

## Master's Thesis

# Developments for the ATLAS ITk Pixel Loaded Local Supports Readout System

prepared by

**Paolo Maria Malatesta**

from Milan, Italy

at the II. Physikalischen Institut

**Thesis number:** II.Physik-UniGö-MSc-2025-04

**Thesis period:** February 22, 2025 until July 15, 2025

**First referee:** Prof. Dr. Arnulf Quadt

**Second referee:** Prof. Dr. Jörn Große-Knetter



# Abstract

The High Luminosity LHC upgrade requires the upgrade of the Inner Detector, which will be replaced by the all-silicon Inner Tracker (ITk). The ITk has two subsystems: the Pixel and the Strip detector. The Pixel detector is composed of around 10,000 modules, i.e. detector units. Groups of modules will be loaded on carbon fibre structures, forming the Loaded Local Supports. This master's thesis presents a preliminary evaluation of the FELIX/YARR readout chain, selected for performing quality control tests on the Loaded Local Supports. The data collected during the readout chain tests has led to identifying necessary developments for the YARR software. Consequently, the Data Acquisition software was upgraded to feature a new design with a parallel multi-module configuration.

**Keywords:** ITk Pixel read-out, Loaded Local Supports, YARR, ITkPixV1



# Contents

|                                                             |           |
|-------------------------------------------------------------|-----------|
| <b>1. Introduction</b>                                      | <b>1</b>  |
| <b>2. The ATLAS Experiment at the Large Hadron Collider</b> | <b>3</b>  |
| 2.1. The Inner Detector . . . . .                           | 4         |
| 2.2. Calorimetry . . . . .                                  | 5         |
| 2.3. Muon System . . . . .                                  | 7         |
| 2.4. Trigger and Data Acquisition . . . . .                 | 8         |
| <b>3. The Inner Tracker (ITk)</b>                           | <b>11</b> |
| 3.1. The Pixel Detector . . . . .                           | 12        |
| 3.2. The Strip Detector . . . . .                           | 14        |
| 3.3. The Local Supports . . . . .                           | 15        |
| 3.3.1. The Pixel Local Supports . . . . .                   | 16        |
| 3.3.2. The Strips Local Supports . . . . .                  | 18        |
| 3.3.3. The Loaded Local Supports (LLS) . . . . .            | 18        |
| <b>4. Data Acquisition for the ITk Pixel detector</b>       | <b>21</b> |
| 4.1. The on-detector readout electronics . . . . .          | 21        |
| 4.1.1. The lpGBT . . . . .                                  | 22        |
| 4.1.2. The GBCR2 . . . . .                                  | 23        |
| 4.1.3. The VTRx+ . . . . .                                  | 24        |
| 4.2. The FELIX Project . . . . .                            | 26        |
| 4.2.1. FELIX hardware . . . . .                             | 27        |
| 4.2.2. FELIX software . . . . .                             | 27        |
| 4.3. YARR: an ITk Pixel readout software . . . . .          | 28        |
| 4.3.1. Design of the Scans . . . . .                        | 29        |
| 4.3.2. Scan console . . . . .                               | 32        |
| <b>5. System Testing</b>                                    | <b>35</b> |
| 5.1. Test setups . . . . .                                  | 35        |
| 5.1.1. Local test setup . . . . .                           | 35        |

|                                                                         |           |
|-------------------------------------------------------------------------|-----------|
| 5.1.2. Test setup at CERN . . . . .                                     | 38        |
| 5.1.3. ITkPixV2 hardware emulator . . . . .                             | 39        |
| 5.2. Digital Scan of a Quad Module . . . . .                            | 41        |
| 5.3. Digital Scan of Multiple Modules . . . . .                         | 43        |
| 5.4. Further Scans of Multiple Modules . . . . .                        | 44        |
| 5.5. Results using the ERF to SMA adapter . . . . .                     | 45        |
| 5.6. Estimating the error . . . . .                                     | 47        |
| 5.7. Comparison with previous results . . . . .                         | 48        |
| 5.8. Considerations on the Scan Timings . . . . .                       | 49        |
| <b>6. Software Development</b>                                          | <b>51</b> |
| 6.1. Motivation . . . . .                                               | 51        |
| 6.2. Configuration Design in the Released Version . . . . .             | 52        |
| 6.2.1. YARR controller libraries . . . . .                              | 52        |
| 6.2.2. YARR Front-end libraries . . . . .                               | 53        |
| 6.2.3. Data flow in the configuration . . . . .                         | 53        |
| 6.3. Parallel Multi-Module Configuration via TX core Wrapping . . . . . | 56        |
| 6.4. Multi-client development . . . . .                                 | 59        |
| 6.4.1. Tests with ITkPixV1 Quad Modules . . . . .                       | 60        |
| 6.4.2. Tests with ITkPixV2 hardware emulator . . . . .                  | 61        |
| 6.5. Other developments for the LLS . . . . .                           | 63        |
| <b>7. Summary, Conclusion and Outlook</b>                               | <b>65</b> |
| <b>A. Additional Figures and Tables</b>                                 | <b>67</b> |

# 1. Introduction

Science is often driven by curiosity; on the basis of this statement, humans have been asking themselves questions about nature and the universe. One of these questions, which the ancient Greeks were already pondering, is: What is the fundamental constituent of matter? In the twentieth century, a theory called the Standard Model (SM) showed to describe in a satisfying way the subatomic world.

The SM describes matter as composed of a few elementary particles, which are called fermions. These particles interact with each other, and bosons mediate their interaction. The bosons are also elementary particles, and they are mediators of the electroweak and strong forces.

The particles described by the SM have different properties, such as mass, charge, and lifetime. Some of them can also combine, and these bound states form, for example, the protons and neutrons of the atoms. Some other particles, like the top quark, have a high mass. Thus, they are unstable and tend to decay into lighter particles, states with a lower potential energy.

In order to study highly massive particles, accelerators are required. The world-leading facility is the Large Hadron Collider (LHC) at CERN [1], which operates at the record centre of mass energy of 13.6 TeV. One of its main experiments is the ATLAS experiment [2]. An upgrade for the LHC collider is scheduled to begin in July 2026, which will increase the luminosity. This will be achieved by increasing the number of collisions per bunch crossing rather than the bunch crossing frequency. The scientific motivation to increase the luminosity is to allow scientists to perform precision measurements of rare events, probing the standard model and intensifying the search for new physics.

The higher particle flux in the detector will pose a challenge to the ATLAS detector, and some of its components will need to be updated. The new innermost detector is the Inner Tracker (ITk) [3]: a finer granularity will keep the tracking efficiency high, avoiding excessive pile-up. In addition, the components are designed to be radiation-hard and will withstand a radiation dose up to  $10^{16} \text{ n}_{\text{eq}}/\text{cm}^2$ . The ITk is composed of two subsystems: the Pixels and the Strips.

The Data Acquisition system (DAQ) will also need to be upgraded to handle the higher

## *1. Introduction*

flux of data from the detector. FELIX [4] was developed as an interface for all the ATLAS subdetectors, and it is designed to be highly flexible for meeting their different needs.

The Readout chain of the Pixels works in the following way: the detector units (modules) are connected via high-speed electrical links to the Optoboard. This board contains lpGBTs that aggregate the electrical signals, GBCRs that enhance them, and a VTRX+ that converts the signal from electrical to optical. The signal is subsequently transmitted with optical fibres to the FELIX PCs. During the production phase, a DAQ software called YARR is employed. Through YARR, it is possible to configure the modules and read the data out. To perform the read-out of the full ITk-Pixel part (around 10,000 modules), a new DAQ software, called ITk-Felix-SW, is currently being developed targeting the new upgraded ITk operation, planned to start in 2030.

In order to be placed in the detector, the modules are mounted on the Local Supports (LS). They are carbon fibre structures that mechanically hold the modules in place, and they house the services (cooling, powering and data cables). After the manual assembly is completed, the Loaded Local Supports need to be tested: a large number of modules (up to 36) are now read out together.

In this thesis, tests to check the serial powering and the FELIX/YARR read-out chain have been completed with up to 6 modules. The results of these investigations showed the need for a parallelisation of the modules' configuration, as they were configured one by one. Therefore, a new development of the YARR software featuring a parallel multi-module configuration was prepared.

## 2. The ATLAS Experiment at the Large Hadron Collider

The Large Hadron Collider (LHC) [1] accelerator was built to allow scientists to investigate events in an energy range that was previously impossible with accelerators. It is located at CERN in Geneva, where an international team of scientists works on one of the most important high-energy physics experiments in the world. The LHC is a proton-proton collider that was first operated at a centre of mass energy of 7 TeV, later expanded to 13.6 TeV. Cutting-edge values of luminosity and centre-of-mass energy allowed very relevant discoveries, the most important being the discovery of the Higgs boson [5]. Four major experiments use the LHC collider: ALICE, ATLAS, CMS and LHCb. ALICE [6] is optimised to investigate heavy ion collisions. CMS [7] is a general-purpose detector, which means that it is used for many studies, among which are Standard Model (SM) measurements and the search for Dark Matter. LHCb [8] studies mainly the bottom quark, constraining and measuring CKM matrix parameters.

The ATLAS experiment [2] is a general-purpose detector. Research topics include measurements of parameters of the SM and tests of Beyond Standard Model (BSM) theories. It is designed to be a  $4\pi$  detector, which means the intention is to capture all possible decays of an interaction between two protons. The geometry chosen for this task is the barrel geometry, as it is possible to see in Figure 2.1. The detector is not fully hermetic, as particles with high pseudorapidity (i.e. emitted almost parallel to the beamline) cannot be detected for technical reasons. The various systems are displayed inside the detector in layers. In the next pages, the current version of the ATLAS detector will be described, starting from the beamline and proceeding outwards.

## 2. The ATLAS Experiment at the Large Hadron Collider

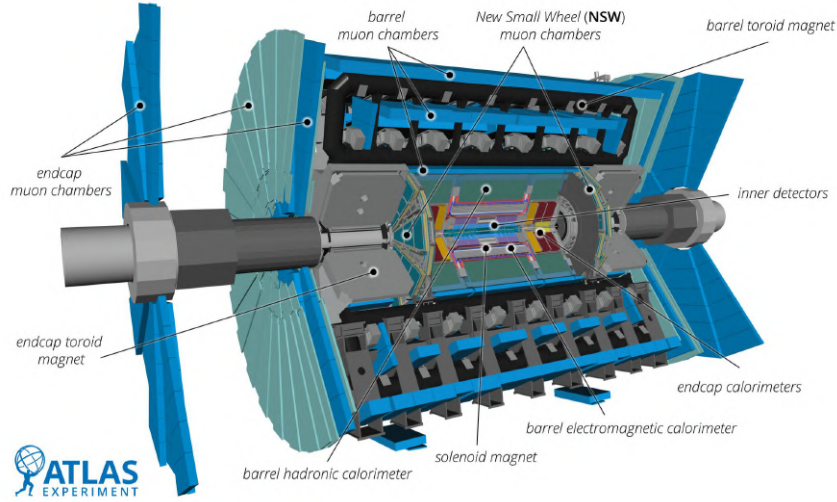


Figure 2.1.: Cut-away view of the ATLAS detector. The dimensions of the detector are 25 m in height and 44 m in length. © CERN, 2022.

### 2.1. The Inner Detector

The Inner Detector is of utmost importance for the experiment as it provides tracking, primary and secondary vertexing, and momentum measurement. It certainly faces the harshest conditions, since it lies only 50 mm away from the beamline, and it requires excellent granularity.

The Inner Detector is composed of the Transition Radiation Tracker (TRT), the SemiConductor Tracker (SCT) and the pixel detector. A view of the detector is seen in Figure 2.2. A solenoid magnet provides a 2 T magnetic field to the detector, which is used to perform momentum measurements and infer the mass and the charge of the particles.

The Pixel detector [9] is the closest to the beamline. All the layers except the innermost, feature the FEI3 module, with a minimum pixel size of  $50 \times 400 \mu\text{m}^2$ . The innermost layer (IBL) was replaced before the start of run 2, and it features FEI4 modules, with an improved pixel size of  $50 \times 250 \mu\text{m}^2$  [10]. The SCT [11] is placed between the TRT and the Pixel detector. It is a silicon strip detector featuring 4 layers in the barrel and 18 end-caps (9 per side). The strip pitch is  $8 \mu\text{m}$ . The outermost layer is the TRT. It is a gas detector composed of straw tubes with a diameter of 4 mm. One of the motivations of this design is its excellent ability to identify electrons: the signal intensity is directly proportional to the Lorentz factor  $\gamma = \frac{E}{mc^2}$ .

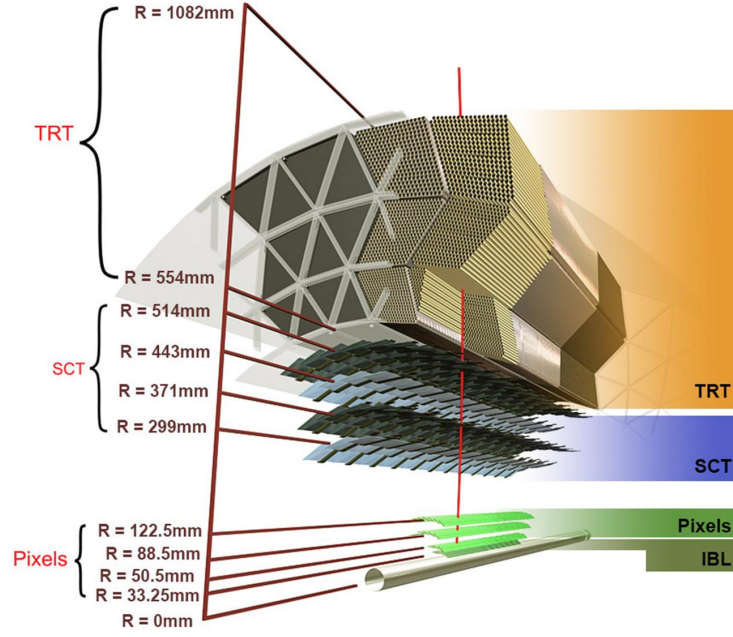


Figure 2.2.: Schema showing the sensors and structural elements traversed by a charged track of  $p_T = 10$  GeV in the barrel inner detector ( $\eta = 0.3$ ). © CERN, 2020.

## 2.2. Calorimetry

A step further inside, there are the calorimeters. The purpose of the calorimeters is to measure the energy of the particles that traverse them. The general principle of a calorimeter is the following: the incoming particles interact with the material (via strong or electromagnetic interaction), and a larger number of lower-energy particles are created. The particles originating from the first interaction, called secondary particles, undergo the same process again. More and more particles are produced until a threshold is met; this process is called showering. The energy of the shower particles is measured, and that of the incoming particle is reconstructed.

There are two types of calorimeters at ATLAS. The innermost is the electromagnetic calorimeter. It is designed to measure the energy of photons and electrons; thus, the particles interact via electromagnetic interaction. The hadronic calorimeter is placed further outside so that electrons and photons have already been stopped. It measures the showers that are initiated by hadrons. A scheme of the calorimetry is seen in Figure 2.3.

The Liquid Argon (LAr) electromagnetic calorimeter is composed of a barrel part ( $|\eta| < 1.475$ ) and the end-caps ( $1.375 < |\eta| < 3.2$ ). The electromagnetic calorimeter at ATLAS is a sampling calorimeter. The absorber is lead embedded in 0.2 mm stainless steel plates, while the ionising material is liquid argon (as the name suggests). The readout electrodes are located in the gaps between the absorbers and consist of three conductive

## 2. The ATLAS Experiment at the Large Hadron Collider

copper layers separated by insulating polyamide sheets.

The hadronic calorimeter has three main components: the Tile calorimeter, the LAr hadronic end-cap calorimeter and the LAr forward calorimeter. The Tile calorimeter covers the barrel region; it is a sampling calorimeter using steel as absorber and scintillating tiles as active material. The LAr hadronic end-cap calorimeter extends coverage in the end-cap region up to  $\eta = 3.1$ ; its absorber material is copper, varying in thickness. The LAr forward calorimeter, designed for higher pseudorapidity values, consists of three distinct layers. The first one serves as an electromagnetic calorimeter, while the two outermost layers are part of the hadronic calorimeter and use tungsten as an absorber.

All particles except neutrinos and muons are stopped in the calorimeters. The first ones are not stopped because they only interact weakly. Although an interaction is possible, its cross-section is very low. Muons are not detected here because the electromagnetic showers are initiated by a Bremsstrahlung process. The cross-section of this process is proportional to  $1/m^2$ ; thus, muons do not effectively emit Bremsstrahlung radiation in the energy range relevant to ATLAS.

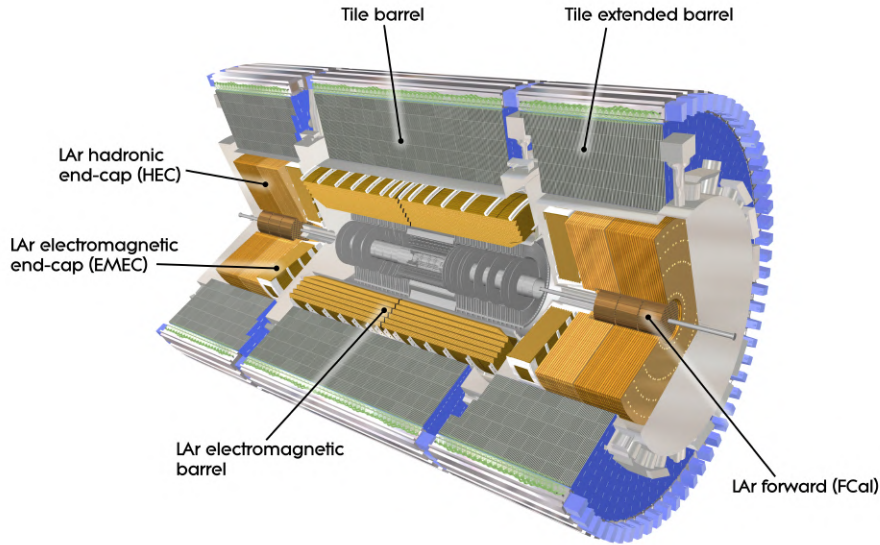


Figure 2.3.: Cut-away view of the ATLAS calorimeter system. © CERN, 2008.

## 2.3. Muon System

The outermost layer of the ATLAS detector is the muon spectrometer. Since it is placed outside the calorimeters, only muons (and neutrinos) can reach it. It covers a pseudorapidity range of  $|\eta| < 2.7$  and operates immersed in a magnetic field to measure precisely the muons' momentum. The system is divided into two parts: the barrel region  $|\eta| \leq 1.2$  and the end-cap region ( $1.2 < |\eta| < 2.7$ ).

The layout of the muon system is seen in Figure 2.4. The barrel region is fitted with cylindrical drift tubes (Monitored Drift Tubes, MDT). In the end-cap region, 2 systems are present: the MDT chambers and the Cathode Strip Chambers (CSCs). The MDT chambers are made of two three-layered MDTs, while the CSCs are multi-wire proportional chambers.

In the region  $|\eta| < 1.4$ , the barrel toroid magnet provides a magnetic field of 0.5 T, while in the region between  $1.6 < |\eta| < 2.7$  the magnetic field is provided by smaller magnets placed in the end-caps. In the region  $1.4 < |\eta| < 1.6$ , a combination of the two fields is present.

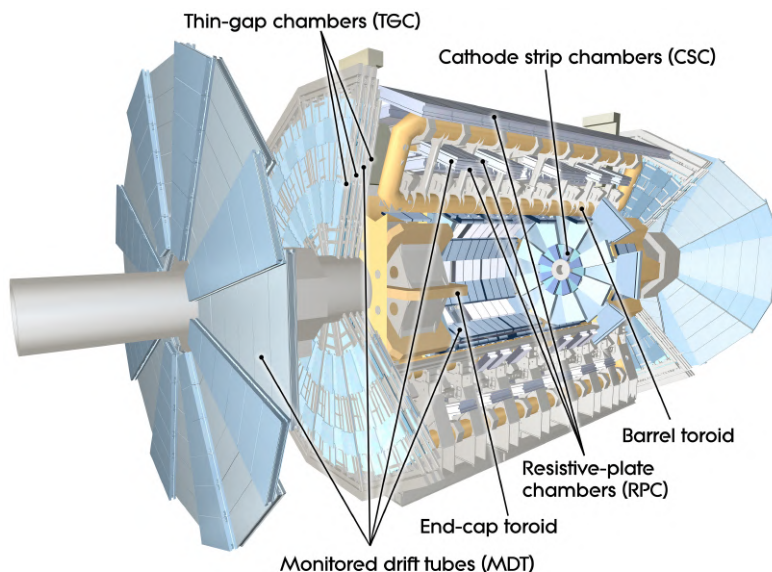


Figure 2.4.: Layout of the ATLAS Muon System. © CERN, 2008.

