Naturwissenschaftlich-Technische Fakultät

Universität Siegen

UNIVERSITÄT SIEGEN

BACHELORARBEIT

Investigation of Noise and Threshold for an ATLAS ITk Pixel Module

---

Tjark Fischer

Arbeit zur Erlangung des akademischen Grades
Bachelor of Science

vorgelegt der

Experimentellen Teilchen- und Astroteilchenphysik
Department Physik

Gutachter
Prof. Dr. Markus Cristinziani

11. Mai 2026

Hiermit versichere ich, dass ich die vorliegende schriftliche Arbeit selbstständig verfasst und keine anderen als die von mir angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinne nach entnommen sind, wurden in jedem Fall unter Angabe der Quellen (einschließlich des World Wide Web und anderer elektronischer Text- und Datensammlungen) kenntlich gemacht. Dies gilt auch für die beigegebenen Zeichnungen, bildlichen Darstellungen, Skizzen und dergleichen. Mir ist bewusst, dass jedes Zuwiderhandeln als Täuschungsversuch zu gelten hat, der die Anerkennung der Arbeit ausschließt und weitere angemessene Sanktionen zur Folge haben kann.

Siegen, ...

Datum Unterschrift

1. Gutachter: Prof. Dr. Markus Cristinziani

Contents

CHAPTER 1

Introduction

The European Organization for Nuclear Research (CERN) operates the Large Hadron Collider (LHC), the world's highest-energy particle accelerator, designed to investigate the fundamental constituents of matter and their interactions. At the LHC, protons are accelerated to near-light speed and collided at centre-of-mass energies of several TeV, enabling the study of processes at the smallest accessible length scales. One of the major experiments at the LHC is ATLAS (A Toroidal LHC ApparatuS), a general-purpose detector designed to explore a broad range of physics topics, including precision measurements of Standard Model processes and searches for physics beyond the Standard Model [1].

The ATLAS detector is composed of several subsystems arranged in a layered cylindrical geometry around the interaction point. Closest to the beam line, the Inner Detector is responsible for reconstructing the trajectories and momenta of charged particles, providing precise vertexing and tracking information that is essential for nearly all physics analyses. Since the start of LHC operation in 2009, the Inner Detector has delivered reliable performance under increasingly demanding conditions. However, the planned upgrade of the LHC to the High-Luminosity LHC (HL-LHC) will significantly increase the instantaneous luminosity and, consequently, the number of simultaneous proton–proton interactions per bunch crossing. These conditions impose stringent requirements on detector granularity, radiation hardness, and data throughput, necessitating a complete replacement of the current Inner Detector by the all-silicon Inner Tracker (ITk). The ITk is designed to maintain high tracking efficiency and precision in an environment with up to around 200 overlapping interactions per event, ensuring the continued physics performance of the ATLAS experiment under HL-LHC conditions [2, 3].

The production and qualification of ITk pixel modules constitute a large-scale international effort, in which the University of Siegen participates as an assembly, testing and quality assurance site. Within this framework, electrical characterisation of modules is performed using standardised Full Quality Control (Full QC) procedures. These measurements are sensitive to subtle effects arising from both the detector hardware and the experimental setup, requiring careful control of systematic influences.

One focus of this thesis is the investigation of a grounding-related issue observed in the multi-module testing setup at the University of Siegen. When operating several modules simultaneously on a shared high-voltage channel, the default grounding configuration was found to be susceptible to the formation of ground loops, potentially affecting measurement stability. A modification of the grounding scheme was implemented by electrically separating different ground references at the data adapter card level. The impact of this modification is evaluated through a comparative analysis of Full QC measurements, with the aim of verifying that the altered configuration mitigates ground loop effects without degrading

measurement quality.

The main focus of this thesis is the analysis of noise and threshold behaviour in ITk pixel modules based on threshold scan data obtained during different stages of the Full QC procedure. Threshold scans provide pixel-resolved information on discriminator thresholds and electronic noise by measuring the response of each pixel to injected calibration charges. The study presented here investigates correlations between noise and threshold values and examines spatial patterns across the front-end chips, with particular emphasis on edge regions where variations in pixel geometry are expected to influence the noise characteristics.

CHAPTER 2

The ATLAS Inner Tracker Upgrade

2.1 Current Inner Detector Layout

The ATLAS Inner Detector (ID) is a key component of the ATLAS experiment at the LHC, responsible for reconstructing the trajectories and momenta of charged particles emerging from proton–proton collisions. Since the start of LHC operations in 2009, the ID has operated reliably under increasingly demanding luminosity conditions [2, 3].

The Inner Detector is designed as a hybrid tracking system composed of three sub-detectors, each optimised for a specific radial region around the beamline:

1. Pixel Detector – providing the highest granularity and spatial resolution, located closest to the interaction point. It consists of multiple barrel layers and end-cap disks of silicon pixel sensors that enable precise vertex reconstruction and efficient identification of secondary decay vertices.

2. Semiconductor Tracker (SCT) – based on silicon micro-strip technology and covering intermediate radii. It provides precise tracking measurements over a larger detector volume and contributes significantly to momentum resolution.

3. Transition Radiation Tracker (TRT) – a system of gas-filled straw tubes interleaved with radiator material, offering continuous tracking at the largest radii and providing additional particle-identification capabilities via transition-radiation detection.

Together, these subsystems form a cylindrical detector shown in Fig. 2.1 inside a 2 T solenoidal magnetic field, covering a pseudorapidity range of $|\eta| < 2.5$ and ensuring precise tracking over the entire solid angle relevant for ATLAS physics analyses.

After more than a decade since the beginning of operation, the Inner Detector faces limitations due to radiation damage and high data rates expected from future LHC luminosity upgrades. With integrated doses already reaching several 10^{14} n_{eq}/cm^2 in the innermost layers, the performance of silicon sensors and front-end electronics is gradually degrading [5].

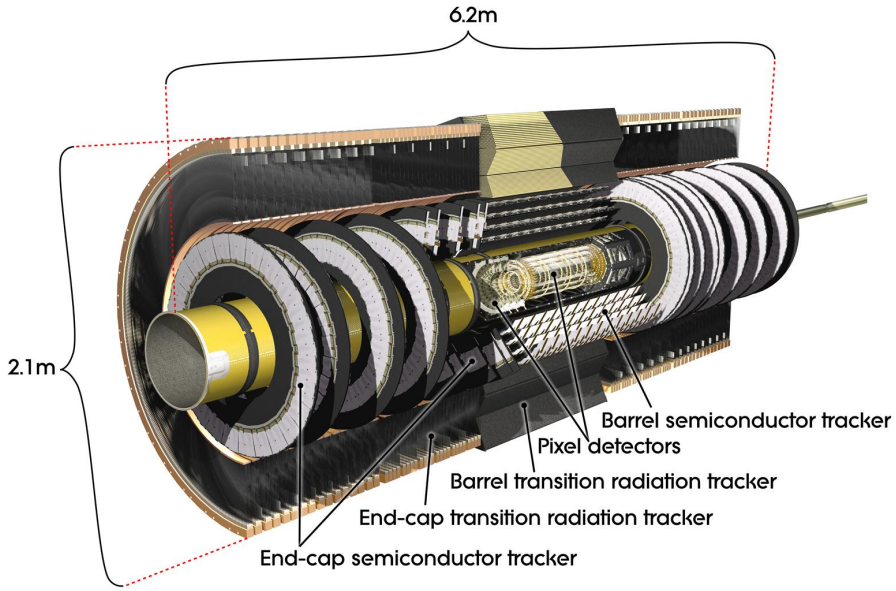

Figure 2.1: Computer generated three-dimensional view of the current ATLAS Inner Detector [4].

2.2 Motivation for the ITk Upgrade

The upcoming High-Luminosity phase of the LHC (HL-LHC) represents the next major step in the accelerator's evolution. Following nearly two decades of successful operation, the LHC will be upgraded in order to significantly increase its instantaneous and integrated luminosity. The HL-LHC aims to reach an instantaneous luminosity of up to $(5\text{–}7.5) \cdot 10^{34}$ cm^{-2}s^{-1}, corresponding to an integrated luminosity of approximately 3 ab^{-1} over its operational lifetime [6, 7].

The motivation for this upgrade lies in the goal of enhancing the physics reach of the ATLAS and CMS experiments. A higher luminosity directly increases the number of proton–proton interactions, thereby improving the statistical precision of measurements and enabling rare processes, such as Higgs boson pair production or potential signatures of new physics beyond the Standard Model, to be observed with sufficient significance. However, the higher instantaneous luminosity will also result in much more challenging experimental conditions.

At the HL-LHC, each bunch crossing will produce up to around 200 simultaneous proton–proton interactions. This extreme particle density places heavy demands on the detector's spatial resolution, radiation hardness, and readout speed. The existing ATLAS Inner Detector, designed for an order of magnitude lower luminosity, would experience excessive occupancy, signal degradation, and data losses under such conditions [8].

To ensure efficient and precise tracking in this new environment, ATLAS will replace its current Inner Detector with a completely new all-silicon tracker known as the Inner Tracker (ITk). The ITk is designed to maintain high tracking efficiency and low fake rates, even in the presence of very high particle multiplicities. Key design objectives include improved granularity, enhanced radiation tolerance, and faster, low-noise readout electronics [8].

In addition, the ITk will play a crucial role in enhancing vertexing and flavour-tagging performance,

both of which are essential for Higgs and heavy-flavour physics at the HL-LHC. The combination of higher granularity and reduced material budget will ensure superior momentum resolution and long-term stability throughout the high-radiation conditions expected over the HL-LHC's projected ten-year running time.

2.3 Detector Concept and Design Goals

The ITk represents a complete redesign of the ATLAS inner tracking system to meet the demands of the HL LHC. Building on the experience of the current ID, the ITk adopts an entirely silicon-based architecture that combines pixel and strip technologies within a unified, lightweight mechanical structure. The system is optimised to provide precise tracking, excellent vertexing, and robust pattern recognition even under the high particle densities expected at the HL-LHC [8].

The overall layout of the ITk consists of five concentric barrel layers complemented by a series of ring-shaped end-cap disks on each side. The pixel detector occupies the innermost region, ensuring high spatial resolution close to the interaction point, while the outer regions are covered by the silicon strip detector, which provides larger area coverage with fewer readout channels [8]. Fig. 2.2 depicts a schematic view of the ITk layout. Together, these two subsystems offer continuous coverage over a pseudorapidity range of $|\eta| < 4$, significantly extending the tracking acceptance in the forward direction compared to the current ID.

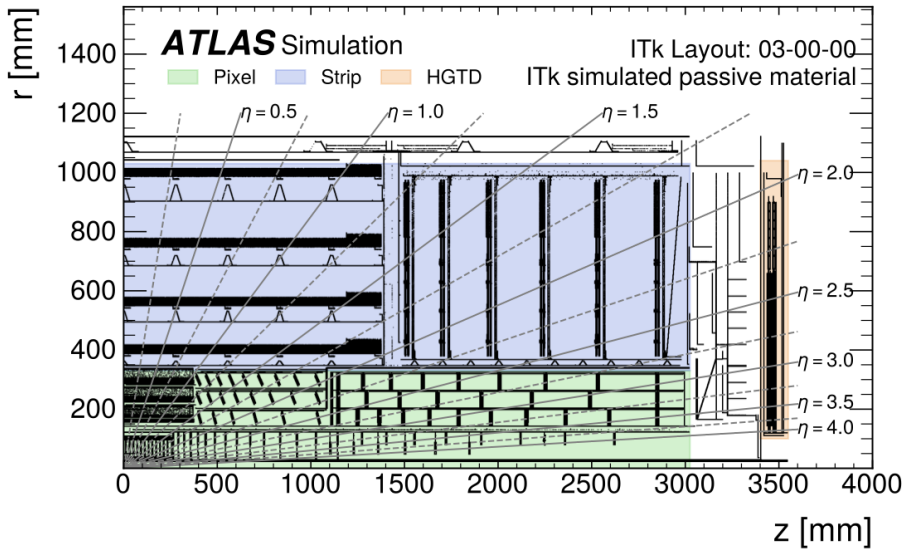

Figure 2.2: Schematic layout of the ITk for the HL-LHC phase of ATLAS. The horizontal axis corresponds to the beam line, with the interaction point at zero, and the vertical axis represents the radial distance from the interaction region up to the inner radius of the barrel cryostat [9].

As with any detector, a central design objective of the ITk is the reduction of the material budget to minimise multiple scattering and improve momentum resolution. This goal is achieved through lightweight mechanical supports, carbon-fibre composite structures, and the extensive use of low-mass electrical and cooling components. The cooling system employs two-phase CO_2 circulation, chosen

for its high efficiency and radiation tolerance, enabling stable operation at low temperatures while minimising added mass [8].

To handle the high radiation doses anticipated at the HL-LHC, the ITk sensors and front-end electronics have been developed with enhanced radiation hardness, capable of withstanding fluences up to 2×10^{16} n_{eq}/cm^2 in the innermost pixel layers. The readout architecture features serial powering and high-bandwidth data transmission to accommodate the increased hit rates while maintaining low noise and power dissipation [8].

Mechanically, the ITk is designed for long-term stability and modular integration. In the barrel region, the strip modules are mounted on staves, while the pixel modules are integrated on longerons. In the end-cap regions, the modules are arranged on ring and inclined-ring structures. These support structures provide precise alignment, mechanical stability, and efficient thermal coupling. The overall mechanical design ensures that the ITk can deliver sustained performance throughout the HL-LHC's decade-long operation period, maintaining the tracking precision and efficiency required for the ATLAS physics programme.

2.4 Implementation and Production Workflow

The production and integration of the ATLAS Inner Tracker (ITk) modules are organised as a global effort across several collaborating institutions. The workflow begins with the fabrication of the silicon sensors, which are then interconnected with readout electronics through bump-bonding. Individual modules are subsequently assembled and undergo rigorous quality assurance (QA) and quality control (QC) procedures, including electrical characterisation, noise measurement, and mechanical inspection [8].

After successful module validation, the components are integrated onto staves and longerons in the barrel region or onto rings and inclined rings in the end-cap regions. These integrated structures are again tested in dedicated system tests to verify alignment, thermal stability, and full readout functionality. The standardised QA/QC protocols ensure that modules produced at different sites meet identical performance specifications, a process referred to as *Site Qualification* [8].

The ITk production is distributed among several qualified institutes, including Siegen, each responsible for defined tasks such as module assembly, electrical testing, and stave or ring integration. The global collaboration ensures efficient workflow management, allowing the detector to be produced in parallel across multiple sites while maintaining uniform quality standards.

A simplified timeline for the ITk project is as follows:

- 2020–2022: Prototyping and design finalisation
- 2023–2025: Pre-production and site qualification
- 2025–2027: Full production and integration
- 2028: Installation during the third long shutdown (LS3) of the LHC

This structured workflow, combining module-level QA with system-level integration tests, is designed to guarantee that the ITk will meet the stringent performance requirements under the challenging HL-LHC conditions.

2.5 Role of the University of Siegen

The University of Siegen is an active member of the German Pixel Community within the ATLAS collaboration. Siegen contributes primarily to the assembly, testing, and quality assurance of the ITk pixel modules [8].

The responsibilities of the Siegen group include module assembly, which encompasses both the mechanical integration and wire bonding of sensors and readout electronics. In addition, the group performs electrical testing and full QC procedures, including applied-voltage (IV) and capacitance-voltage (CV) characterisation, noise measurements and source scans to ensure that each module meets the stringent performance specifications of the ITk.

Siegen carries out full QC measurements according to standardised ITk protocols. The results of these measurements are uploaded to the central ITk Production Database (PDB), which allows the collaboration to track the performance and yield of modules produced at different sites.

CHAPTER 3

Module Architecture

The fundamental building blocks of the ATLAS Inner Tracker (ITk) pixel detector are the 9,164 silicon pixel modules, which convert the passage of charged particles into electrical signals. For the majority of the ITk pixel detector layers, the design known as the *quad module* is employed (8,266 quad modules in total). The discussion of the module architecture is particularly relevant for understanding spatial variations in pixel response, such as the noise and threshold behaviour analysed in subsequent chapters [10].

3.1 Overview of Quad Modules

A quad module consists of a single planar silicon sensor tile onto which four identical front-end (FE) readout ASICs are bump-bonded. Each FE chip, based on the ITkPix design developed for the HL-LHC upgrade, measures approximately 20 mm $\times$ 20 mm and comprises a matrix of 400 columns and 384 rows of individual pixel cells, resulting in 153,600 readout channels per chip. The four chips are arranged in a 2×2 array on the sensor tile, forming an actively instrumented area of roughly 40 mm $\times$ 40 mm per module. The planar sensor itself is produced using radiation-hard n-in-p silicon technology [10].

The FE chips are wire-bonded to a flexible printed circuit (flex) that provides power, data links, and control signals, creating a fully functional module ready for electrical testing and integration into the detector support structures [10].

Fig. 3.1 depicts a quad module with power and data pigtails connected to it.

3.2 Pixel Cell Organisation and Edge Effects

Within each sensor tile, the individual pixel cells are arranged in a regular grid with a nominal pitch of 50 µm in each of the two in-plane dimensions. This fine segmentation ensures high spatial resolution for particle tracking and high hit density operation under the High-Luminosity LHC (HL-LHC) conditions. The uniform pixel matrix covers the majority of the sensor area under each FE chip, and this structure is repeated for all four chips on the quad module.

However, a quad module necessarily comprises regions where the active sensor area spans the gap between adjacent FE chips. To avoid dead zones in the sensitive area and maintain full coverage in what

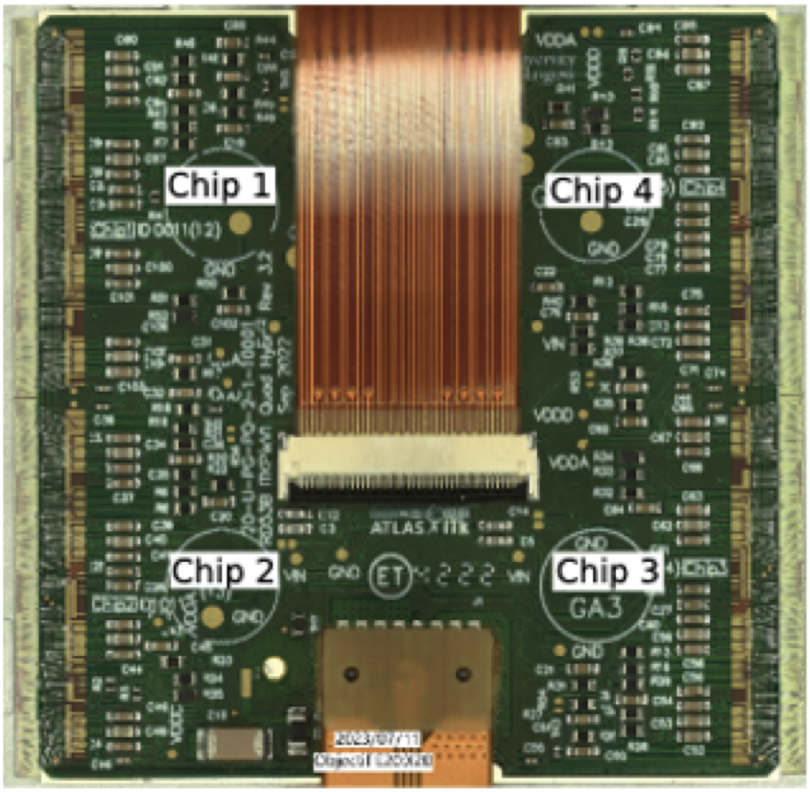

Figure 3.1: This is an image of the readout side of a quad module as described in Section 3.1 [11].

is referred to as the *inter-chip regions*, sensor pixels that lie across chip boundaries are fabricated with slightly modified geometries. This results in two additional effective pixel sizes: Larger 100×100 µm^2 pixels and elongated 100×50 µm^2 pixels (see Fig. 3.2). The elongated pixels bridge the gap where two FE chips meet by having, on each side, two sensor pixels in the inter-chip region connected to the outermost pixel of one FE chip. This leaves the sensor pixel under the outermost FE-chip pixel unmatched, which is solved by connecting it to the closest FE-chip pixel in the second row of FE-chip pixels — in addition to the sensor pixel under the second row FE-chip pixel, thus forming a second row of effectively elongated 100×50 µm^2 pixels. This layout, shown in Fig. 3.2, can be repeated except for the centre of the module, where all four FE chips meet, and therefore the largest inter-chip region forms. Here, a 4×4 array of 100×100 µm^2 pixels is utilised, where the four larger sensor pixels of each quadrant of the 4×4 array are connected to the respective readout pixels at the corner of the FE chip.

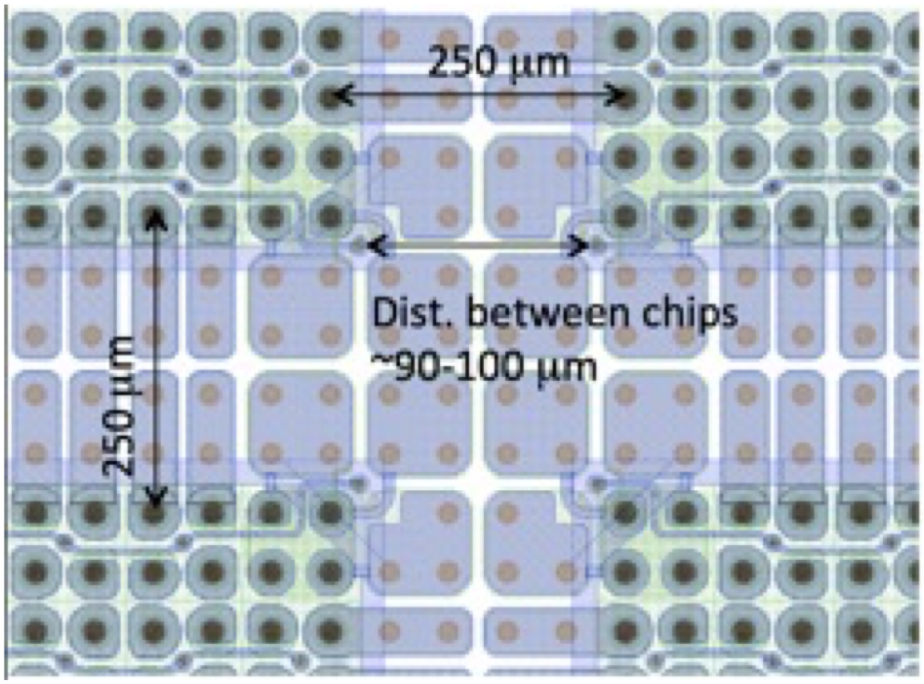

Figure 3.2: Interchip region of a quad module showing the different pixel sizes necessary to maintain full coverage. The typical distance between the chip edges of two FE chips is approximately 90-100 µm [10].

