 62 in Figures 5.6, 5.7 and 5.8.

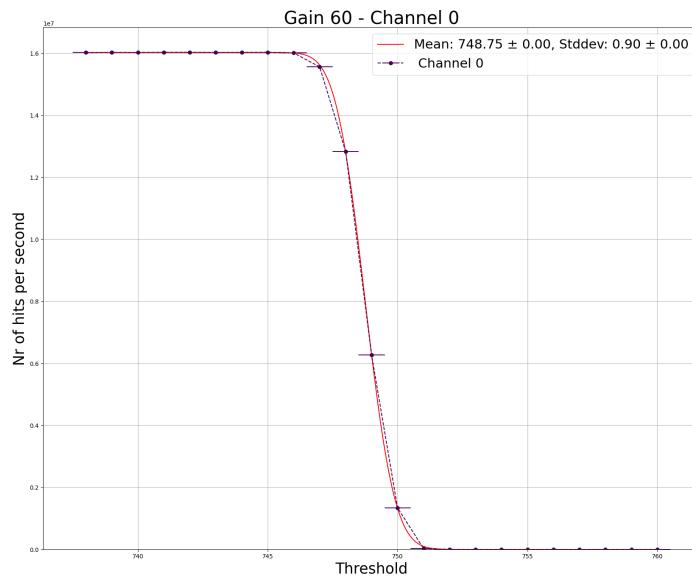


Figure 5.6.: Results of the S-curve analysis for channel 0 of Citiroc1A 1 of FPGA 1 with gain 60.

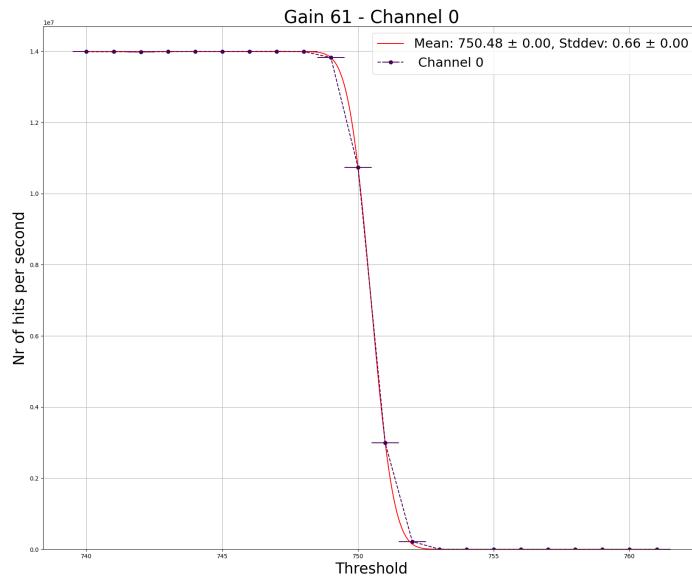


Figure 5.7.: Results of the S-curve analysis for channel 0 of Citiroc1A 1 of FPGA 1 with gain 61.

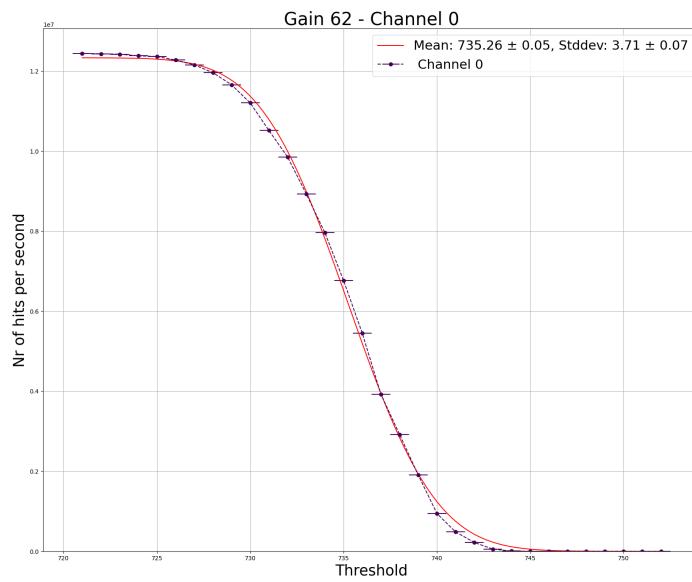


Figure 5.8.: Results of the S-curve analysis for channel 0 of Citiroc1A 1 of FPGA 1 with gain 62.

The S-curve fit, for all channels, for the high gain of 62 needed for the PRM experiment is shown in Figure 5.9.