During the Long Shutdown 3, part of the muon system was updated in preparation for the higher luminosities of HL-LHC. The Small Wheels, placed in the end-cap region, were updated to the New Small Wheels [12]. The 10 m diameter NSW are equipped with completely new detector technologies: the Micro Pattern Gaseous Detectors (MPGD) are here employed for the first time in a large-scale detector. More updates for the HL-LHC

## 2. The ATLAS Experiment at the Large Hadron Collider

are planned, namely on the electronics and in the transition area between the barrel and the end-caps [13].

### 2.4. Trigger and Data Acquisition

A fundamental part of ATLAS is the Trigger and Data Acquisition (TDAQ) system. The trigger plays a key role since it would be impossible to save all the data of each collision at ATLAS. The TDAQ system collects information about the event and, if it complies with determined preset parameters, saves it. As seen in Figure 2.5, the TDAQ system has many components. There are two main components: the L1 trigger, which is a hardware trigger and the High Level Trigger (HLT), which is software-based.

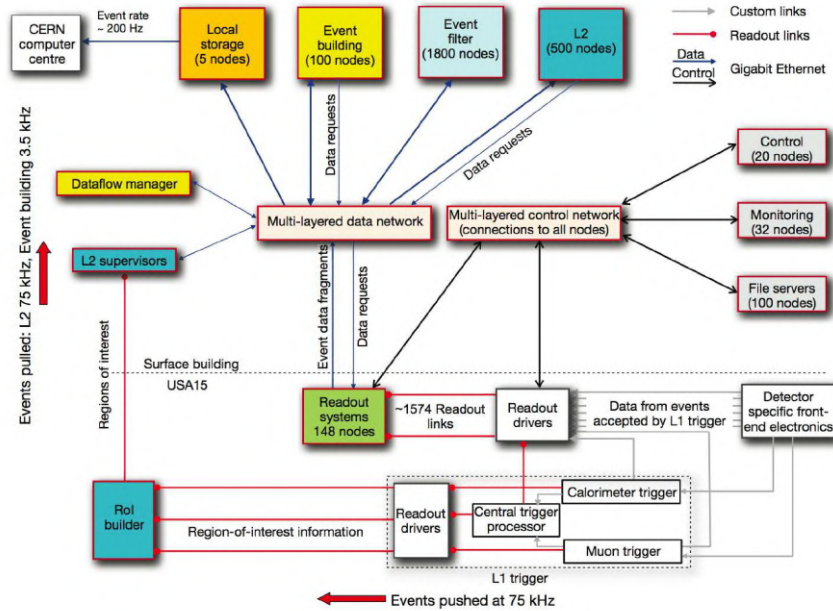


Figure 2.5.: Block diagram of the ATLAS TDAQ system [2].

#### The L1 trigger

The first level of the trigger is the L1 trigger; it selects high transverse momentum ( $p_T$ ), muons, electrons, tau leptons (hadronic decay), as well as events with high total and missing transverse energy. These selection criteria comply with the research targets of the ATLAS experiment. Leptons with high transverse momentum typically come from the decay of heavy particles (e.g. Z, W bosons). The study of heavy particles, e.g. top

quark and Higgs, is a crucial part of the ATLAS agenda. Events with a large missing transverse energy could originate from yet-unknown particles (New Physics). The L1 trigger is composed of three subsystems: the Calorimeter trigger, the Muon Trigger and the Central Trigger Processor (CTP).

The calorimeter trigger identifies particles with high transverse energy, including electrons, muons, photons, jets, and hadronically decaying taus. It is also capable of detecting events characterised by significant missing or total transverse energy. The Level-1 Calorimeter Trigger (L1 Calo) processes data originating from approximately 7,000 analogue trigger towers, each with a reduced granularity of at least  $0.1 \times 0.1$  in  $\Delta\eta \times \Delta\phi$ .

The muon trigger reads out its data from dedicated areas of the muon spectrometer: the RPC's and the TGC's. These subsystems are finely segmented and have a faster response time (providing unambiguous identification of the bunch-crossing containing the muon candidate). The purpose of the muon trigger is to identify high transverse momentum muons: it is achieved using an algorithm which requires the coincidence of different trigger stations from the interaction point through the detector. The width of the path of the muon (road) is proportional to the transverse momentum.

The Central Trigger Processor (CTP) is a decision-making unit that processes the output of, among others, the muon trigger and the L1 Calo. It combines the information coming from the subsystems and implements the trigger logic, which includes selection criteria for high-priority physics events. It is also responsible for transmitting the L1 Accept (L1A), i.e. the information about the events that were accepted by the L1 trigger, as well as the timing for all ATLAS detector systems. The L1 decision latency is about 2.5  $\mu$ s.

### The High-Level Trigger

The High-Level Trigger (HLT) in the ATLAS experiment includes the Level-2 (L2) trigger and the Event Filter (EF). Its primary goal is to efficiently process and select events of interest for further analysis, reducing the event rate to a manageable level while retaining events of significant importance for physics.

The Level-2 (L2) trigger processes events identified as potentially interesting by the first-level trigger. It performs detailed analyses on specific regions of the detector, called Regions Of Interest (ROIs), using additional detector data. It processes the data against predefined physics criteria/simulations. Events that meet these criteria are forwarded for further examination, while the majority are rejected to reduce the event rate. The

## *2. The ATLAS Experiment at the Large Hadron Collider*

L2 system is designed to handle high data rates efficiently by temporarily storing detector data, distributing the processing workload across multiple computing units, and employing fast decision-making algorithms.

The Event Filter (EF) is the third and final stage of the ATLAS experiment's trigger system, following the Level-1 (L1) and Level-2 (L2) triggers. It uses algorithms akin to those used in offline data analysis to select events of interest. The comprehensive work of the trigger system reduces the event rate to approximately 3 kHz during run 3 [14].

### 3. The Inner Tracker (ITk)

The High Luminosity LHC (HL-LHC) is designed to allow scientists to study relevant topics of physics [15]. The first stream is represented by Higgs physics. Some Higgs decays are currently inaccessible at LHC. All of them are sensitive, in different ways, to Beyond Standard Model (BSM) Physics. In addition, Higgs studies are required to better understand and investigate the electroweak symmetry-breaking mechanism. The second stream is represented by the quest for BSM phenomena and processes that cannot be explained by the Standard Model (SM), e.g. dark matter. Among BSM studies, some of them suggest that the top quark plays a special role. Its special properties of high mass and the fact that it is not found in bound states could hint at this. Lastly, studies about QCD at high density and temperature, as well as better measurements of Parton Distribution Functions (PDFs), could be performed.

The HL-LHC [16] is programmed to increase the luminosity to an instantaneous luminosity up to  $7.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  and an integrated luminosity to  $3000 \text{ fb}^{-1}$ . These values are optimistic, considering a pile-up of about 200. The minimum requirement for the HL-LHC is an integrated luminosity of  $2500 \text{ fb}^{-1}$ .

The Inner Tracker (ITk) is designed to withstand the challenging conditions caused by the increased luminosity. A higher resistance to radiation is essential to avoid damage in these circumstances. In addition, a higher granularity is required to maintain high tracking efficiency. The design chosen is a full silicon detector composed of two subsystems, as seen in Figure 3.1. The new Pixel design features several end-caps per side, allowing a pseudorapidity coverage up to  $\eta = 4$ . On the other hand, the strips offer coverage up to  $\eta = 2.7$ .

### 3. The Inner Tracker (ITk)

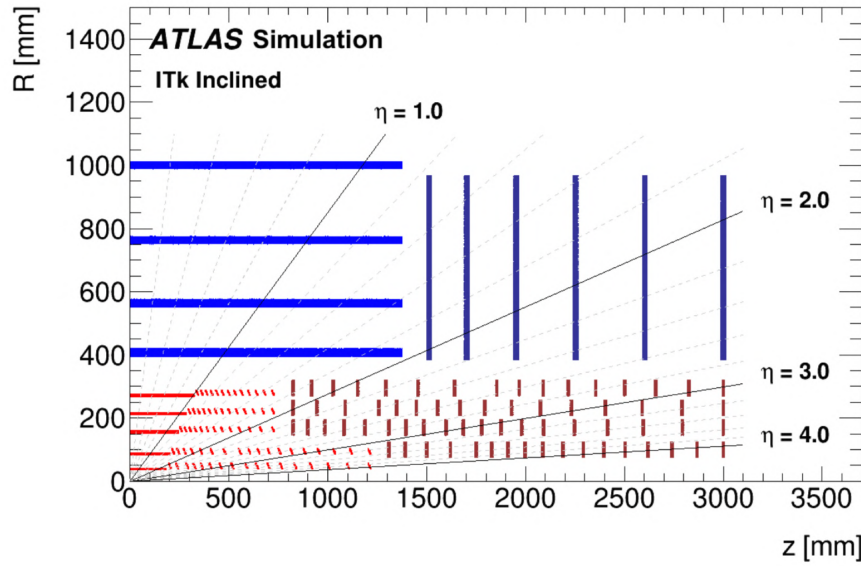


Figure 3.1.: Schematic layout of the ITk detector [3]. In red, the pixel detector and in blue, the strips.

#### 3.1. The Pixel Detector

The Pixel detector [17] is the closest detector to the beam line. For this reason, it poses a huge challenge in terms of radiation resistance and particle density. The pixel detector features 5 (barrel) layers, numbered 0 to 5 from the beamline outwards. Layer 0 and layer 1 will be replaced after half-life, i.e. after  $2000 \text{ fb}^{-1}$  of operation.

Different sensors are employed depending on the layer. Layer 0 employs 3D sensors. The design, seen in Figure 3.2, is intrinsically more radiation-hard than planar sensors. The short path between the point where the charges are created and the charge-collecting electrodes (columns) reduces the trapping from radiation-induced defects. To reduce the material budget but still maintain enough collected charge at the end of life, an active thickness of  $150 \mu\text{m}$  is achieved. Furthermore, the pixel size is reduced to  $50 \times 50 \mu\text{m}^2$  in the barrel and  $25 \times 100 \mu\text{m}^2$  in the end-caps, which is crucial given the high particle density.

The planar sensors are a very well-tested technology, and they will be employed in layers 1-4. They use the n in p technology with a  $p^+$  implant to guarantee an Ohmic contact of the bulk and the bias voltage. They will be fabricated with different active thicknesses, from  $100 \mu\text{m}$  to  $150 \mu\text{m}$ , depending on the location of the detector. The pixel size is set to  $50 \times 50 \mu\text{m}^2$ .

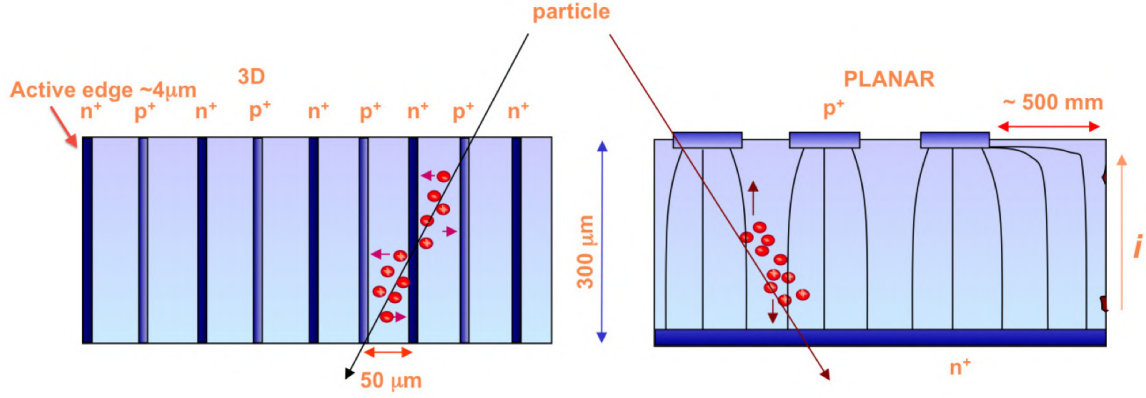


Figure 3.2.: Schematic view of electrode arrangement and charge collection in planar and 3D pixel sensors [17].

The pixel module is composed of three parts: the sensor, the readout chip and the flexible PCB. A sketch is available in Figure 3.3(b). The first two are bump-bonded together in a process called hybridisation, forming the bare module. The Front End (FE) chips are then wire-bonded to the flexible PCB, which provides the interface between the bare module and the power and data service cables. There are two different configurations of modules: quad modules and triplets [18]. As the names suggest, they consist of 4 and 3 readout chips bump-bonded to a single sensor. The quad modules are used in layers 1-4, i.e. in the outer barrel, in the outer end-cap rings and in layer 1 of the Inner System. The triplets are used in the innermost barrel layer. A view of a Quad Module can be found in Figure 3.3(a).

The readout chip was developed by the RD53 collaboration, which was established in 2013 to develop pixel readout chips for the Phase-2 upgrade of ATLAS and CMS [19]. The main requirements are radiation resistance, high hit rate, high data readout, and low power consumption ( $<1 \text{ W cm}^{-2}$ ).

In terms of features, the RD53 chips [20] are configured through a downlink, at a speed of 160 Mb/s. The data is read out via high-speed links using the Aurora 64b/66b encoding [21], with a maximum speed, using 1 lane, of 1.28 Gb/s per each FE (5.12 Gb/s for a quad module). Another key feature of the RD53 chip is serial powering, for the first time employed in a large-scale detector. It will be possible to power up to 16 modules in a serial chain, thus allowing to keep the material budget low. The core of the serial powering is the Shunt-LDO (SLDO) regulator, which provides a regulated supply voltage from a constant input current [22].

### 3. The Inner Tracker (ITk)

The pre-production chips, known as RD53B, were submitted in March 2020, tested, and returned for evaluation. The ATLAS version is referred to as ITkPixV1. The Production chips (RD53C/ITkPixV2) were submitted to the foundries in March 2023.

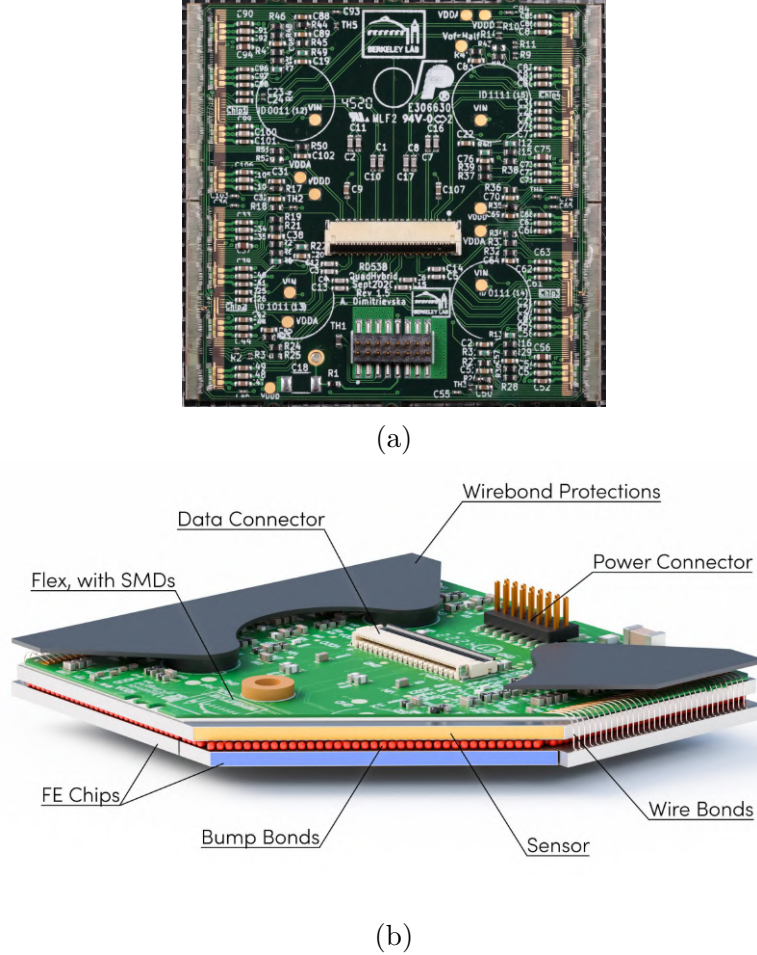


Figure 3.3.: (a) A picture of the ITkPixV1 Quad Module. (b) A scheme of an ITkPix outer barrel module. Drawn by Ruben Förster.

## 3.2. The Strip Detector

The silicon strip detector [3] for the ITk is situated just outside the pixel detector. The strip detector is equipped with 4 concentric barrel cylinders in the central region and with one end-cap per side in the forward region. Each end-cap contains 6 disks.

The fundamental unit of the ITk Strip is the silicon strip module. A module consists of one sensor and one or two PCBs containing part of the readout electronics. A view of one of the ITk strips module is provided in Figure 3.4. The design foresees eight

different shapes for the strip modules, six for the end-caps and two for the barrel region. In the central part of the barrel region, short strips (24.1 mm) are used due to the higher expected fluence. On the other hand, at higher pseudorapidity values, longer modules (48.2 mm) are employed. The strip pitch (75.5  $\mu\text{m}$ ) is uniform in the whole subsystem.

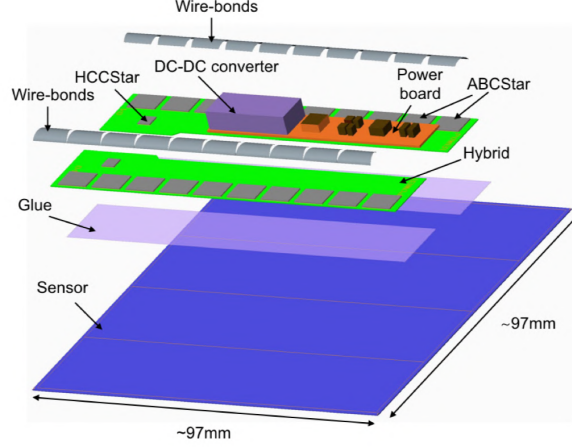


Figure 3.4.: Exploded view of a short-strip barrel module with all relevant components [3]. HCCStar and ABCStar are readout Application Specific Circuits (ASICs) employed in the ITk Strips.

### 3.3. The Local Supports

The Local Supports (LS) are carbon-based structures which are designed to house the services and to support the modules structurally. The modules are mounted on them, forming a fully functional detector unit called the Loaded Local Support (LLS).

The LS provide precise, stable, and rigid mounting for the modules during the experiment's operation; they ensure high-speed electrical connectivity between the modules and the on-detector electronics; they deliver the necessary cooling performance for the silicon modules; and they supply the required electrical power to the modules. All of these requirements should be accomplished while keeping the material budget as low as possible.

The fundamental design idea consists of a carbon-fibre-based, sandwich-like structure, which embeds a cooling pipe in the middle. In the next sections, the design for each subsystem is presented.

#### 3.3.1. The Pixel Local Supports

The LS have different shapes and (slightly) different designs depending on the area of the pixel detector [23] [17].

In the Outer Barrel (OB), which is relevant for this thesis, the LS are designed so that the individual modules are screwed into the support. This provides full flexibility for replacing individual modules, the downside is that an additional step during the sub-system integration is required. The material used to manufacture the LS is carbon-fibre-reinforced polymers (CFRP).