3.3 Implications for Noise and Threshold Measurements

Because the noise performance of a pixel cell depends on its capacitance and coupling to the readout electronics, the larger sensor area per readout pixel in the inter-chip regions typically results in a higher noise level compared to standard pixel cells. This effect influences threshold measurements, particularly in scans where the threshold is varied and noise is extracted from the S-curve response of each pixel. In threshold scans, an increased noise contribution broadens the S-curve explained in Section 4.4, potentially resulting in a systematic shift and a larger spread in the threshold values for pixels in inter-chip regions. The different pixel sizes inform the interpretation of the observations of the threshold and noise data investigated in Chapter 6.

CHAPTER 4

Testing Procedure

This chapter describes the electrical testing procedures applied to ITk pixel modules at the University of Siegen. It establishes the experimental context for the investigations presented in the following chapters. Chapter 5 examines a modification of the standard testing setup motivated by observations made during Full QC measurements, while Chapter 6 analyses data obtained from threshold scans at different tuning stages. The focus of this chapter is placed on the Full Quality Control (Full QC) workflow, with particular emphasis on threshold scans performed at different stages of the tuning procedure. The basis is an internal document of the ATLAS collaboration on electrical specifications and QC procedures for ITk Pixel modules [12].

4.1 Overview of Electrical Tests Performed at the University of Siegen

The University of Siegen performs electrical quality control measurements on fully assembled pixel modules. The local testing programme follows the centrally defined ITk Pixel electrical QC procedures and is designed to verify that each module meets the electrical specifications required for operation in the ATLAS detector.

The tests conducted at Siegen include visual inspection, electrical functionality checks, HV characterisation, calibration measurements, pixel performance tests, and detailed pixel and module tuning procedures. These measurements are performed using standardised hardware, readout systems, and analysis tools built on site or partly provided by the collaboration to ensure consistency across all production and testing sites. The results are uploaded to the central ITk Production Database and contribute to the overall module grading and acceptance process.

4.2 Experimental Setup

Electrical tests are carried out using a dedicated test stand consisting of low-voltage and high-voltage power supplies, a power adapter card, a readout adapter card, power and data pigtails, a ^{90}Sr radioactive source and precision multimeters for voltage and current measurements. The module under test is connected to a data adapter card with a readout via DisplayPort cables and is controlled using the Yet Another Rapid Readout (YARR) data acquisition framework [13, 14].

Particular care is taken to ensure a well-defined grounding scheme and reproducible cabling configuration, as the electrical measurements are sensitive to noise pickup via potential ground loops. A schematic view of the test setup is shown in Fig. 4.1, which illustrates the connection of the modules to the power supplies, readout electronics, and monitoring equipment.

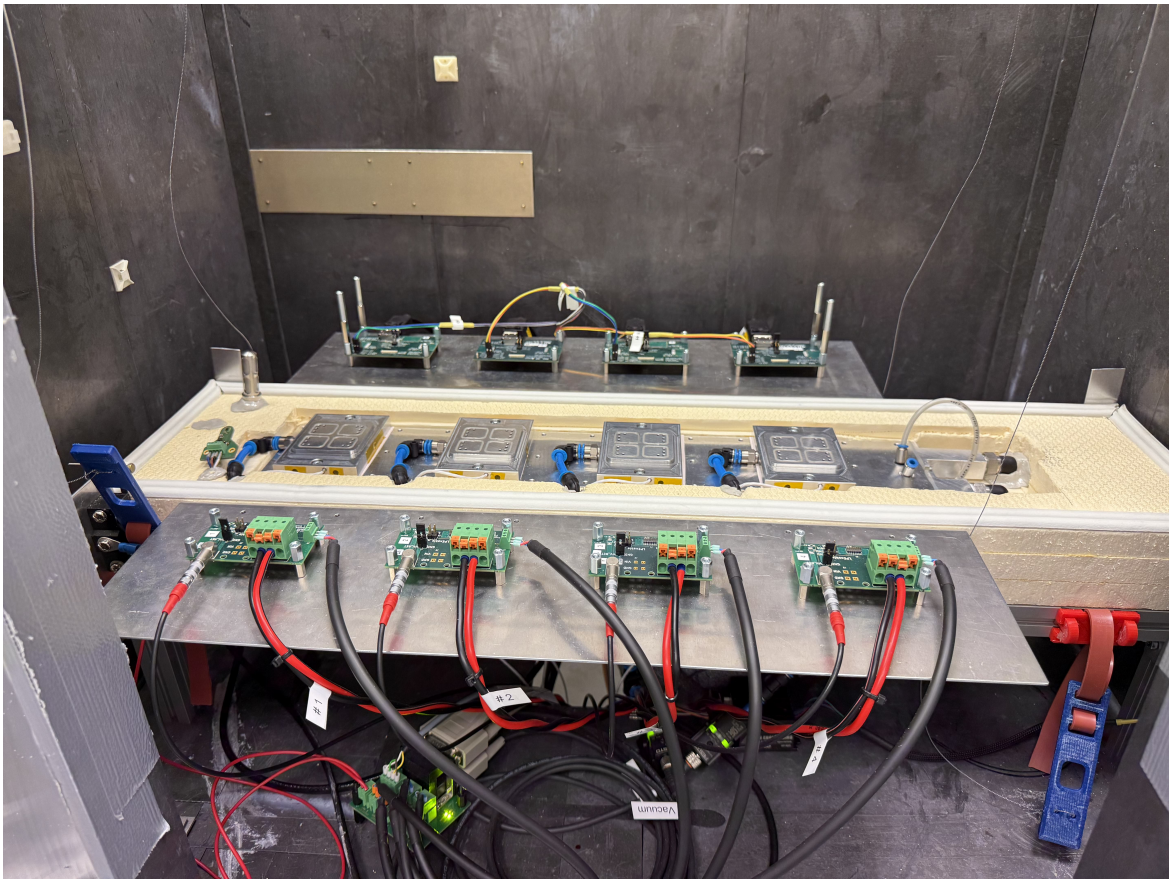

Figure 4.1: The test stand used for the Full Quality Control procedure at Siegen. Up to four modules can be tested at once, each sitting on a separate vacuum chuck. The power pigtail of each module connects to the power adapter card (front) while the data pigtail connects to the readout adapter card (back).

4.3 Full Quality Control Procedure

The Full Quality Control (Full QC) constitutes a comprehensive series of electrical tests performed on an ITk pixel module and is typically executed after module assembly for final module characterisation after thermal cycling. Its primary purpose is to verify the electrical integrity, stability, and pixel-level performance of the module to approve it for operation in the ATLAS detector. The Full QC procedure follows a standardised sequence defined by the ITk Pixel collaboration and is identical across all test sites, including the University of Siegen site.

The Full QC begins with the Minimum Health Test (MHT), which serves as an initial functionality check. During the MHT, the module is powered up and basic communication with all front-end chips is verified. Configuration registers are written and read back to ensure reliable data transmission, and a first threshold scan is performed to confirm that the majority of pixels respond to charge injection. The MHT is intended to identify severe defects, such as non-responsive chips or large groups of failing pixels, at an early stage.

Following the MHT, a series of calibration and monitoring measurements is carried out. The ADC calibration determines the response of the on-chip analogue-to-digital converters, which are used to digitise internal voltage and current monitoring signals. This calibration establishes the conversion between ADC counts and physical quantities, enabling accurate monitoring of operating parameters throughout the test.

The analogue readback measurements exploit dedicated monitoring circuits within the front-end chips to read back analogue quantities such as supply voltages, bias voltages, and reference currents. These measurements are used to verify the correct operation of internal regulators and biasing circuits and to detect potential instabilities or deviations from nominal operating conditions.

A key component of the Full QC is the qualification of the shunt-low-dropout (SLDO) regulators. The SLDOs provide locally regulated supply voltages to the front-end electronics and are essential for stable operation, especially in serial powering schemes. During SLDO qualification, the regulation behaviour, voltage stability, and current handling capability are tested under defined load conditions to ensure reliable performance.

Additional calibration steps include the calibration of the input voltage and injection capacitance measurements, which characterise the charge injection circuitry used for threshold and noise measurements. These calibrations ensure that the injected charge is well understood and consistent across pixels and front-end chips.

The Full QC includes a complete tuning of the module. This process involves adjusting global and pixel-level analogue-to-digital converter settings such that the discriminator thresholds are centred on the target value and the threshold dispersion across the pixel matrix is minimised. Threshold scans are performed before and after tuning to quantify the tuning performance and to extract pixel-wise threshold and noise values. These data form the basis for the detailed noise and threshold analyses presented in Chapter 6.

Following the tuning procedure, a source scan using a ^{90}Sr radioactive source is performed as part of the Pixel Failure Analysis. In this measurement, electrons originating from the β^- decay of the source traverse the sensor and generate signals in the pixels, allowing the response of each pixel to be evaluated. The recorded hit statistics are used to identify defective pixels. The source scan thereby provides a complementary assessment of pixel functionality, in particular revealing issues that may not be visible in purely electrical or injection-based measurements.

Throughout the Full QC procedure, all measurement results are recorded and uploaded to the central ITk Production Database. The combination of functionality checks, calibration measurements, and tuning ensures that only modules meeting the electrical performance requirements are accepted for integration into the ITk detector.

4.4 Threshold Scans and Tuning Stages

Threshold scans play a central role in the Full QC, as they provide pixel-by-pixel information on threshold dispersion, electronic noise, and overall tuning quality. In a threshold scan, a known calibration charge is injected into each pixel while the discriminator threshold is varied, allowing the extraction of S-curves (see Fig. 4.2) from which the threshold and noise values are determined.

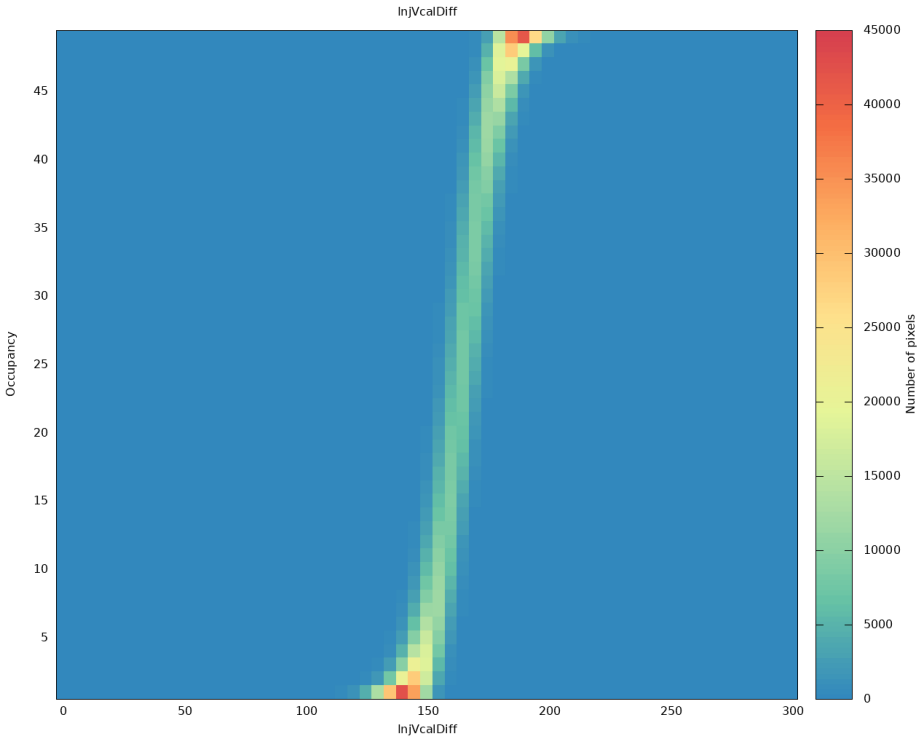

Figure 4.2: This is an exemplary two-dimensional heat map obtained from the superposition of all pixel S-curves of a single front-end chip, measured during a threshold scan. In such a scan, a known calibration charge is injected into each pixel while the discriminator threshold is varied, and the hit probability is recorded. For an ideal, noise-free pixel, the resulting S-curve would correspond to a step function at the true threshold value. In practice, electronic noise leads to a smearing of this transition, resulting in a characteristic error function shape whose width is determined by the noise. The horizontal axis of the shown heat map represents the injected calibration charge, while the vertical axis shows the corresponding occupancy. The colour scale indicates the number of pixels contributing to each bin.

Within the Full QC, threshold scans are performed at several distinct stages. An initial scan is taken during the Minimum Health Test to assess basic pixel functionality. Additional scans are recorded before and after the tuning procedure. The scans before tuning reflect the intrinsic dispersion of threshold values and noise values with default or globally tuned settings, while the post-tuning scans demonstrate the effectiveness of the pixel-level tuning in equalising thresholds and reducing threshold dispersion.

CHAPTER 5

Ground Loop Issue with Testing Setup

5.1 Testing Setup Configuration and Ground Loop Hypothesis

At the University of Siegen, the ITk pixel module testing setup shown in Fig. 4.1 is designed to allow the simultaneous electrical qualification of up to four modules. This parallel operation is essential for efficient throughput during the production and qualification phases of modules but introduces additional complexity in the distribution of power and reference potentials. In particular, all modules share a common high-voltage (HV) ground, while the HV phase is switched by a relay card and low-voltage (LV) powering, control and readout are provided via dedicated data adapter cards.

In the default configuration of the data adapter card, two ground-related contacts labelled DAQ_CMD GND and GND are electrically connected via removable jumpers. The GND contact serves as the common electrical ground reference for the module power and readout electronics, while DAQ_CMD GND provides the ground reference for command and control signals originating from the data acquisition system. Under standard operating conditions, connecting these two contacts ensures a well-defined common reference potential between the DAQ system and the module electronics, which is required for stable communication and reliable signal integrity.

When operating multiple modules simultaneously on the shared HV channel, it was observed that the default grounding scheme is susceptible to the formation of a ground loop. In such a scenario, multiple electrical return paths between different ground references exist, potentially picking up small parasitic currents and allowing them to circulate. In sensitive mixed-signal systems, such effects can manifest themselves as increased noise, baseline fluctuations, or unstable analogue measurements, particularly in threshold- and noise-related scans.

Based on this consideration, a modification of the grounding scheme was proposed. The jumpers connecting DAQ_CMD GND and GND on the data adapter card shown in Fig. 5.1 were removed, thereby electrically separating the command ground from the main module ground at the data adapter card level. This intervention was intended to interrupt potential ground loop paths while preserving a well-defined ground reference for both the module electronics and the DAQ system through their respective power supply connections. Importantly, no other aspects of the test stand configuration or measurement procedure were altered.

To evaluate the impact of this modification, a comparative measurement strategy was adopted. Full Quality Control (Full QC) measurements of a module operated individually on an unmodified test stand were compared to measurements of the same module operated in the modified grounding

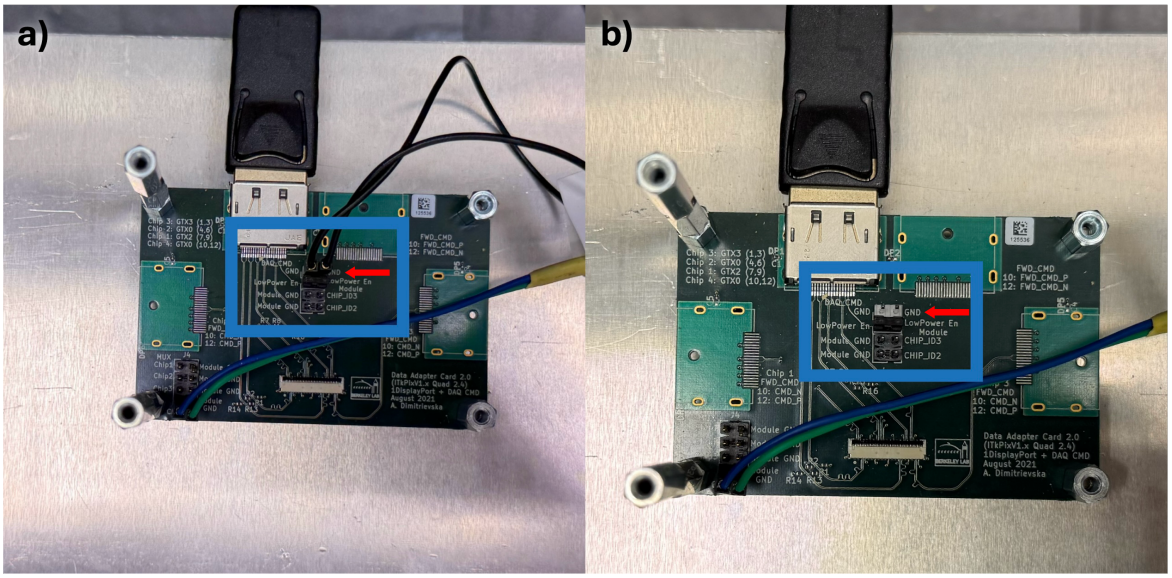

Figure 5.1: This image shows the data adapter card for one module with: a) the original configuration including a jumper between DAQ_CMD GND and GND, b) the modified configuration without the jumper.

configuration, together with three additional modules on the test stand. In both cases, the same two FE chips were active. The additional modules primarily served to reproduce the electrical conditions of simultaneous multi-module operation. The comparison focused on the consistency of electrical performance indicators, aiming to verify that the grounding modification neither degrades measurement quality nor introduces systematic deviations.

5.2 Comparison of Electrical Performance in the Original and Modified Testing Setup

For the vast majority of investigated parameters, no significant differences are observed between the original and the modified testing setup. Only three parameters exhibit noticeable deviations, which are discussed in the following.

5.2.1 Deviation in the Analog-to-Digital Converter (ADC) Calibration

One of these deviations concerns the ADC_CALIBRATION_OFFSET values listed in Tab. 5.1. For both FE chips, lower offset values are observed in the modified configuration compared to the original setup.

The Analogue-to-Digital Converter (ADC) calibration establishes the relationship between an analogue input signal and the corresponding digital output of the on-chip ADC. In particular, the ADC calibration offset corresponds to the digital ADC value associated with a zero analogue input signal and compensates for systematic offsets introduced by the analogue front-end circuitry.

FE chip	ADC_CAL_OFFSET (orig.)	ADC_CAL_OFFSET (mod.)	QC criteria
0x17537	16.0	10.0	[-9, 31]
0x17567	8.0	3.0	[-9, 31]

Table 5.1: ADC_CALIBRATION_OFFSET values for Module 20UPGM22201249 (internally referred to as Si_011) with two active FE chips in the original and modified configuration.

Although the modified setup yields consistently lower ADC_CALIBRATION_OFFSET values, these remain well within the allowed QC range. Consequently, the observed difference is not considered problematic and does not indicate a degradation of the measurement quality.

5.2.2 Deviation in the Shunt Low-Dropout Regulator (SLDO)

The second deviation concerns the parameter SLDO_ISHUNTA corresponding to the shunt current of the SLDO supplying the analogue domain. This current represents the fraction of the input current that is not consumed by the load but is instead diverted through the internal shunt element in order to stabilise the output voltage. The value of SLDO_ISHUNTA is accessed through the ADC using the analogue readback functionality.

The parameter SLDO_VINA_VIND provides information on the input voltage conditions of the SLDO and corresponds to the difference between the input voltages of the analogue and digital regulators. Monitoring this quantity allows potential imbalances or irregularities in the supplied input voltages to be identified, which could otherwise affect the stability of the on-chip voltage regulation.

In an initial comparison, a pronounced difference between the original and modified testing setup is observed, with the SLDO_ISHUNTA values in the original configuration partially falling below the lower QC limit, as shown in Tab. 5.2.

FE chip	SLDO_ISHUNTA (orig.)	SLDO_ISHUNTA (mod.)	QC criteria
0x17537	0.005	0.291	[0.005, 999]
0x17567	0.004	0.280	[0.005, 999]

Table 5.2: SLDO_ISHUNTA values for Module 20UPGM22201249 (internally referred to as Si_011) with two active FE chips in the original and modified configuration.

To further investigate this behaviour, an additional measurement was performed using Module 20UPGM22211010 (internally referred to as Si_015), this time with four active front-end chips. The results obtained with both the original and modified setups were compared to an earlier Full QC measurement of the same module. As summarised in Tab. 5.3, the SLDO_ISHUNTA values are consistently low, showing hardly any change across all three measurements and violating the QC criteria

for one of the four FE chips, 0x172b4, in each case. Furthermore, the parameter SLDO_VINA_VIND exhibits a QC criteria violation for FE chip 0x17235, as shown in Tab. 5.4.