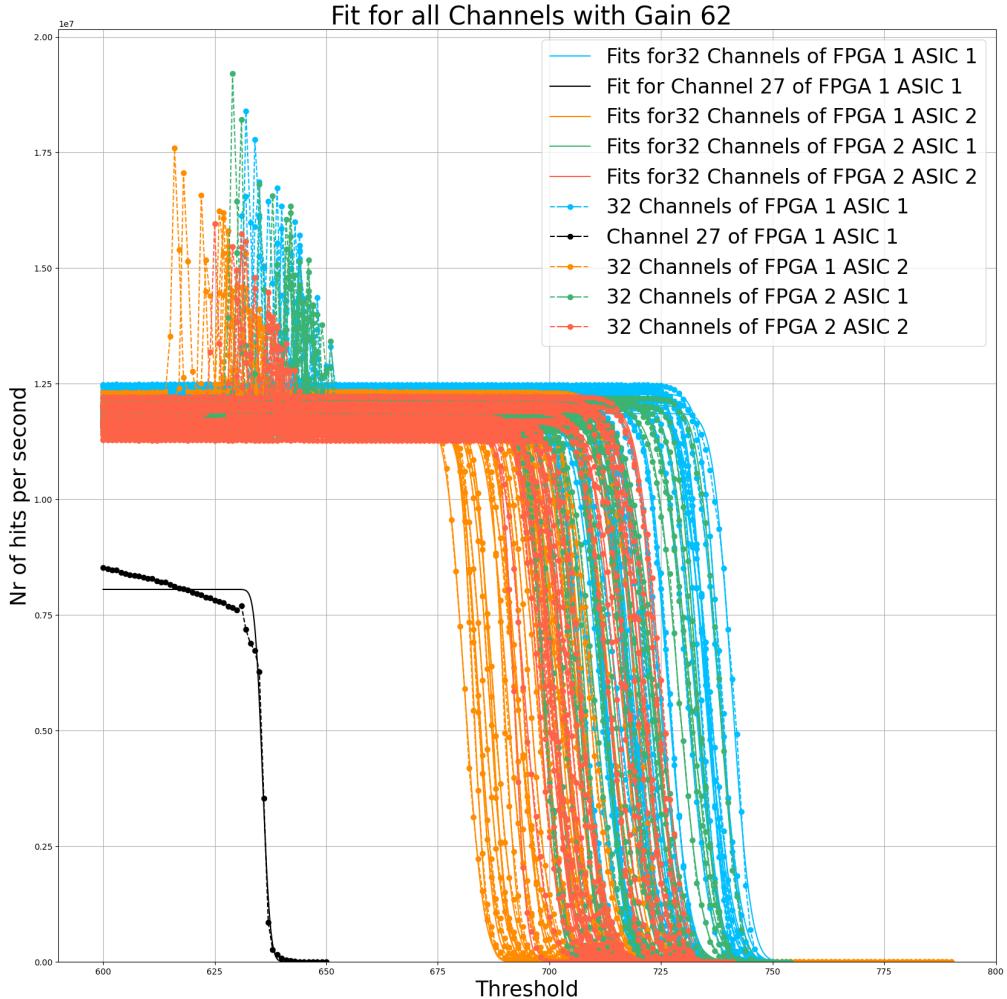


Figure 5.9.: Results of the S-curve analysis for all channels of the Citiroc1A ASICs of FPGA 1 and 2 with gain 62.

The mean  $\mu$  and the standard deviation  $\sigma$  of the noise distribution that were determined from the S-curve fits and are shown in Appendix B. They are plotted against the gain for all channels in Figure 5.10.

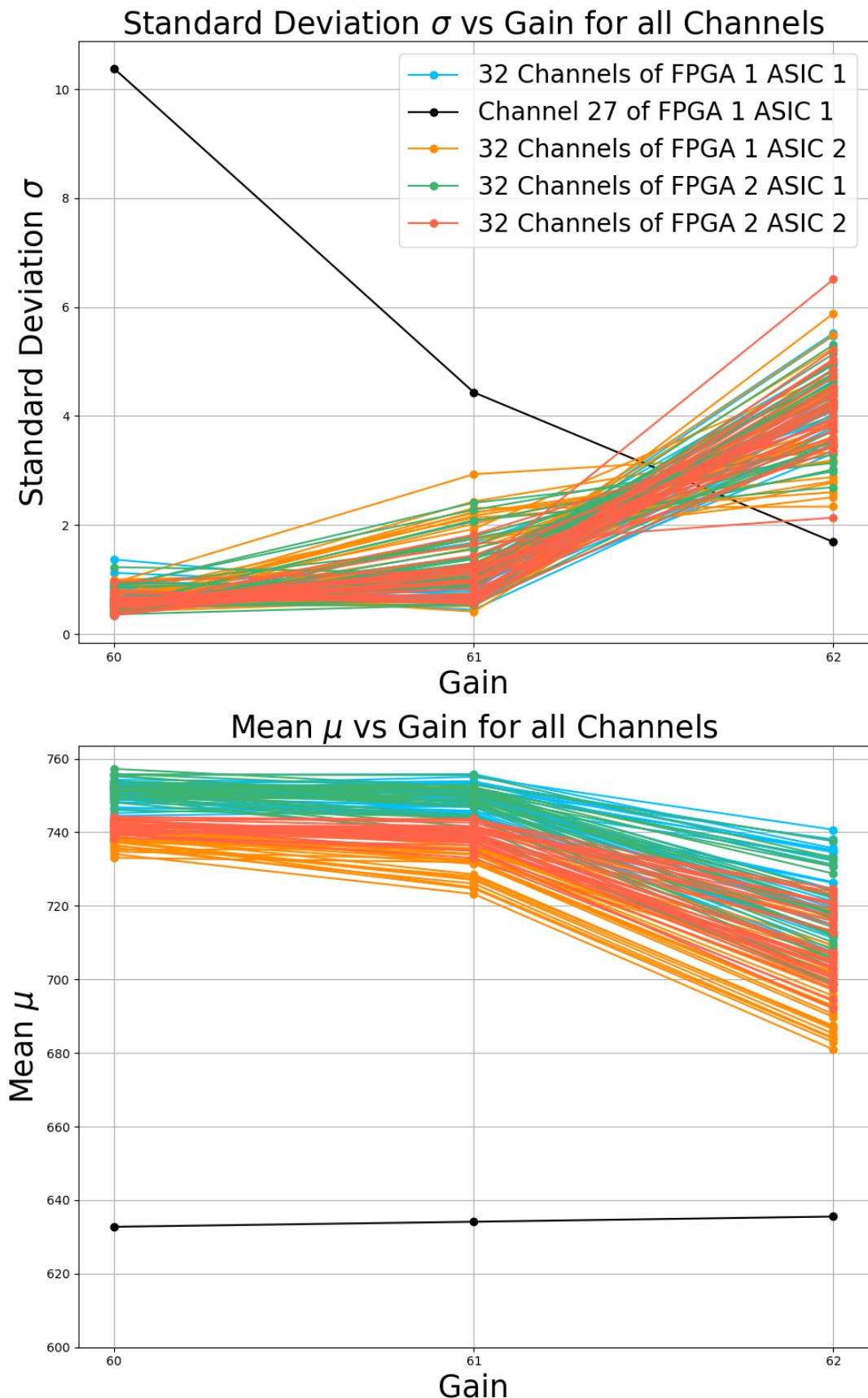


Figure 5.10.: Mean  $\mu$  and standard deviation  $\sigma$  determined from the S-curve fits for all channels of the Citiroc1A ASICs of FPGA 1 and 2 plotted vs the gain.

### 5.2.3. Comparison with Evaluation Board

A comparison of the threshold scan to one performed with the same configuration of the Citiroc1A ASIC on a by Weeroc company provided Evaluation Board with gain 58 is shown in Figure 5.11.