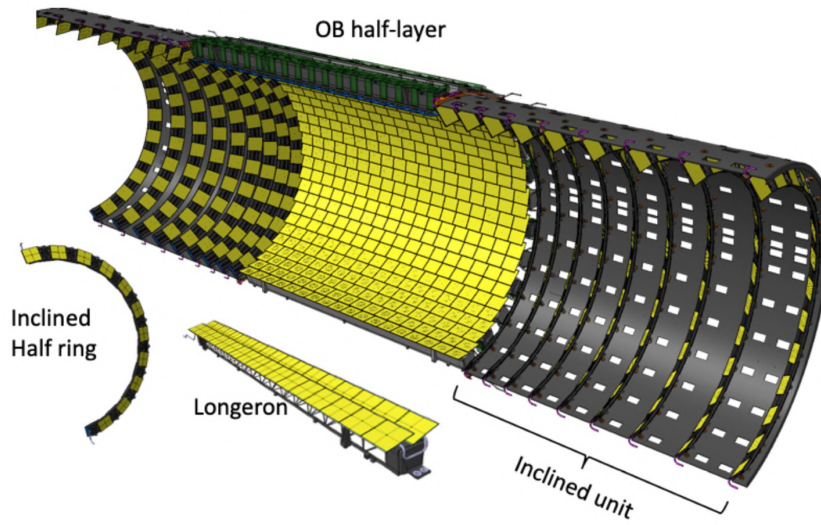


Figure 3.5.: The LS for the OB and a view of the OB half-layer [23].

There are two different LS for the OB: the Longeron and the Inclined Half Ring. The two geometries and a cut-view of the assembly are seen in Figure 3.5. The structure of the LS accommodates the cooling pipes as well as the services, such as the power cables and the data cables. Figure 3.6 shows a sketch of a Longeron as an example of this.

The Outer Endcaps (OEC) local support is seen in Figure 3.7. There are three concentric cylinders, which are filled with Half Rings. The Half Rings are manufactured in three different sizes, and they differ in the supports of the OB as the material employed is a mixture of carbon foam, pre-impregnated carbon fibre layers and plastic. The modules are loaded on both sides of the rings, with their position on one side corresponding to the space between modules on the other side. In this way, the full coverage is achieved, with a geometry which resembles that of a chessboard.

Lastly, the Inner System (IS) local support design is similar to that of the OB. The main difference is the material. Here, carbon fibre is co-cured with carbon foam layers

embedding the cooling pipe. In the barrel region, the support structures are Staves, which are elongated structures of different sizes for layers 0 and 1. As for the endcaps, the support structures are Rings, which come in three different flavours.

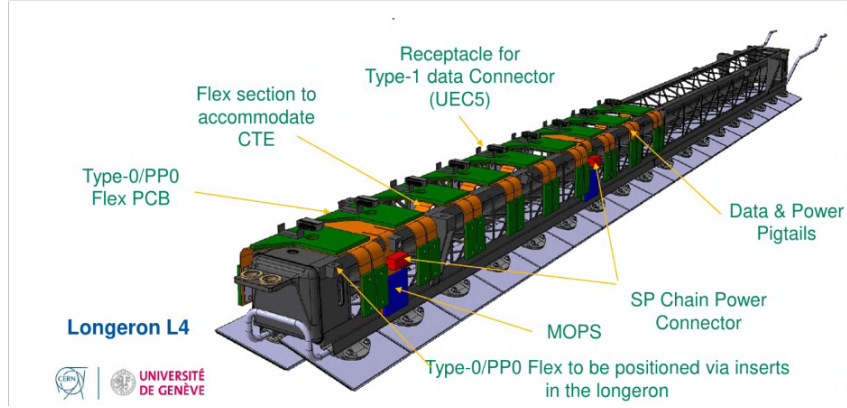


Figure 3.6.: View of a longeron.

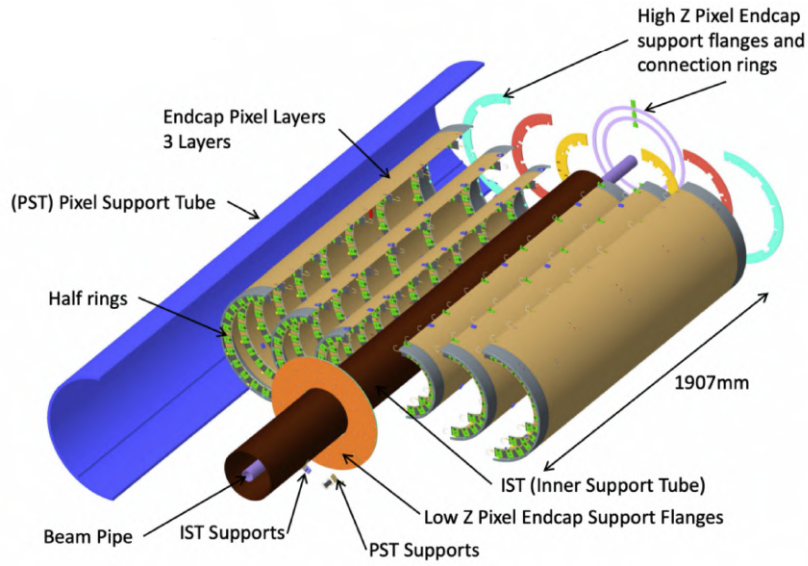


Figure 3.7.: Full layout of the Outer Endcap Local support [23].

### 3. The Inner Tracker (ITk)

#### 3.3.2. The Strips Local Supports

The most innovative side of the ITk strips is the mechanical structure, which greatly optimises the material budget [24]. There are two different LS for the barrel and the end-caps, a sketch can be seen in Figure 3.8.

In the barrel region, the detector layers are arranged in concentric cylinders. The LS in the barrel region are known as Staves, and they have a rectangular shape. Since there are 4 barrels in the strips, 4 flavours (i.e. dimensions) of staves are required. Each stave houses 28 strip modules, 14 per side. As for the end-caps, the detector layers are arranged in disks. The LS for this region have more or less the form of a circular sector, thus they are called Petals. There is only one flavour of petals, and each one houses 12 strip modules (6 per side).

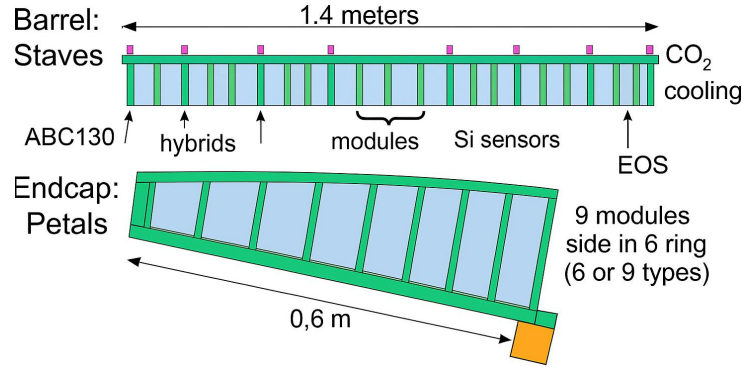


Figure 3.8.: Sketch of the strips Local supports: a stave and a petal.

#### 3.3.3. The Loaded Local Supports (LLS)

After the modules have undergone Quality Controls (QC), they are loaded on the Local Supports. In this process, the single modules are mechanically attached to the Carbon structures, and the electric and data wires, as well as the cooling pipes, are connected. There are specific coded procedures for each part of the detector.

As for the Outer Barrel [25], the cells are loaded manually by two operators, due to the complexity of the geometry. The LS is populated by many so-called quad module-loaded cells, interconnected with each other. For loading, the cells are picked up and placed onto the cooling block. The two are glued to each other, and the correct glue thickness is verified. The previous steps are performed very differently by different institutes/clusters, different loading methods, glue pattern and glue thickness assurance methods are used. Later on, the power and data cables are connected, and the LLS is ready to test.

The QC tests for the LLS are performed to check that the modules were not damaged

### *3.3. The Local Supports*

during loading. The tests are minimal, and they are expected to include only a functionality test for the digital and analogue circuitry (digital and analogue scan). These tests will be performed at the integration sites; the one for the German cluster is located in Bonn.



## 4. Data Acquisition for the ITk Pixel detector

The Data Acquisition (DAQ) system of the ITk pixel will read out around 10,000 modules during operation, summing up to approximately 5 billion pixels. To do so, the system will need to have an adequate bandwidth and low latency. The DAQ scheme is seen in Figure 4.1. The ITkPix modules (triplets or quad modules) are connected, through an adapter, to the Optoboard, where the signal is converted from electrical to optical. The resulting optical signal is elaborated by FELIX, which acts as an interface between the detector electronics and the DAQ. The actual readout is carried out on software, the one employed by the module QC and the Loaded Local Supports is YARR.

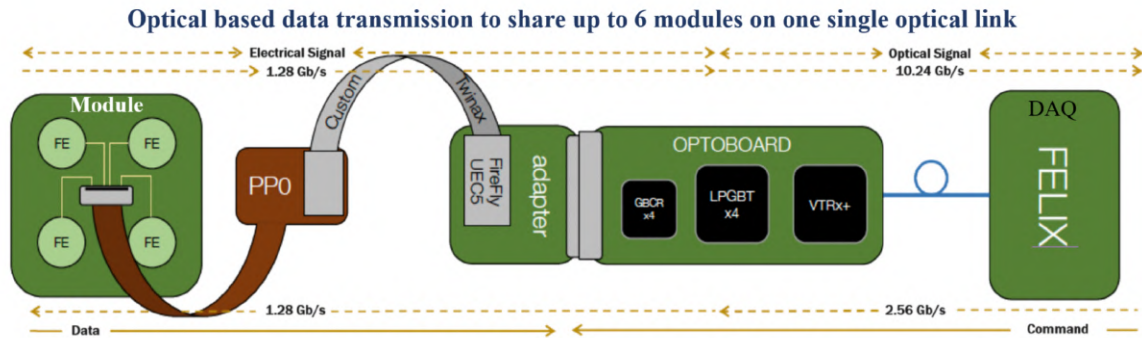


Figure 4.1.: Schematics of ITk Pixel Data Acquisition system [26].

### 4.1. The on-detector readout electronics

The on-detector electronics, as the name suggests, is placed inside the ATLAS cavern and has to withstand a high radiation dose, up to  $10^{16} \text{ n}_{\text{eq}}/\text{cm}^2$  at the HL-LHC. For this reason, Application Specific Integrated Circuits (ASICs) are required.

The Optoboard [27] is a PCB that contains the on-detector electronics; its overall role is to convert the signal from electrical to optical and vice versa. The aim is achieved

#### 4. Data Acquisition for the ITk Pixel detector

using three different ASICs: the lpGBT, which aggregates the data coming from the modules; the GBCR2, which enhances the electrical signal; and the VTRx+, which converts commands from electrical to optical. Different versions of the Optoboard house a different number of these chips; version 2.1 houses four lpGBTs, four GBCR2s, and a VTRx+.

The downlinks provide timing distribution, detector control and trigger distribution, from FELIX to the FEs. The uplinks transport acquired data to FELIX for off-detector processing and storage.

##### 4.1.1. The lpGBT

The low-power GigaBit Transceiver (lpGBT) is a high-speed, radiation-hard multiplexing chip designed in-house by CERN [28]. On the downlink side, the chip can de-multiplex a serial input link into many channels (e-links). On the uplink side, it can multiplex many e-links in a common uplink. To help the reader, the architecture of the lpGBT is provided in Figure 4.2.

The lpGBT features two sets of links: the high-speed electrical links from the back-end to the lpGBT and the electrical FE links from the lpGBT to the FE. The first ones connect the lpGBT to the VTRx+. The uplink speed is 5.12 Gb/s or 10.24 Gb/s, while the downlink speed is limited to 2.56 Gb/s. The links are designed to be asymmetric because the data rate is expected to be higher than the configuration/trigger rate. The second ones connect the lpGBT to the GBCR2. Each lpGBT can handle up to 6 uplinks at full speed (1.28 Gb/s). More links are available when the uplinks are operated at a lower speed. The downlinks are all handled by the master lpGBT, and the chosen speed for operation is 160 Mb/s.

The electrical FE link building blocks are called e-ports. The downlink ones (ePortTx) operate at a speed between 80 and 320 Mb/s. Depending on the data rate, a different number of e-links can be operated. For the 160 Mb/s speed, a maximum of 8 downlinks can be operated. Since the downlinks are operated through the master lpGBT, an optoboard with 4 lpGBTs will have 8 downlinks. An extra link is also present: the External Control (EC) output. This link is used for experiment control and lpGBT configuration. For the downlink signal, it is possible to set the driving strength, pre-emphasis, and polarity.

The ePortRx have numerous operation modes. Firstly, the number of groups depends on the uplink Forward Error Correction (FEC) code. FEC5 or FEC12 are available, allowing up to 5 and 12-bit errors, respectively. Secondly, the number of links per group depends on the high-speed link data rate and on the group data rate. The highest speed,

foreseen for ITk pixels, is reached using FEC-12, high-speed uplink at 10.24 Gb/s with 6 electrical uplinks (with individual speed of 1.28 Gb/s). The signal of these links can be controlled through equalisation settings.

The eClock is a frequency-programmable clock operating from the lpGBT to the FE. There are 29 eClocks available with frequencies of 40 – 1280 MHz, fitting all different data rates. The signal is tuned in the same way as for ePortTx.

The lpGBT can be configured in three different ways. It is always possible to configure the Optoboard using an I2C slave interface. As a second option, the IC channel via high-speed downlink is available, in the TRX (Transceiver) mode only. Lastly, the EC can be used to configure the Optoboard via the dedicated ePort, in RX (Receiver) and TX (Transmitter) mode only. EFuses, i.e. small fuses where the settings are permanently stored, are also present, but they have been found to be unreliable, so their use is not recommended.

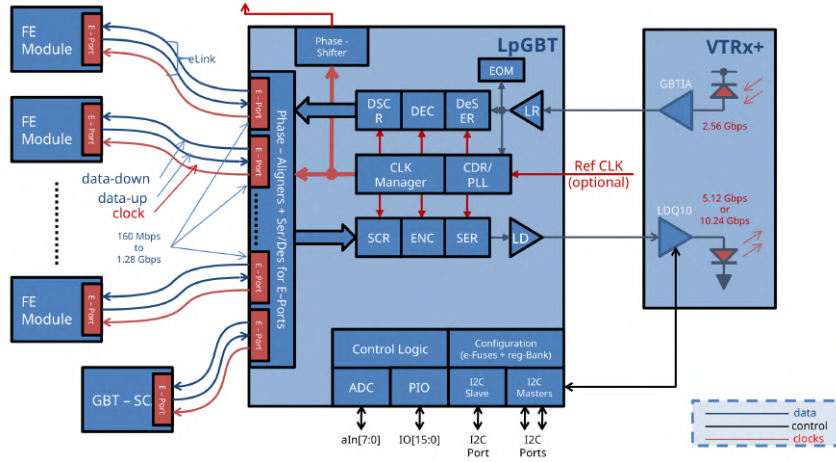


Figure 4.2.: lpGBT architecture [28].

#### 4.1.2. The GBCR2

The GBCR2 [29] is a chip that enhances the electrical signal between the lpGBT and the ITkPix FE chip. It is needed since the data is transmitted up to 6 m with Twinax cables and the flex, causing the loss of the high-frequency component of the signal. The distance between the lpGBT and the FE chip is not fixed and varies depending on the design, which must be carefully considered to achieve optimal performance. Therefore, GBCR2 is programmable. It hosts seven uplink channels and two downlink channels, and it is configured with the I2C slave interface, similar to the lpGBT.

As seen in Figure 4.3, the uplink channel consists of a passive attenuator, an equaliser,

#### 4. Data Acquisition for the ITk Pixel detector

re-timing logic, DC offset cancellation and a Current Mode Logic (CML) driver. The role of the passive attenuator is to reduce the intensity of the incoming signal in order to avoid distortion. In the equaliser lies the core functionality, as it compensates for the losses of the high-frequency components of the signal along the cables. The re-timing circuit uses the 1.28 GHz external clock and a phase shifter to recover the signal integrity. The DC offset cancellation eliminates the DC component of the signal. Lastly, the CML driver is used to transmit the high-speed data to the lpGBT.

The downlink driver has fewer components since the lower speed allows for a simpler design. It features a passive attenuator, a pre-emphasiser and a CML driver. The pre-emphasiser amplifies the high-frequency components of the signal, compensating for future signal losses.

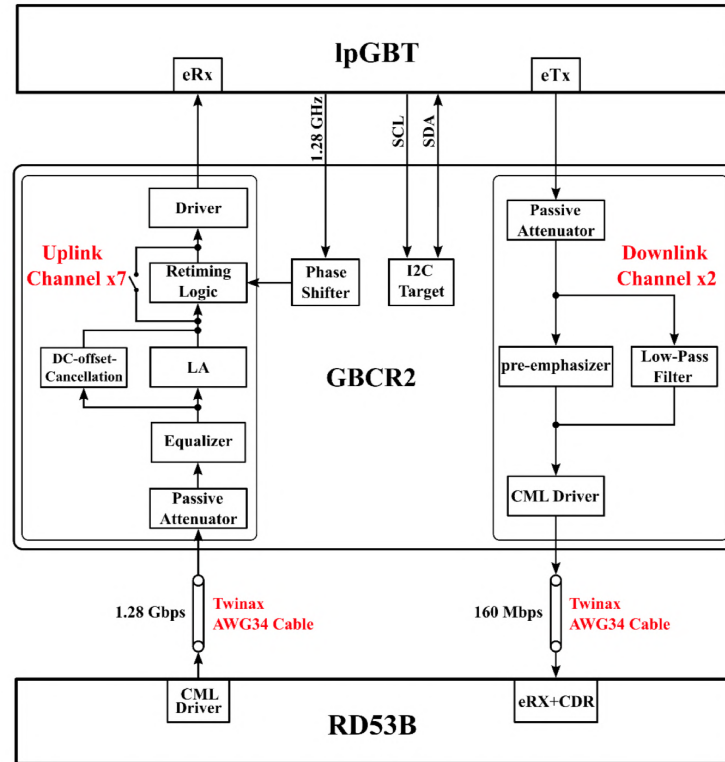


Figure 4.3.: GBCR2 architecture [30].

##### 4.1.3. The VTRx+

The Versatile Link Plus project is a development of the Versatile Link project [31]. The Versatile Link Plus Transceiver (VTRx+) [32] will be placed on-detector, which poses

some requirements. First of all, it will need to be radiation hard; the total dose is expected to be 1 MGy. Secondly, it will also need to operate over a wide range of temperatures ( $-35^{\circ}\text{C}$  to  $65^{\circ}\text{C}$ ). In the end, it will need to be a low-profile device to reduce the material budget. The data rate for operation is 10.24 Gb/s (reducible to 5.12 Gb/s) for the uplink and 2.56 Gb/s for the downlink.

The VTRx+ converts the electric signal into optical and vice versa; a schema is seen in Figure 4.4. It consists of the following sub-components. In the uplink channel, the Laser Diode Driver (LDD) performs the high-speed signal conditioning to properly bias and drive a VCSEL-type laser. There are a few versions of the VTRx+ available, featuring a different number of uplinks. Versions with an asymmetrical number of uplinks/downlinks are also available. In the downlink channel, the optical signal is converted using a PIN photodiode (PD). The electrical output signal is then amplified by the Transimpedance and Limiting Amplifier (TIA).

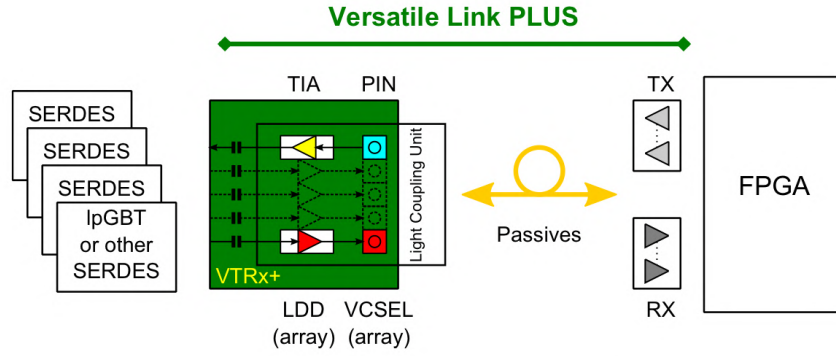


Figure 4.4.: Simplified block diagram of the VTRx+ architecture [32].