FE chip	ISHUNTA (orig.)	ISHUNTA (mod.)	ISHUNTA (old)	QC criteria
0x172b4	0.002	0.002	0.002	[0.005, 999]
0x17225	0.012	0.012	0.013	[0.005, 999]
0x17235	0.009	0.009	0.010	[0.005, 999]
0x17294	0.013	0.013	0.013	[0.005, 999]

Table 5.3: SLDO_ISHUNTA values for Module 20UPGM22211010 (internally referred to as Si_015) with four active FE chips in the original and modified configuration as well as an older Full QC measurement of the same module.

FE chip	VINA_VIND (orig.)	VINA_VIND (mod.)	VINA_VIND (old)	QC criteria
0x172b4	-0.00	-0.00	-0.00	[-0.05, 0,05]
0x17225	-0.01	-0.01	-0.01	[-0.05, 0,05]
0x17235	0.06	0.06	0.06	[-0.05, 0,05]
0x17294	-0.01	-0.01	-0.01	[-0.05, 0,05]

Table 5.4: SLDO_VINA_VIND values for Module 20UPGM22211010 (internally referred to as Si_015) with four active FE chips in the original and modified configuration as well as an older Full QC measurement of the same module.

Since the observed deviations in both SLDO_ISHUNTA and SLDO_VINA_VIND are present independently of the grounding configuration and are also observed in earlier measurements, they cannot be attributed to the modification of the testing setup. These parameters are therefore not further investigated within the scope of this thesis.

5.2.3 Deviation in the Pixel Failure Analysis (PFA)

The third deviation is observed in the source scan as part of the PFA, where a significantly higher number of disconnected pixels (meaning pixels that recorded less than 10 hits) are identified on FE chip 0x17567 in the modified configuration, as summarised in Tab. 5.5.

An inspection of the spatial distribution of the affected pixels, shown in Fig. 5.2, reveals a pronounced accumulation in the top-right region of the chip. This area is partially shielded from the radioactive source by large electrical components, most notably the power connector. As a consequence, some pixels in this region did not record a sufficient number of hits during the source scan to satisfy the acceptance criteria and were therefore classified as disconnected.

Parameter	Analysis result (orig.)	Analysis result (mod.)	QC criteria
ELECTRICALLY_FAILED	72	87	[0, 7680]
DISCONNECTED_PIXELS	21	209	[0, 7680]
FAILING_PIXELS	93	296	[0, 7680]

Table 5.5: Pixel failures for FE chip 0x17567 of Module 20UPGM22201249 (internally referred to as Si_011) in the original and modified configuration after differently timed source scans.

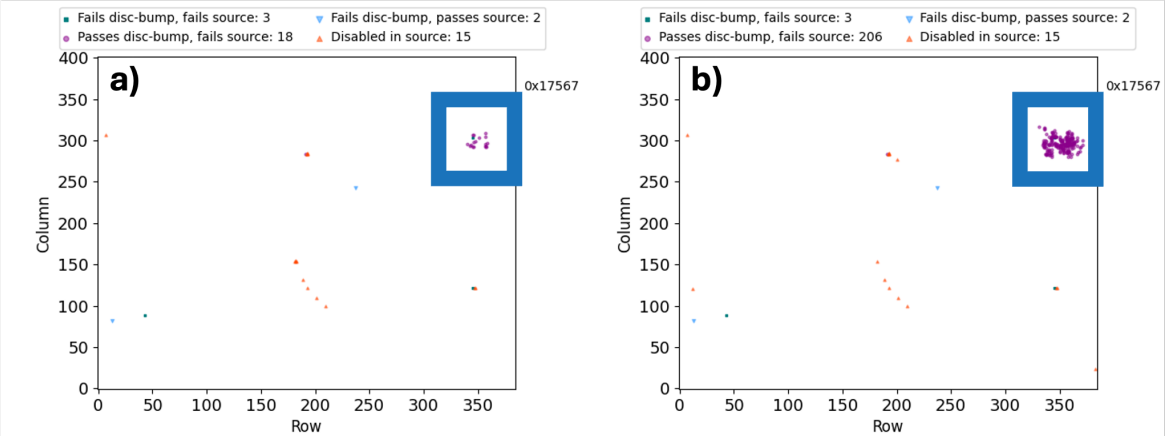

Figure 5.2: These are the failed pixels after source scan of FE chip 0x17567 of Module 20UPGM22201249 (internally referred to as Si_011). a) shows the results for the original setup with a source scan duration of 45 minutes. b) shows the results for the modified setup with a source scan duration of 30 minutes.

The difference in the number of disconnected pixels between the two configurations is most likely explained by the different source scan durations. While a scan time of 45 minutes was used for the original setup, the modified configuration employed a reduced duration of 30 minutes. The shorter exposure time increases the probability that pixels located in shielded regions fail to accumulate the required number of hits. A longer source scan duration, for example 60 minutes, would likely mitigate this effect and reduce the number of falsely identified disconnected pixels in these regions.

Therefore, these differences cannot be attributed to the modification of the testing setup as well and are not further investigated within the scope of this thesis.

5.3 Conclusion on the Grounding Modification

The comparative analysis of the electrical performance obtained with the original and the modified testing setup shows no evidence of degradation in measurement quality resulting from the removal of the jumper between `DAQ_CMD GND` and `GND`. For the vast majority of evaluated parameters, the results obtained in both configurations are consistent and remain well within the defined Quality Control criteria.

The few observed deviations, namely in selected SLDO readback quantities and the Pixel Failure Analysis, can be attributed to effects unrelated to the grounding modification, such as intrinsic chip behaviour or differences in source scan duration.

The removal of the jumper thereby constitutes a suitable and effective measure to mitigate ground loop effects in the multi-module testing setup without introducing adverse side effects on the measurement results. Based on these findings, the modified configuration without the jumper will henceforth be used consistently for electrical testing at the University of Siegen.

CHAPTER 6

Noise-Threshold Correlation

6.1 Expected Noise Patterns

Since the pixels on the connecting edges of the FE chips have a larger surface area than the others (as described in Chapter 3), their noise values are expected to be larger as well. Therefore, an accumulation of pixels with above-average noise levels on the said edges is expected to be found.

6.2 Analysis Strategy

To test this assumption, the Full QC test results from June 18, 2025, of Module 20UPGM23211410 (internally referred to as Si_030) with FE chips 0x240bb, 0x2403a, 0x24162 and 0x24174 are used. To distinguish between the different threshold scans during the Full QC procedure, *preTun* refers to the one during Minimum Health Test (MHT), *noTun* is used as a label for the scan during the tuning procedure before global or pixel threshold tuning and the scan *postTun* runs after global and pixel threshold tuning are finished. preTun runs with tuning parameters of the previous tuning still active, while for noTun, these parameters are reset and the characteristics therefore come closest to an untuned module. Hence, the focus lies on noTun and postTun. From now on, preTun, noTun and postTun are referred to as states.

The code setup described in detail in Appendix A reads the threshold scan data of the different states and analyses the distribution of noise and threshold values. As the goal is to investigate the connecting edges of the FE chips for any irregularities, both the noise and threshold data sets are cut at certain deviations from their respective means to classify outliers beyond these cuts. The threshold data is cut using the lower (1 350 e) and upper (1 650 e) bounds of the QC criteria centred around the targeted tuning value of 1 500 e. Every pixel outside that range is registered as an outlier. From the noise data, every pixel further than $3\,\sigma$ from the mean contributes to the outlier mask. Combining the outlier mask of the threshold distribution and the noise distribution, a common outlier mask without duplicates is created, from which the total number of outliers can be counted. By applying this procedure, a set of characteristics, as exemplarily shown for FE chip 0x240bb in Tab. 6.1, is obtained.

This can be visualised by a set of plots (Fig. 6.1) for which again FE chip 0x240bb is chosen as an example, in particular the preTun data. In Fig. 6.1 a) and d), the noise and threshold distributions are plotted as histograms and the cuts are shown as dashed grey lines. In Fig. 6.1 b), they are then

state		mean [e]	std. dev. [e]	lower cut [e]	upper cut [e]	outliers [count]	tot. outliers [count]
preTun	noise	108.5	11.30	74.6	142.4	1457	1937
	threshold	1 492.7	42.68	1 350.0	1 650.0	502	
noTun	noise	100.3	11.72	65.1	135.4	1242	153171
	threshold	2 492.6	331.85	1 350.0	1 650.0	153169	
postTun	noise	108.5	11.38	74.3	142.6	1397	1872
	threshold	1 495.7	40.60	1 350.0	1 650.0	484	

Table 6.1: This table shows the characteristics of the noise and threshold distributions of FE chip 0x240bb in each state.

combined into a single correlation plot. Here, it can already be observed which cuts produce which amounts of outliers. Fig. 6.1 c) shows the position of said outliers on the FE chip. A strong tendency towards the bottom and left edge of the FE chip is obvious.

To further investigate the connection between the placement of an outlier on the FE chip and the cuts that classified it as such, each kind of outlier (determined by the cuts it has been classified by) is represented in its own mapping. This results in eight maps of the FE Chip surrounding the correlation plot from Fig. 6.1 b). Fig. 6.2 colour codes which maps belong to which kind of outlier.

6.3 Comparison of the States

As mentioned in Section 6.2, the threshold scan at the preTun state is performed with tuning parameters from the previous tuning. Therefore, the most valuable comparison is between noTun, being similar to an untuned module, and postTun, with proper global and pixel tuning. This section shows this comparison for the FE chip 0x240bb. The figures and tables for FE chips 0x2403a, 0x24162 and 0x24174 can be found in Appendix B.

Tab. 6.1 indicates that while noTun does not show large differences in the noise characteristics compared to postTun, the threshold characteristics reveal a substantial increase in outliers classified by the threshold cuts, with the mean of the distribution being well above the upper cut. This can also be seen in Fig. 6.3, rendering Fig. 6.3 c) illegible.

In contrast, the effects of the tuning process on the threshold distribution become visible when compared with Fig. 6.4. With a substantially smaller standard deviation and the targeted tuning value of 1 500 e within 0.11 σ of the mean value, the tuning process can be considered successful. This rapidly reduces the number of outliers classified by the cuts on the threshold distribution. A concise summary of the distributions of each state is visualised in Fig. 6.5. It can be seen that the noise and threshold distributions for preTun and postTun are nearly identical, indicating that no significant changes to the tuning parameters occurred between the previous and the most recent tuning procedure.

The comparison of the noise distributions of noTun and postTun does not deliver any interesting insights, as the differences are marginal.

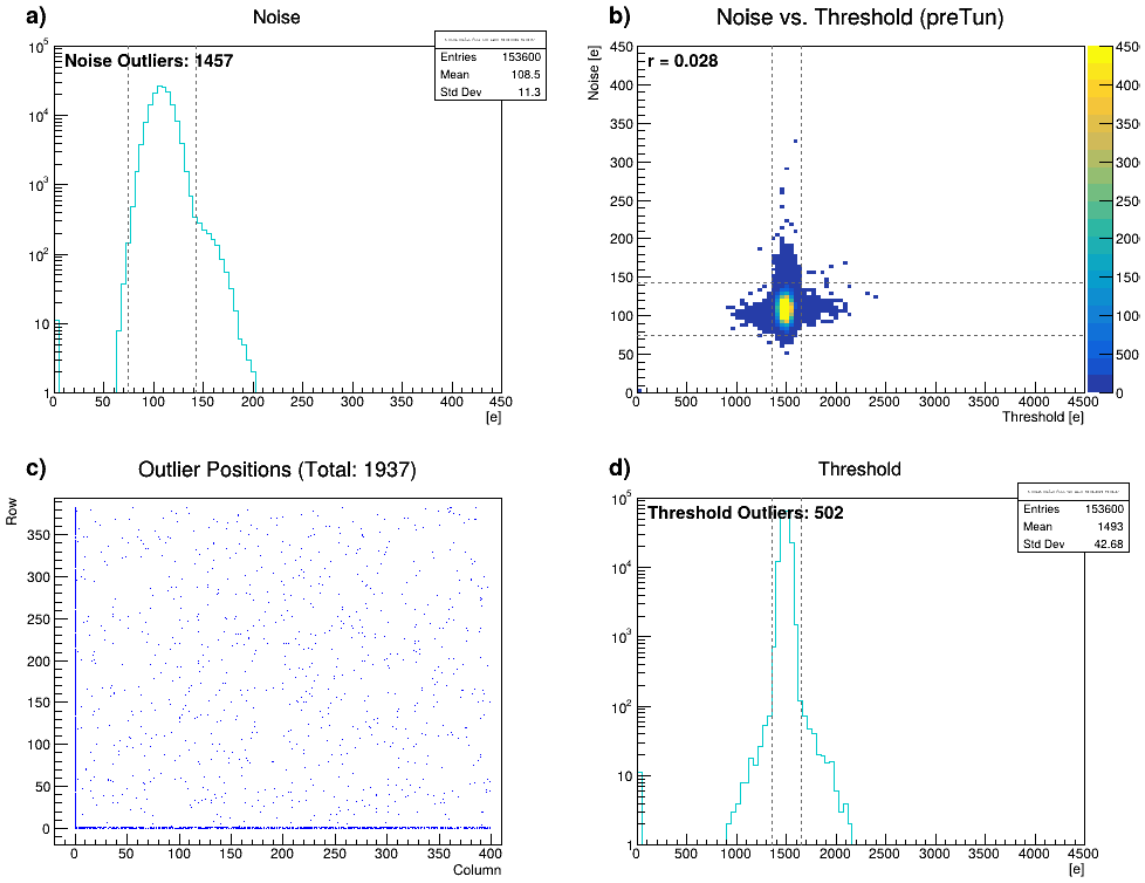

Figure 6.1: FE Chip 0x240bb, preTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

6.4 Location of Outliers on the FE Chip

Further inspecting the location of outliers on the FE chip by using the visualisation explained along Fig. 6.2, Fig. 6.6 shows outliers distributed randomly and equally across the FE chip, except for the region of larger noise. Here, an accumulation of outliers located in the two lowest rows and the two leftmost columns emerges.

This matches the expectations derived from the wiring of sensor pixels to readout channels in Section 3.2. The sensor pixels of the interchip regions are wired to the readout channels on the edge of the respective FE chips and thereby, for each edge of a FE chip facing an interchip region, two rows of readout channels effectively triple their assigned sensor area. This makes them more susceptible to electronic noise and highlights them in the analysis of threshold scan data in this chapter.

Although only discussed for FE chip 0x240bb in postTun, the findings are consistent across FE chips and states. The respective figures are shown in Appendix B.

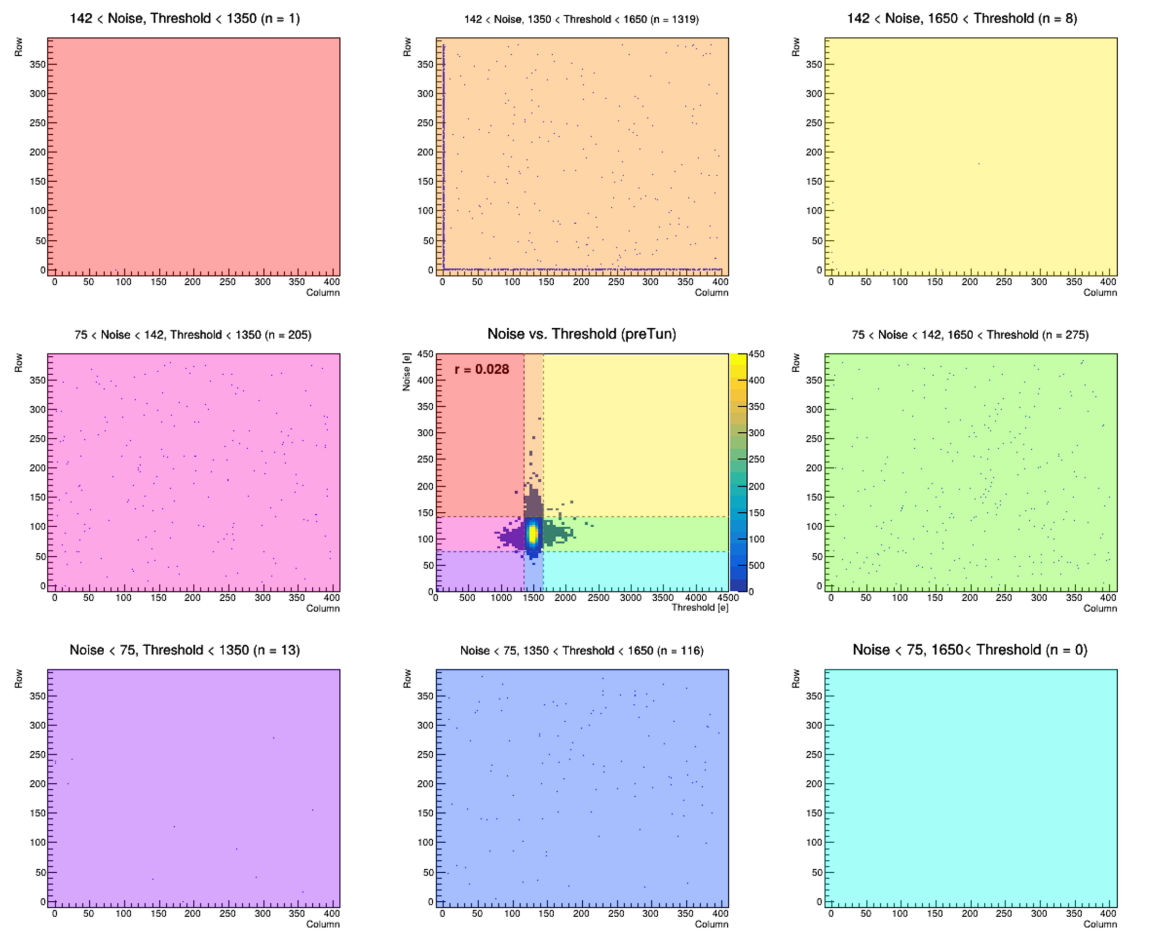

Figure 6.2: Different kinds of outliers distributed on separate regions for FE Chip 0x240bb in state preTun (colour coded for visualisation, cuts in units of e).

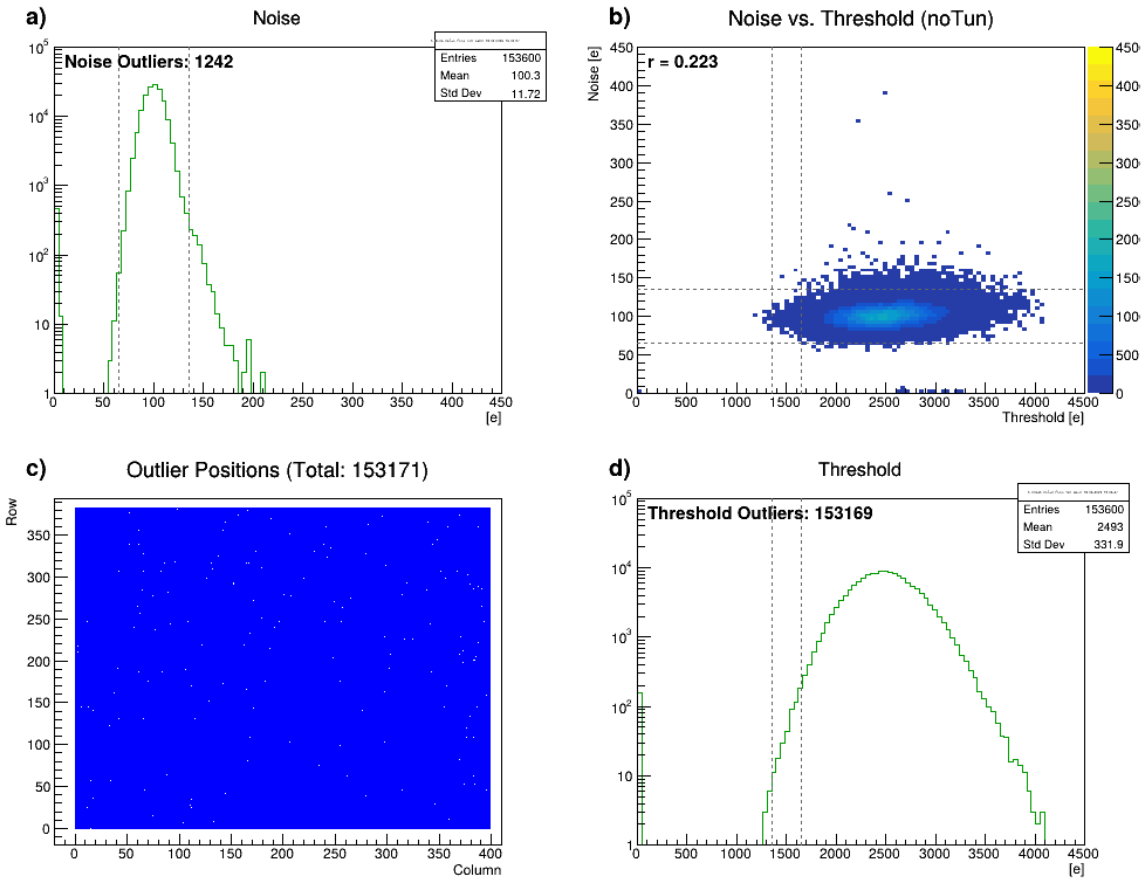

Figure 6.3: FE Chip 0x240bb, noTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