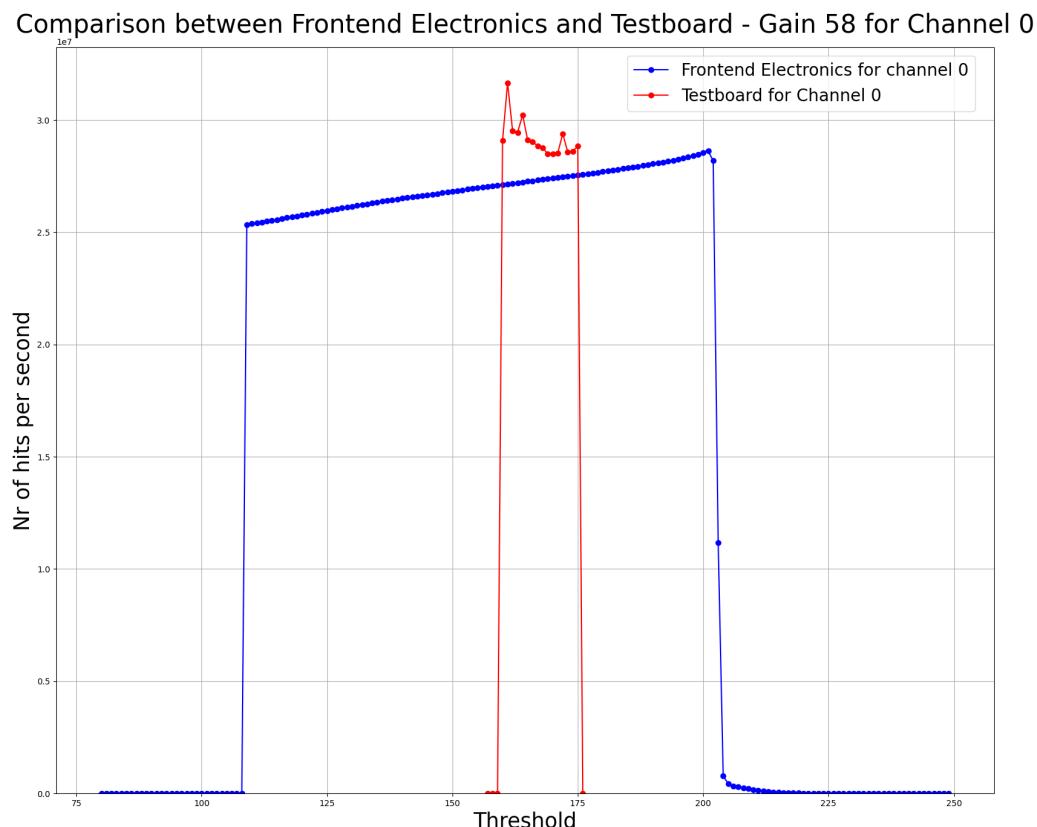


Figure 5.11.: Comparison of the threshold scan of the frontend electronics to the threshold scan performed on a by Weeroc company provided Evaluation Board with a gain of 58.

The pedestal of the threshold scan performed with the frontend electronics is far larger than the pedestal of the threshold scan performed with the Evaluation Board. This difference in the observed behavior of the frontend electronics can currently not be explained. The possibility of the Citiroc1A being wrongly configured was investigated and ruled out by testing for specific behaviors.

### 5.2.4. Behavior at different Gain Setting

The threshold scan of the Citroc1A ASICs in the frontend electronics shows different behavior for gain ranges of 0 to 58 (Figure 5.1) and 60 to 62 (Figures 5.3, 5.4 and 5.5), with the threshold scan at gain 59 showing a transition between the two behaviors depicted in Figure 5.2. This difference is also currently not understood and is suspected to have the same cause as the difference in the threshold scan compared to the Evaluation Board measurement.

A general trend of the standard deviation  $\sigma$  of the S-curve increasing and the mean  $\mu$  decreasing with higher gain is observed in Figure 5.10.

### 5.2.5. Channel Behavior Analysis

From the S-curve analysis it can be observed that the channels of the Citroc1A ASICs show different behavior in the threshold scan.

Especially channel 6 of Citroc1A 1 of FPGA 1 shows a significantly different behavior compared to the other channels, as can be seen in Figure 5.10 and in the results of the threshold scan shown in the Figures 5.3, 5.4 and 5.5.

Furthermore, for the threshold scan with gain 59 depicted in Figure 5.2 , all the channels show diverging behavior, this is also suspected to be because of the transition between the two gain regions.

The behavior of channel 6 of Citroc1A 1 of FPGA 1 seems to have a different cause than the gain transition and needs further investigation. This could be investigated by performing a threshold scan with another FEE PCB and comparing the results.

### 5.2.6. Summary of the Characterisation

The threshold scan and the S-curve analysis show that the frontend electronics of the SFH do not perform as expected. The noise behavior at different gains is not consistent with the change of the gain value.

Due to this, a calibration of the individual channels of the frontend electronics is not yet possible. The cause of the observed behavior is currently under investigation. The following sources are being considered:

- Ripple noise on the power supplies of the frontend electronics.
- Software error, that causes the ASIC to be configured with a wrong configuration.
- Mistake in the circuit diagram or the layout of the PCBs.

## 5.2. Frontend Electronics Characterization

---

Furthermore, the behavior of channel 6 of Citiroc1A 1 of FPGA 1 seems to have a different cause than the gain transition and also needs further investigation.



# CHAPTER 6

---

## Conclusion and Outlook

---

### 6.1. Personal Contribution

In this thesis, I developed the firmware and software, for the three FPGAs of the front-end electronics of the scintillating fiber hodoscope (SFH) responsible for the control and provisional readout of the Citiroc1A ASICs.

For this, I structured the firmware hierarchically into different modules, responsible for the configuration of the Citiroc1A ASICs, communication with the controlling computer and the readout of the ASICs and developed it in VHDL using Xilinx Vivado.

I tested the firmware and characterized the frontend electronics with a threshold scan and an S-curve analysis.

### 6.2. Conclusion

The threshold scan shows that the Citiroc1A ASICs can be successfully configured, but that the front-end electronics show abnormal behaviour. I conclude that the developed firmware allows for the configuration and provisional readout of the Citiroc1A ASICs, but that the frontend electronics of the SFH are not performing as desired.

### 6.3. Outlook

The next steps in the development of the frontend electronics for the SFH are, the integration of this part of the firmware with the rest of the frontend electronics and the final

## Chapter 6. Conclusion and Outlook

---

readout of the SFH. Moreover, additional investigation into the abnormal behaviour of the frontend electronics observed during the noise measurement is required.

This could be accomplished using the internal probing capabilities of the Citroc1A ASICs. Furthermore, tests with different FEE PCBs could be performed to investigate the origin of the observed behaviour.

By conducting these proposed steps and completing the development of the frontend electronics, the scintillating fiber hodoscope will be one step closer to being fully operational and performing its role in the PRM experiment and help solve the proton radius puzzle.

## APPENDIX A

---

### Configurable Registers of the Citiroc1A ASIC

---

Table A.1.: Configurable registers of the slow control register<sup>[7]</sup>

<b>Field</b>	<b>Bits</b>	<b>Default</b>	<b>Position</b>	<b>Description</b>
<b>Register:channel_thr_time</b>				
ch_0	4	0	0	Channel-dependent 4-bit threshold for time discriminator.
:				
ch_31	4	0	-	-
<b>Register:channel_thr_charge</b>				
ch_0	4	0	128	Channel-dependent 4-bit threshold for charge discriminator.
:				
ch_31	4	0	-	-
<b>Register:discriminator_power</b>				
discriminator_charge_en	1	0	256	Enable charge discriminator.
discriminator_charge_pp	1	0	-	Power pulse for charge discriminator.
discriminator_latched_output	1	0	-	1: latched, 0: direct output.