In the next lines, a brief description of the custom radiation-hard components of the VTRx+ is provided. A custom LDD [33] [34] was developed: redundancy of components (triplicated I2C components) and the use of 65 nm technology allowed to meet the radiation hardness requirements keeping the dimensions to  $1750 \times 400 \mu\text{m}^2$ . In addition, a programmable pre-emphasis is available to correct the asymmetry that the VCSELs typically exhibit for rising and falling edges.

The GBTIA [35] consists of a low-noise, high-bandwidth TIA and a high-performance Limiting Amplifier (LA). To achieve high transimpedance, high bandwidth and low noise, a differential cascode structure with series inductive peaking is employed in the TIA. The LA is composed of a cascade of four limiting amplifier stages followed by a  $50 \Omega$  output stage. Each limiting stage employs a modified Cherry-Hooper structure with resistive loading and active inductive peaking to enhance the bandwidth. Commercial PD and VCSEL were proven to meet the requirements.

## 4.2. The FELIX Project

The Front-End Link eXchange (FELIX) [36] [37] is an interface between the trigger/detector electronics and commodity-switched networks for the ATLAS experiment at CERN. It is designed to be detector agnostic. Thus, it will be employed in phase II for all the sub-detectors at ATLAS. The system is also designed to be modular, which allows for independent upgrade of specific aspects of the system (e.g. computing resources or network technology).

FELIX is employed for three main tasks. It serves as the interface to all detector-specific electronics using custom point-to-point serial links. It also acts as the interface for data handlers, monitoring, control, configuration, and the Detector Control System (DCS). In addition, FELIX receives information from the Timing Trigger and Control (TTC) and relays it to the FE electronics. In Figure 4.5, a possible architecture of the readout system is displayed. Changes are still possible as the project is still being developed. The FELIX project developed a hardware part with its firmware and a software part.

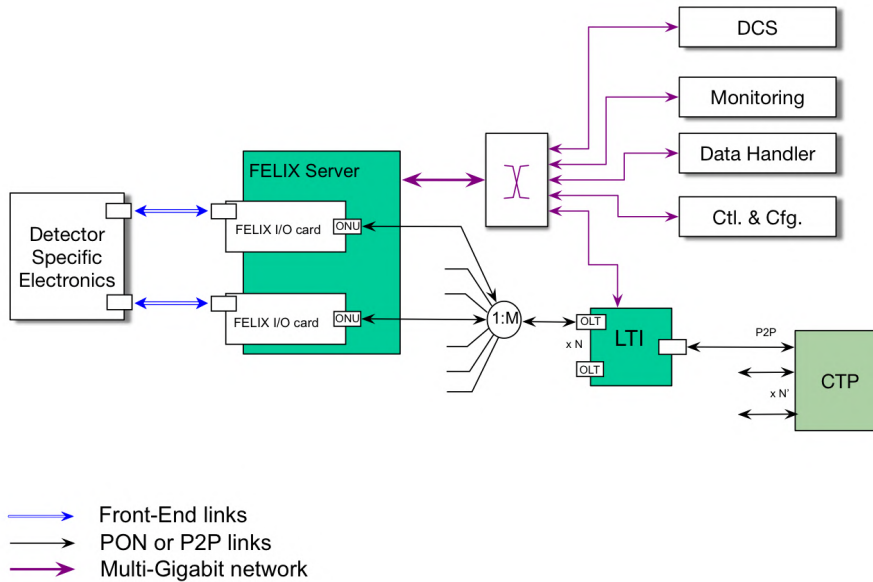


Figure 4.5.: Possible Readout system architecture for the HL-LHC upgrade [37].

### 4.2.1. FELIX hardware

The version of the FELIX hardware which will be employed for the entire integration QC is the FLX-712 card (FELIX I/O card), see Figure 4.6. In agreement with the flexible, reprogrammable philosophy, the FLX-712 is an FPGA-based card. It features a Xilinx Ultrascale FPGA with two 8-lane PCIe Gen 3 interfaces. The card is also equipped with an 8 Avagio Minipod transceiver for connecting the optical fibres from/to the detector electronics and a mezzanine card for the TTC link and the BUSY output. The BUSY system dynamically manages the data flux to avoid a system overload.

The system was tested, and the throughput of the PCIe interface is 100 Gb/s, while the optical links were characterised for 9.6 and 12.8 Gb/s. The card has 48 links, and the bit-error rate was measured to be less than  $10^{-15}$  for each of them.



Figure 4.6.: FELIX PCIe card, FLX-712 [36].

The FELIX firmware supports the GBT, lpGBT, and FULL mode protocols. The GBT protocol [38] is a high-speed, bidirectional communication protocol operating at up to 4.8 Gb/s, designed for reliable data transmission in noisy, radiation-prone environments. It features Reed-Solomon Forward Error Correction (FEC) and framing. The lpGBT protocol [39] is an evolution of the GBT one. It operates with a downlink speed of 2.56 Gb/s and an uplink speed of 5.12 Gb/s or 10.24 Gb/s. It enables all 48 links, while the other protocols use a 24-link version. The FULL mode is a custom 8b/10b encoded protocol operating at 9.6 Gb/s. Each sub-detector operates with its dedicated firmware. This thesis focuses on testing the ITkPix firmware, which enables the lpGBT protocol.

### 4.2.2. FELIX software

The FELIX software controls and configures the connected hardware and transfers the data from the detectors to the DAQ system. The FLX driver configures and controls the connected hardware (accesses the register map), while the *cmem\_rcc* driver reserves memory space to transfer large amounts of data. The data are accessed via DMA.

#### 4. Data Acquisition for the ITk Pixel detector

The FELIX-star application is the routing tool; it distributes the data between the buffers and subscribed applications. As seen in Figure 4.7, it is organised into three sub-applications: FELIX-tohost, FELIX-toflx (sometimes referred to as FELIX-fromhost) and FELIX-registers. The subscribed applications (e.g. YARR) use a library known as Net-IO-next to communicate with FELIX-star.

To control and configure the FELIX server, low-level tools are available [40]. Important commands are: **flx-init** command, which initialises the system; **flx-config**, which writes and reads registers; **flx-info**, which shows the system information and with the **elink** argument shows the lpGBT e-link alignment.

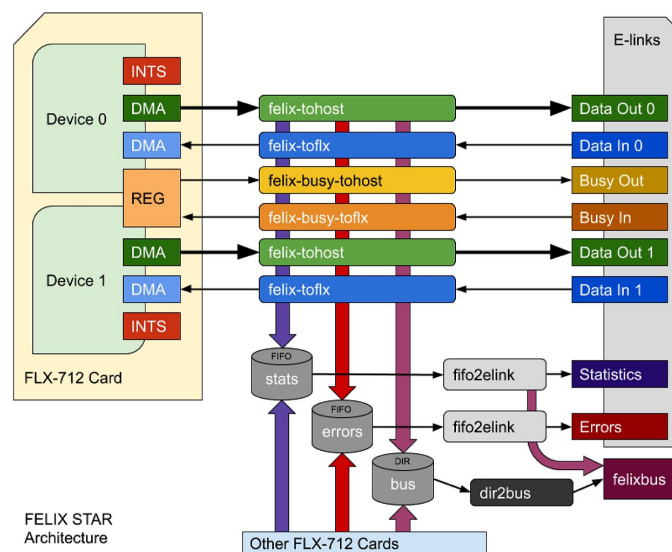


Figure 4.7.: Architecture of the felix-star data routing tool [36].

### 4.3. YARR: an ITk Pixel readout software

Yet Another Rapid Readout (YARR) [41] was one of the candidates for the final ITk DAQ system. In 2024, a decision was made, and a new software will be developed instead. Nevertheless, YARR was appointed to be used in the module and local support for electrical QC. The latter will be using YARR in combination with the FELIX FLX-712 card.

YARR features an innovative design compared to previous readout architectures since it performs the DAQ completely in software. To fully understand the novelty of the system, the reader can refer to Figure 4.8. The traditional architecture (Figure 4.8(a)) poses some

issues. Firstly, the Readout Drivers' (ROD) hardware is very specialised, requiring it to be custom-made. The software used by the ROD is deeply entangled with the firmware, which is updated less often than the software and whose programming language is hardly ever understood by the users. Secondly, and most importantly for scientists, the data is compressed by creating histograms. This process, while necessary due to the limited bandwidth between the ROD and the host computer, results in data loss.

YARR is deploying a different readout concept, as shown in Figure 4.8(b). The data processing and the scan execution are performed by the host computer. This is possible thanks to the PCIe FPGA card, which features a high-speed, low-latency connection. The FPGA firmware (FELIX firmware) is kept to the essentials, and it is detector-specific. It only contains the information for routing the data from the detector to the DAQ software.

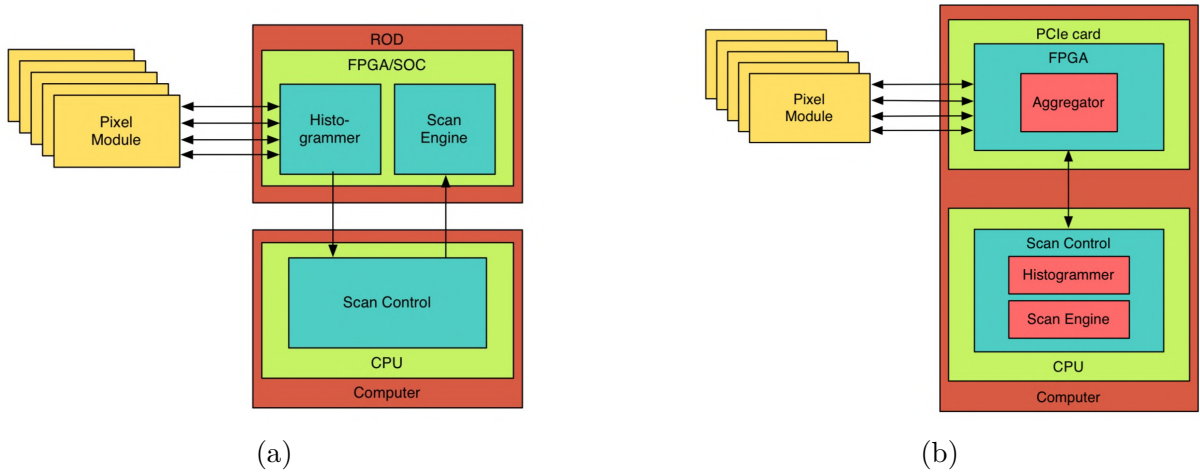


Figure 4.8.: (a) Traditionally used readout architecture (b) Readout architecture enabled by YARR [41].

The software was originally designed for a different hardware than the FELIX card. However, the advent of FELIX necessitated the enhancement and modification of YARR's functionalities. Consequently, ongoing efforts have been made to adjust and refine the software to ensure its compatibility with the new hardware, trying to meet the performance, functionality, and integration standards essential for the ITk pixel readout system.

#### 4.3.1. Design of the Scans

In order to better understand the working principle of a scan, it is necessary to first understand the general working principle of an FE chip. A drawing of its main components

#### 4. Data Acquisition for the ITk Pixel detector

is seen in Figure 4.9. The preamplifier amplifies the signal coming from the sensor (or internally generated). The shaper, as the name suggests, changes the shape of the signal, optimising the signal-to-noise ratio. The discriminator compares the incoming signal with a threshold value, digitising the incoming signal: 1 in case of a hit, and 0 if no hit is detected.

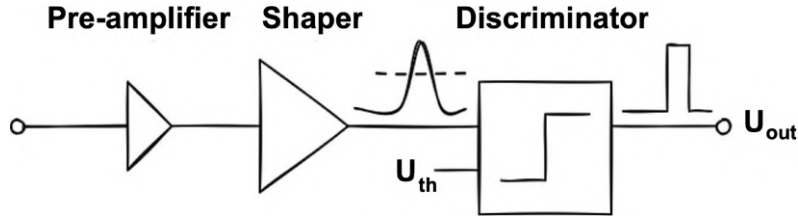


Figure 4.9.: A drawing of the general working principle of a FE chip. The conceptual source is the detector's book of Wormes and Kolanoski [42].

Although there are different scans available in YARR, they are all based on two fundamental ones: the digital and the analogue scan. An in-depth explanation of these two scans is necessary to understand the working principle of YARR.

The general structure of a scan, see Figure 4.10, is the following: there are three nested loops which are executed. The outermost loop is the masking loop. It is required since it would be impossible to read out the data from all the pixels due to bandwidth limitations. Therefore, each chip is divided into 64 parts, and each one is read out in a different step. In the second loop, some of the FE parameters are changed, depending on the type of scan. The innermost loop is the trigger loop; its frequency is set by default to 5kHz, and it can be increased by modifying the settings of the scan. The trigger loop sends a sequence of 16 triggers to each pixel 100 times, to simulate the data throughput during operation. If the chip is functional, 15 out of the 16 triggers are empty, and one contains the injected event. The occupancy map of a perfectly working FE sees 100 hits per pixel, and the total number of triggers sent is  $64 \cdot 5 \cdot 100 \cdot 16 = 512000$ . At the end, a loop for data collection is called, which collects all the pixels' data, synchronises the loops and checks that all the data is collected before passing to the next step.

One of the available scans in YARR is the digital scan. It is used in LLS and module QC, especially to test the communication between the module and the readout. It employs a calibration injection pulse that bypasses the analogue circuitry. The digital scan has 3 nested loops: the mask loop, the core column loop and the trigger and data collection loop. The core column loop changes the parameters EnCol0-EnCol3 present in the chip configuration. It enables five core columns at a time, looping from column 0 to column 50 (with a step of 5). In a digital scan, the input signal is generated by the FE chip itself,

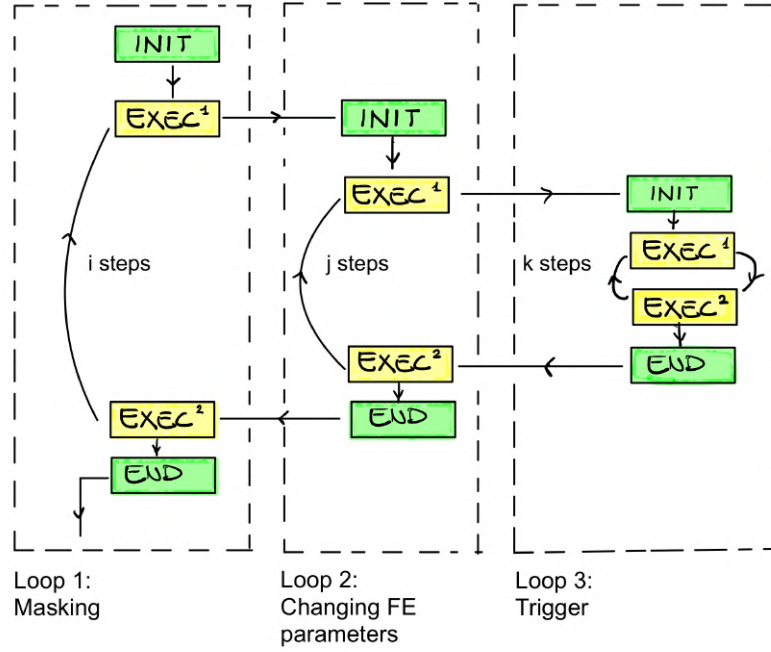


Figure 4.10.: The structure of a scan in YARR.

for test purposes. In the data collection loop, the digital output signal (coming from the discriminator) is read out. A successful digital scan shows an occupancy map with 100 hits for each pixel.

The structure of the analogue scan is the same as that of the digital scan. The difference is in the hit generation and in the data taking. Firstly, in the analogue scan, the hits are generated in the analogue circuitry. In addition, while the digital scan only reads the hit/no hit information, the analogue scan calculates the Time Over Threshold (TOT), based on the injected charge.

The other scans employed in this project are: noise, TOT, merged bump bonds, disconnected bump bonds and threshold scan.

The noise scan is used to find noisy pixels; it runs without any injection and checks the occupancy of each pixel. If the occupancy is greater than the threshold value, the pixel is marked as noisy.

The TOT scan has the same structure as the analogue scan. The difference is that while the analogue scan varies the injected charge, in the TOT scan, the injection charge is kept constant. In addition, as the name suggests, it generates a distribution of the TOT.

The merged bump bond scan is designed to identify bump bonds that are merged (e.g. due to a manufacturing issue). This scan uses the usual loops and checks the response of neighbouring pixels; a simultaneous occupancy of two neighbouring pixels can be an indication of a merged bump bond. In this scan, the FEs are read out one by one,

#### 4. Data Acquisition for the ITk Pixel detector

since the masking step happens in series. This is specific to the merged bump bond and disconnected bump scans.

The disconnected bump bond scan identifies a missing connection between the sensor and the FE chip. The structure is similar to that of the merged bump bond scan, but the goal of the analysis algorithm is to find pixels that do not respond to the trigger (occupancy 0).

Lastly, the threshold scan is designed to identify the injection threshold for each pixel. It can be seen as a series of analogue scans executed with different injection charges.

##### 4.3.2. Scan console

The Scan Console is the program used to run scans on the FE chips. It requires three configuration files. The first one is the controller file, which specifies the back end and its settings. Indeed, different FPGA cards can be used in YARR. The second one is a connectivity file: here, the FEs are listed with their addresses (RX and TX channels) and a link to each FE chip configuration file, which stores the FE chip registers. YARR can actively change this file during the scan, as it automatically tries to optimise the scan. Lastly, a scan definition is needed, which includes the type of scan that is performed, which is specific for each FE chip type.

```
[12:14:39:081][ info ][ ScanConsole ][839782]: Scan done!
[12:14:39:081][ info ][ ScanConsole ][839782]: Waiting for processors to finish ...
[12:14:39:082][ info ][ Rd53bDataProcessor ][839872]: [0x154b8] Finished raw data processor thread
[12:14:39:082][ info ][ Rd53bDataProcessor ][839875]: [0x15478] Finished raw data processor thread
[12:14:39:082][ info ][ Rd53bDataProcessor ][839875]: [0x15478] Unfinished streams (no EOS): 0
[12:14:39:082][ info ][ Rd53bDataProcessor ][839875]: [0x15478] Expect new stream with NS=0: 0
[12:14:39:082][ info ][ Rd53bDataProcessor ][839875]: [0x15478] Out-of-range bit requests: 0
[12:14:39:082][ info ][ Rd53bDataProcessor ][839872]: [0x154b8] Unfinished streams (no EOS): 0
[12:14:39:082][ info ][ Rd53bDataProcessor ][839872]: [0x154b8] Expect new stream with NS=0: 0
[12:14:39:082][ info ][ Rd53bDataProcessor ][839872]: [0x154b8] Out-of-range bit requests: 0
[12:14:39:082][ info ][ ScanConsole ][839782]: Processor done, waiting for histogrammer ...
[12:14:39:340][ info ][ HistogramAlgorithm ][839871]: Histogrammer done!
[12:14:39:347][ info ][ StdAnalysis ][839870]: [0][0x154b8] Total number of failing pixels: 0
[12:14:39:359][ info ][ StdAnalysis ][839870]: [0][0x154b8][0] ToT Mean = 7 +- 0
[12:14:39:365][ info ][ HistogramAlgorithm ][839874]: Histogrammer done!
[12:14:39:365][ info ][ ScanConsole ][839782]: Processor done, waiting for analysis ...
[12:14:39:372][ info ][ StdAnalysis ][839873]: [1][0x15478] Total number of failing pixels: 0
[12:14:39:384][ info ][ StdAnalysis ][839873]: [1][0x15478][0] ToT Mean = 7 +- 0
[12:14:39:587][ info ][ AnalysisAlgorithm ][839873]: Analysis done!
[12:14:39:765][ info ][ AnalysisAlgorithm ][839870]: Analysis done!
[12:14:39:765][ info ][ ScanConsole ][839782]: All done!
[12:14:39:765][ info ][ ScanConsole ][839782]: #####
[12:14:39:765][ info ][ ScanConsole ][839782]: ## Timing ##
[12:14:39:765][ info ][ ScanConsole ][839782]: #####
[12:14:39:765][ info ][ ScanConsole ][839782]: -> Configuration: 3928 ms
[12:14:39:765][ info ][ ScanConsole ][839782]: -> Scan: 5219 ms
[12:14:39:765][ info ][ ScanConsole ][839782]: -> Processing: 1 ms
[12:14:39:765][ info ][ ScanConsole ][839782]: -> Analysis: 682 ms
```

Figure 4.11.: Part of the screen logging of a YARR scan.