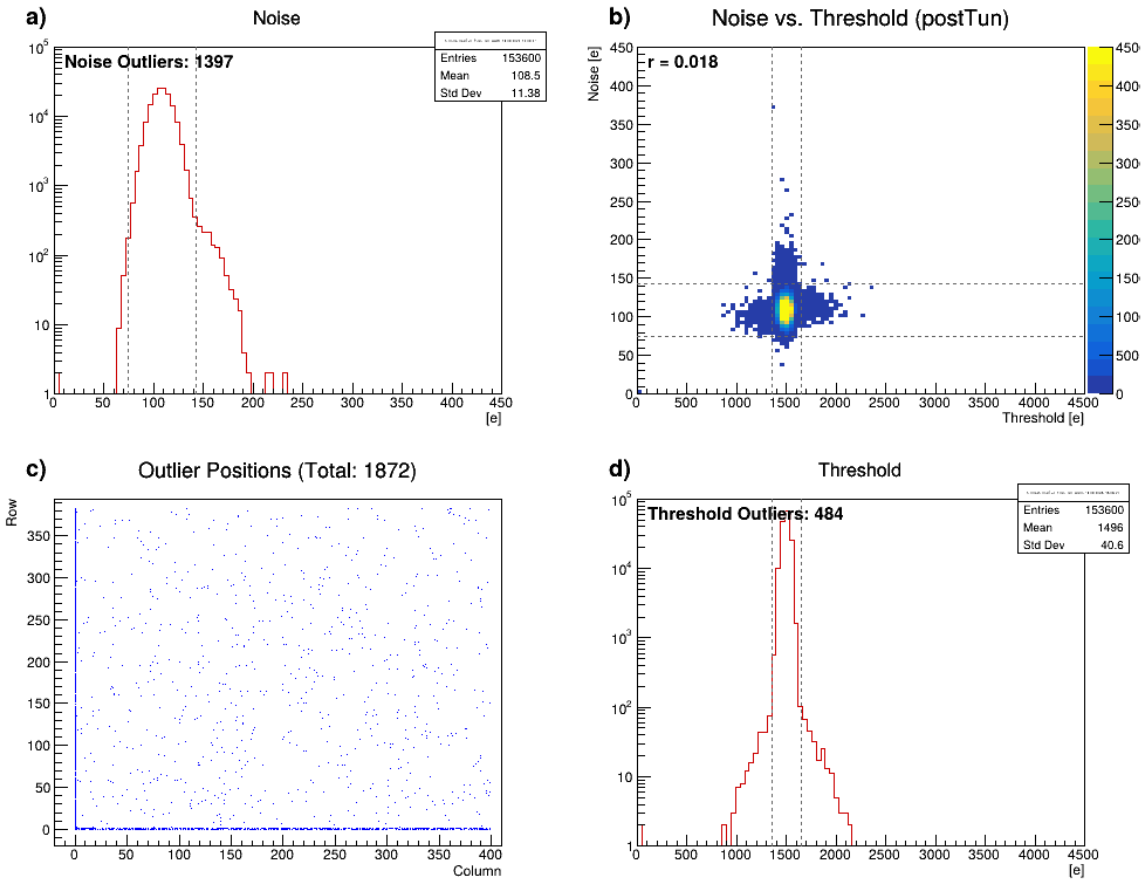

Figure 6.4: FE Chip 0x240bb, postTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

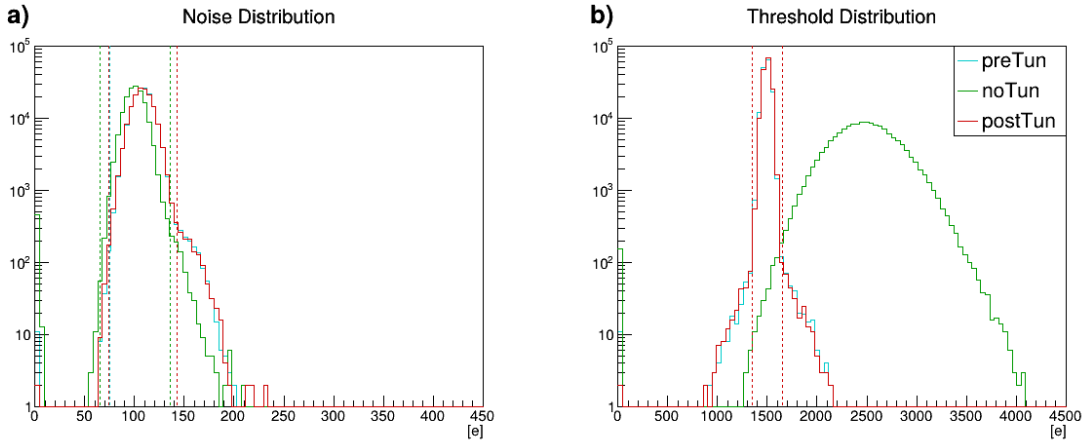

Figure 6.5: Noise and threshold distributions of FE Chip 0x240bb, all states overlayed.

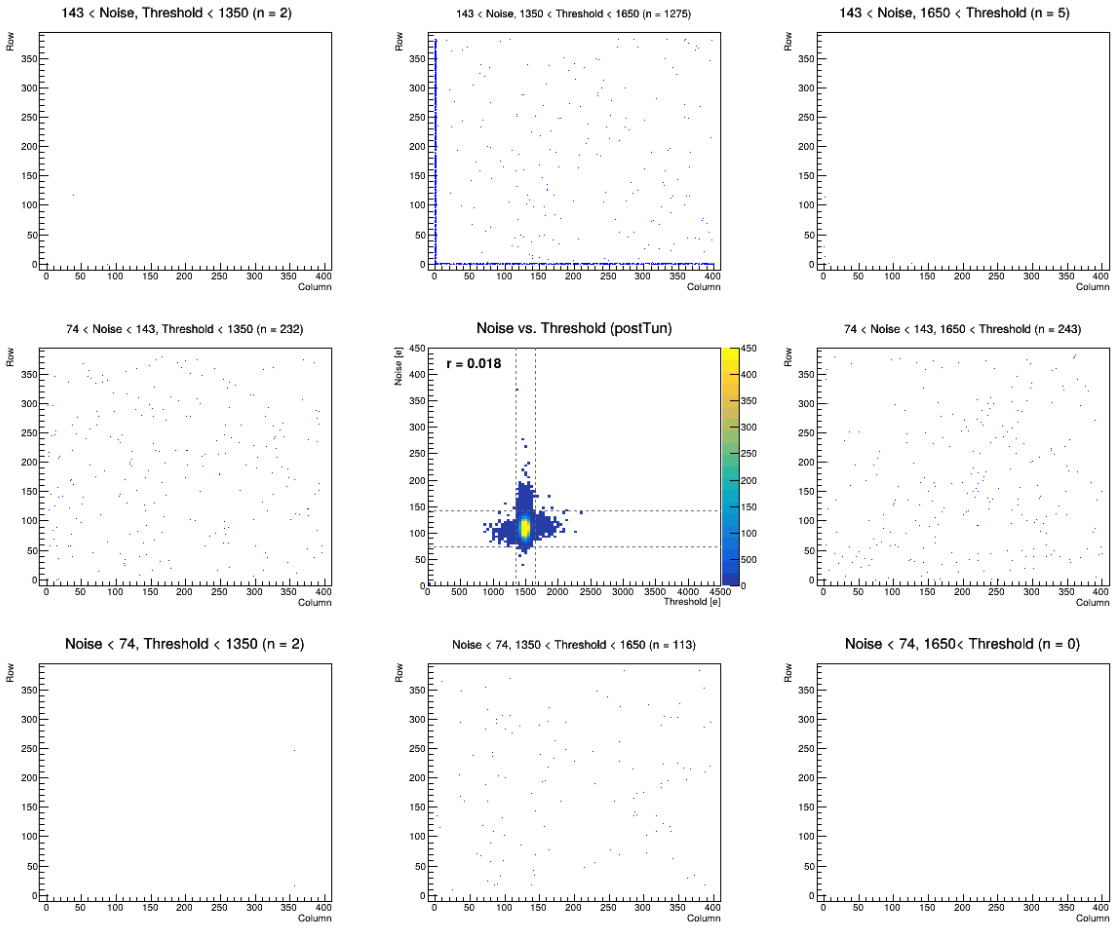

Figure 6.6: Location of outliers separated by classifying cuts, FE Chip 0x240bb, postTun (cuts in units of e).

CHAPTER 7

Summary and Outlook

This thesis addressed two aspects of the testing and analysis procedures associated with the Full Quality Control (Full QC) of ITk pixel modules at the University of Siegen. Chapter 5 focused on the investigation of a grounding-related issue in the multi-module testing setup, while Chapter 6 analysed noise and threshold behaviour based on threshold scan data.

Previous studies of the grounding configuration demonstrated that the default setup, in which the `DAQ_CMD GND` and `GND` contacts on the data adapter card are connected, can lead to conditions that are susceptible to the formation of ground loop effects when operating multiple modules simultaneously on a shared high-voltage channel. A modified configuration, in which these contacts are electrically separated by removing the corresponding jumpers, was implemented and evaluated.

The comparison of Full QC measurements between the original setup, with only one module on the test stand, and the modified setup, with all four slots occupied, showed no indication of degradation in measurement quality. The vast majority of analysed parameters remained consistent within the defined quality control criteria. Observed deviations in specific quantities, such as SLDO-related measurements and pixel failure rates, were found to be attributable to effects unrelated to the grounding modification, including module-specific behaviour and differences in measurement conditions. These results indicate that the removal of the jumpers constitutes a suitable and robust measure to mitigate potential ground loop effects without compromising the reliability of the electrical characterisation. The modified grounding scheme can therefore be considered a viable configuration for multi-module operation in the ITk pixel module testing environment at the University of Siegen.

A spatial analysis of noise values across the front-end chips showed a clear accumulation of higher noise levels at the edges of the chips. This behaviour is consistent with the increased effective pixel size in these regions, which leads to higher input capacitance and, consequently, increased electronic noise. The observed patterns confirm the initial hypothesis and demonstrate that geometric variations in the pixel layout have a measurable impact on noise performance. The combination of spatial mapping and outlier classification proved to be a useful tool for identifying regions with atypical behaviour and for diagnosing potential defects or systematic effects.

From a broader perspective, the results presented in this thesis contribute to the validation and optimisation of testing procedures for ITk pixel modules. The grounding study provides a practical improvement to the testing infrastructure, which are applied to enhance measurement stability in multi-module setups. The noise-threshold analysis, on the other hand, offers insight into the interplay between sensor geometry and electronic performance, which is relevant for both quality assurance and

the interpretation of detector behaviour.

A more detailed statistical analysis of noise and threshold behaviour with larger datasets comprising multiple modules could allow for a more comprehensive assessment of systematic trends and module-to-module variations.

APPENDIX A

Noise-Threshold Correlation Analysis Algorithm

The following algorithm used for the analysis of the noise and threshold data first finds the respective JSON files on the measurement PC for predefined test results and reads them into numpy arrays.

```
import json
import numpy as np
import ROOT
import os
from pathlib import Path
import glob

# selection of module, FE Chip, measurement and state
module = "Si030_20UPGM23211410"
frontendchip = "0x240bb"
measurements = ["FULL_QC_warm_18-06-2025_15-06-27"]
states = ["preTun", "noTun", "postTun"]

file_map = {
    "preTun":  f"{frontendchip}_ThresholdMap-0_0_500_0_0_0.json",
    "noTun":  f"{frontendchip}_ThresholdMap-0_0_500_0_0_0.json",
    "postTun": f"{frontendchip}_ThresholdMap-0_0_300_0_0_0.json"
}

noise_file_map = {
    "preTun":  f"{frontendchip}_NoiseMap-0_0_500_0_0_0.json",
    "noTun":  f"{frontendchip}_NoiseMap-0_0_500_0_0_0.json",
    "postTun": f"{frontendchip}_NoiseMap-0_0_300_0_0_0.json"
}

state_paths = {
    "preTun":  ("MHT", "_std_thresholdscan_hr"),
    "noTun":  ("TUN", "_std_thresholdscan_hr"),
    "postTun": ("TUN", "_std_thresholdscan_hd")
}

```

```
nbins = 100
xmin, xmax = 0, 4500
ymin, ymax = 0, 450
y_log_min = 1
y_log_max = 1e5
padding = 10

colors = {
    "preTun": ROOT.kCyan + 1,
    "noTun": ROOT.kGreen + 2,
    "postTun": ROOT.kRed + 1
}

base = Path("..").resolve()
project_root = base / module / "initial"

for measurement in measurements:
    h_noise_all = {}
    h_thresh_all = {}
    cut_lines = {}

    for state in states:
        output_dir =
            f"results/{module}/{frontendchip}/{measurement}"
        os.makedirs(output_dir, exist_ok=True)

        subfolder, suffix = state_paths[state]
        state_dir = project_root / measurement / subfolder
        matching_dirs = list(state_dir.glob(f"*{suffix}"))

        if len(matching_dirs) == 1:
            target_dir = matching_dirs[0]
            print(f"{measurement}, {state}, Found folder: {target_dir}")
        else:
            raise RuntimeError(f"{measurement}, {state}: Exactly one
    ↪ matching folder expected, found: {len(matching_dirs)}, {
    ↪ matching_dirs}")

        with open(target_dir / file_map[state], "r") as f_thresh:
            threshold_data = json.load(f_thresh)
        with open(target_dir / noise_file_map[state], "r") as f_noise:
            noise_data = json.load(f_noise)

        threshold_matrix = np.array(threshold_data["Data"]).T
        noise_matrix = np.array(noise_data["Data"]).T

        threshold_flat = threshold_matrix.flatten()
        noise_flat = noise_matrix.flatten()
```

Continuing inside the loop over the states outliers are classified using the cuts described in 6.2. Every pixel outside of that range contributes to the outlier mask.

```
        n_mean, n_std = np.mean(noise_flat), np.std(noise_flat)

        n_low, n_high = n_mean - 3 * n_std, n_mean + 3 * n_std
        t_low, t_high = 1350, 1650
        cut_lines[state] = (t_low, t_high, n_low, n_high)

        outlier_mask_thresh = (threshold_flat < t_low) | (threshold_flat
    ↪ > t_high)
        outlier_mask_noise = (noise_flat < n_low) | (noise_flat > n_high)
        outlier_mask = outlier_mask_thresh | outlier_mask_noise
        count_outliers_thresh = np.sum(outlier_mask_thresh)
        count_outliers_noise = np.sum(outlier_mask_noise)
        count_outliers_total = np.sum(outlier_mask)
```

The noise and threshold distribution each are placed in a 1D plot with their respective cut lines. The noise data is also plotted against the threshold data in a 2D correlation plot with cut lines from both. Then all outliers are mapped to their true positions on the FE Chip.

```
        corr_coeff = np.corrcoef(threshold_flat, noise_flat)[0, 1]

        corr_hist = ROOT.TH2F(f"corr_{state}", f"Noise vs. Threshold ({
    ↪ state})", nbins, xmin, xmax, nbins, ymin, ymax)
        for t, n in zip(threshold_flat, noise_flat):
            corr_hist.Fill(t, n)
        corr_hist.SetMaximum(4500)
        corr_hist.SetMinimum(0)

        n_rows, n_cols = threshold_matrix.shape
        g_outliers = ROOT.TGraph()
        for idx, is_outlier in enumerate(outlier_mask):
            if is_outlier:
                y = idx // n_cols
                x = idx % n_cols
                g_outliers.SetPoint(g_outliers.GetN(), x, y)

        h_thresh = ROOT.TH1F(f"h_thresh_{state}_{measurement}", "
    ↪ Threshold", nbins, xmin, xmax)
        h_noise = ROOT.TH1F(f"h_noise_{state}_{measurement}", "Noise",
    ↪ nbins, ymin, ymax)
        for t in threshold_flat:
            h_thresh.Fill(t)
        for n in noise_flat:
            h_noise.Fill(n)

        h_thresh.SetLineColor(colors[state])
```

```
        h_noise.SetLineColor(colors[state])

        h_thresh_all[state] = h_thresh
        h_noise_all[state] = h_noise

        canvas = ROOT.TCanvas(f"canvas_{state}_{measurement}", f"
    ↪ Threshold vs. Noise ({state})", 1000, 800)
        canvas.Divide(2, 2)

        canvas.cd(1)
        ROOT.gPad.SetLogy()
        h_noise.SetMinimum(y_log_min)
        h_noise.SetMaximum(y_log_max)
        h_noise.GetXaxis().SetTitle("[e]")
        h_noise.Draw()
        line_n_low = ROOT.TLine(n_low, y_log_min, n_low, y_log_max)
        line_n_low.SetLineColor(ROOT.kGray + 2)
        line_n_low.SetLineStyle(2)
        line_n_low.Draw()
        line_n_high = ROOT.TLine(n_high, y_log_min, n_high, y_log_max)
        line_n_high.SetLineColor(ROOT.kGray + 2)
        line_n_high.SetLineStyle(2)
        line_n_high.Draw()
        text = ROOT.TLatex()
        text.SetTextSize(0.045)
        text.DrawLatexNDC(0.12, 0.85, f"Noise Outliers: {
    ↪ count_outliers_noise}")
        label = ROOT.TLatex()
        label.SetNDC()
        label.SetTextSize(0.06)
        label.DrawLatex(0.05, 0.95, "a)")

        canvas.cd(2)
        corr_hist.GetXaxis().SetTitle("Threshold [e]")
        corr_hist.GetYaxis().SetTitle("Noise [e]")
        corr_hist.SetStats(False)
        corr_hist.Draw("COLZ")
        vline_t_low = ROOT.TLine(t_low, ymin, t_low, ymax)
        vline_t_low.SetLineColor(ROOT.kGray + 2)
        vline_t_low.SetLineStyle(2)
        vline_t_low.Draw()
        vline_t_high = ROOT.TLine(t_high, ymin, t_high, ymax)
        vline_t_high.SetLineColor(ROOT.kGray + 2)
        vline_t_high.SetLineStyle(2)
        vline_t_high.Draw()
        hline_n_low = ROOT.TLine(xmin, n_low, xmax, n_low)
        hline_n_low.SetLineColor(ROOT.kGray + 2)
        hline_n_low.SetLineStyle(2)
        hline_n_low.Draw()
        hline_n_high = ROOT.TLine(xmin, n_high, xmax, n_high)
```

```
        hline_n_high.SetLineColor(ROOT.kGray + 2)
        hline_n_high.SetLineStyle(2)
        hline_n_high.Draw()
        text_corr = ROOT.TLatex()
        text_corr.SetTextSize(0.045)
        text_corr.DrawLatexNDC(0.12, 0.85, f"r = {corr_coeff:.3f}")
        label = ROOT.TLatex()
        label.SetNDC()
        label.SetTextSize(0.06)
        label.DrawLatex(0.05, 0.95, "b)")

        canvas.cd(3)
        frame = ROOT.TH2F("frame", f"Outlier Positions (Total: {
    ↪ count_outliers_total})", n_cols + 2 * padding, -2 * padding,
    ↪ n_cols + padding, n_rows + 2 * padding, -2 * padding, n_rows +
    ↪ padding)
        frame.GetXaxis().SetTitle("Column")
        frame.GetYaxis().SetTitle("Row")
        frame.SetStats(False)
        frame.Draw()
        g_outliers.SetMarkerStyle(1)
        g_outliers.SetMarkerColor(ROOT.kBlue)
        g_outliers.Draw("P SAME")
        label = ROOT.TLatex()
        label.SetNDC()
        label.SetTextSize(0.06)
        label.DrawLatex(0.05, 0.95, "c)")

        canvas.cd(4)
        ROOT.gPad.SetLogy()
        h_thresh.SetMinimum(y_log_min)
        h_thresh.SetMaximum(y_log_max)
        h_thresh.GetXaxis().SetTitle("[e]")
        h_thresh.Draw()
        line_t_low = ROOT.TLine(t_low, y_log_min, t_low, y_log_max)
        line_t_low.SetLineColor(ROOT.kGray + 2)
        line_t_low.SetLineStyle(2)
        line_t_low.Draw()
        line_t_high = ROOT.TLine(t_high, y_log_min, t_high, y_log_max)
        line_t_high.SetLineColor(ROOT.kGray + 2)
        line_t_high.SetLineStyle(2)
        line_t_high.Draw()
        text2 = ROOT.TLatex()
        text2.SetTextSize(0.045)
        text2.DrawLatexNDC(0.12, 0.85, f"Threshold Outliers: {
    ↪ count_outliers_thresh}")
        label = ROOT.TLatex()
        label.SetNDC()
        label.SetTextSize(0.06)
        label.DrawLatex(0.05, 0.95, "d)")
```

```
208
209         canvas.SaveAs(f"{output_dir}/noise_vs_threshold_{state}.png")
```