Configurable registers of the slow control register continued on next page

## Appendix A. Configurable Registers of the Citroc1A ASIC

<b>Field</b>	<b>Bits</b>	<b>Default</b>	<b>Position</b>	<b>Description</b>
discriminator_time_en	1	1	-	Enable time discriminator.
discriminator_time_pp	1	1	-	Power pulse for time discriminator.
4bit_dac_charge_en	1	0	261	Enable 4-bit charge DAC.
4bit_dac_charge_pp	1	0	-	Power pulse for 4-bit charge DAC.
4bit_dac_time_en	1	1	-	Enable 4-bit time DAC.
4bit_dac_time_pp	1	1	-	Power pulse for 4-bit time DAC.
ch_0	1	1	265	0: masked, 1: unmasked.
:				
ch_31	1	1	-	-
<b>Register:track_and_hold_power</b>				
high_gain_pp	1	0	297	Enable high gain.
high_gain_en	1	0	-	-
low_gain_pp	1	0	-	Power pulse for low gain.
low_gain_en	1	0	-	Enable low gain.
weak_bias	1	0	-	1: weak bias (600kHz max), 0: high bias (5MHz max).
<b>Register:peak_detector_power</b>				
high_gain_pp	1	0	302	Enable high gain for peak detector.
high_gain_en	1	0	-	-
low_gain_pp	1	0	-	Power pulse for low gain.
low_gain_en	1	0	-	Enable low gain for peak detector.
<b>Register:select_peak_sensing</b>				
high_gain_th	1	0	306	0: peak detector, 1: track and hold.
low_gain_th	1	0	-	-
peak_sensing_cell_bypass	1	0	-	0: cell active, 1: bypass peak sensing cell.
peak_sensing_external_trigger	1	0	-	0: internal trigger, 1: external trigger.

Configurable registers of the slow control register continued on next page

Field	Bits	Default	Position	Description
<b>Register:shaper</b>				
fast_shaper_follower_pp	1	0	310	Power pulse for fast shaper follower.
fast_shaper_en	1	1	-	Enable fast shaper.
fast_shaper_pp	1	1	-	Power pulse for fast shaper.
low_gain_slow_shaper_pp	1	0	-	Power pulse for low gain slow shaper.
low_gain_slow_shaper_en	1	0	-	Enable low gain slow shaper.
low_gain_slow_shaper_time_const	3	0	-	See the table above for values.
high_gain_slow_shaper_pp	1	0	-	Power pulse for high gain slow shaper.
high_gain_slow_shaper_en	1	0	-	Enable high gain slow shaper.
high_gain_slow_shaper_time_const	3	0	-	See the table above for values.
<b>Register:pre_amp_power</b>				
low_gain_weak_bias	1	0	323	0: normal bias, 1: weak bias.
high_gain_pp	1	1	-	Power pulse for high gain preamp.
high_gain_en	1	1	-	Enable high gain preamp.
low_gain_pp	1	0	-	Power pulse for low gain preamp.
low_gain_en	1	0	-	Enable low gain preamp.
fast_shaper_low_gain	1	0	-	0: fast shaper on high gain.
<b>Register:input_dac</b>				
dac_en	1	1	329	Input DAC for bias correction.
dac_ref	1	1	-	Voltage ref: 1 = internal 4.5V, 0 = internal 2.5V, depends on vdd_dac.

Configurable registers of the slow control register continued on next page

## Appendix A. Configurable Registers of the Citiroc1A ASIC

<b>Field</b>	<b>Bits</b>	<b>Default</b>	<b>Position</b>	<b>Description</b>
ch_0	8	255	-	VSipm = V_HV - V_DAC (check what makes sense here).
ch_0_en	1	1	-	Enable channel 0 input DAC.
:				
ch_31	8	255	-	Same as ch_0 for channel 31.
ch_31_en	1	1	-	Enable channel 31 input DAC.
<b>Register:channel_preamp</b>				
ch_0_hg	6	62	619	High gain preamp setting.
ch_0_lg	6	0	-	Low gain preamp setting.
ch_0_ctest_hg	1	0	-	1: Connect injection capacitance for test signal.
ch_0_ctest_lg	1	0	-	1: Connect low gain injection capacitance.
ch_0_disable	1	0	-	1 disables preamp for channel 0.
:				
ch_31_hg	6	62	-	High gain preamp setting.
ch_31_lg	6	0	-	Low gain preamp setting.
ch_31_ctest_hg	1	0	-	1: Connect injection capacitance for test signal.
ch_31_ctest_lg	1	0	-	1: Connect low gain injection capacitance.
ch_31_disable	1	0	-	1 disables preamp for channel 0.
<b>Register:service_blocks</b>				
temp_pp	1	1	999	Enable power pulse for temperature monitoring.
temp_en	1	1	-	Enable temperature monitoring.
band_gap_pp	1	1	-	Enable power pulse for band gap reference.

Configurable registers of the slow control register continued on next page

<b>Field</b>	<b>Bits</b>	<b>Default</b>	<b>Position</b>	<b>Description</b>
band_gap_en	1	1	-	Enable band gap reference.
<b>Register:threshold_dac</b>				
charge_dac_en	1	0	1103	Enable charge threshold DAC.
charge_dac_pp	1	0	-	Power pulse for charge threshold DAC.
time_dac_en	1	1	-	Enable time threshold DAC.
time_dac_pp	1	1	-	Power pulse for time threshold DAC.
charge_threshold	10	0	-	Charge threshold value.
time_threshold	10	-	-	Time threshold value (e.g., 200 for 1 cell min, 250 for 2 cells).
<b>Register:otaq_power</b>				
high_gain_en	1	1	1127	Enable high gain for OTAQ.
high_gain_pp	1	1	-	Power pulse for high gain OTAQ.
low_gain_en	1	0	-	Enable low gain for OTAQ.
low_gain_pp	1	0	-	Power pulse for low gain OTAQ.
debug_probe_en	1	1	-	Enable debug probe.
debug_probe_pp	1	1	-	Power pulse for debug probe.
<b>Register:input_output</b>				
output_buffer_bias	1	0	1133	Output OTA buffer bias: 0 = auto bias, 1 = force on.
val_event_receiver_en	1	1	-	Enable validation event receiver.
val_event_receiver_pp	1	1	-	Power pulse for validation event receiver.
raz_chn_en	1	1	-	Enable RAZ channel.
raz_chn_pp	1	1	-	Power pulse for RAZ channel.

Configurable registers of the slow control register continued on next page

## Appendix A. Configurable Registers of the Citiroc1A ASIC

<b>Field</b>	<b>Bits</b>	<b>Default</b>	<b>Position</b>	<b>Description</b>
digital_output_en	1	1	-	Enable digital multiplexed output.
or32_output_en	1	1	-	Enable OR32 output.
or32_oc_output_en	1	1	-	Enable OR32 over-current output.
trigger_polarity	1	0	-	Trigger polarity: 0 = positive (rising edge), 1 = negative (falling edge).
or32_t_oc_en	1	1	-	Enable OR32 timeout over-current.
32_triggers_en	1	1	1143	Enable 32 triggers.