Once the program is called, providing all the necessary configuration files, the hardware

is initialised, the configuration files are loaded, and the scan is set up and run. During the scan, the mask stages are shown on the terminal video, and errors are also shown here in case of corrupted/incomplete data being received by YARR. At the end of the scan, plots are generated (on request) and YARR provides a summary of the timing of the scan (see Figure 4.11). This information is detailed for the configuration, scan, processing and analysis time. The configuration time refers to the period required to configure the FE chips. The scan time is the time needed to perform the scan - that is, to send triggers and acquire data through all the mask stages. The analysis time covers the processing of the output data and the creation of the final files. Finally, the processing time accounts for saving the results, including configuration files and any plots generated.



## 5. System Testing

The Loaded Local Supports (see Section 3.3.3) must be tested after assembly, which takes place at the integration sites; the German site is located in Bonn. In Bonn, the Outer Barrel LLS will undergo quality control to verify the functionality of all modules. During these tests, numerous modules are read out simultaneously (36 QMs for the longeron). Therefore, these tests are demanding in terms of bandwidth and require a fast read-out chain.

The objective of the first half of this thesis was to test the ITk Pixel DAQ chain for the LLS and to provide information about the timing of the YARR scans. A test of a fully populated optoboard (6 QMs) was performed, a preliminary test paving the way for a future test of the loaded longeron.

### 5.1. Test setups

#### 5.1.1. Local test setup

The test setup located at the II Physikalisches Institut at the University of Göttingen reproduces the DAQ chain for ITk Pixel LLS, which is explained in detail in Chapter 4. In Figure 5.1, a scheme of the setup is seen. The FELIX FLX-712 card is connected using optical fibres to the Optoboard version 2.1, which is equipped with 4 lpGBTs, 4 GBCRs and a VTRX+. It allows the connection of a maximum of 6 QMs (or 24 FEs). The Optoboard is connected to the FEs through electrical links, making use of adapters.

Two different layouts have been tested. The first one uses two adapters. The optoboard is first connected to an ERF-to-SMA adapter developed by the University of Bern, shown in Figure 5.2. Several SMA cables (two per link, one for each polarity) are then connected to an SMA-to-DP adapter. Finally, a DP cable connects to the module data adapter, where the pigtail is plugged in. This setup involves many adapters, which may make it difficult to handle the high-speed signals from the ITkPix modules. However, it allows flexibility in selecting which links to use. For example, each FE of a quad module could

## 5. System Testing

be read out by a different lpGBT.

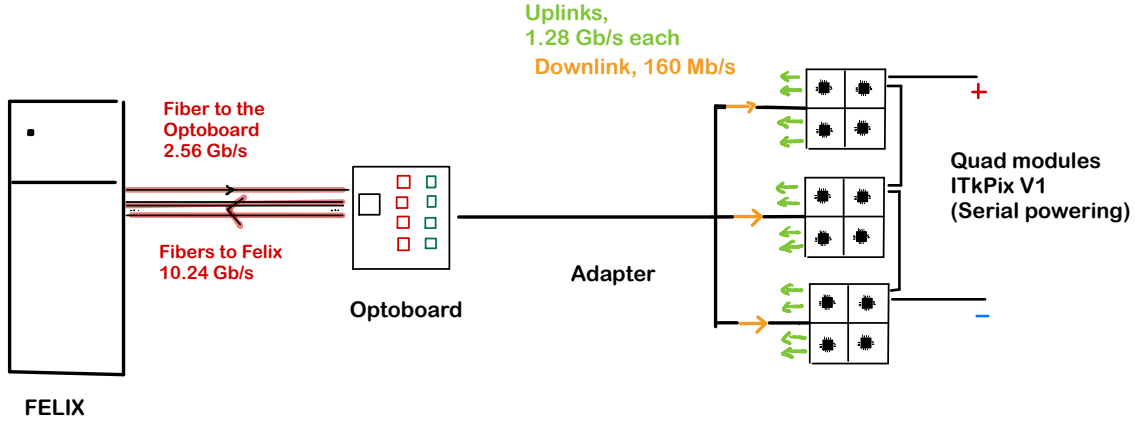


Figure 5.1.: A scheme of the laboratory setup.

In the second layout, the optoboard is inserted in the ERF to DP adapter, which was developed by Zaza Chubinidze (INFN) and is seen in Figure 5.3. A DP cable is connected directly to the module’s data adapter. This solution simplifies the setup and reduces the signal loss, at the cost of having only 6 DP port sockets with specific addresses. An additional advantage of this adapter is that it is well documented, also providing specific parameters to be set in the optoboard configuration file. This adapter has been more reliable and easier to operate in comparison to the ERF to SMA adapter. Therefore, it has been the default setup, the measurements presented in the next sections have been taken with this setup if it is not stated otherwise.

At the time of the tests, the ITk Pixel collaboration was still in pre-production. For this reason, 5 ITkPixV1 pre-production modules have been used, together with a digital module. The digital module is an ITkPixV1 module, which does not have a sensor. In the use case of this work, which focuses on the quality and the timing of the communication between the hardware and the read-out, it is indifferent whether or not a sensor is present.

In Figure 5.4, a picture of the setup with the DP to ERF adapter is seen. The modules are connected from one side to the power supply through the power adapter and from the other to the DAQ through the data adapter.

The setup employs two Techtronic PL303QMD-P devices, which are quad-mode dual power supplies. The range is 30 V, 3 A / 6 A. The first one is used to power up the Optoboard. The optoboard requires two input voltages: one at 2.5 V and the second one at 1.25 V. The second power supply is used to power up the modules. Each module needs to be powered at around 1.6 – 1.7 V, with a current of 5.88 A and more modules



Figure 5.2.: The ERF to SMA adapter.

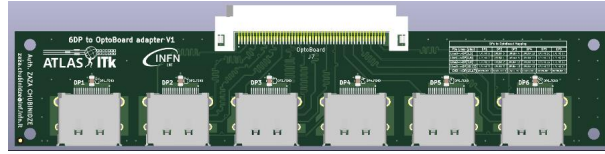


Figure 5.3.: A scheme of the ERF to DP adapter.

are powered in series. Both the Optoboard and the modules need to have the same ground. This is achieved by connecting the ground of the three outputs (2 outputs for the optoboard, 1 for the modules). At this point, to avoid ground loops, the ground of the module and that of the data must not be connected in the data adapters.

In this test setup, the cooling of the modules is provided by fans. The modules are attached to the heat sinks, and air is blown to provide cooling. The temperature is constantly checked, using an Arduino Mega and labRemote. The module power adapters are provided with a pin for the temperature sensor to be read out.

The version of YARR used in this chapter is version 1.5.2, with FELIX software version 5.0.4. In the next chapter, FELIX versions 5.0.4 and 5.1.4 (emulators) were used, using YARR v1.5.2/v1.5.5 as a reference for the new developments.

## 5. System Testing

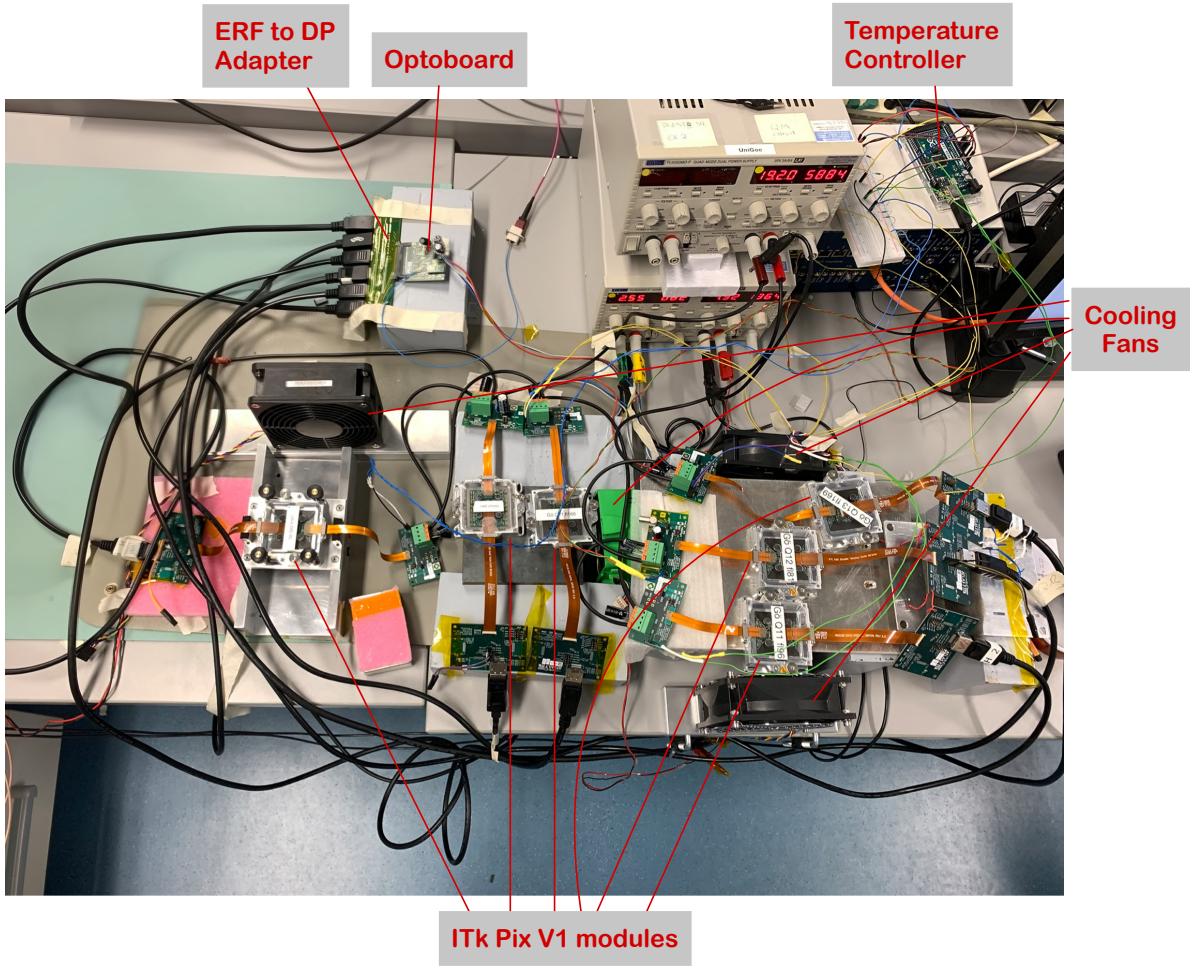


Figure 5.4.: A picture of the lab setup employing the ERF to DP adapter. Relevant pieces of the setup are labelled. The FELIX Hardware is connected through the Optoboard with optical fibres, and it is not visible as it is placed under the desk.

### 5.1.2. Test setup at CERN

Although most of the measurements were performed in the test setup described in the previous section, towards the end of the project, a smaller number of ITkPixV1 modules were available in the local setup. For this reason, some tests of a new software development, which is described in the next chapter, were performed at the CERN testing facilities for the ITk DAQ, located at SR1. The access to another setup also allows to cross-check the results.

The setup is similar to the one in Göttingen, it employs a FELIX FLX-712 card and an updated version of the Optoboard (version 4), with 4 lpGBTs and 4 GBCR2s. This setup makes use of a DP to ERF adapter connected to a serial powering chain with 4 ITkPixV1

modules, one of which with a non-functioning FE. An ITkPixV2 module is also available and powered separately. The core difference of this setup lies in the cooling, which uses a gas-cooled transfer plate, as seen in Figure 5.5.

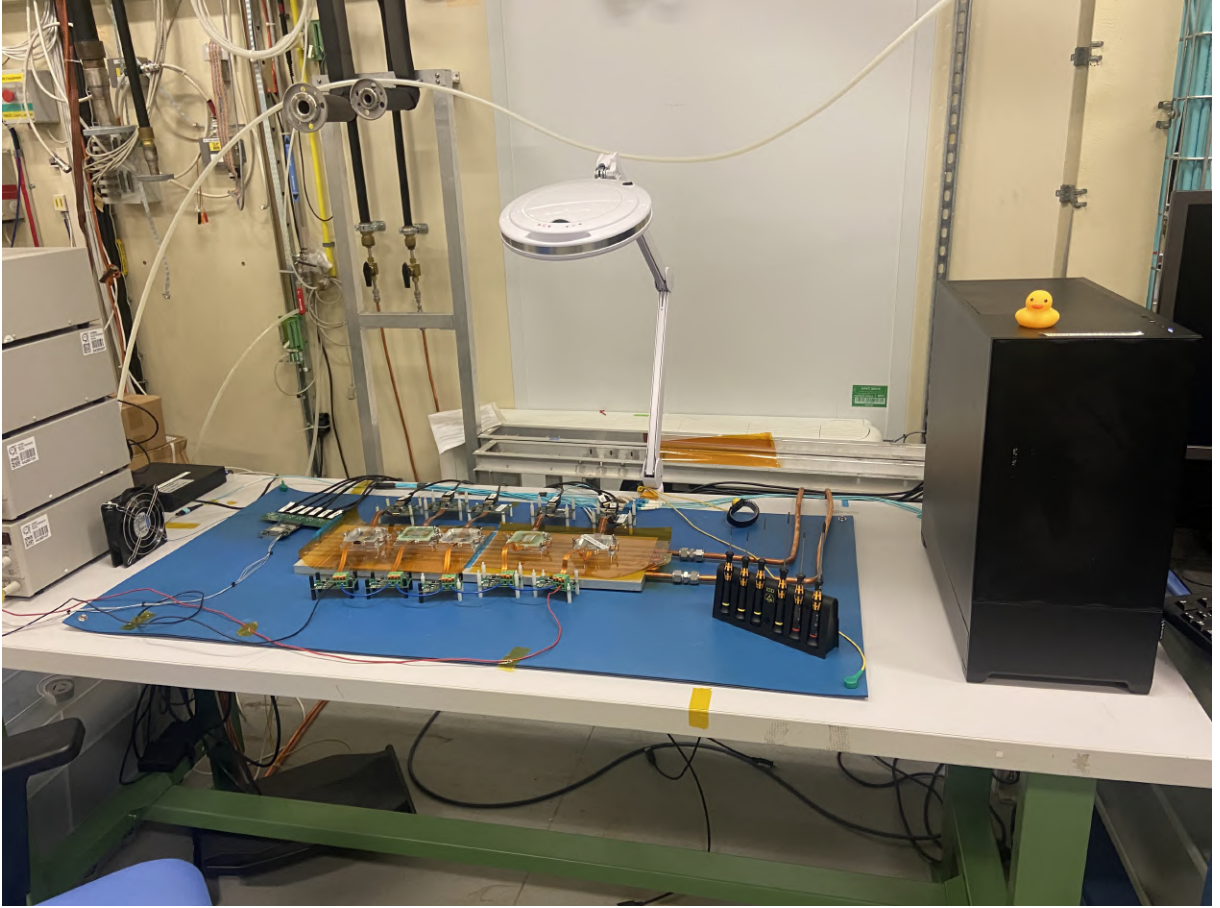


Figure 5.5.: The SR1 multi-module test facility, with 5 serial powered modules. At the moment of testing, there were 4 ITkPixV1 in the serial powering chain and an ITkPixV2 powered separately.

### 5.1.3. ITkPixV2 hardware emulator

In the context of the ITk detector, emulators play an essential role by allowing realistic simulations of hardware behaviour during system development and testing. The idea is to simulate the data throughput of the ITk pixel detector to stress-test specific parts of the read-out chain, such as the FEs or the ASICs. In fact at a QC stage there is a shortage of hardware parts as they are still under development or in production phase. However, the DAQ and FELIX need to be stress-tested to check their behaviour in real-like conditions. As a consequence the emulators are developed to allow researchers to perform these tests: the emulators send data similarly to the real hardware. In this way tests with a large

## 5. System Testing

number of chips can be simulated.

There are two types of emulators: software and hardware emulators. The first type reproduces in software the operation of a specific component, e.g. a FE. An example is the ITkPixV2 emulator available in YARR. The second type is hardware emulators, which are the ones employed in this thesis. They use FPGAs, which are programmed to reflect the behaviour of specific hardware chips.

A hardware emulator has been developed to stress test FELIX: it reproduces the FELIX/YARR read-out chain [43]. The FEs (RD53A) and the optoboard (lpGBT, GBCR and VTRX+) are emulated, carefully reproducing the data throughput during operation. This work has been further developed to feature the ITkPixV2 FE-chip. The new development employs FELIG, an emulator based on the FELIX FLX-712 card. To operate the setup, FELIG and FELIX are connected via optical fibres, see Figure 5.6. It is then possible to launch a scan through the scan console of YARR, in the same way as for a real FE (differences in the settings of the connectivity and FE chip apply).



Figure 5.6.: FELIG is connected to FELIX, allowing to use of the ITkPixV2 hardware emulator.

The emulator is useful to investigate the scan timing with a higher number of FEs, and in this thesis it has been employed to test a development of the YARR software, which is described in the next chapter. However, the testing performed with emulators is partial as it does not fully reproduce a real FE. For example, an FE can be configured with a wrong or incomplete configuration file, which would result in read-out errors. Using the emulators, the values of the configuration are read, but since they are not applied to a

real FE, the scan will succeed even if the set values are wrong.

## 5.2. Digital Scan of a Quad Module

The first natural step to test the LLS readout chain is to verify the functionality of a digital scan employing a single module.

The first step is to run the scan on each FE, verifying that it is correctly tuned and communicating with the DAQ. It has happened that pre-production modules successfully tested in module QC would not be read out by the FELIX/YARR setup. The reason lies in the fact that module QC uses a different setup (a different FPGA board and only electrical links), as well as different cables/cable lengths. After analysing the signal using a DP spy board and an oscilloscope, see Figure 5.7, it was clear that the signal was showing overshooting. The solution was found in tuning the CmlBias parameter of the ITkpix FE chip, which regulates the pre-emphasis of the signal. The default values (from the RD53B manual [19]) were reset.

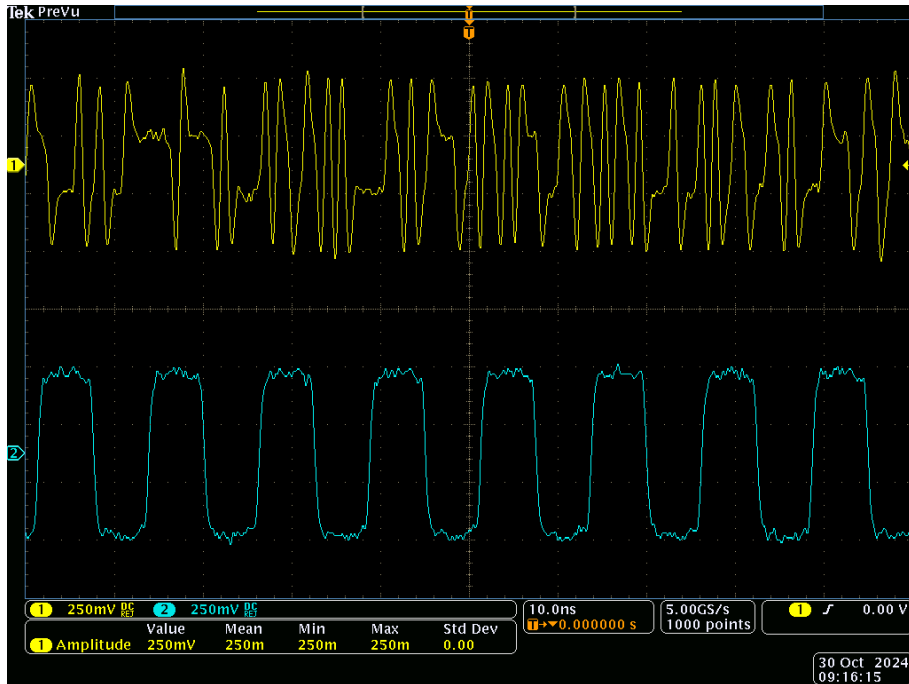


Figure 5.7.: An image taken with the oscilloscope, showing the signal of the data output of a FE (yellow line) and the configuration IDLE (blue line).