Determined by the classifying cuts the outliers get distributed over eight plot regions, each being a map of the selected FE Chip.

```
210         # --- Outlier Classification 3x3 Grid ---
211
212         t_low_r = round(t_low)
213         t_high_r = round(t_high)
214         n_low_r = round(n_low)
215         n_high_r = round(n_high)
216
217         canvas_outliers = ROOT.TCanvas(f"c_outliers_{state}_{measurement}
    ↪ ", f"Outlier Types ({state})", 1200, 1000)
218         canvas_outliers.Divide(3, 3)
219
220         # Initialize 8 Graphs + Counter
221         g_00 = ROOT.TGraph(); c_00 = 0
222         g_01 = ROOT.TGraph(); c_01 = 0
223         g_02 = ROOT.TGraph(); c_02 = 0
224         g_10 = ROOT.TGraph(); c_10 = 0
225         g_12 = ROOT.TGraph(); c_12 = 0
226         g_20 = ROOT.TGraph(); c_20 = 0
227         g_21 = ROOT.TGraph(); c_21 = 0
228         g_22 = ROOT.TGraph(); c_22 = 0
229
230         # Fill Graphs
231         for idx, (t, n) in enumerate(zip(threshold_flat, noise_flat)):
232             if not ((t_low <= t <= t_high) and (n_low <= n <= n_high)):
233                 x = idx % n_cols
234                 y = idx // n_cols
235                 if n > n_high and t < t_low:
236                     g_00.SetPoint(g_00.GetN(), x, y); c_00 += 1
237                 elif n > n_high and t_low <= t <= t_high:
238                     g_01.SetPoint(g_01.GetN(), x, y); c_01 += 1
239                 elif n > n_high and t > t_high:
240                     g_02.SetPoint(g_02.GetN(), x, y); c_02 += 1
241                 elif n_low <= n <= n_high and t < t_low:
242                     g_10.SetPoint(g_10.GetN(), x, y); c_10 += 1
243                 elif n_low <= n <= n_high and t > t_high:
244                     g_12.SetPoint(g_12.GetN(), x, y); c_12 += 1
245                 elif n < n_low and t < t_low:
246                     g_20.SetPoint(g_20.GetN(), x, y); c_20 += 1
247                 elif n < n_low and t_low <= t <= t_high:
248                     g_21.SetPoint(g_21.GetN(), x, y); c_21 += 1
249                 elif n < n_low and t > t_high:
250                     g_22.SetPoint(g_22.GetN(), x, y); c_22 += 1
251
```

```
252         titles = [
253             f"{n_high_r} < Noise, Threshold < {t_low_r} (n = {c_00})",
254             f"{n_high_r} < Noise, {t_low_r} < Threshold < {t_high_r} (n =
    ↪  {c_01})",
255             f"{n_high_r} < Noise, {t_high_r} < Threshold (n = {c_02})",
256             f"{n_low_r} < Noise < {n_high_r}, Threshold < {t_low_r} (n =
    ↪ {c_10})",
257             f"{n_low_r} < Noise < {n_high_r}, {t_high_r} < Threshold (n =
    ↪  {c_12})",
258             f"Noise < {n_low_r}, Threshold < {t_low_r} (n = {c_20})",
259             f"Noise < {n_low_r}, {t_low_r} < Threshold < {t_high_r} (n =
    ↪ {c_21})",
260             f"Noise < {n_low_r}, {t_high_r}< Threshold (n = {c_22})"
261         ]
262
263         f_00 = ROOT.TH2F("f_00", titles[0], n_cols + 2 * padding, -
    ↪ padding, n_cols + padding, n_rows + 2 * padding, -padding, n_rows
    ↪ + padding)
264         f_00.GetXaxis().SetTitle("Column")
265         f_00.GetYaxis().SetTitle("Row")
266         f_00.SetStats(False)
267
268         f_01 = ROOT.TH2F("f_01", titles[1], n_cols + 2 * padding, -
    ↪ padding, n_cols + padding, n_rows + 2 * padding, -padding, n_rows
    ↪ + padding)
269         f_01.GetXaxis().SetTitle("Column")
270         f_01.GetYaxis().SetTitle("Row")
271         f_01.SetStats(False)
272
273         f_02 = ROOT.TH2F("f_02", titles[2], n_cols + 2 * padding, -
    ↪ padding, n_cols + padding, n_rows + 2 * padding, -padding, n_rows
    ↪ + padding)
274         f_02.GetXaxis().SetTitle("Column")
275         f_02.GetYaxis().SetTitle("Row")
276         f_02.SetStats(False)
277
278         f_10 = ROOT.TH2F("f_10", titles[3], n_cols + 2 * padding, -
    ↪ padding, n_cols + padding, n_rows + 2 * padding, -padding, n_rows
    ↪ + padding)
279         f_10.GetXaxis().SetTitle("Column")
280         f_10.GetYaxis().SetTitle("Row")
281         f_10.SetStats(False)
282
283         f_12 = ROOT.TH2F("f_12", titles[4], n_cols + 2 * padding, -
    ↪ padding, n_cols + padding, n_rows + 2 * padding, -padding, n_rows
    ↪ + padding)
284         f_12.GetXaxis().SetTitle("Column")
285         f_12.GetYaxis().SetTitle("Row")
286         f_12.SetStats(False)
287
```

```
288         f_20 = ROOT.TH2F("f_20", titles[5], n_cols + 2 * padding, -
    ↪ padding, n_cols + padding, n_rows + 2 * padding, -padding, n_rows
    ↪ + padding)
289         f_20.GetXaxis().SetTitle("Column")
290         f_20.GetYaxis().SetTitle("Row")
291         f_20.SetStats(False)
292 
293         f_21 = ROOT.TH2F("f_21", titles[6], n_cols + 2 * padding, -
    ↪ padding, n_cols + padding, n_rows + 2 * padding, -padding, n_rows
    ↪ + padding)
294         f_21.GetXaxis().SetTitle("Column")
295         f_21.GetYaxis().SetTitle("Row")
296         f_21.SetStats(False)
297 
298         f_22 = ROOT.TH2F("f_22", titles[7], n_cols + 2 * padding, -
    ↪ padding, n_cols + padding, n_rows + 2 * padding, -padding, n_rows
    ↪ + padding)
299         f_22.GetXaxis().SetTitle("Column")
300         f_22.GetYaxis().SetTitle("Row")
301         f_22.SetStats(False)
302 
303         collection = [
304             (1, f_00, g_00),
305             (2, f_01, g_01),
306             (3, f_02, g_02),
307             (4, f_10, g_10),
308             (6, f_12, g_12),
309             (7, f_20, g_20),
310             (8, f_21, g_21),
311             (9, f_22, g_22)
312         ]
313 
314         for pad, frame, graph in collection:
315             canvas_outliers.cd(pad)
316             frame.Draw()
317             graph.SetMarkerStyle(1)
318             graph.SetMarkerColor(ROOT.kBlue)
319             graph.Draw("P")
320 
321         canvas_outliers.cd(5)
322         corr_hist.GetXaxis().SetTitle("Threshold [e]")
323         corr_hist.GetYaxis().SetTitle("Noise [e]")
324         corr_hist.SetStats(False)
325         corr_hist.Draw("COLZ")
326         vline_t_low.Draw()
327         vline_t_high.Draw()
328         hline_n_low.Draw()
329         hline_n_high.Draw()
330         text_corr = ROOT.TLatex()
331         text_corr.SetTextSize(0.05)
```

```
        text_corr.DrawLatexNDC(0.15, 0.83, f"r = {np.corrcoef(
    ↪ threshold_flat, noise_flat)[0,1]:.3f}")

        canvas_outliers.SaveAs(f"{output_dir}/outlier_types_{state}.png")
```

Now leaving the loop over the states but continuing inside the loop over the measurements, the noise and threshold distributions of all states and their respective cuts are plotted.

```
    # Collective Plot for all Marginals
    legend = ROOT.TLegend(0.7, 0.7, 0.9, 0.9)
    c_marginals = ROOT.TCanvas(f"c_marginals_{measurement}", f"Marginals
    ↪ {measurement}", 1200, 500)
    c_marginals.Divide(2, 1)
    lines_noise = []
    lines_thresh = []

    c_marginals.cd(1)
    ROOT.gPad.SetLogy()
    for i, state in enumerate(states):
        h = h_noise_all[state]
        h.SetStats(False)
        h.SetTitle("Noise Distribution")
        h.SetMinimum(y_log_min)
        h.SetMaximum(y_log_max)
        h.GetXaxis().SetTitle("[e]")
        if i == 0:
            h.Draw()
        else:
            h.Draw("SAME")
        for val in [cut_lines[state][2], cut_lines[state][3]]:
            line = ROOT.TLine(val, y_log_min, val, y_log_max)
            line.SetLineColor(colors[state])
            line.SetLineStyle(2)
            line.Draw()
            lines_noise.append(line)  # Save Reference
        legend.AddEntry(h, state, "l")
    label = ROOT.TLatex()
    label.SetNDC()
    label.SetTextSize(0.06)
    label.DrawLatex(0.05, 0.95, "a)")

    c_marginals.cd(2)
    ROOT.gPad.SetLogy()
    for i, state in enumerate(states):
        h = h_thresh_all[state]
        h.SetStats(False)
        h.SetTitle("Threshold Distribution")
        h.SetMinimum(y_log_min)
        h.SetMaximum(y_log_max)
```

```
        h.GetXaxis().SetTitle("[e]")
        if i == 0:
            h.Draw()
        else:
            h.Draw("SAME")
        for val in [cut_lines[state][0], cut_lines[state][1]]:
            line = ROOT.TLine(val, y_log_min, val, y_log_max)
            line.SetLineColor(colors[state])
            line.SetLineStyle(2)
            line.Draw()
            lines_thresh.append(line)  # Save Reference
    legend.Draw()
    label = ROOT.TLatex()
    label.SetNDC()
    label.SetTextSize(0.06)
    label.DrawLatex(0.05, 0.95, "b)")

    c_marginals.SaveAs(f"{output_dir}/combined_marginals.png")
```

APPENDIX B

Other Noise-Threshold Correlation Plots

B.1 FE Chip 0x240bb

The characteristics of the noise and threshold distributions of FE chip 0x240bb were listed in Tab. 6.1. Fig. 6.1, Fig. 6.3 and Fig. 6.4 display these distributions, while Fig. B.1, Fig. B.2 and Fig. 6.6 show the outliers of each region. In Fig. 6.5, the noise and threshold distributions of all states are overlaid.

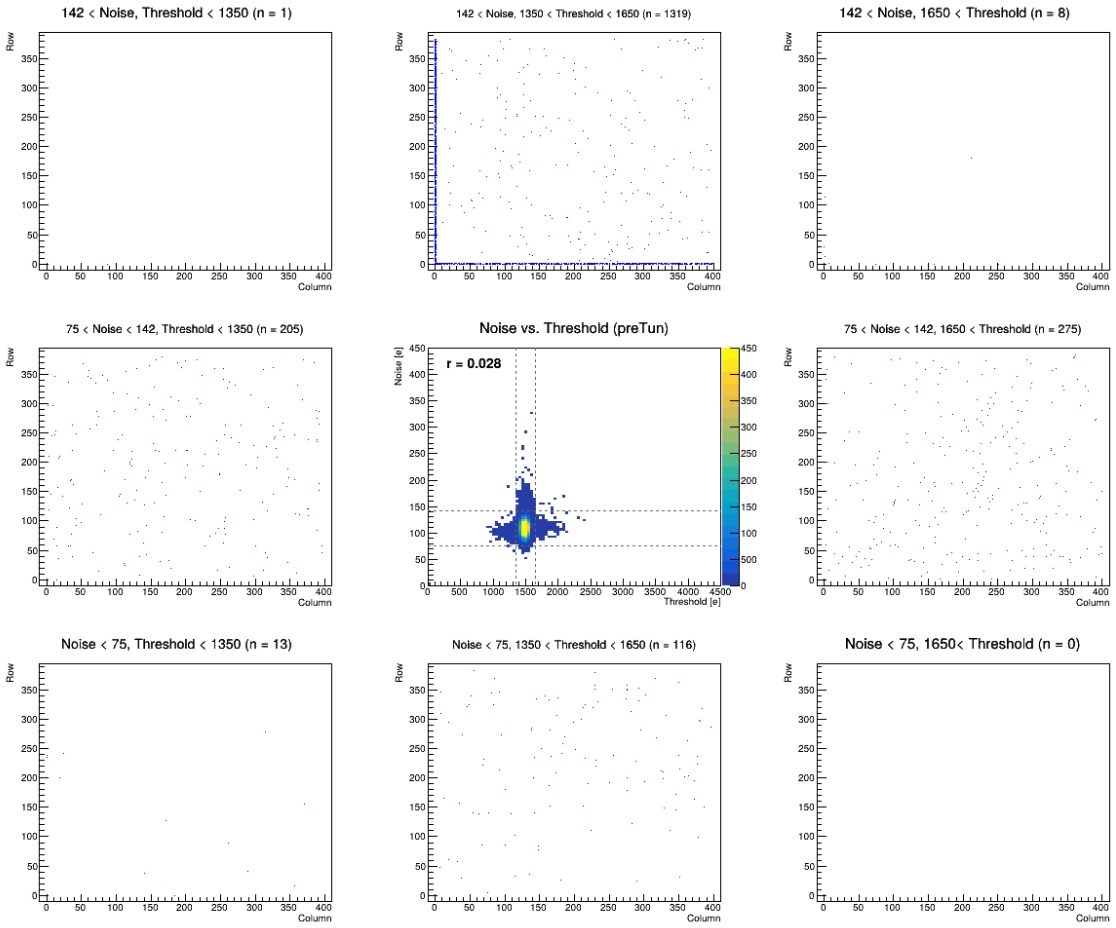

Figure B.1: Location of outliers separated by classifying cuts, FE Chip 0x240bb, preTun (cuts in units of e).

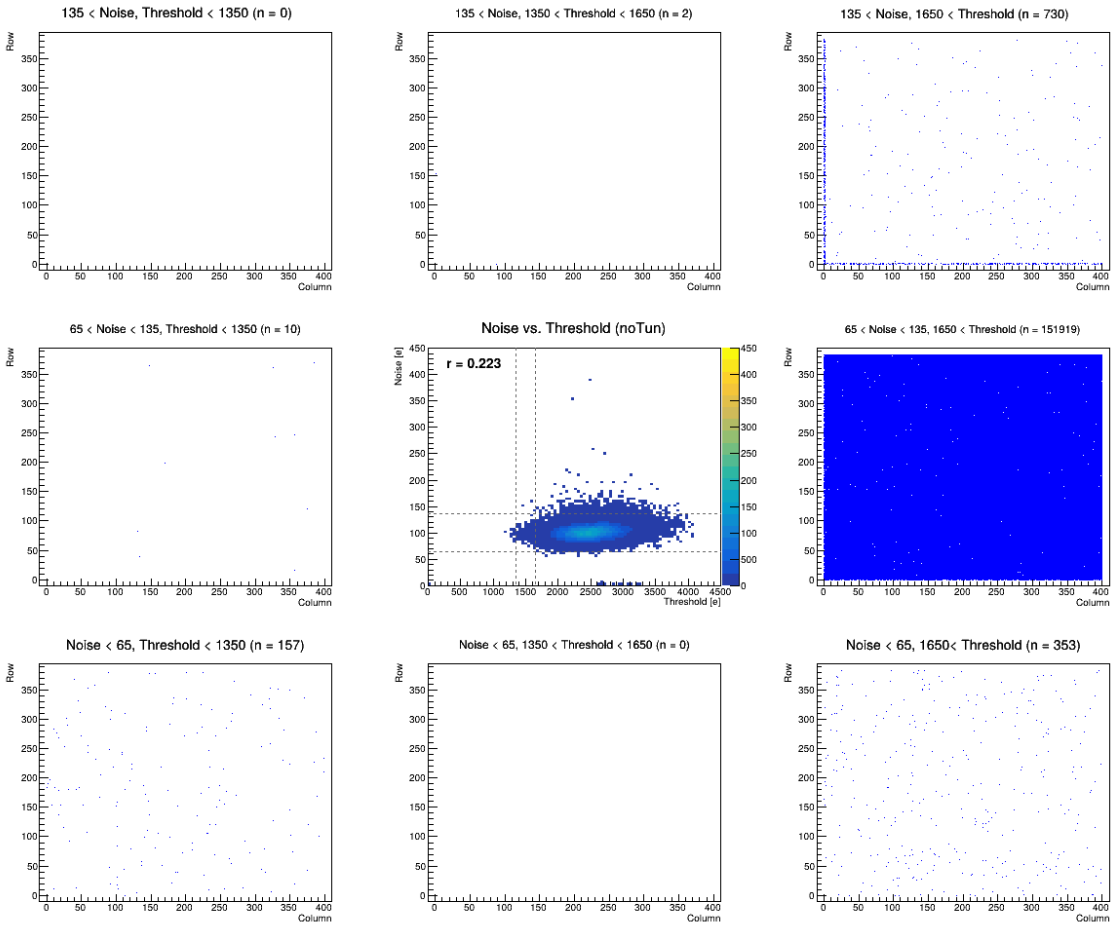

Figure B.2: Location of outliers separated by classifying cuts, FE Chip 0x240bb, noTun (cuts in units of e).

B.2 FE Chip 0x2403a

The characteristics of the noise and threshold distributions of FE chip 0x2403a are listed in Tab. B.1. Fig. B.3, Fig. B.4 and Fig. B.5 display these distributions, while Fig. B.6, Fig. B.7 and Fig. B.8 show the outliers of each region. In Fig. B.9, the noise and threshold distributions of all states are overlaid.

state		mean [e]	std. dev. [e]	lower cut [e]	upper cut [e]	outliers [count]	tot. outliers [count]
preTun	noise	108.3	11.54	73.7	142.9	1503	1747
	threshold	1 497.0	40.98	1 350.0	1 650.0	265	
noTun	noise	99.6	11.02	66.6	132.7	1272	149374
	threshold	2 222.1	320.97	1 350.0	1 650.0	159362	
postTun	noise	108.4	11.66	73.4	143.3	1480	1699
	threshold	1 496.9	41.06	1 350.0	1 650.0	241	

Table B.1: This table shows the characteristics of the noise and threshold distributions of FE chip 0x2403a in each state.

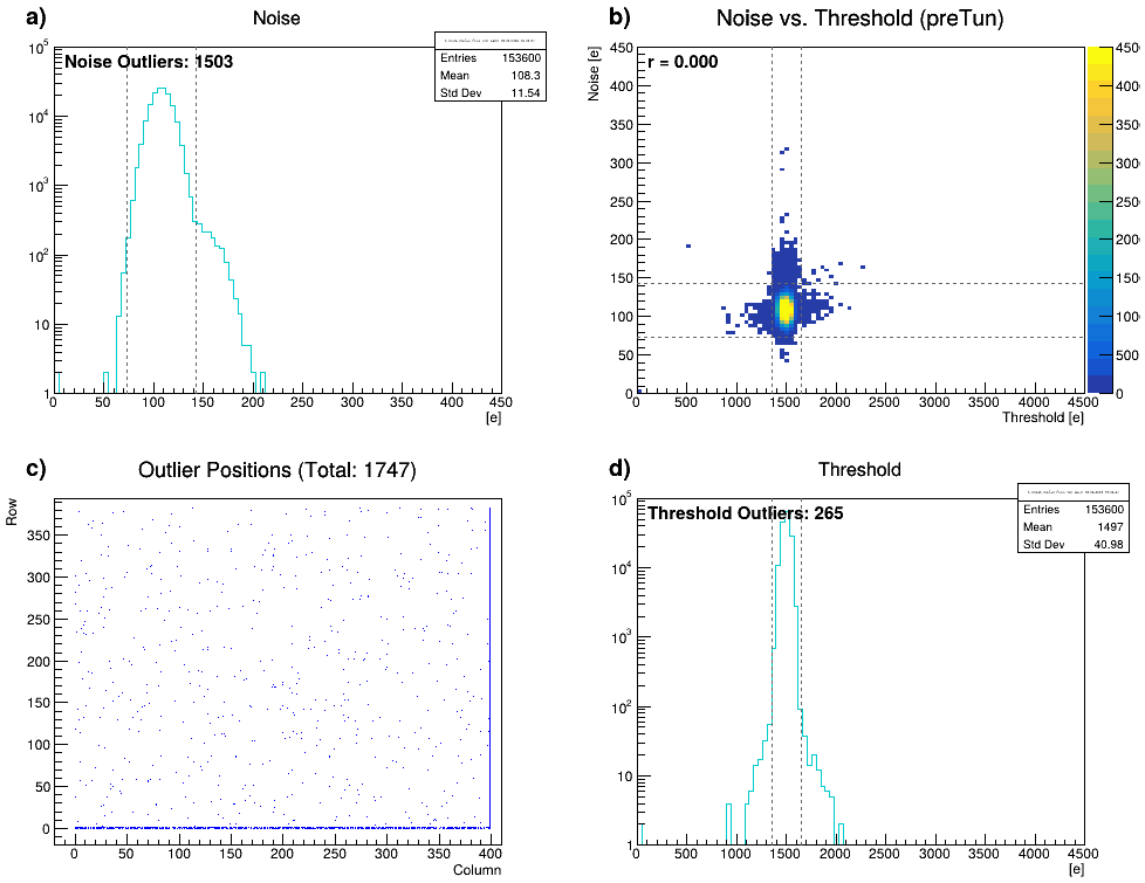

Figure B.3: FE Chip 0x2403a, preTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

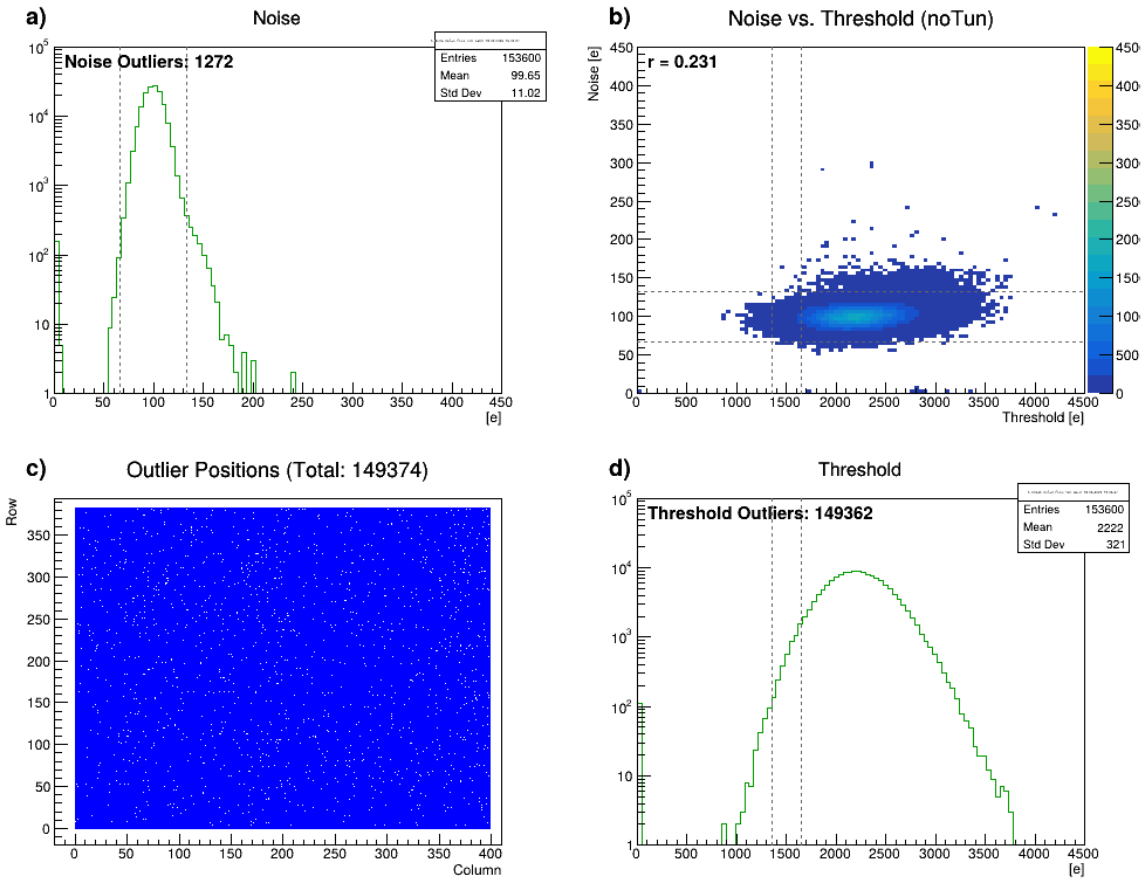

Figure B.4: FE Chip 0x2403a, noTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