---

Table A.2.: Configurable registers of the probe register[7]

<b>Register</b>	<b>Field</b>	<b>Bits</b>	<b>Position</b>	<b>Type</b>
out_probe_fast_shaper	ch_0 ⋮ ch_31	1 ⋮ 1	0 ⋮ 31	Analog - Out_probe
out_probe_slow_shaper_lg	ch_0 ⋮ ch_31	1 ⋮ 1	32 ⋮ 63	Analog - Out_probe
digital_probe_peak_sense_lg	ch_0 ⋮ ch_31	1 ⋮ 1	64 ⋮ 95	Digital - Digital_probe
out_probe_slow_shaper_hg	ch_0 ⋮ ch_31	1 ⋮ 1	96 ⋮ 127	Analog - Out_probe
digital_probe_peak_sense_hg	ch_0 ⋮ ch_31	1 ⋮ 1	128 ⋮ 159	Digital - Digital_probe
out_probe_preamp_hg	ch_0 ⋮ ch_31	1 ⋮ 1	160 ⋮ 191	Analog - Out_probe
out_probe_preamp_lg	ch_0 ⋮ ch_31	1 ⋮ 1	192 ⋮ 223	Analog - Out_probe
input_dac_probe	ch_0 ⋮ ch_31	1 ⋮ 1	224 ⋮ 255	Analog - Out_probe_dac_5V



## APPENDIX B

---

### Fit Parameters of the S-Curve Analysis

---

Gain	Channel	$\mu$ of C1 F1	$\sigma$ of C1 F1	$\mu$ of C2 F1	$\sigma$ of C2 F1	$\mu$ of C1 F2	$\sigma$ of C1 F2	$\mu$ of C2 F2	$\sigma$ of C2 F2
58	ch_0	203.83	0.07	209.59	0.05	215.71	0.06	207.77	0.06
	ch_1	203.76	0.10	209.83	0.07	215.71	0.06	207.76	0.06
	ch_2	203.39	0.27	209.76	0.08	210.68	5.50	207.76	0.06
	ch_3	203.43	0.26	209.75	0.08	211.28	5.46	207.76	0.06
	ch_4	203.84	0.08	209.29	0.20	215.61	0.08	204.03	4.72
	ch_5	203.45	0.06	209.73	0.09	215.79	0.06	207.76	0.06
	ch_6	203.45	0.06	209.79	0.07	215.68	0.08	207.76	0.06
	ch_7	203.78	0.09	209.33	0.19	215.79	0.06	205.11	4.59
	ch_8	203.30	0.34	209.78	0.08	208.09	6.26	207.77	0.06
	ch_9	203.76	0.10	209.80	0.07	215.66	0.09	207.76	0.06
	ch_10	203.80	0.09	209.74	0.09	215.37	0.17	207.76	0.06
	ch_11	203.45	0.06	209.73	0.09	215.67	0.09	207.77	0.06
	ch_12	203.45	0.05	207.83	2.42	215.66	0.09	199.08	5.04
	ch_13	203.78	0.09	208.52	1.73	215.82	0.04	200.69	4.97
	ch_14	203.45	0.05	209.77	0.08	215.71	0.08	207.76	0.06
	ch_15	203.78	0.10	209.37	0.21	215.68	0.08	207.87	0.03
	ch_16	203.57	0.00	209.79	0.10	215.67	0.08	207.76	0.06
	ch_17	203.33	0.29	209.83	0.08	208.97	5.96	207.77	0.06
	ch_18	203.45	0.06	209.82	0.07	215.71	0.09	207.77	0.06
	ch_19	203.77	0.10	209.83	0.07	215.71	0.06	207.77	0.06
	ch_20	203.38	0.26	209.41	0.20	209.18	6.06	207.69	0.08
	ch_21	203.45	0.00	209.48	0.20	215.76	0.06	207.70	0.07
	ch_22	203.57	0.02	207.98	2.39	215.70	0.09	198.81	5.24
	ch_23	203.45	0.06	209.78	0.10	215.71	0.06	207.77	0.06
	ch_24	203.78	0.09	209.81	0.10	215.50	0.00	207.77	0.06
	ch_25	203.57	0.04	209.58	0.22	215.71	0.06	207.76	0.07
	ch_26	203.44	0.06	209.86	0.06	215.90	0.03	207.77	0.06
	ch_27	203.45	0.06	209.82	0.07	215.71	0.08	207.71	0.09
	ch_28	203.45	0.06	209.73	0.09	215.67	0.08	207.76	0.06
	ch_29	203.58	0.05	208.87	1.27	215.71	0.06	201.26	5.41
	ch_30	203.45	0.06	209.80	0.08	215.71	0.08	207.70	0.08
	ch_31	203.45	0.05	209.36	0.35	215.73	0.07	202.52	5.44

Table B.1.: Mean  $\mu$  and standard deviation  $\sigma$  of the noise distribution gained from the fit of S-curve for gain 58.

## Appendix B. Fit Parameters of the S-Curve Analysis

Gain	Channel	$\mu$ of C1 F1	$\sigma$ of C1 F1	$\mu$ of C2 F1	$\sigma$ of C2 F1	$\mu$ of C1 F2	$\sigma$ of C1 F2	$\mu$ of C2 F2	$\sigma$ of C2 F2
59	ch_0	NAN	NAN	286.45	0.14	196.76	6.06	192.84	3.08
	ch_1	NAN	NAN	731.10	1.85	193.94	14.47	194.52	3.94
	ch_2	NAN	NAN	722.27	3.14	260.12	3.10	193.59	1.78
	ch_3	NAN	NAN	174.42	30.04	734.90	3.12	194.29	3.28
	ch_4	NAN	NAN	722.99	2.58	197.46	7.03	191.72	5.85
	ch_5	NAN	NAN	730.69	1.83	741.43	1.99	193.63	3.68
	ch_6	NAN	NAN	724.90	2.67	739.22	2.39	194.28	3.66
	ch_7	NAN	NAN	724.84	2.60	746.93	1.77	192.11	8.38
	ch_8	NAN	NAN	727.13	2.09	741.23	1.66	NAN	NAN
	ch_9	NAN	NAN	731.51	1.81	742.72	1.59	194.91	6.82
	ch_10	NAN	NAN	199.37	17.71	747.08	1.45	193.87	2.34
	ch_11	NAN	NAN	725.48	2.44	745.86	1.50	195.06	5.87
	ch_12	NAN	NAN	729.47	1.65	741.21	1.81	730.21	1.78
	ch_13	NAN	NAN	728.01	1.60	194.60	11.81	186.52	18.82
	ch_14	NAN	NAN	196.86	18.67	739.68	3.27	193.76	1.92
	ch_15	NAN	NAN	731.88	1.58	742.00	1.69	268.74	0.09
	ch_16	NAN	NAN	731.34	1.29	188.54	25.05	194.89	7.43
	ch_17	NAN	NAN	726.70	1.71	746.58	1.49	195.17	7.70
	ch_18	NAN	NAN	734.32	1.00	743.01	1.31	194.84	10.03
	ch_19	NAN	NAN	732.88	1.02	197.00	8.90	726.84	2.15
	ch_20	NAN	NAN	730.04	1.73	737.53	1.87	731.45	1.52
	ch_21	NAN	NAN	731.85	1.47	193.33	14.32	190.32	12.80
	ch_22	NAN	NAN	730.34	1.09	749.33	1.37	730.83	1.75
	ch_23	NAN	NAN	734.84	0.94	196.37	12.73	232.74	0.22
	ch_24	NAN	NAN	732.42	1.12	740.63	2.39	729.67	2.21
	ch_25	NAN	NAN	733.04	0.89	196.11	7.34	731.96	0.86
	ch_26	NAN	NAN	729.33	1.76	736.03	3.03	194.91	5.42
	ch_27	NAN	NAN	735.99	0.98	740.27	2.40	732.01	1.74
	ch_28	NAN	NAN	724.77	2.47	189.68	17.66	195.04	6.91
	ch_29	NAN	NAN	735.02	1.11	197.14	6.10	181.79	29.53
	ch_30	NAN	NAN	728.13	1.48	741.47	2.17	729.51	1.61
	ch_31	NAN	NAN	735.61	0.83	737.23	3.25	726.48	2.24