As a second step, more FEs are run together. This step would often be problematic, as the received data can be incomplete. In this thesis, the problem is referred to as

## 5. System Testing

*communication issues*, and it refers to a specific situation: when YARR returns an error indicating that the data is incomplete ("Expected new event while NS = 0"). This type of error can arise from various causes, such as incorrect chip configuration, improper module powering, or misconfiguration of the Optoboard. This problem is also visible in the resulting plots: in Figure 5.8, the occupancy map of a FE that faced *Communication issues* is seen; the diagonal pattern is sensible considering the loop structure of YARR, see Section 4.3.1.

After investigating the issue, the most effective solution was to adjust the *SldoTrimA* and *SldoTrimD* parameters of the FE chip. These parameters determine how each front-end is powered within the module. It is essential to set them to similar values across all four FEs in a module, as significant differences can lead to a power imbalance. This is because the FEs within a QM are powered in parallel.

Another source of *communication issues* is the core column issue, a known defect of the RD53B chips. In this case, the malfunction of a core-column (a group of 8 columns of the chip) would cause a general misbehaviour of the FE. The solution is to mask the affected core column.

Once the chip configurations have been set correctly for all four FEs, a successful digital scan can be achieved, i.e. all the enabled pixels have an occupancy of 100. The resulting occupancy maps, for a fully enabled FE and a FE with a disabled core column, are seen in Figure 5.9.

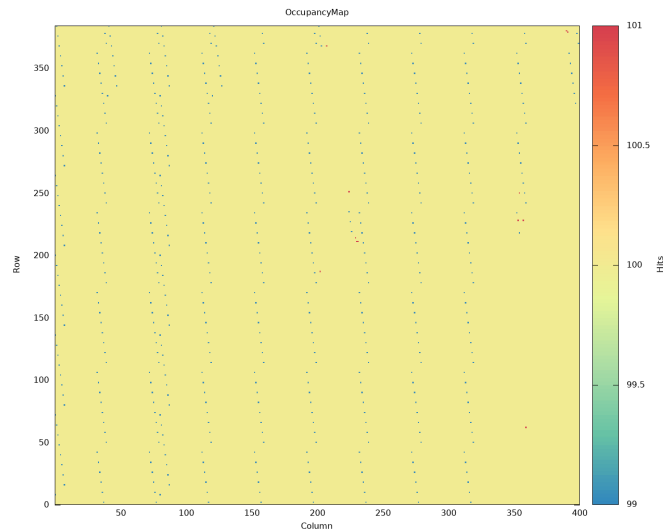


Figure 5.8.: The occupancy map of a FE, showing a diagonal pattern due to *communication issues*.

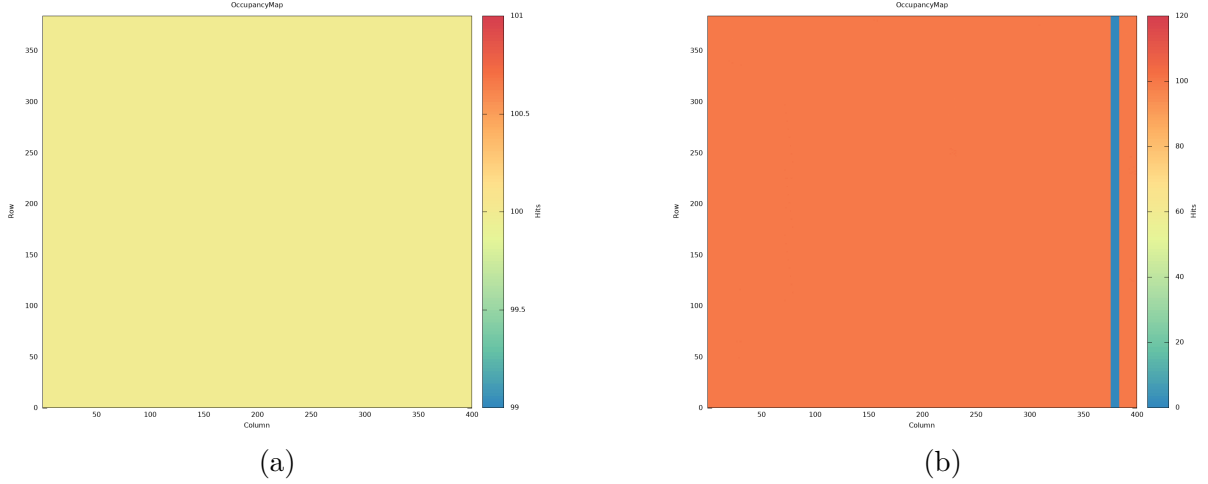


Figure 5.9.: (a) The occupancy map of a fully enabled FE. (b) The occupancy map of an FE with one disabled core column [41].

### 5.3. Digital Scan of Multiple Modules

There are two major issues to address when using more than one module. The first is serial powering. Although it is possible to power the modules from different power supplies, the choice to only use serial powering was made, as this is the projected powering mode inside the detector. The correct settings for serial powering are a current of 5.88 A, and a voltage of  $n(3.2\text{V})$ , where  $n$  is the number of modules serially powered. The value of voltage is setup-specific, as it has been verified that with this setting, a voltage of 1.6 – 1.7 V is measured on the power adapter of the modules. To improve the stability of the input current, capacitors have been put on the power adapters.

The second change that has to be made is the signal coupling. The Optoboard accepts both DC-coupled and AC-coupled signals. The latter is desired for operation in serial powering, as the DC offset will be different for each module. The AC coupling can be enabled via an Optoboard register.

Scans with an increasing number of modules have been performed. Up to 10 FEs, the digital scans were successful. By further increasing the number of enabled FEs, it was not possible to run a successful digital scan. The problem was identified in the trigger. YARR supports three types of triggers: the software trigger (default), the firmware trigger, and an external trigger (for example, TTC). In the software trigger, the trigger signal is generated inside YARR and sent to the FEs, while in the firmware trigger, the trigger signal is generated inside the FELIX card and sent to the FEs. The solution was to enable the firmware trigger with a higher frequency of 10 kHz (the default setting of the trigger

## 5. System Testing

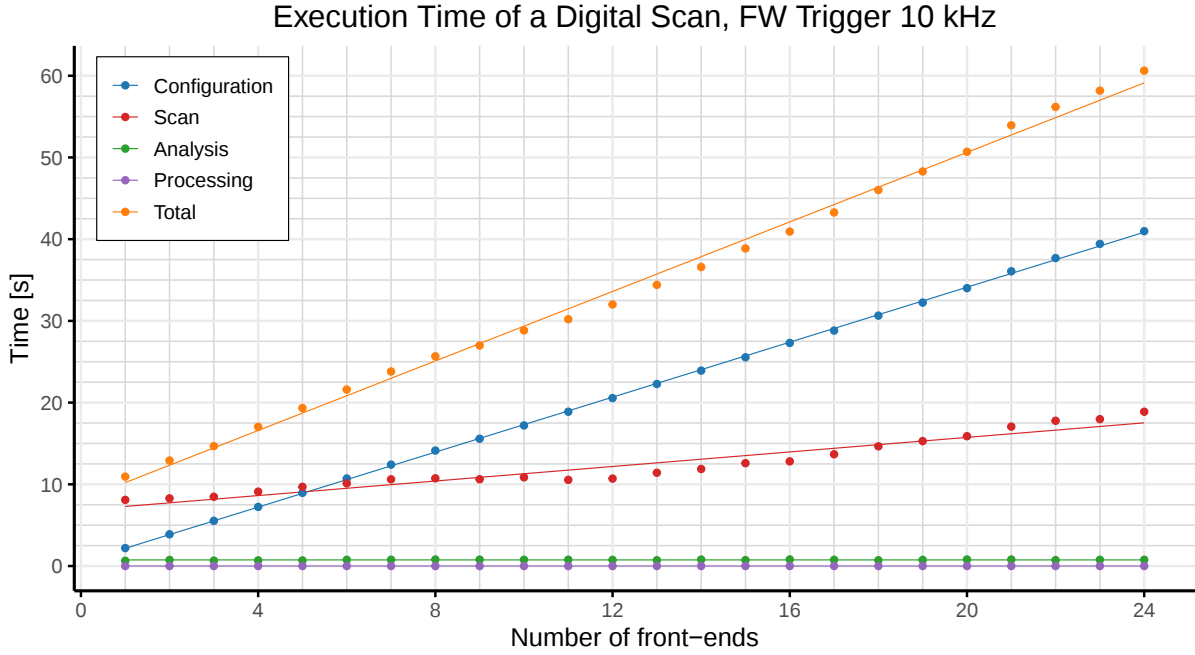


Figure 5.10.: The Execution time of a digital scan as a function of the number of enabled FEs.

is 5 kHz. This has solved the problem, most likely because of the lower delay compared to the software trigger, especially when many channels are enabled.

Eventually, six QMs were successfully read out, fully populating the Optoboard V 2.1. A digital scan was performed with a gradual increase in the number of enabled FEs, from 1 to 24 in steps of one. The timing is noted down for each step; the resulting plot is seen in Figure 5.10.

### 5.4. Further Scans of Multiple Modules

The digital scan is the natural point to start, as it is the simplest scan that can be performed. The next step is to check the correct behaviour of the DAQ software when performing other scans. The scans that have been performed are: analogue, disconnected bump bonds, merged bump bonds, noise, TOT, and threshold scans. A definition of the scans can be found in Section 4.3.1. The following issues have been observed: the threshold scan prints many errors, and although the scan can be concluded and the results are often not affected, the timing is not relevant any more. This is a software issue known to the YARR developers and users. Secondly, it has not been possible to perform a successful analogue scan with more than 9 FEs; a higher number of enabled FEs causes a stream of errors to be observed. This problem has not been fully understood, but it could

also explain the problem with the threshold scan, as a threshold scan can be seen as a sequence of analogue scans with different parameters. The other scans were successfully performed up to 24 FEs.

The plot of the timings as a function of the number of enabled FEs is seen in Figure 5.11, the y-axis scale is logarithmic. The full results can be found in the Appendix A. It is apparent that the merged and disconnected bump bond scans have a longer duration compared to the other. For these scans, the scan timings are calculated as  $n T_1$ , where  $n$  is the number of enabled FEs and  $T_1$  is the scan time for one FE. This is because the core column loop is not parallelised in this scan. As for the noise scan, the scan time component is constant (set to 60 s), and the increase in the total time is only due to the configuration time. The analogue, digital, and ToT timings are very similar to each other, which is consistent with expectations.

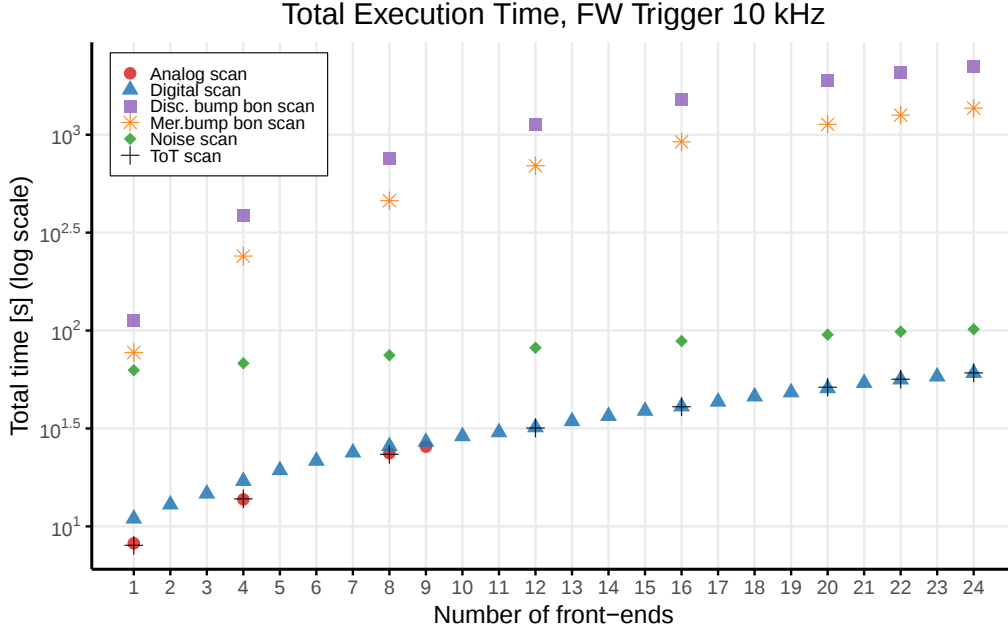


Figure 5.11.: The execution time of the scans as a function of the number of enabled FEs. Logarithmic scale.

## 5.5. Results using the ERF to SMA adapter

The ERF to SMA adapter has been employed in the early measurements, as the ERF to DP adapter was not yet available in the laboratory facilities. After performing the steps described in section 5.2, a successful digital scan was performed with the digital module and some of the preproduction modules. Subsequently, a multi-module setup was set and

## 5. System Testing

tested with a digital scan. It has been very difficult to reach a stable operation point with this layout. The plausible reasons for this behaviour seem to be two. The multiple adapters employed in this setup and the different lengths of the tracks and the cables could cause signal loss and, most importantly, the loss of synchronisation of the signal. In second place, some issues have been observed with the grounding, with the shielding of the adapter not being grounded.

An operation with up to 2 QMs (8 FEs), using the software trigger, was possible, despite it not being consistently reproducible. A higher number of enabled FEs would show *Communication issues*. The timing of the digital scan is seen in Figure 5.12.

Six other scans were tested on this setup with 1 QM: analogue, disconnected bump bonds, merged bump bonds, noise and threshold scans. The scans were performed, enabling one to four FEs and noting the timings. All the scans were successful and did not show any particular issue. The total execution scan time for each scan as a function of enabled FE is seen in Figure 5.13.

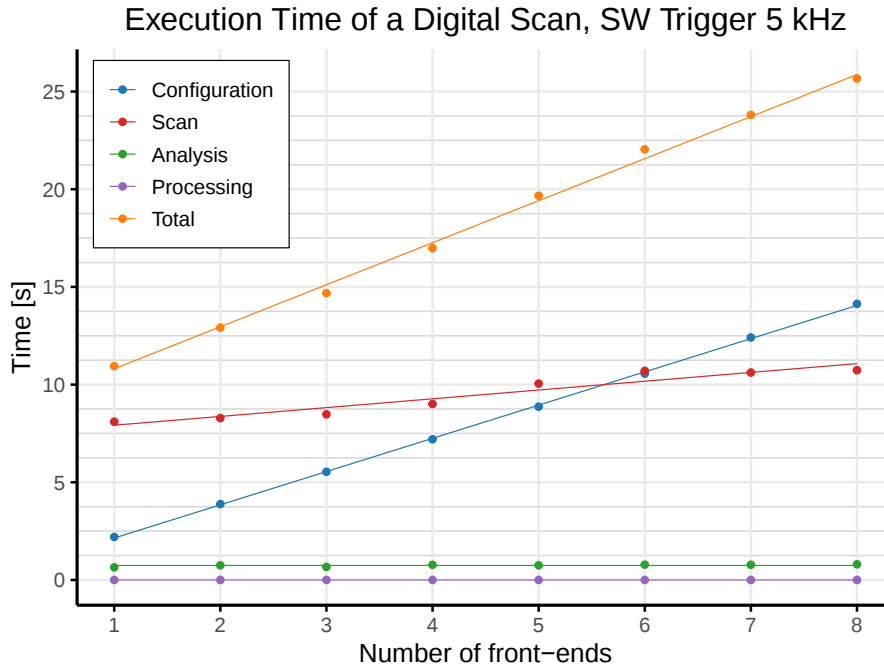


Figure 5.12.: The Execution time of a digital scan as a function of the number of enabled FEs.

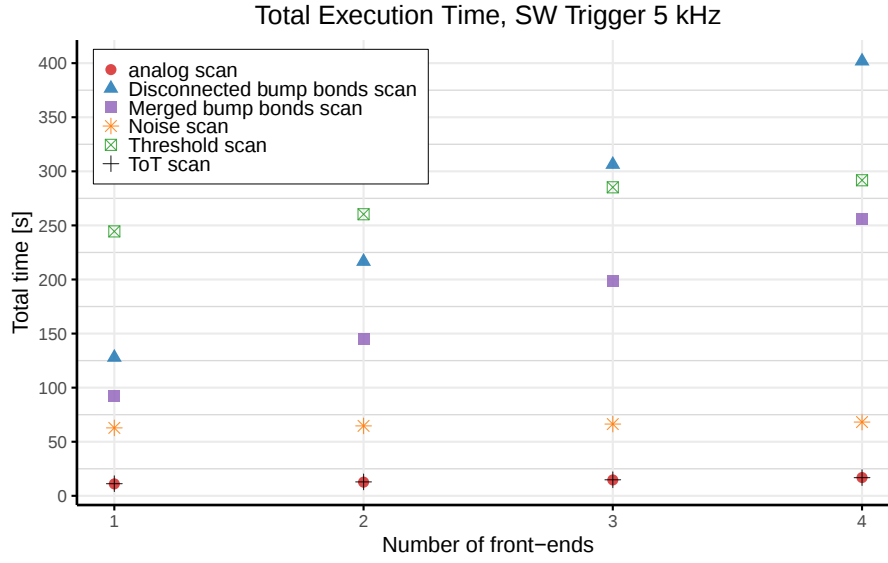


Figure 5.13.: The Total Execution time of analogue, disconnected bump bonds, merged bump bonds, threshold, ToT and noise scans as a function of the number of enabled FEs.

## 5.6. Estimating the error

The results shown in the previous sections are obtained by running the code once. Indeed, the timing of a YARR scan should not change if repeated in the same conditions. In order to verify this statement and estimate the order of magnitude of the error, a digital scan is repeated 5 times, and the results are compared. This was done, with 1 FE and 10 FEs, to check the dependence on the number of enabled FEs.

The results are shown in Table 5.1. Starting with the configuration time, the error increases with the increasing number of FEs; it is reasonable to assume that the FEs are individual objects with variable configuration times. The scan time is most affected by the increase in the number of FEs. In theory, these changes should not be observed as the scan procedure is the same for each of the FEs, and the information is broadcast. This might be explained by the uncontrolled timing interactions of all running computer task processes. The analysis time always shows a particularly broad variability in the results, but does not show an increase in variability with a higher number of enabled FEs. Lastly, the total time shows to even out the fluctuations of all the different components and to have a constant standard deviation of around 100 ms.

It is necessary to highlight that these are only estimations and that a sample standard deviation of 5 values is more subject to the effect of outlier points. Despite this, it is safe to say that the error (calculated as  $\frac{\sigma}{n-1}$ ) on each of the points is lower than 100 ms.

## 5. System Testing

Indeed, the highest error value was found to be the scan time error for 10 FEs, calculated to be 77 ms.

| n_Fe   | $\sigma$ Cfg. time [ms] | $\sigma$ Scan time [ms] | $\sigma$ Ana. time [ms] | $\sigma$ Tot. time [ms] |
|--------|-------------------------|-------------------------|-------------------------|-------------------------|
| 1 FE   | 7                       | 0.7                     | 97                      | 96                      |
| 10 FEs | 40                      | 154                     | 74                      | 97                      |

Table 5.1.: Sample standard deviation, from 5 measurements.

## 5.7. Comparison with previous results

This work is the natural continuation of the work which Wael Alkakhri did on the FELIX YARR DAQ system [44]. Therefore, it is important to compare the results. In Figure 5.14, the timing of a digital scan of 3 QMs (ITkPixV1), using 5 kHz software trigger, is seen. The scan is performed only up to 10 FEs, which is the maximum with this trigger setting.