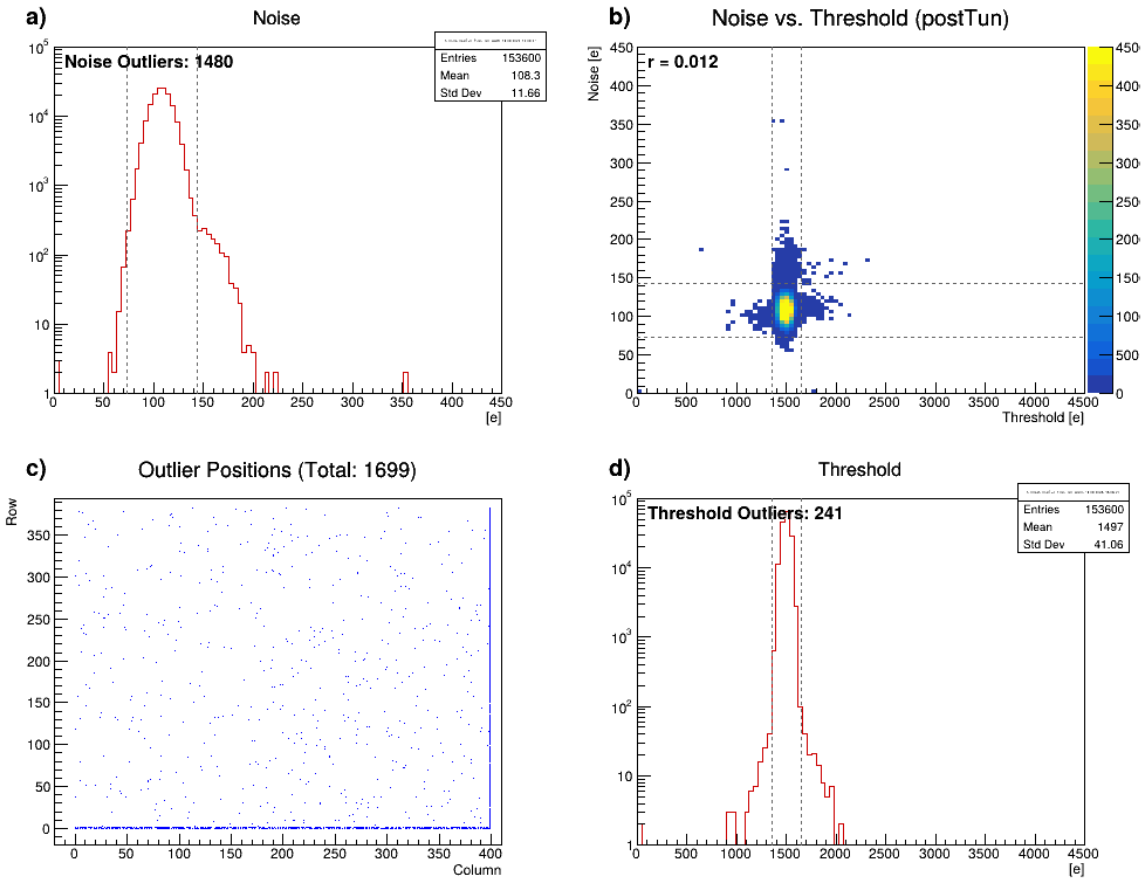

Figure B.5: FE Chip 0x2403a, postTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

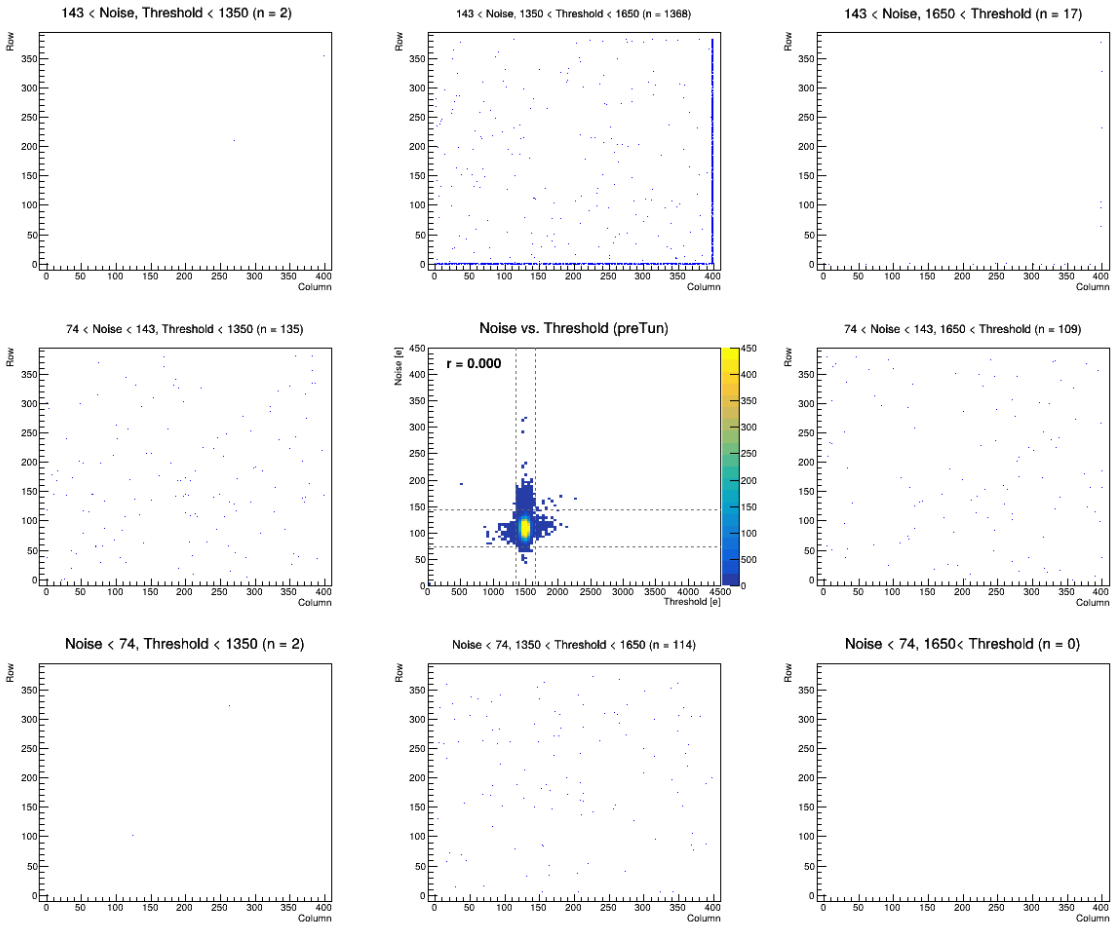

Figure B.6: Location of outliers separated by classifying cuts, FE Chip 0x2403a, preTun (cuts in units of e).

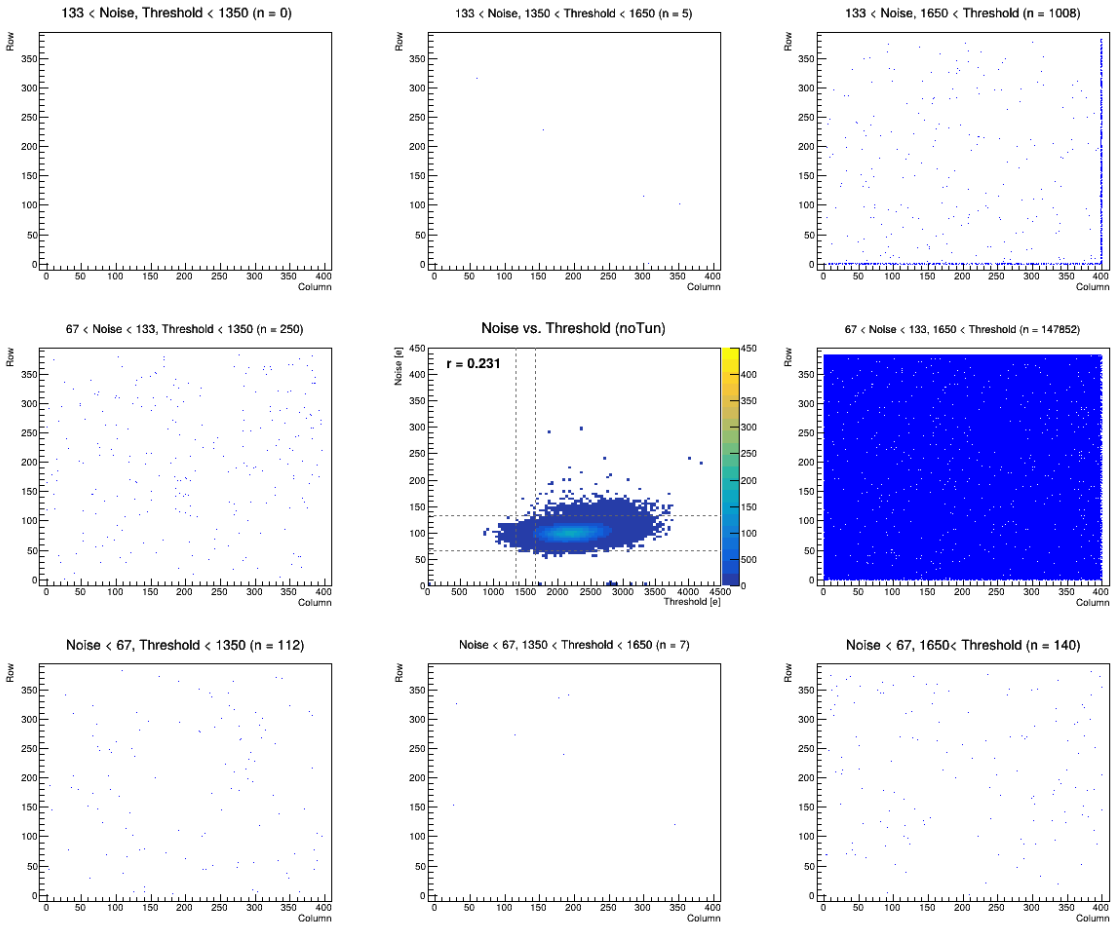

Figure B.7: Location of outliers separated by classifying cuts, FE Chip 0x2403a, noTun (cuts in units of e).

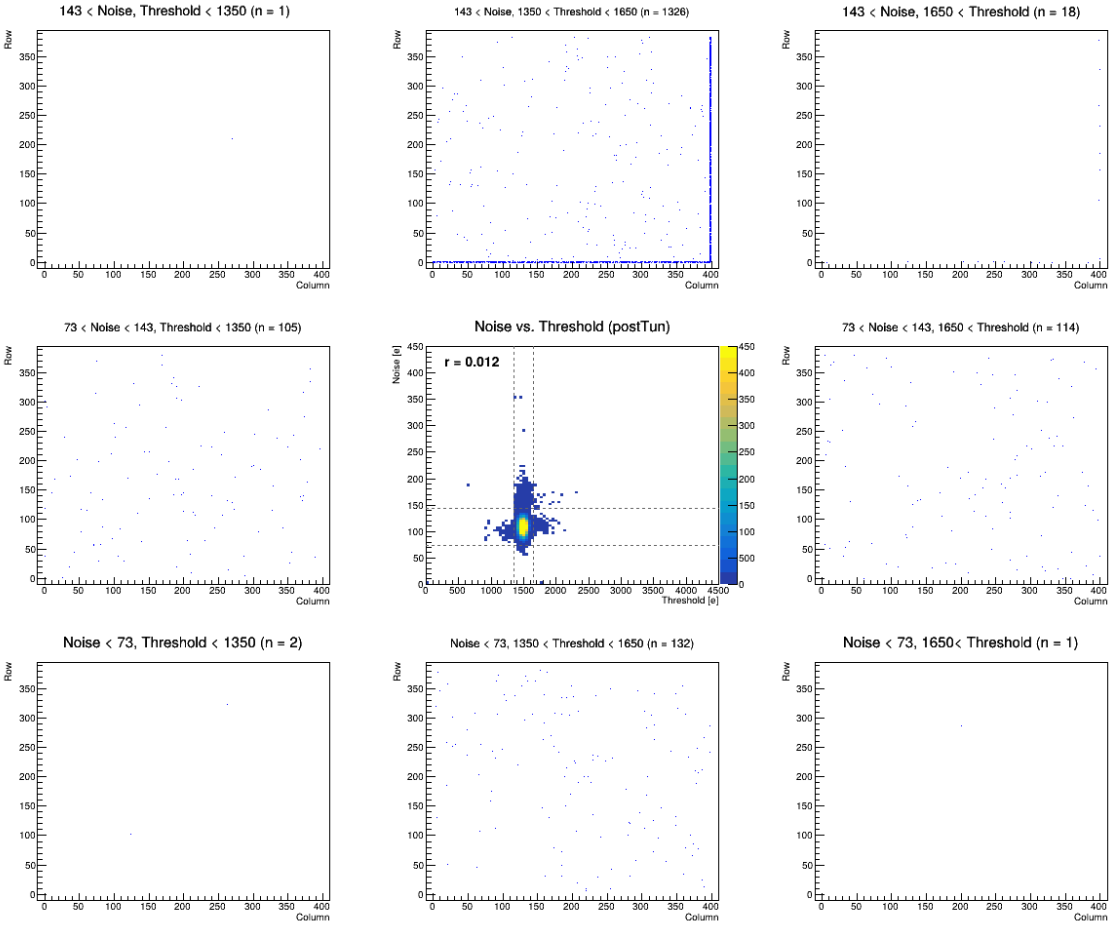

Figure B.8: Location of outliers separated by classifying cuts, FE Chip 0x2403a, postTun (cuts in units of e).

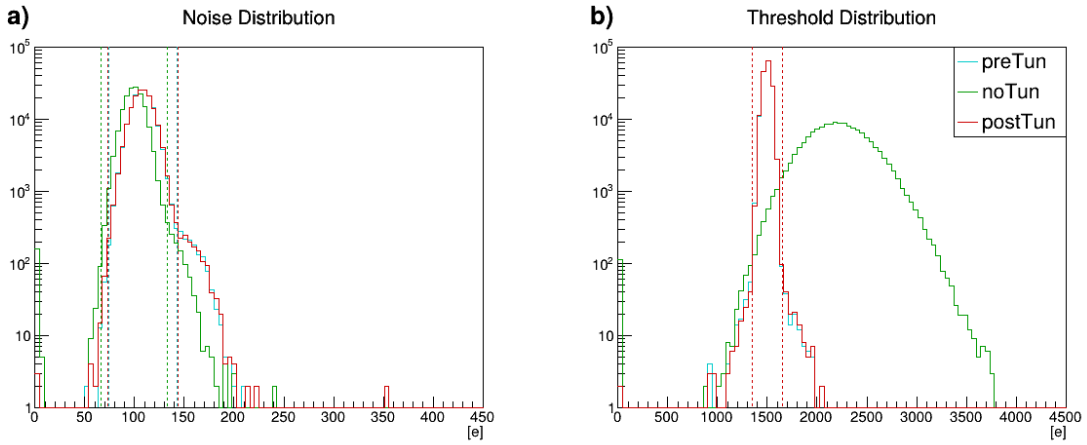

Figure B.9: Noise and threshold distributions of FE Chip 0x2403a, all states overlayed.

B.3 FE Chip 0x24162

The characteristics of the noise and threshold distributions of FE chip 0x24162 are listed in Tab. B.2. Fig. B.10, Fig. B.11 and Fig. B.12 display these distributions, while Fig. B.13, Fig. B.14 and Fig. B.15 show the outliers of each region. In Fig. B.16, the noise and threshold distributions of all states are overlaid.

state		mean [e]	std. dev. [e]	lower cut [e]	upper cut [e]	outliers [count]	tot. outliers [count]
preTun	noise	112.6	11.66	77.6	147.6	1444	1725
	threshold	1 497.6	42.93	1 350.0	1 650.0	304	
noTun	noise	103.6	11.14	70.2	137.0	1248	149695
	threshold	2 238.5	319.43	1 350.0	1 650.0	149674	
postTun	noise	112.6	11.69	77.5	147.6	1475	1724
	threshold	1 499.4	41.62	1 350.0	1 650.0	267	

Table B.2: This table shows the characteristics of the noise and threshold distributions of FE chip 0x24162 in each state.

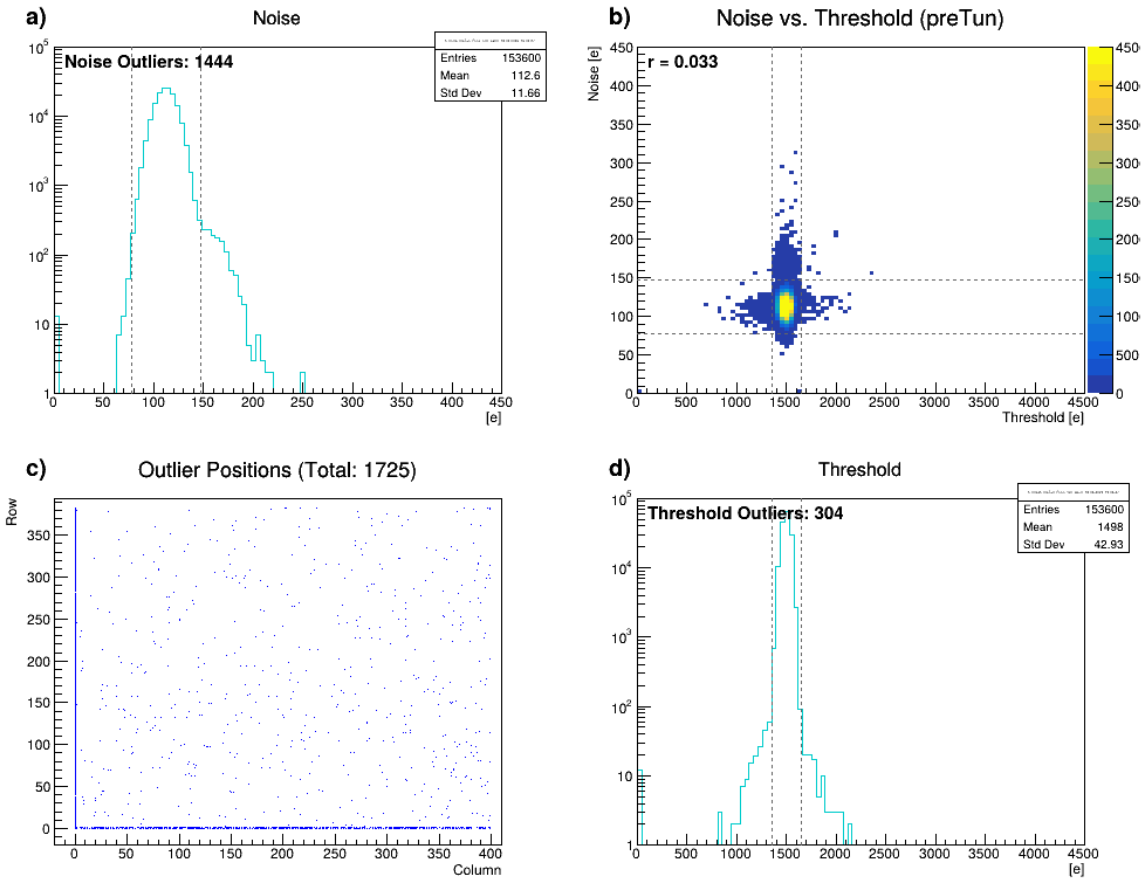

Figure B.10: FE Chip 0x24162, preTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