Table B.2.: Mean  $\mu$  and standard deviation  $\sigma$  of the noise distribution gained from the fit of S-curve for gain 59.

Gain	Channel	$\mu$ of C1 F1	$\sigma$ of C1 F1	$\mu$ of C2 F1	$\sigma$ of C2 F1	$\mu$ of C1 F2	$\sigma$ of C1 F2	$\mu$ of C2 F2	$\sigma$ of C2 F2
60	ch_0	748.75	0.90	739.59	0.50	752.26	0.51	741.58	0.65
	ch_1	746.83	0.81	742.14	0.41	751.30	0.42	743.27	0.65
	ch_2	754.29	0.60	737.67	0.60	752.57	0.45	739.84	0.69
	ch_3	753.39	0.53	733.06	0.99	749.16	0.55	741.91	0.65
	ch_4	749.22	0.91	737.72	0.49	752.06	0.52	741.33	0.65
	ch_5	749.76	0.73	742.25	0.40	751.68	0.41	744.02	0.63
	ch_6	752.20	0.69	737.48	0.57	750.47	0.87	744.07	0.62
	ch_7	748.26	0.62	738.36	0.76	755.89	0.43	740.46	0.58
	ch_8	753.40	0.48	738.13	0.79	749.29	0.50	741.97	0.54
	ch_9	748.14	0.73	740.39	0.50	751.86	0.40	740.63	0.63
	ch_10	749.23	0.63	734.75	0.97	755.47	0.34	741.00	0.75
	ch_11	752.25	0.59	738.39	0.90	753.68	0.55	741.95	0.60
	ch_12	749.39	0.76	738.74	0.72	750.23	0.88	743.13	0.59
	ch_13	751.94	0.92	736.30	0.50	749.76	0.69	739.40	0.42
	ch_14	752.00	0.85	735.27	0.81	755.80	0.37	739.42	0.79
	ch_15	750.85	0.64	740.24	0.46	751.31	0.39	741.93	0.51
	ch_16	752.72	0.74	737.30	0.54	751.89	0.49	740.65	0.60
	ch_17	753.24	0.52	734.09	0.95	755.20	0.34	742.07	0.49
	ch_18	748.85	0.76	740.65	0.50	750.67	0.86	742.04	0.49
	ch_19	750.04	0.80	738.94	0.73	746.05	1.23	739.39	0.49
	ch_20	753.85	0.48	738.02	0.62	748.09	0.53	743.13	0.59
	ch_21	751.55	0.94	739.92	0.43	750.13	0.86	744.28	0.57
	ch_22	752.28	0.66	736.52	0.48	757.25	0.35	743.35	0.62
	ch_23	753.32	0.63	740.73	0.48	752.25	0.53	742.45	0.35
	ch_24	753.86	0.49	737.67	0.54	752.68	0.43	742.34	0.36
	ch_25	744.47	1.13	737.79	0.51	751.27	0.46	741.47	0.54
	ch_26	750.64	0.53	742.20	0.41	751.58	0.36	740.17	0.60
	ch_27	634.24	8.11	742.30	0.41	752.36	0.51	743.06	0.58
	ch_28	752.81	0.67	737.42	0.55	753.74	0.62	738.47	0.96
	ch_29	745.20	1.37	742.67	0.40	749.86	0.94	743.65	0.65
	ch_30	755.56	0.57	735.74	0.58	753.72	0.56	742.27	0.37
	ch_31	746.41	0.95	741.40	0.52	752.53	0.52	740.73	0.48

Table B.3.: Mean  $\mu$  and standard deviation  $\sigma$  of the noise distribution gained from the fit of S-curve for gain 60.

Gain	Channel	$\mu$ of C1 F1	$\sigma$ of C1 F1	$\mu$ of C2 F1	$\sigma$ of C2 F1	$\mu$ of C1 F2	$\sigma$ of C1 F2	$\mu$ of C2 F2	$\sigma$ of C2 F2
61	ch.0	750.48	0.66	738.00	0.74	752.27	0.63	741.17	0.61
	ch.1	744.52	1.14	739.01	0.94	747.48	1.24	743.65	0.63
	ch.2	753.41	0.66	735.73	0.59	750.14	1.18	740.94	0.56
	ch.3	750.89	0.94	731.88	1.15	743.42	1.15	741.58	0.69
	ch.4	751.64	0.73	735.50	0.56	752.99	0.54	741.33	0.60
	ch.5	746.79	1.11	739.61	1.10	744.60	1.57	743.23	0.73
	ch.6	749.02	1.06	734.19	1.16	743.82	1.62	743.47	0.71
	ch.7	744.56	1.22	735.16	0.98	751.80	1.04	740.21	0.65
	ch.8	747.88	1.23	733.91	1.20	739.11	2.08	738.07	1.14
	ch.9	745.26	1.25	734.10	1.55	746.84	1.35	738.18	0.96
	ch.10	744.42	1.39	732.65	1.17	748.15	1.75	741.68	0.61
	ch.11	747.46	1.22	735.52	0.86	748.37	1.42	740.75	0.73
	ch.12	747.35	1.05	732.21	1.61	741.51	2.24	739.72	0.94
	ch.13	753.57	0.54	728.04	1.79	748.57	0.90	734.98	1.29
	ch.14	753.01	0.63	735.53	0.41	755.50	0.54	740.69	0.58
	ch.15	745.09	1.66	735.59	0.98	743.70	1.76	739.18	0.94
	ch.16	755.05	0.45	727.06	2.41	750.54	0.99	737.11	1.22
	ch.17	749.95	0.89	723.25	2.89	749.48	1.38	739.96	0.84
	ch.18	745.86	1.17	732.13	1.72	739.44	2.39	737.26	1.43
	ch.19	751.54	0.77	728.58	2.11	744.19	1.11	733.06	1.80
	ch.20	750.49	1.01	731.41	1.93	739.78	2.05	738.54	1.16
	ch.21	752.19	0.84	734.99	1.22	748.81	1.04	742.61	0.73
	ch.22	749.45	1.04	724.72	2.29	752.30	1.40	738.79	0.96
	ch.23	753.85	0.63	732.30	1.97	751.88	0.72	738.53	1.01
	ch.24	750.66	1.01	727.46	2.14	750.05	1.15	737.67	1.24
	ch.25	745.71	0.86	725.03	2.22	751.89	0.65	734.94	1.65
	ch.26	749.46	0.82	740.97	0.63	748.70	1.06	739.20	0.71
	ch.27	634.66	3.42	733.96	1.41	748.30	1.29	739.11	1.09
	ch.28	751.71	1.01	735.34	0.51	751.35	1.04	736.48	1.17
	ch.29	747.79	0.80	736.37	1.15	750.14	0.87	741.53	0.83
	ch.30	755.82	0.62	726.08	2.19	751.84	0.89	738.74	1.09
	ch.31	747.03	0.87	731.70	1.72	751.26	0.87	737.39	1.18