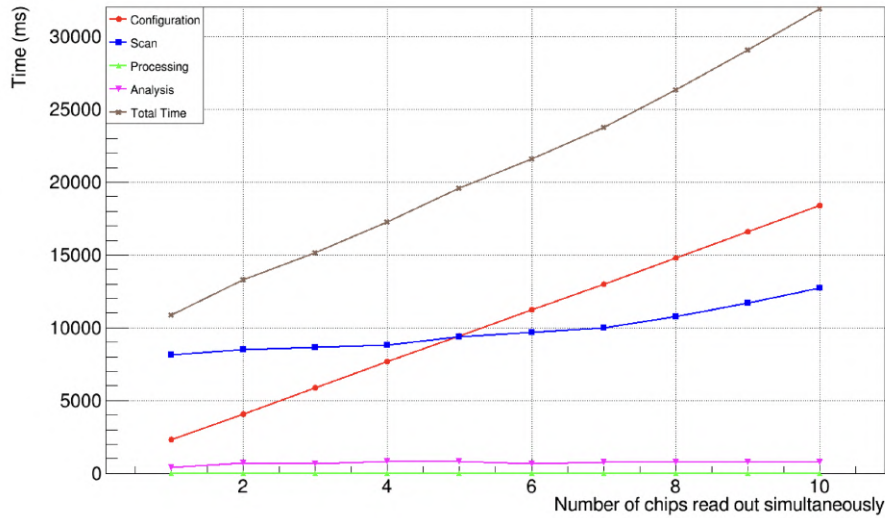


Figure 5.14.: The Time consumed by YARR processes as a function of the number of FEs under scan [44].

These results can be compared with the results obtained with the ERF to SMA adapter, which employs 5 kHz software trigger. Comparing Figure 5.14 and Figure 5.12, it is clear that they follow the same trend. The results are very similar, and the variations are probably due to slight differences in the setup or effects of the computing resources.

On the other hand, comparing Figure 5.14 and Figure 5.10, it can be noticed that the configuration timings follow a similar trend while the scan timings are slightly higher

in the published results. This meets expectations as the use of firmware trigger with a higher frequency is expected to reduce the scan timings, while it should not affect the configuration ones.

## 5.8. Considerations on the Scan Timings

A recent investigation carried out by Leander Teich, a Bachelor's Student in the AG Quadt, showed that the scan timing reported by YARR is not accurate. More in detail, it showed that the process timing reported by YARR is circa 30% lower than the actual one.

Despite this, the results presented in this thesis are still important and comparable with the work of other research groups, as the timings reported by YARR are usually the ones reported to the collaboration meetings. However, this is undoubtedly important news when scheduling the projected time for the LLS QC tests.



## 6. Software Development

### 6.1. Motivation

After successfully testing the ITk Pixel FELIX/YARR read-out chain, the need for an improvement in the scan timing became clear. Indeed, this read-out chain will be employed at the LLS. Two possible scenarios are under discussion: reading out a serial powering chain (12 QMs) or reading out an entire Longeron or Inclined Half Ring (up to 36 QMs). Based on the data collected from the digital scan timings, predictions can be made for the timings of a digital scan in the two projected scenarios.

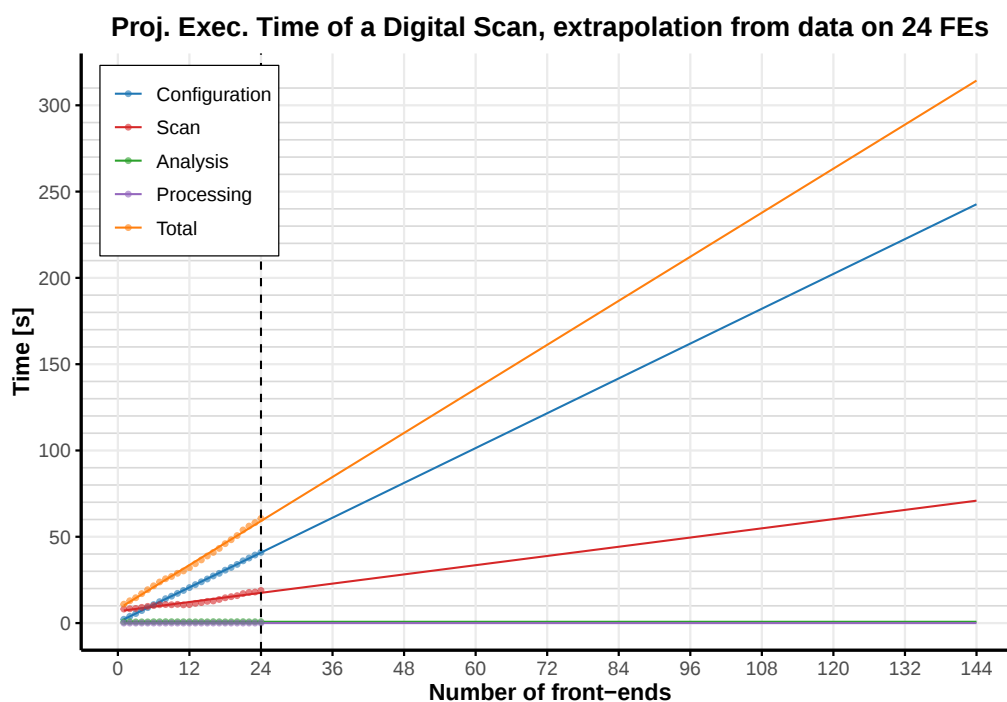


Figure 6.1.: Projected scan timings for a fully loaded longeron (36 QMs). To the left of the dashed line are the measured points for six ITkPixV1 modules (YARR v1.5.2); to the right, the values are extrapolated from the collected data.

## 6. Software Development

In Figure 6.1, the projected scan timings for a loaded longeron (36 QMs) are seen. The prediction is based on the data collected with the ITkPixV1 modules, which are reported in Section 5.3. The data points are fitted, and the fit lines are extended up to 144 FEs. This method only allows for a rough estimation as the extrapolation over a large interval is affected by sizeable errors. The configuration and scan time curves employ a linear fit, while the analysis and processing curves employ a constant one. The total time curve is the sum of the individual fits. The predicted total timings exceed 350 s, with the configuration making up roughly 70% of this time. Additionally, for a serial powering chain (12 QMs), the total time for a digital scan is expected to be around 115 s.

If the software is not updated, the QC testing at the LLS would take a very long time, thus conflicting with the schedule of the ITk project. Two paths are being followed. The first one is to fully parallelise the operation during the scan, to get the scan timings to be constant with respect to the number of enabled FEs. The second one is to parallelise the configuration of the modules (inside the module itself, the configuration must happen in series, as its FEs share the same TX link). An improvement on this side is very important and desired as it could eventually improve the total timings by up to 70%. This thesis focuses on the second approach, whose development and testing are described in the following sections.

## 6.2. Configuration Design in the Released Version

### 6.2.1. YARR controller libraries

YARR is a read-out software which was originally designed for the SPEC board. It was later expanded to be compatible with many types of read-out hardware (controllers) as well as many types of FEs. This means that the software is very flexible and it can be employed in many different setups. For example, in the module QC, the SPEC board is employed, while for the LLS, since the number of FEs will be very high, FELIX will be used.

YARR is a software written in the C++ language, using generic virtual classes, whose methods are then explicitly written in specific daughter classes. As for the controllers, the HwController is the generic virtual class. YARR supports the Spec, Bdaq and FelixClient controllers. Each of the controllers has its own library, which can be compiled when needed. The Felix Client library allows the use of YARR with FELIX via the NetIO library, a custom interface library developed by the FELIX collaboration. The

working principle of the Felix Client library is now briefly explained. Firstly, there is a FelixController object, which takes care of the initialisation of the registers and the communication with the FELIX card: the configuration of FELIX is set, and the client is created. The controller object contains two key components: the TX core and the RX core.

### 6.2.2. YARR Front-end libraries

YARR is designed to function with several types of FEs, including the chips that are currently employed in the IBL (FEI4) and the chips that are going to be employed in the ITk (RD53A, RD53B and ITkPixV2).

Similarly to the controllers, each FE has its own class, which is independent from the others: the software is developed to be FE agnostic, i.e. it functions regardless of the connected hardware. This means that any further improvement on the controller side should not modify the FE classes.

The FE classes are fairly complex, as the communication to the FE is the core of the whole software. Each FE has its own FE object, with its ID, and they are all managed by a bookkeeper. The bookkeeper orders them and allows access to specific ones. In addition, an object called *GlobalFrontEnd* is also available, which enables access and communication to all the FEs at the same time, which is useful when broadcasting a message.

Among other tasks, the FE class includes a data processor, which interprets the data received from FELIX and converts them into events; an encoder, which encodes the information for communicating to the FEs; an analysis tool, which analyses the data that is taken (e.g. plotting); and the command, loop, mapping and trigger tools, which are fundamental for completing a scan. More importantly for this project, the configuration of the FEs is also handled inside the FE class. The *configure()* method employs the TX core in the controller to send the data through FELIX to the FE chips.

### 6.2.3. Data flow in the configuration

The configuration of the FEs is done in series in the released version. The reason is the design, which was thought to maximise the broadcasting of information to the FEs. Although it is necessary to broadcast the trigger and other commands to all the FEs, the configuration is (mostly) FE-specific.

In order to better understand how the code was developed, it is necessary to explain briefly how the configuration is performed. In Figure 6.2, the 2 main objects relevant for

the configuration are seen. The first one is the FE-class object, which is one per each real FE. This object holds information about the connectivity (e.g. TX and RX addresses), has loaded the chip's configuration file and has a smart pointer to the FELIX controller. The second important object is the FELIX controller, which has two specialised parts: one for the communication to the module (TX core) and one for the data taking (RX core). The TX core is the relevant part for the configuration of the chips.

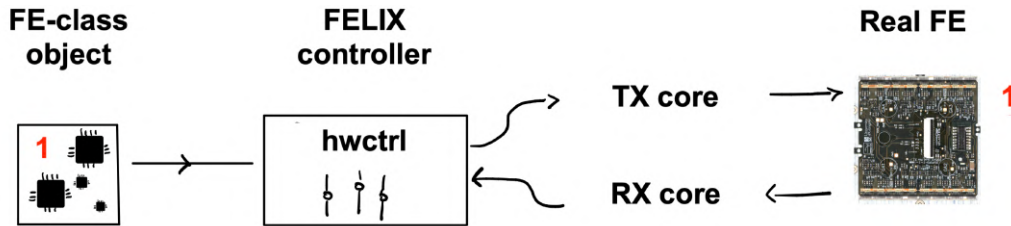


Figure 6.2.: Scheme of the data flow for the YARR configuration.

In the released version of YARR (v1.5.5), there is one TX core, which is responsible for the communication with all the modules. It is designed to have a single data buffer (FIFO) segmented for each TX channel. The FELIX TX core class also contains all the methods which are relevant to handling the FIFO. By using these methods, the information contained in the chip's config (and stored in the FE-class object) is written in the FIFO and sent to the FELIX Client. A scheme of the design of the TX core and its interaction with the FE class objects is seen in Figure 6.3. The FELIX logo represents the FELIX client, which is unique for all FEs and accessed through the FELIX core objects (both TX and RX share the same client).

Now that the general design has been outlined, some details on how the configuration is done are given. The configuration of the chip consists of sending information from the FE class object to the actual FE. This is effectively done using 3 methods: **WriteFifo**, which writes the value of the configs in the TX core buffer; **ReleaseFifo**, which releases the TX core buffer when full; and **SendFifo**, which sends the data to the FELIX client. These methods were developed with broadcast architecture in mind; therefore, the channel is not passed as input. Considering **WriteFifo**, whose implementation is seen in Listing 6.1, it can be seen that if the broadcast flag is on, the information is written to the broadcast channel, but if the broadcast flag is set to false, the information is sent anyway to all the enabled channels. The workaround the developers found to send each configuration file to the correct FE is to disable all the channels except the FE's one. In this way, the information is sent only to the correct channel, as it is the only active one. This comes at the cost of being able to configure only one FE at a time.

```

1 void FelixTxCore::writeFifo(uint32_t value)
2 {
3     if (m_broadcast && m_numEnabledChns > 1) {
4         auto fid_broadcast = fid_from_channel(BroadcastChn);
5         fillFifo(m_fifo[fid_broadcast], value);
6     } else {
7         for (auto& [chn, buffer] : m_fifo) {
8             if (m_enables[chn]) {
9                 fillFifo(buffer, value);
10            }
11        }
12    }
13 }

```

Listing 6.1: The writeFifo function implementation in YARR v1.5.5.

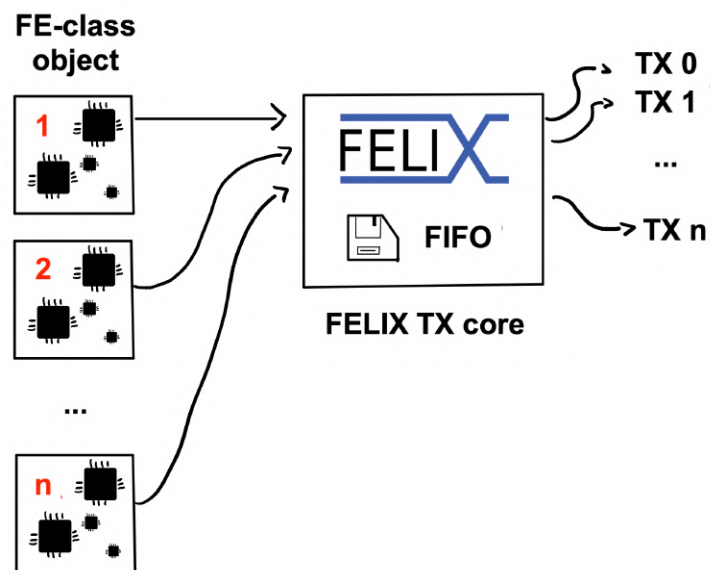


Figure 6.3.: Scheme of the design of the FELIX TX core object and its interaction with the FE class objects.

### 6.3. Parallel Multi-Module Configuration via TX core Wrapping

The first attempt to allow the configuration of multiple modules in parallel is to develop a sort of wrapper for the TX core object. The idea is to create  $n$  single TX core objects, one for each of the TX channels. All objects use the same FELIX Client, and they use the segmented FIFO of the FELIX TX core. A scheme of the design is seen in Figure 6.4.

Once the  $n$  `SingleChFelixTxCore` objects are created, they are assigned to each FE-Class object. The relevant methods (e.g. `WriteFifo` and `ReleaseFifo`) are overridden so that the information is only written to one specific channel. This mechanism works by storing the TX channel information in a private member variable of the class, together with a pointer to the FELIX TX core that holds the FIFO buffers. For example, as shown in Listing 6.2, the `WriteFifo` function retrieves the channel identifier from the `m_channel` member and writes the data directly into the FIFO managed by the FELIX TX core, accessed via the `m_tx` pointer.

```

1 void SingleChFelixTxCore::writeFifo(uint32_t value) {
2     auto fid = m_tx.fid_from_channel(m_channel);
3     if (m_tx.m_enables[fid]) {
4         m_tx.fillFifo(m_tx.m_fifo[fid], value);
5     } else {
6         sctlog->warn("Attempted to write to FIFO for disabled channel 0x
7             {:x}", fid);
8     }
9 }

```

Listing 6.2: The `writeFifo` function implementation in the `SingleChFelixTxCore` class.

The actual parallelisation of the configuration is achieved with a worker thread. The Listing 6.3 shows a simplified version of the worker thread: a queue is created with all modules, and it is accessed safely (using `std::mutex`). In this way, many modules can be configured in parallel; the number of threads is chosen to be the minimum between the hardware cores and the number of modules to configure. Inside the worker loop, the for loop reassures that the FE chips of the same module are configured in series, as they share the same physical TX channel. The `SingleChFelixTxCore` object is assigned to the FEs before entering the worker thread, so that the configuration is written to the correct FIFO channel of `FelixTxCore`. The process of sending the data to FELIX is performed in the same way as in the released version.

```

1  auto worker = [&]() {
2      while (true) {
3
4          {
5              std::unique_lock<std::mutex> lock(queueMutex);
6              module = chipQueue.front();
7              chipQueue.pop_front();
8          }
9
10         for (unsigned id : module) {
11             auto feCfg = bookie->getFeCfg(id);
12             bookie->getFe(id)->configure();
13         }
14     }
15 };

```

Listing 6.3: A simplified version of the worker thread.

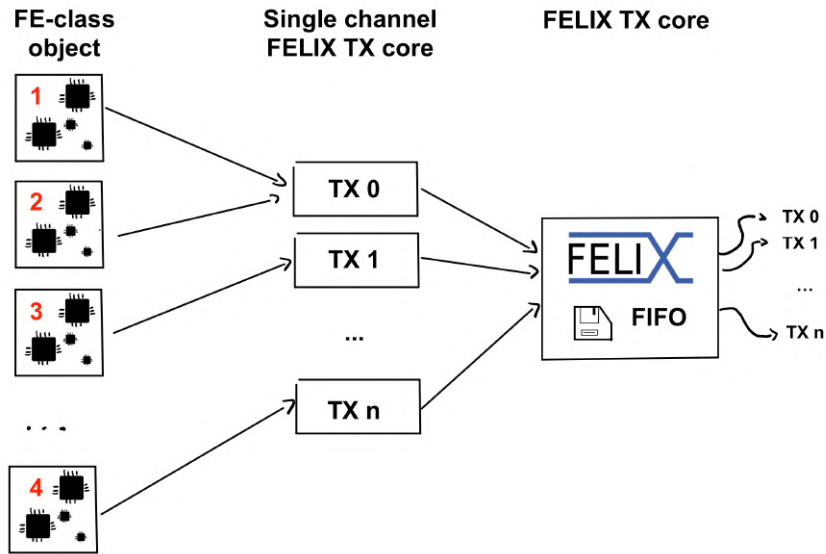


Figure 6.4.: Scheme of the design of the SingleChFelixTxCore object and its interaction with the FE class objects.

The new design was tested using the lab setup (with the ERF to DP adapter), and the results are seen in Figure 6.5. It is apparent that the configuration time ramps up to 4 FEs and then flattens, as expected.

Although the new design shows encouraging results on the configuration time side, it has shown problems when configuring only some FEs in a quad module: The reason for the missing points in Figure 6.5 is that it was not possible to reach a stable configuration

## 6. Software Development

for that specific number of FEs. The observed problems may be related to the fact that the `WriteFifo` function is parallelised in such a way that it only writes the data to the specific channel, but since the channels share the same FIFO and FELIX client, the `SendFifo` function is sending the data written to all channels. This may cause problems when the configuration of the FEs is not synchronised between all the FEs, as incomplete data could be sent.

However, the deviation of the points in the scan timing at 19 and 20 FEs may be due to the simultaneous use of resources on the PC that is running YARR. Unfortunately, it was not possible to repeat the measurements as the modules used were no longer available for testing.

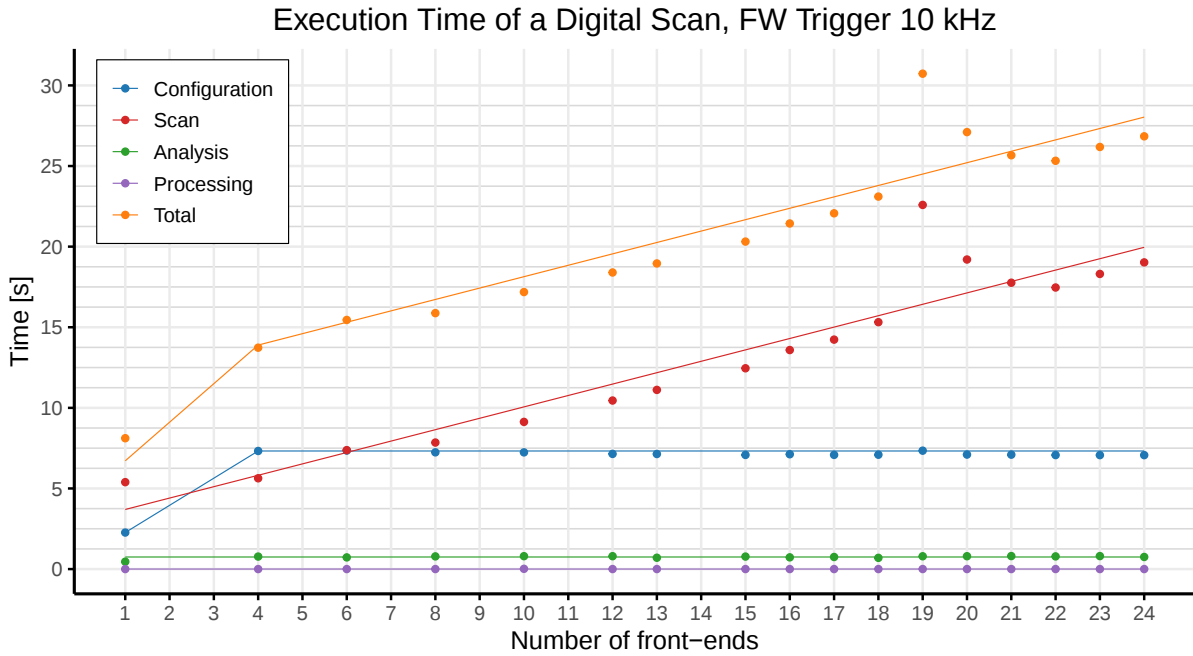


Figure 6.5.: Execution time of a digital scan as a function of the number of enabled FEs, using the `SingleChFelixTxCore` class.