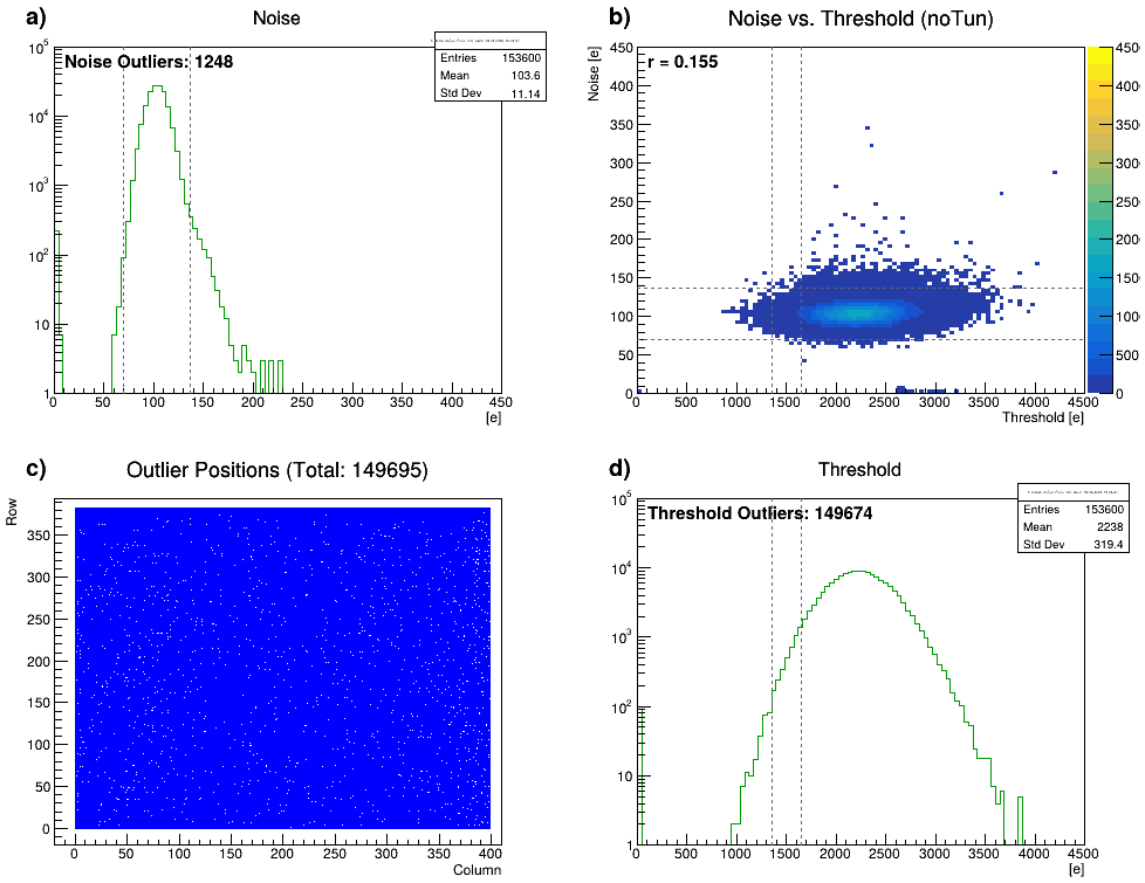

Figure B.11: FE Chip 0x24162, noTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

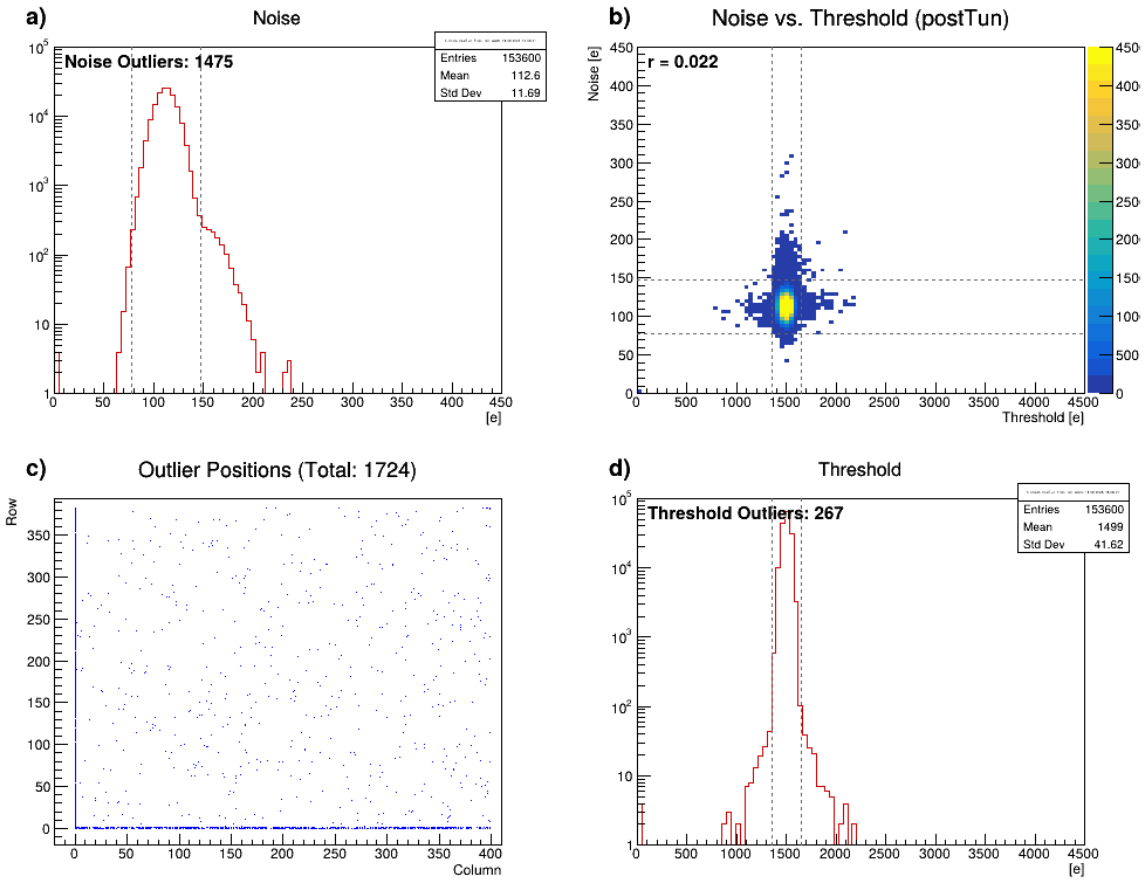

Figure B.12: FE Chip 0x24162, postTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

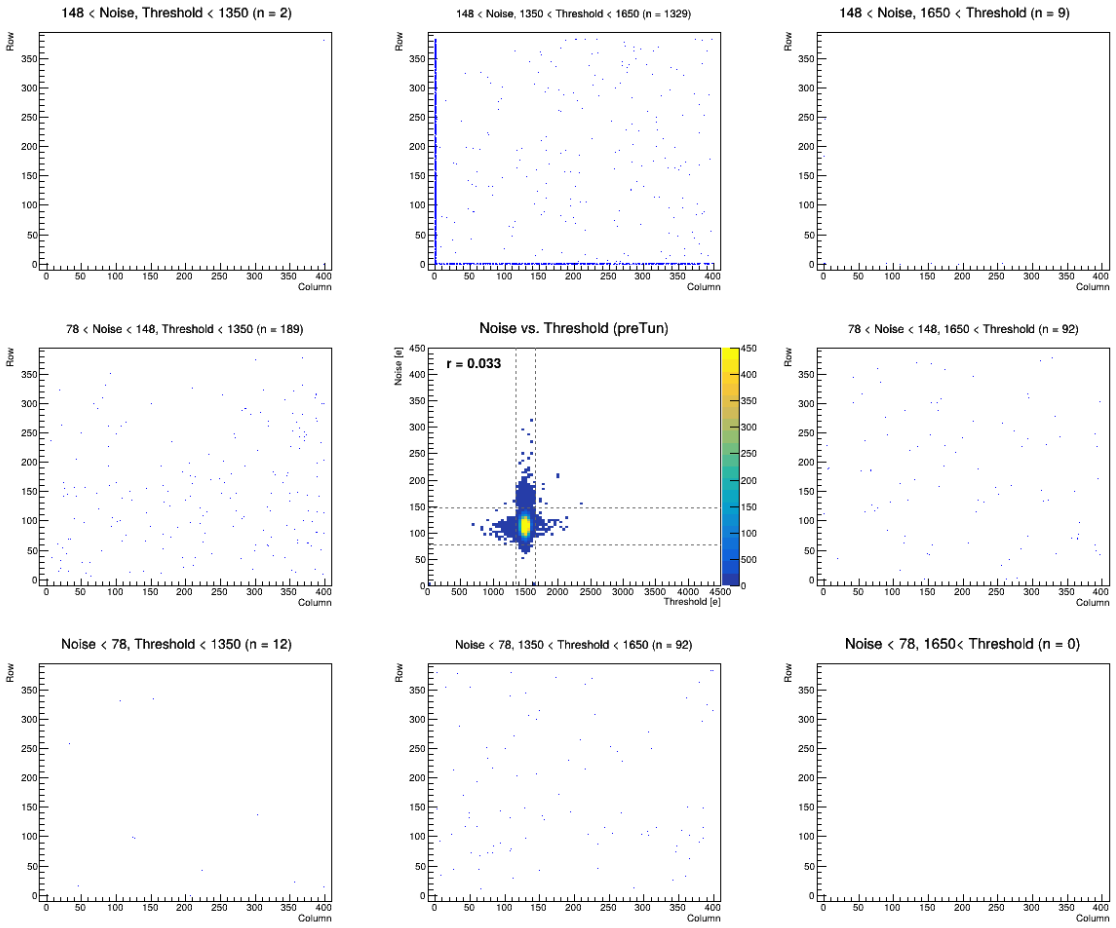

Figure B.13: Location of outliers separated by classifying cuts, FE Chip 0x24162, preTun (cuts in units of e).

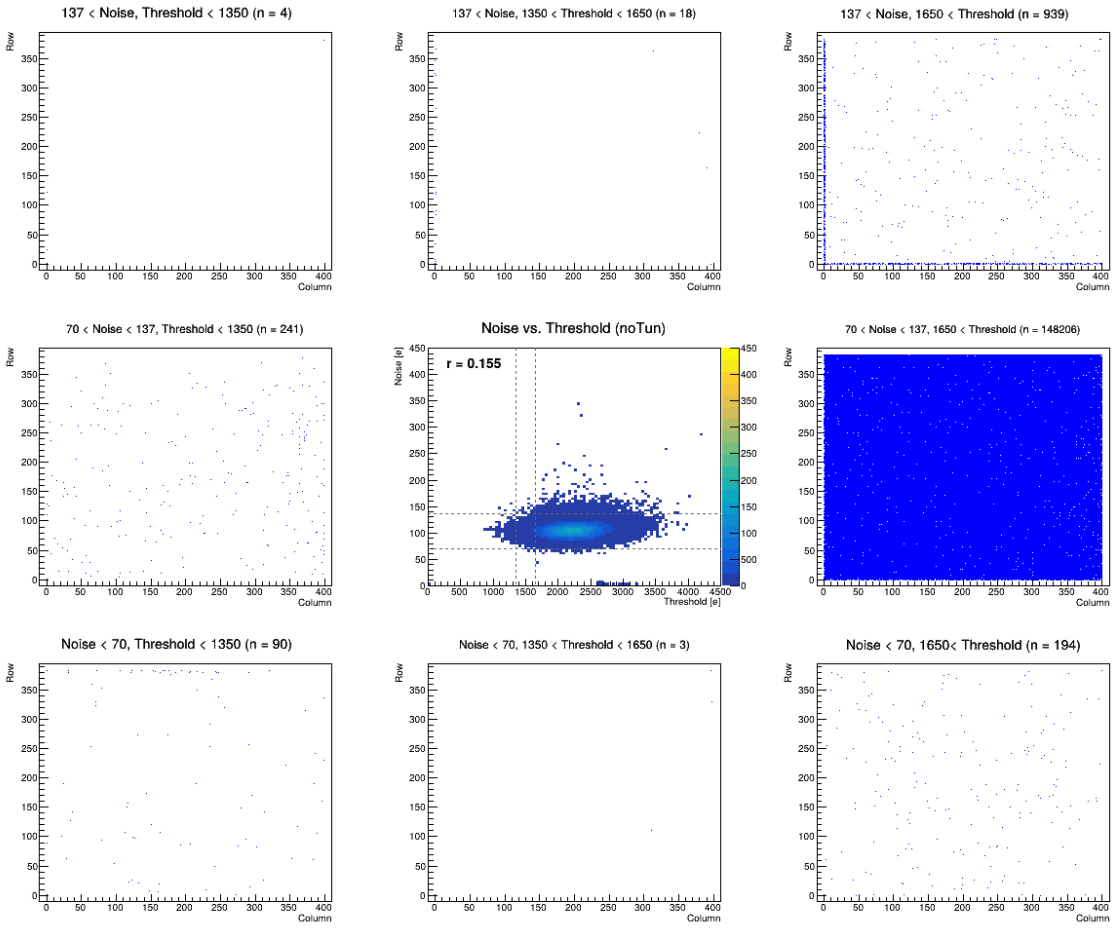

Figure B.14: Location of outliers separated by classifying cuts, FE Chip 0x24162, noTun (cuts in units of e).

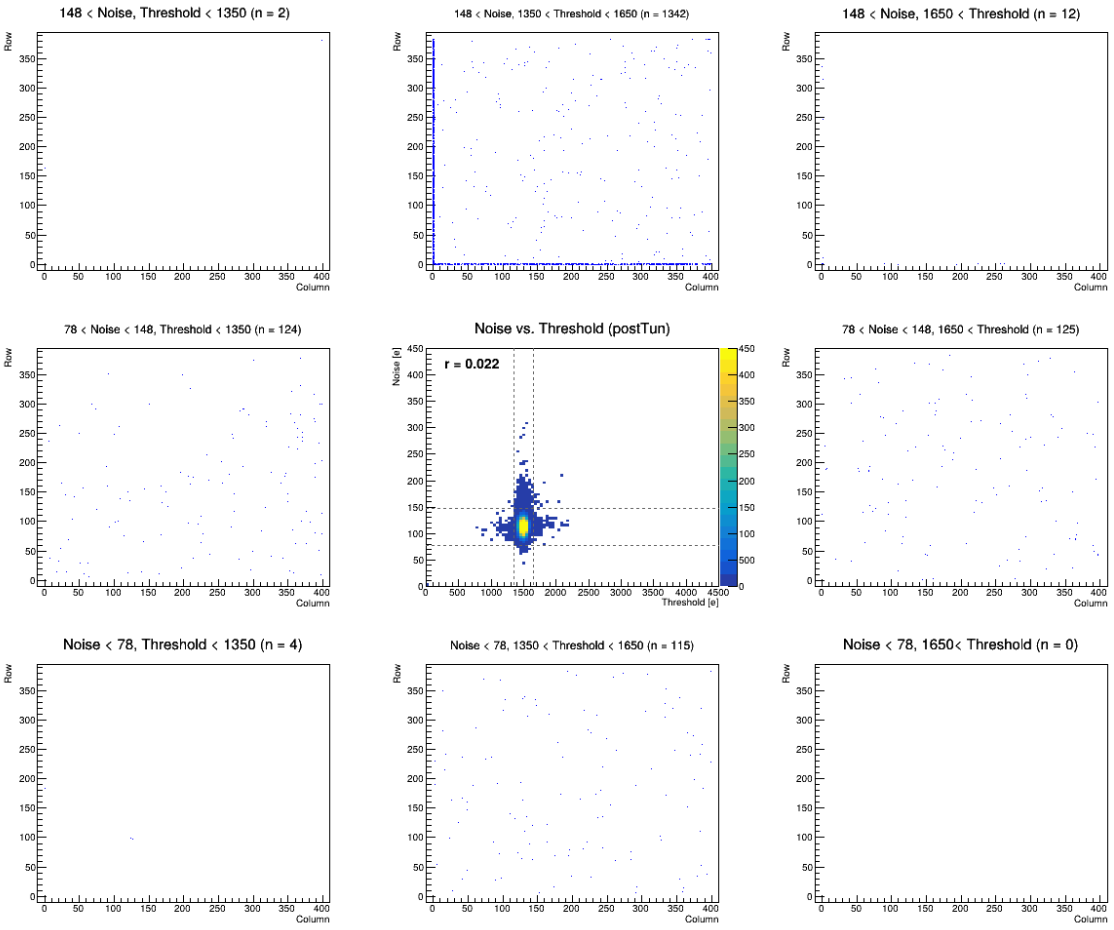

Figure B.15: Location of outliers separated by classifying cuts, FE Chip 0x24162, postTun (cuts in units of e).

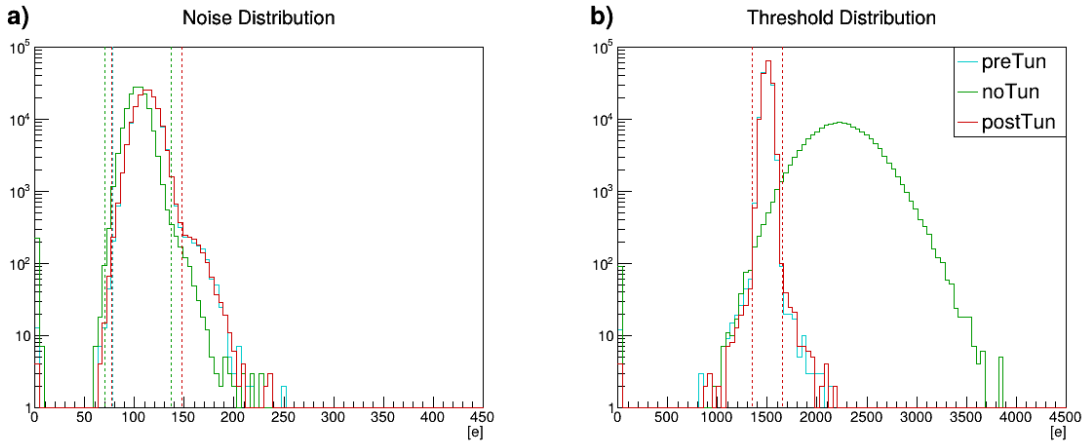

Figure B.16: Noise and threshold distributions of FE Chip 0x24162, all states overlayed.

B.4 FE Chip 0x24174

The characteristics of the noise and threshold distributions of FE chip 0x24174 are listed in Tab. B.3. Fig. B.17, Fig. B.18 and Fig. B.19 display these distributions, while Fig. B.20, Fig. B.21 and Fig. B.22 show the outliers of each region. In Fig. B.23, the noise and threshold distributions of all states are overlaid.

state		mean [e]	std. dev. [e]	lower cut [e]	upper cut [e]	outliers [count]	tot. outliers [count]
preTun	noise	114.4	12.02	78.4	150.5	1510	2047
	threshold	1 493.2	54.59	1 350.0	1 650.0	636	
noTun	noise	105.7	11.97	69.8	141.6	1299	151258
	threshold	2 328.5	335.72	1 350.0	1 650.0	151254	
postTun	noise	114.5	11.90	78.8	150.2	1505	2006
	threshold	1 492.0	43.42	1 350.0	1 650.0	525	

Table B.3: This table shows the characteristics of the noise and threshold distributions of FE chip 0x24174 in each state.

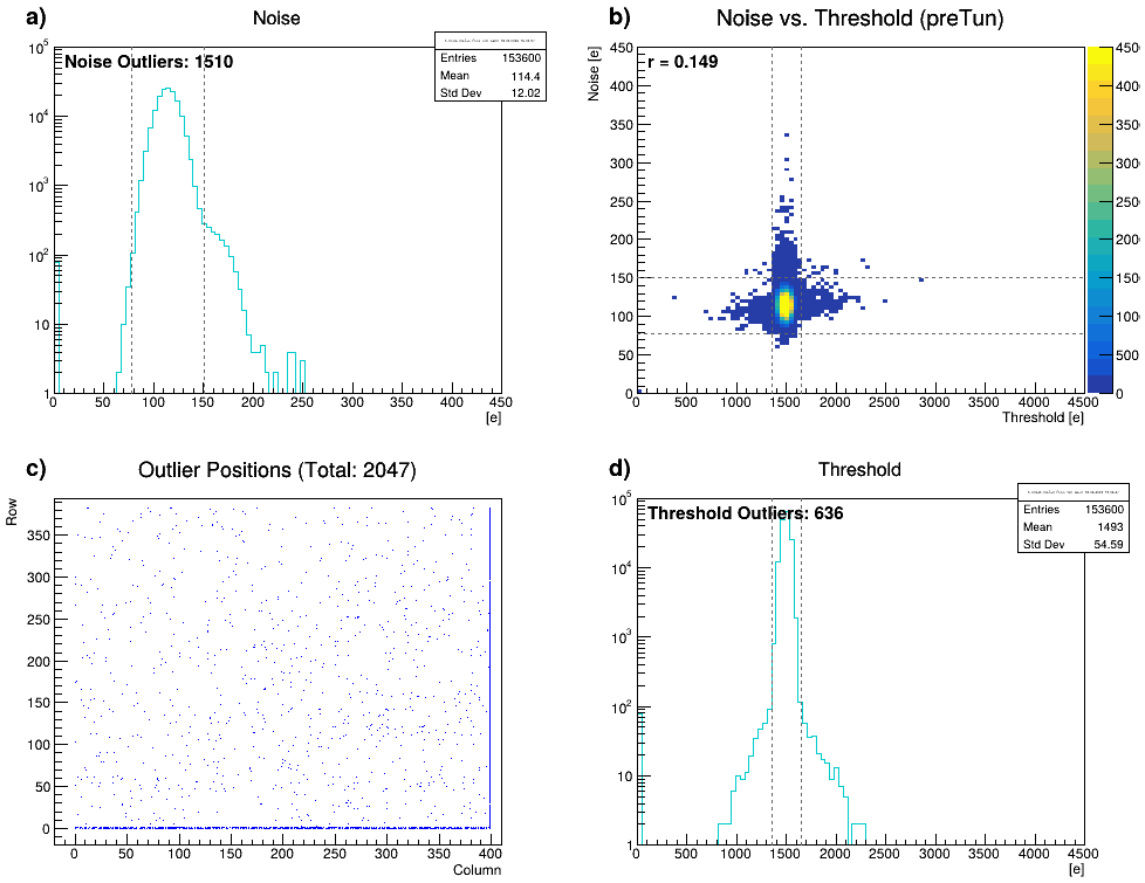

Figure B.17: FE Chip 0x24174, preTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