Table B.4.: Mean  $\mu$  and standard deviation  $\sigma$  of the noise distribution gained from the fit of S-curve for gain 61.

Gain	Channel	$\mu$ of C1 F1	$\sigma$ of C1 F1	$\mu$ of C2 F1	$\sigma$ of C2 F1	$\mu$ of C1 F2	$\sigma$ of C1 F2	$\mu$ of C2 F2	$\sigma$ of C2 F2
62	ch.0	735.26	3.71	713.25	4.66	731.30	4.69	717.16	4.33
	ch.1	712.19	4.44	707.30	3.73	714.10	4.74	723.34	4.69
	ch.2	730.69	4.04	708.57	4.55	717.47	4.59	723.90	3.40
	ch.3	722.21	4.23	705.94	5.43	705.84	3.38	720.48	4.91
	ch.4	738.21	3.32	706.72	4.42	737.82	3.61	719.20	4.18
	ch.5	714.86	5.56	709.42	4.45	706.49	3.04	717.28	4.72
	ch.6	716.92	4.55	701.70	4.33	705.45	4.07	720.55	4.24
	ch.7	709.14	4.97	702.83	4.98	717.67	3.47	716.01	5.01
	ch.8	712.22	3.43	698.26	4.24	698.83	2.71	702.71	3.86
	ch.9	711.24	4.44	695.52	3.73	710.03	4.36	705.11	4.53
	ch.10	707.39	4.39	702.32	5.19	708.00	3.02	721.05	3.82
	ch.11	712.53	4.09	701.87	5.85	711.97	4.17	713.00	4.49
	ch.12	717.93	5.52	692.80	3.86	701.20	3.44	704.06	4.14
	ch.13	737.67	4.15	687.40	3.57	722.67	4.92	697.49	3.66
	ch.14	735.12	3.26	718.96	4.38	733.07	3.57	724.83	4.10
	ch.15	707.32	3.95	700.24	4.55	703.29	3.53	706.64	4.31
	ch.16	740.84	3.34	685.21	3.51	722.10	4.75	703.03	3.60
	ch.17	718.00	3.76	680.97	3.33	712.25	3.49	706.71	4.73
	ch.18	711.97	4.43	692.07	2.82	698.94	3.16	698.39	3.95
	ch.19	734.75	3.96	686.94	2.60	712.96	4.94	694.37	3.92
	ch.20	718.51	4.55	692.56	4.84	698.83	3.32	700.33	3.77
	ch.21	732.29	4.36	697.01	4.25	720.14	5.30	714.82	4.18
	ch.22	717.88	4.88	682.96	2.34	716.58	4.25	700.42	3.41
	ch.23	734.83	4.56	690.54	3.62	728.87	4.13	700.98	4.24
	ch.24	719.34	4.02	686.47	3.20	718.08	4.52	699.47	4.45
	ch.25	726.40	3.93	683.87	2.89	732.71	4.13	692.14	2.13
	ch.26	721.32	4.59	716.23	3.83	716.49	4.25	713.04	4.34
	ch.27	635.71	1.36	694.05	2.76	711.69	4.19	701.18	4.00
	ch.28	724.57	4.05	705.01	4.20	719.93	4.66	704.64	6.43
	ch.29	733.40	3.66	698.91	3.47	730.83	4.41	707.22	3.45
	ch.30	735.73	4.28	684.10	3.17	723.99	5.16	703.25	4.18
	ch.31	726.70	3.60	689.76	2.52	723.08	4.71	700.16	5.00

Table B.5.: Mean  $\mu$  and standard deviation  $\sigma$  of the noise distribution gained from the fit of S-curve for gain 62.



---

## List of Figures

---

2.1. Previous measurements of the proton radius from electron proton scattering experiments and the Lamb shift in muonic and ordinary hydrogen, the measurements differ from each other by five standard deviations. An arbitrary result for the proton radius by the PRM experiment, with the expected uncertainty is included.[1] . . . . .	4
2.2. General setup of the Amber experiment with new detectors for PRM.[7] . . . . .	5
2.3. Unified tracking station (UTS) with three layers of pixilized silicon detectors (ALPIDEs) and the scintillating fiber hodoscope (SFH).[7] . . . . .	6
2.4. Scintillating fiber hodoscope (SFH) with two of the four layers of scintillating fibers installed. The frontend electronics are not attached.[7] . . . . .	7
3.1. Sideview of the frontend electronics that will be attached on the sides of the SFH, the fiber holders will be attached to the fibers.[7] . . . . .	10
3.2. The analog frontend electronics (FEE) PCB with the six Citiroc1A ASICs. The output of the Citiroc1A ASICs is transmitted to the iFTDC over three flex PCBs.[7] . . . . .	11
3.3. The signals used by the FPGA to control and readout the two Citiroc1A ASICs.[14] . . . . .	11
3.4. General ASIC block scheme of the Citiroc1A.[14] . . . . .	12
3.5. High gain amplification of the Citiroc1A. The gain is adjustable from 0 to 1575 fF in 25 fF steps.[14] . . . . .	13
3.6. The two configurable registers of the Citiroc1A are selected using the <b>Select</b> signal. The FPGA communicates with the Citiroc1A through the signals <b>Clk_sr</b> , <b>Rstb_sr</b> , <b>Srin_sr</b> , and <b>Load_sr</b> , while the <b>Sroutr</b> signal is sent back from the Citiroc1A to the FPGA for verification.[14] . . . . .	13