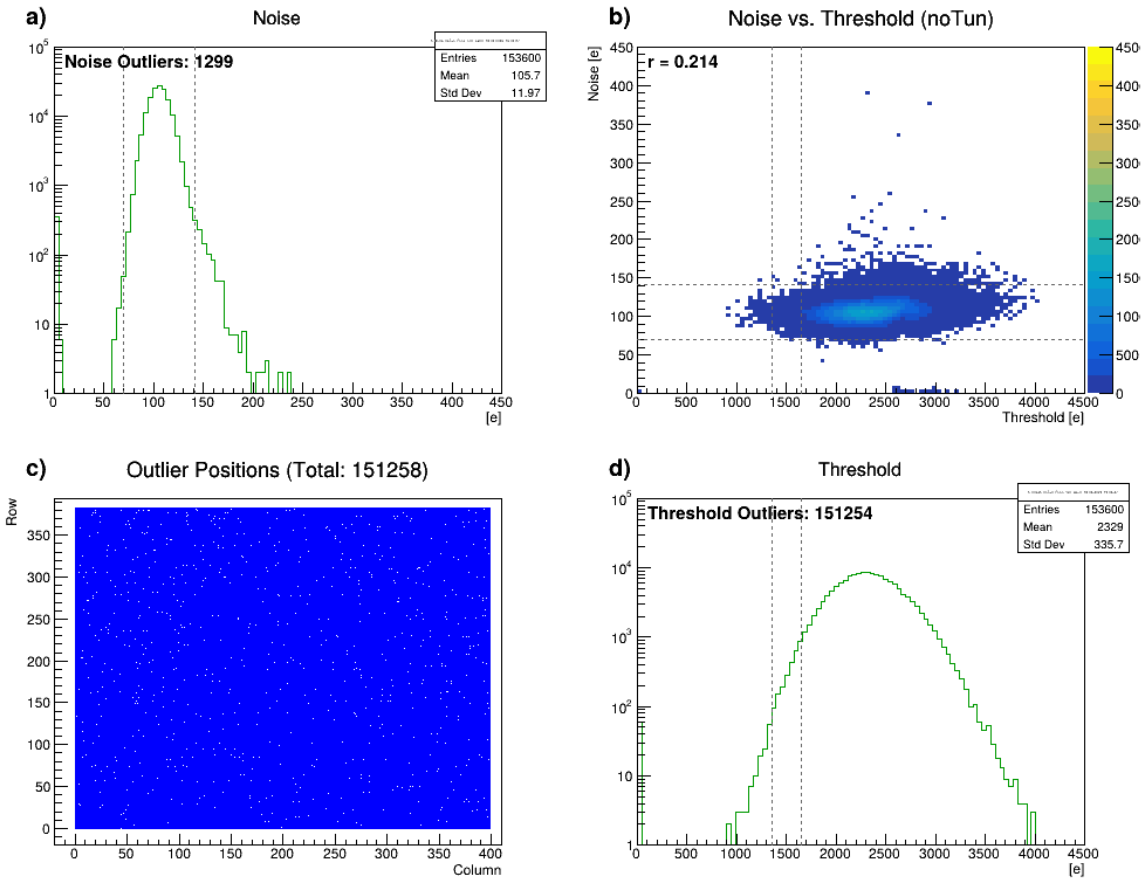

Figure B.18: FE Chip 0x24174, noTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

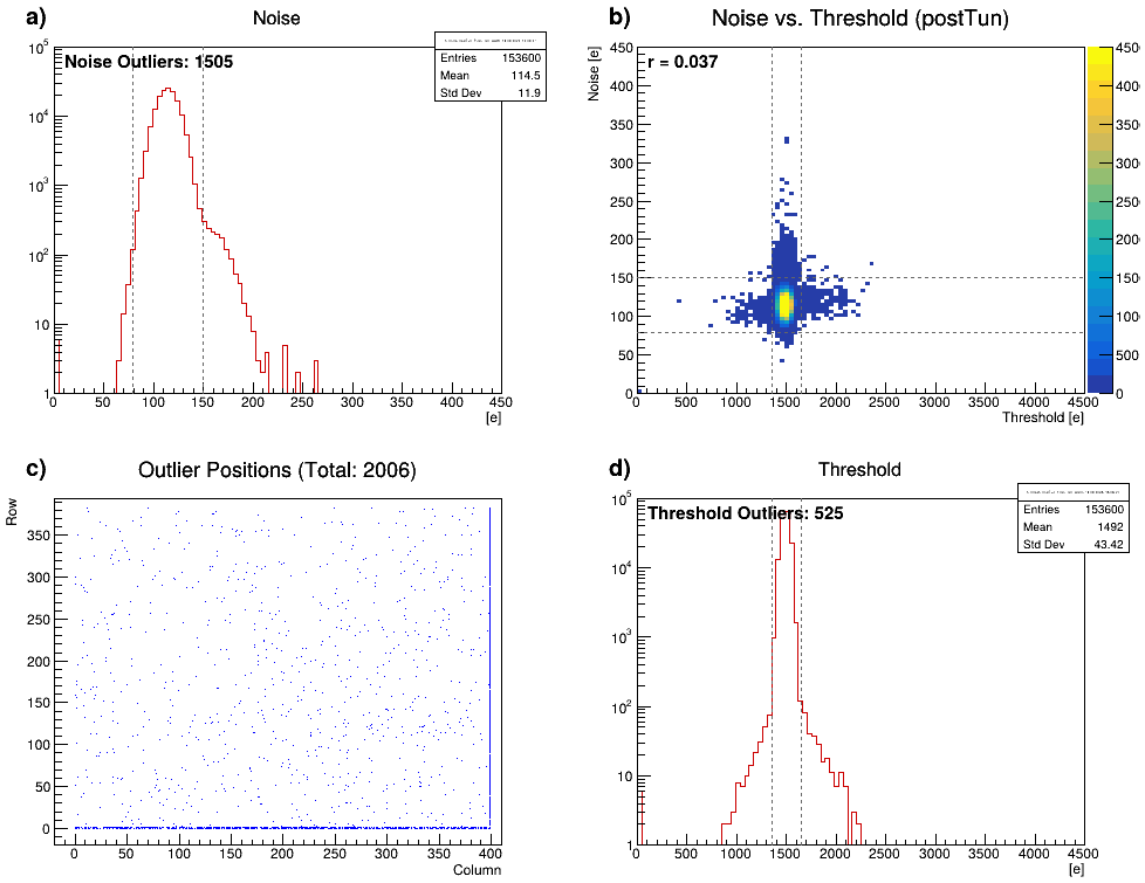

Figure B.19: FE Chip 0x24174, postTun: a) noise distribution, b) noise-threshold correlation, c) outlier positions on FE Chip, d) threshold distribution.

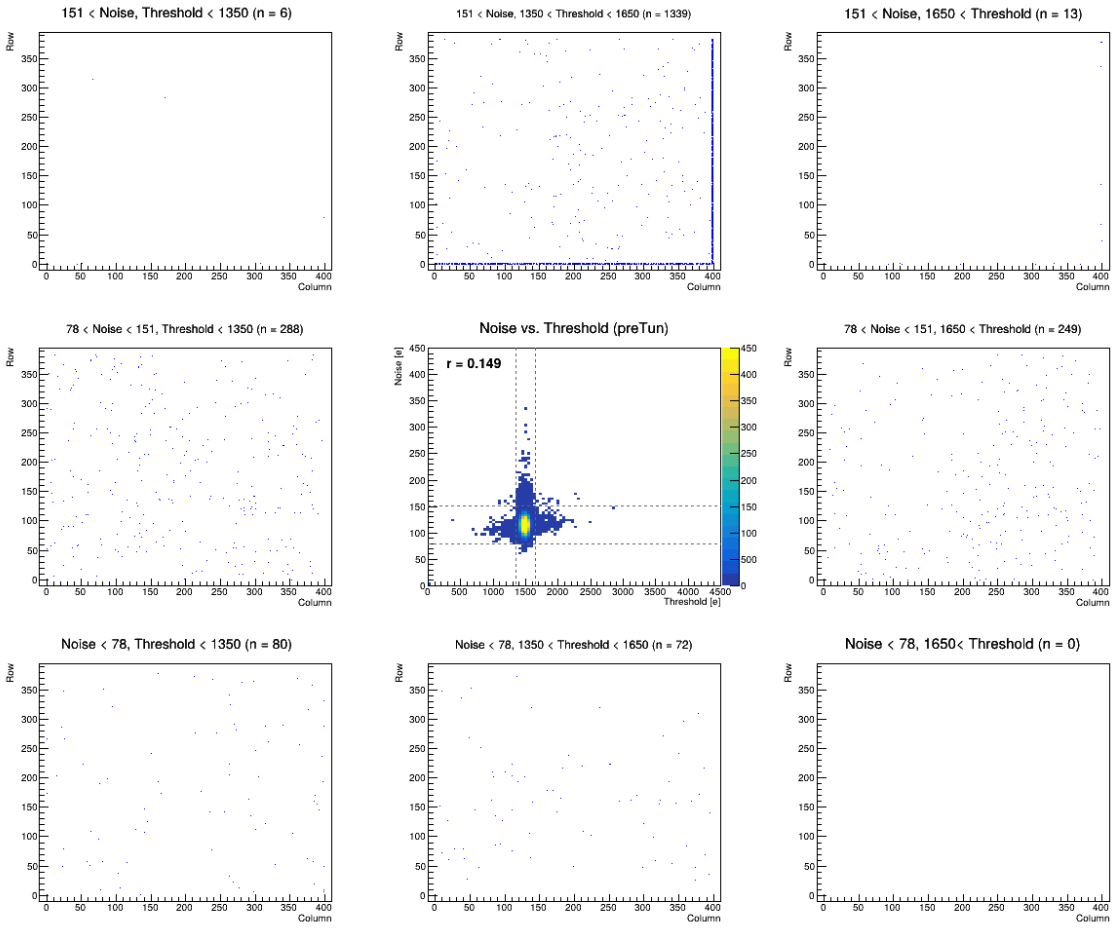

Figure B.20: Location of outliers separated by classifying cuts, FE Chip 0x24174, preTun (cuts in units of e).

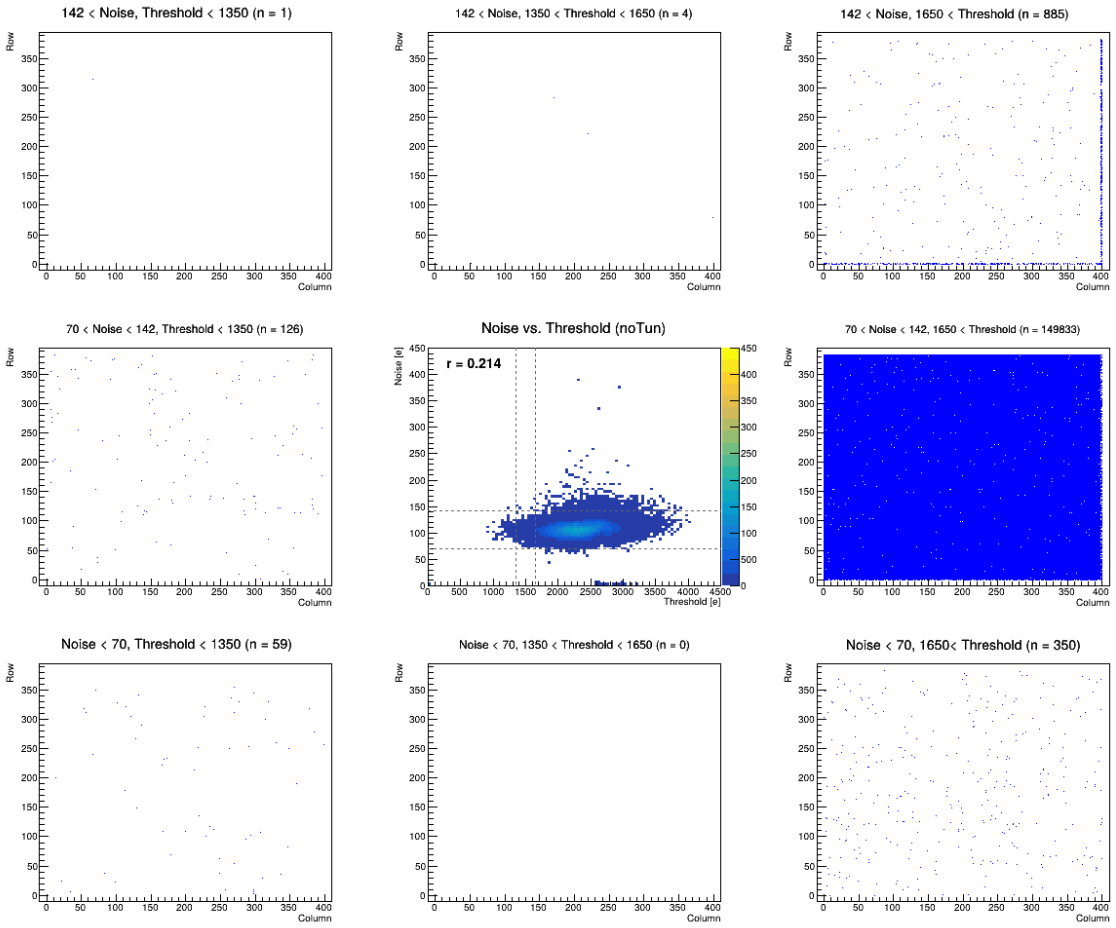

Figure B.21: Location of outliers separated by classifying cuts, FE Chip 0x24174, noTun (cuts in units of e).

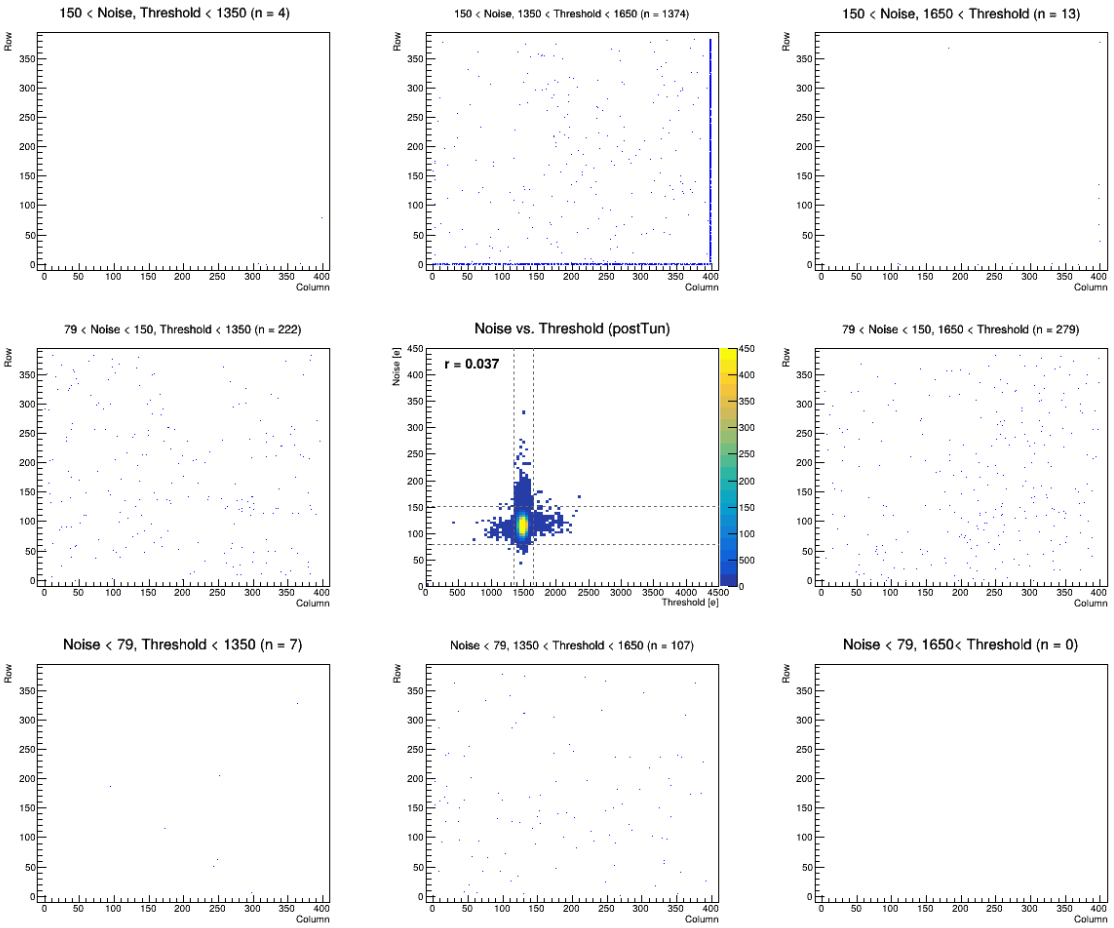

Figure B.22: Location of outliers separated by classifying cuts, FE Chip 0x24174, postTun (cuts in units of e).

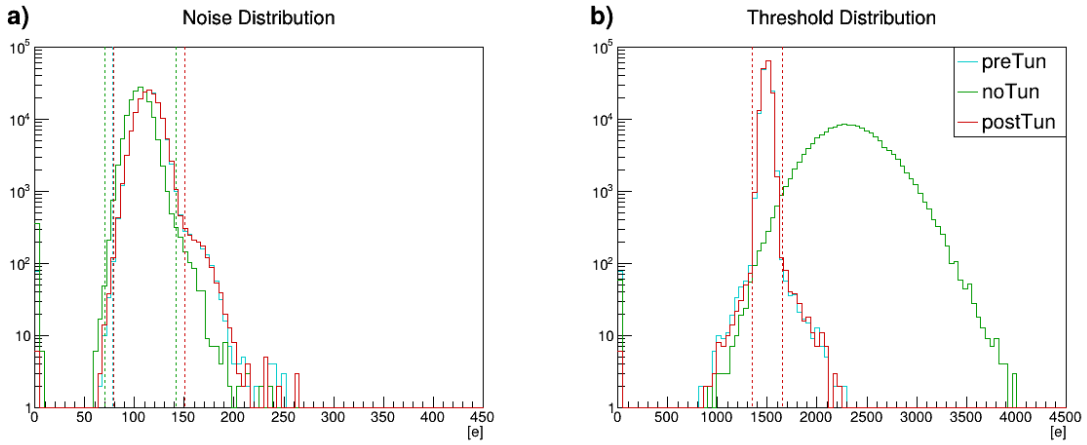

Figure B.23: Noise and threshold distributions of FE Chip 0x24174, all states overlayed.

Bibliography

[1] Wikipedia contributors, *Large Hadron Collider*, URL: https://en.wikipedia.org/wiki/Large_Hadron_Collider (visited on 12/04/2026).

[2] ATLAS Collaboration, *ATLAS Inner Detector Technical Design Report*, 4, 1997, URL: https://cds.cern.ch/record/331063.

[3] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, Journal of Instrumentation **3** (2008), URL: https://doi.org/10.1088/1748-0221/3/08/S08003.

[4] ATLAS Collaboration, *The ATLAS Inner Detector commissioning and calibration*, Eur. Phys. J. C **70** (2010) 787, URL: https://doi.org/10.1140/epjc/s10052-010-1366-7.

[5] ATLAS Collaboration, *Operational Experience and Performance with the ATLAS Pixel Detector at the LHC*, Presented at ICHEP2022, Bologna, Italy, 6–13 July 2022, ATL-INDET-SLIDE-2022-325, URL: https://cds.cern.ch/record/2816415/files/ATL-INDET-SLIDE-2022-325.pdf.

[6] ATLAS Collaboration, *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report*, CERN Yellow Reports: Monographs, 2020, URL: https://cds.cern.ch/record/2749422.

[7] ATLAS Collaboration, *The High Luminosity Large Hadron Collider*, 2nd ed., WORLD SCIENTIFIC, 2024, URL: https://www.worldscientific.com/doi/abs/10.1142/13487.

[8] ATLAS Collaboration, *ATLAS Inner Tracker Pixel Detector Technical Design Report*, 2018, URL: https://cds.cern.ch/record/2285585.

[9] ATLAS Collaboration, *Expected tracking performance of the ATLAS Inner Tracker at the High-Luminosity LHC*, Journal of Instrumentation **20** (2025), URL: https://doi.org/10.1088/1748-0221/20/02/P02018.

[10] ATLAS Collaboration, *ATLAS ITk Pixel Detector Overview*, Proceedings for CORFU2024, Corfu, Greece, 2024, ATL-ITK-PROC-2025-011, URL: https://cds.cern.ch/record/2928802/files/ATL-ITK-PROC-2025-011.pdf.

[11] ATLAS Collaboration, *ATLAS ITk Pixel Detector Overview*,
Proceedings for ICHEP2024, Prague, Czech Republic, 2024, ATL-ITK-PROC-2024-038,
URL: https://cds.cern.ch/record/2914047/files/ATL-ITK-PROC-2024-038.pdf.

[12] ATLAS Collaboration, *Electrical Specification and QC Procedures for ITkPixV1.1 Modules*,
tech. rep. AT2-IP-QA-0025, 2025.

[13] ATLAS Collaboration,
YARR - A PCIe based Readout Concept for Current and Future ATLAS Pixel Modules,
J. Phys.: Conf. Ser. **898** (2017) 032053,
URL: https://doi.org/10.1088/1742-6596/898/3/032053.

[14] ATLAS Collaboration,
Upgrade of the YARR DAQ System for the ATLAS Phase-II Pixel Detector Readout Chip,
Proceedings for TWEPP-17, Santa Cruz, California, 2017,
URL: https://doi.org/10.22323/1.313.0076.

List of Figures

List of Tables

Acknowledgements

I am particularly grateful for the patient and consistently helpful guidance and advice provided by my supervisor, Dr. Wolfgang Walkowiak, throughout this thesis, both during laboratory measurements and the writing process.

I am also very grateful to Prof. Dr. Markus Cristinziani for giving me the opportunity to carry out this thesis within the Experimental Particle and Astroparticle Physics Group at the University of Siegen, where I benefited from a supportive and pleasant working environment.