## List of Figures

---

3.7. The slow control chronogram, depicting the bitstream writing process controlled by <b>Clk_sr</b> the clock signal and <b>Srin_sr</b> the data signal. A rising edge of <b>Load_sr</b> is required, after successful verification with the <b>Sroutr</b> signal to load the slow control register.[14]	14
3.8. Scheme block of internal probing system, allowing the routing of internal signals to probe pins for debugging purposes. It is configured via the probe register.[14]	15
3.9. The iFTDC with three Artix-7 FPGA, the three flex PCBs that connect the iFTDC with the FEE PCB and the connected power suply.	17
4.1. Hierarchical structure of the firmware, where each module contains its sub-modules. The black arrows indicate connections that are semantically correct but are routed through higher-level modules.	20
4.2. Structure of the status and control register of the Citiroc1A interface. The specified commands are executed when the bits are set to 1.	22
4.3. Finite state machine for configuring the Citiroc1A ASIC. States associated with the same processes are highlighted using the same color.	23
4.4. Finite state machine for verifying the configuration of the Citiroc1A ASIC. States belonging to the same processes are represented with the same color.	25
4.5. Experimental setup for the noise measurement of the frontend electronics of the scintillating fiber hodoscope.	27
4.6. Experimental setup for the noise measurement with an Evaluation Board provided by Weeroc company.	28
5.1. Threshold scan of the Citiroc1A ASICs of FPGA 1 and 2 for gain 58 from threshold 0 to 800.	32
5.2. Threshold scan of the Citiroc1A ASICs of FPGA 1 and 2 for gain 59 from threshold 0 to 800.	33
5.3. Threshold scan of the Citiroc1A ASICs of FPGA 1 and 2 for gain 60 from threshold 0 to 800.	34
5.4. Threshold scan of the Citiroc1A ASICs of FPGA 1 and 2 for gain 61 from threshold 0 to 800.	35
5.5. Threshold scan of the Citiroc1A ASICs of FPGA 1 and 2 for gain 62 from threshold 0 to 800.	36
5.6. Results of the S-curve analysis for channel 0 of Citiroc1A 1 of FPGA 1 with gain 60.	37
5.7. Results of the S-curve analysis for channel 0 of Citiroc1A 1 of FPGA 1 with gain 61.	38

5.8. Results of the S-curve analysis for channel 0 of Citiroc1A 1 of FPGA 1 with gain 62. . . . .	38
5.9. Results of the S-curve analysis for all channels of the Citiroc1A ASICs of FPGA 1 and 2 with gain 62. . . . .	39
5.10. Mean $\mu$ and standard deviation $\sigma$ determined from the S-curve fits for all channels of the Citiroc1A ASICs of FPGA 1 and 2 plotted vs the gain. . . . .	40
5.11. Comparison of the threshold scan of the frontend electronics to the thresh- old scan performed on a by Weeroc company provided Evaluation Board with a gain of 58. . . . .	41



---

## Bibliography

---

- [1] B Adams et al. *COMPASS++/AMBER: Proposal for Measurements at the M2 beam line of the CERN SPS Phase-1: 2022-2024*. Tech. rep. Geneva: CERN, 2019. URL: <https://cds.cern.ch/record/2676885> (cit. on pp. 1, 3–6).
- [2] B. Adams et al. *Letter of Intent: A New QCD facility at the M2 beam line of the CERN SPS (COMPASS++/AMBER)*. 2019. arXiv: 1808.00848 [hep-ex]. URL: <https://arxiv.org/abs/1808.00848> (cit. on p. 4).
- [3] Maxim Alexeev et al. “Design and Testing of a new Tracking System for the Proton Charge Radius Measurement with the AMBER Experiment at CERN”. In: *PoS VERTEX2023* (2024), p. 049. DOI: 10.22323/1.448.0049 (cit. on p. 5).
- [4] E. E. Chambers and R. Hofstadter. “Structure of the Proton”. In: *Phys. Rev.* 103 (5 Sept. 1956), pp. 1454–1463. DOI: 10.1103/PhysRev.103.1454 (cit. on p. 1).
- [5] The AMBER Collaboration. *AMBER Status Report 2024*. Tech. rep. Geneva: CERN, 2024. URL: <https://cds.cern.ch/record/2907624> (cit. on p. 7).
- [6] Procedia Cpi. “Classifying FPGA Technology in Digital Signal Processing: A review”. In: *International Journal of Engineering & Technology Sciences* 2023 (July 2024), p. 10 (cit. on p. 9).
- [7] Karl Eichhorn. Private Communication. Nov. 2024 (cit. on pp. 5–8, 10, 11, 47, 53).
- [8] J M Friedrich and O Denisov. *AMBER Status Report 2022*. Tech. rep. Geneva: CERN, 2022. URL: <https://cds.cern.ch/record/2810822> (cit. on pp. 6, 7).
- [9] Werner Heisenberg. *Physics and Philosophy: The Revolution in Modern Science*. Ed. by Ruth Nanda Anshen. Amherst, N.Y.: Prometheus Books, 1958 (cit. on p. 1).

## Bibliography

---

- [10] P. Kmon, P. Maj, P. Grybos, and R. Szczygiel. “An Effective Multilevel Offset Correction Technique for Single Photon Counting Pixel Detectors”. In: *IEEE Transactions on Nuclear Science* 63.2 (2016), pp. 1194–1201. DOI: 10.1109/TNS.2016.2527834 (cit. on p. 29).
- [11] Igor Konorov. Private Communication. Oct. 2024 (cit. on pp. 2, 9–12, 17).
- [12] C. Ghabrous Larrea et al. “IPbus: a flexible Ethernet-based control system for xTCA hardware”. In: *Journal of Instrumentation* 10.02 (Feb. 2015), p. C02019. DOI: 10.1088/1748-0221/10/02/C02019 (cit. on pp. 19, 20).
- [13] Barrie M. Peake. “The discovery of the electron, proton, and neutron”. In: *Journal of Chemical Education* 66.9 (1989), p. 738. DOI: 10.1021/ed066p738 (cit. on p. 1).
- [14] Weeroc. *citiroc1a-datasheet-v2-53*. Dec. 2019. URL: <https://www.weeroc.com/~documents/products/citiroc-1a/citiroc1a-datasheet-v2-53/?layout=file> (cit. on pp. 11–16).
- [15] R.L. Workman et al. *Review of Particle Physics: 2022*. Vol. 2022. Oxford: Oxford University Press, 2022. DOI: 10.1093/ptep/ptac097 (cit. on pp. 1, 3).

---

## Eidesstattliche Erklärung

---

Ich versichere hiermit an Eides statt, dass ich die von mir eingereichte Arbeit bzw. die von mir namentlich gekennzeichneten Teile selbständig verfasst und ausschließlich die angegebenen Hilfsmittel benutzt habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum

Unterschrift