## 6.4. Multi-client development

After the new design with the TX core wrapper was presented to the FELIX developers, they expressed some concerns about the possibility of handling numerous modules using only one FELIX client. They claimed that this could eventually create memory problems for FELIX.

The new design, seen in Figure 6.6, uses multiple FELIX Clients (4 by default). A new class called `FelixTxThread` is created, which is very similar to the `FelixTxCore` class. It has its own FELIX client and FIFO, as well as most of the methods implemented in `FelixTxCore`. The latter is not dropped, but rather used for the broadcast methods, the ones that employ the *GlobalFrontEnd* object; see Section 6.2.2.

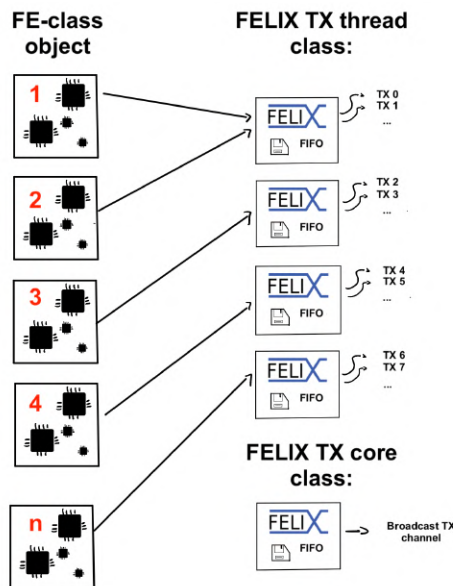


Figure 6.6.: Scheme of the new design employing multiple FELIX clients with their own FIFO.

On the other hand, the `FelixTxThread` objects are used during the configuration only. Each of these objects has a certain number of TX channels assigned, depending on the total number of enabled channels and the number of FELIX clients chosen by the user. The configuration is performed in a very similar way to the released version: All channels assigned to `FelixTxThread` are turned off except for the one of interest, and the methods write to the FIFO in the specific channel. The *SendFifo* method is now independent for each thread, sending the data to its own client, eliminating the risk of sending incomplete information. In addition, once the configuration of each FE is finished, the FIFO is sent and emptied. The parallel configuration uses a worker thread, as in Section 6.3. The

## 6. Software Development

number of threads is equal to the number of FELIX client instances.

The `FelixTxThread` and `FelixTxCore` are connected with each other: although they maintain independent FIFOs, the data is ultimately sent via the former, where the client is stored. This is made possible by the fact that the `FelixTxCore` class inherits from `FelixTxThread` and holds pointers to all the `FelixTxThread` objects created. In Listing 6.4, the `SendFifo` of `FelixTxCore` is seen, where the sending of the FIFO is delegated to the `FelixTxThread` object.

```
1 void FelixTxCore::sendFifo(FelixID_t fid, std::vector<uint8_t>& fifo) {  
2     prepareFifo(fifo);  
3     auto it = m_fid2thread.find(fid);  
4     if (it == m_fid2thread.end()) {  
5         ftlog->error("No FelixTxThread assigned for fid 0x{:x}", fid);  
6     } else {  
7         bool flush = true;  
8         it->second->send(fid, fifo.data(), fifo.size(), flush);  
9     }  
10  
11     fifo.clear();  
12 }
```

Listing 6.4: The `SendFifo` function implementation in the `FelixTxThread` class.

### 6.4.1. Tests with ITkPixV1 Quad Modules

This version of YARR was preliminarily tested using the local setup with two ITkPixV1 modules. The results were promising, and further tests with a larger number of modules were carried out at the CERN SR1 facilities, they are described in Section 5.5.

The results obtained at SR1 testing facilities are seen in Figure 6.7. The Multi-Client development proved to be more stable than the previous one, while still reducing the configuration time. If the results with the multi-client development are compared with the ones obtained with the TX-core wrapper (Section 6.3), the configuration times are comparable. Indeed, the effect of the parallel configuration of only 4 modules instead of 12 (given by the cores of the PC) would be only visible starting from 17 FEs.

Although the configuration of the chip is succeeding, some of the scans showed a peculiar behaviour in which the scan was completed successfully (no failing pixels and plots with 100 hits of occupancy), but a stream of errors would be shown during the scan. The same behaviour was also observed by other scientists working with YARR and FELIX after the release of YARR version 1.5.4. The conclusion is that these errors are not

connected with the multi-client development, but rather derive from the software update. The community is aware of this problem, but the reason is not yet clear. This issue is also visible in Figure 6.7, where the scan timings fluctuate around the fit line. The fluctuations can be explained by the fact that the streams of errors delay the scan and that this issue usually only happens for a specific number of FEs. In particular, the scans performed with 2, 8, 11, 12 and 13 FEs successfully completed the scans but with a high number of streams of errors during the scan phase, which explains the longer recorded scan timings.

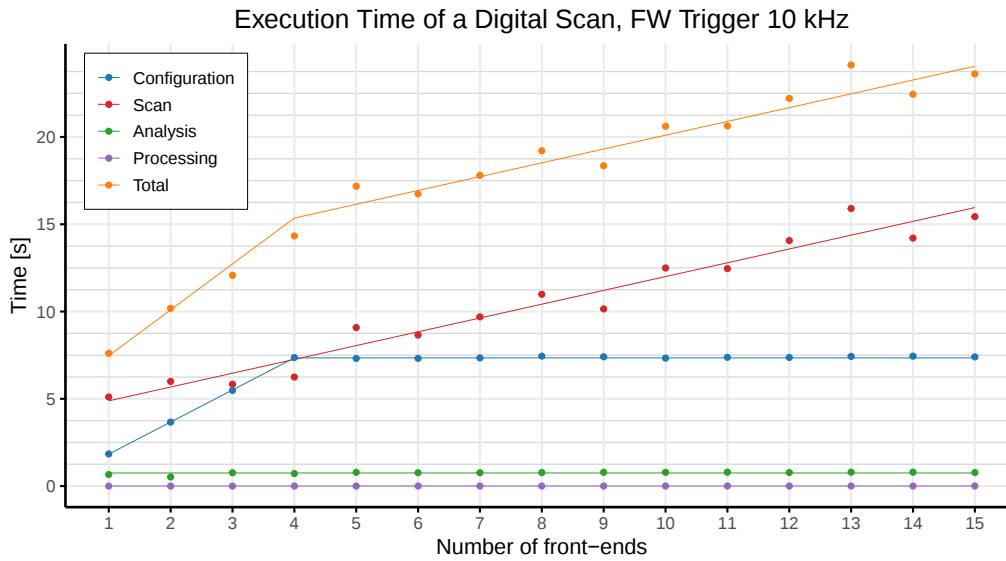


Figure 6.7.: Execution time of a digital scan as a function of the number of enabled FE chips, using the Multi-Client development. The measurement was performed on the SR1 setup. Preliminary results obtained with the local setup are shown in the Appendix, Figure A.6.

### 6.4.2. Tests with ITkPixV2 hardware emulator

In order to try to get results with a higher number of modules, the ITkPixV2 hardware emulator has been employed, its working principle is described in Section 5.1.3.

The emulators have been developed to have one FE for each TX channel. This situation does not correctly describe the real situation, where four FEs share the same TX channel, and they cannot be configured simultaneously. A workaround has been found by adding an FE, with the same name and address, multiple times in the connectivity file.

In the upper part of Figure 6.8, the results are presented, where it is evident that the configuration happens in parallel. The modules are configured in groups of 4; thus, it is possible to see an increase in configuration time once 4 modules (i.e. 16 FEs) are

## 6. Software Development

configured. This is consistently visible across the plot, except for the interval between 68 and 72 FEs. This is most likely due to the wrong allocation of the TX channels.

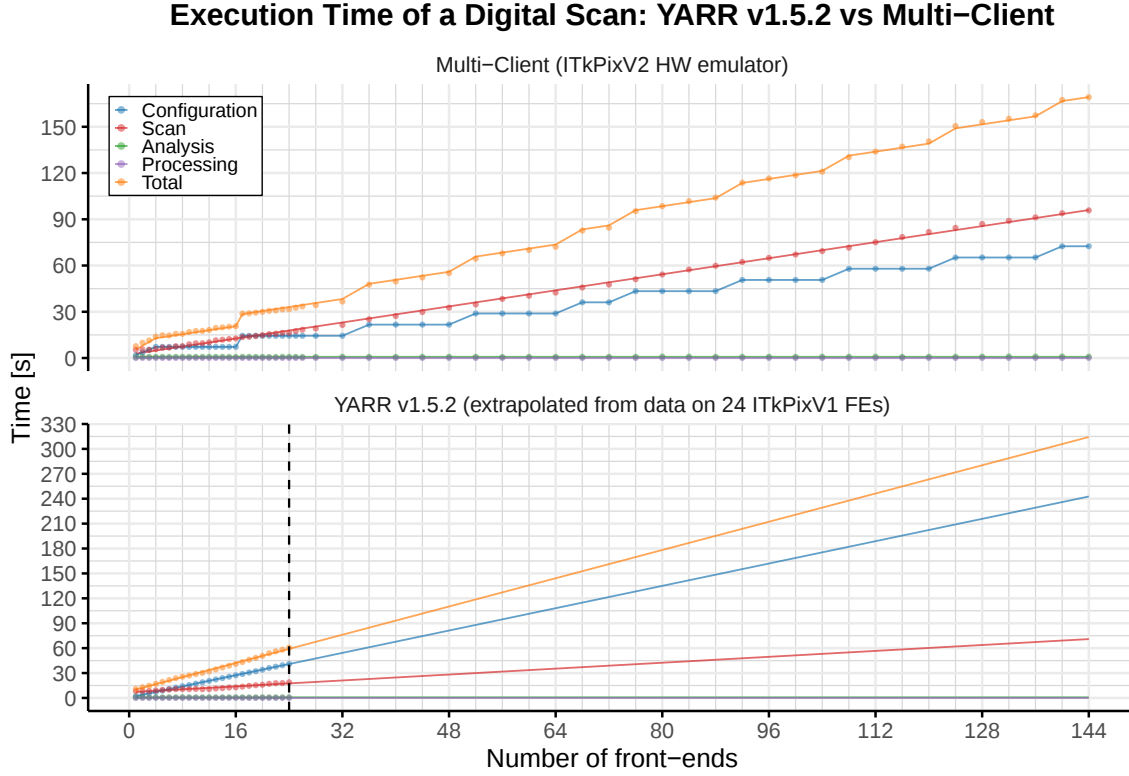


Figure 6.8.: Comparison of the timings for a digital scan using YARR v1.5.2 (projected) and the Multi-Client development (ITkPixV2 Hardware Emulator). The projected times for YARR v1.5.2 are extrapolated from data collected with 24 ITkPixV1 FE chips; the dashed line on the lower graph separates the extrapolated region from the measured points.

It is now possible to compare the new development with the projected timings for the released version. First of all, it is necessary to highlight that the timing for YARR v1.5.2 is obtained with an extrapolation starting from the data collected on 6 ITkPixV1 modules. On the other hand, the data available for the multi-client development was obtained using the ITkPixV2 Hardware emulator. Although it would be better to compare the results of the same version of the chip, this is not possible because there are not enough ITkPixV2 modules to perform a large-scale read-out chain testing, and the ITkPixV1 emulators were not developed. However, the ItkPixV1 and ItkPixV2 FE-chips are similar to each other, with minor changes made between the two versions, which should not affect the timings.

Comparing the timings in Figure 6.8, it is possible to see that the configuration times are reduced to 73s in the multi-client development, while the released version is projected

to take 242s for the configuration, i.e. the new development is roughly 70% faster. In addition, this result can be further improved by increasing the number of FELIX client instances, which is now set to four.

Comparing the scan timings, which should be the same for the multi-client development and the released version, it can be seen that they are lower for the multi-module development, taken with the emulators. If the measured data of both sets are compared, i.e. up to 24 FEs, on the left side of the dashed line, the data follow the same trend. The deviation only shows up at a higher number of FEs, and it is compatible with the fact that extrapolating the fit for such a large range is affected by a sizeable error.

Lastly, the analysis and processing time, which are hardly visible in the lower part of the plots, are found to be more or less constant regardless of the number of FEs, which meets with expectations.

Regarding the total time, the Multi-Client development is roughly 50% faster than the projected time for YARR v1.5.2.

## 6.5. Other developments for the LLS

At the LLS, some parameters of the FE chip configuration file will need to be changed for all the FEs at the same time. For example, when changing the speed of the readout, the value of the *CdrClk* has to be changed for all the FEs. Manually opening up 144 configuration files for changing a single parameter is definitely not efficient. Therefore, the possibility of having a global configuration file that changes the parameters for all the FEs was demanded.

The new development allows for adding a *GlobalOverwrite* file path for each of the chips in the connectivity file of YARR. The values are read in the file and overwritten in the chip's configuration file. The implementation has been designed to perform multiple checks on the file given by the user, e.g. checking the structure of the file and the existence of the parameter.



## 7. Summary, Conclusion and Outlook

During the master's thesis work, familiarity was developed with the FELIX/YARR read-out chain for the Loaded Local Supports. Firstly, an understanding of how to configure and operate the FELIX, Optoboard and ITkPixV1 chips was achieved. This aspect required attention at the beginning of the thesis work, as all three systems have their own configuration files, which must be mutually compatible to operate successfully. It was also necessary to understand how to correctly tune the chip configuration to enhance the communication: the CmlBias parameters have been identified as key for this.

The next step was to verify the correct functionality of the ITk Pixel read-out chain for the Loaded Local Supports: numerous tests have been completed. The most relevant result is the operation of six Quad Modules simultaneously. It was the highest number of ITkPixV1 FEs ever read out in a FELIX/YARR setup.

The data about the timing of the digital scan and other scans is important to predict the execution time with a higher number of modules and to schedule the necessary time for integration. In addition, this information helps the collaboration to select the most necessary upgrades for the YARR Software.

It became apparent that the most urgent upgrade before integration is the parallelisation of the configuration of the modules. A new design employs multiple FELIX Clients, preventing memory problems on the FELIX side when operating numerous modules. The use of multiple data buffers (FIFOs) allows a completely parallel operation on the TX side.

This development was tested both using ITkPixV1 FEs and using the ITkPixV2 emulator. The results show that the new development reduces the configuration time for a loaded longeron by circa 70%, and the total scan time by more than 50%.

### Outlook

The natural continuation of the system testing part of this work is the Loaded Local Supports QC testing, which is scheduled for Autumn 2025 in Bonn. The loaded Inclined

## 7. *Summary, Conclusion and Outlook*

Half Rings and longerons will undergo QC procedures, testing the cooling and electrical services (serial powering). On this occasion, similar tests to the ones presented in this thesis are performed, with a larger number of modules (up to 36).

Regarding the development aspect, further improvements can still be achieved. Firstly, the number of FELIX client instances could be increased. However, the maximum number of FELIX client instances that can run simultaneously has not yet been determined; therefore, a dedicated investigation should be completed to establish this limit. In addition, the multi-client design should be extended to the RX side to avoid overloading the FELIX capabilities. This is even more relevant than for the TX side, as the bandwidth on the RX side is roughly 10 times larger than that on the TX side. Lastly, the operation of the scan loops should be fully parallelised, and any dependence on the number of enabled FEs should be removed.

## A. Additional Figures and Tables

|               | Intercept [s]         | Slope [s/FE]      |
|---------------|-----------------------|-------------------|
| Configuration | $0.49 \pm 0.06$       | $1.682 \pm 0.004$ |
| Scan          | $6.8 \pm 0.4$         | $0.445 \pm 0.026$ |
| Analysis      | $0.752 \pm 0.009$     | 0 (fixed)         |
| Processing    | $0.00129 \pm 0.00022$ | 0 (fixed)         |

Table A.1.: Fit values of Figure 6.1.

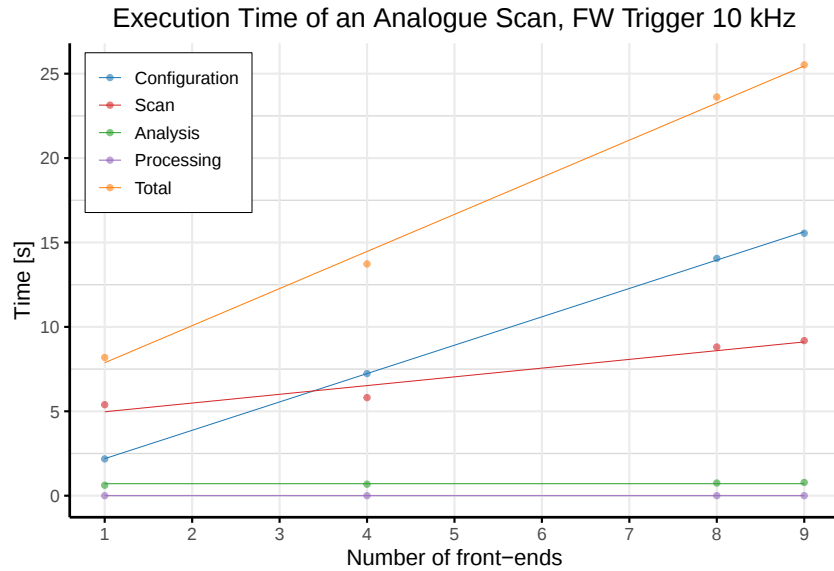


Figure A.1.: The Execution time of an analogue scan as a function of the number of enabled FEs. Performed with YARR released version.

## A. Additional Figures and Tables

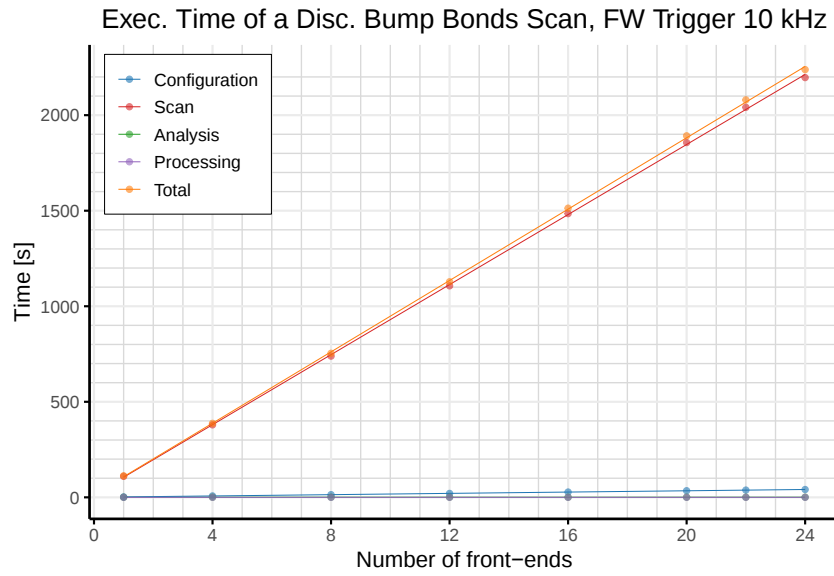


Figure A.2.: The Execution time of a disconnected bump bonds scan as a function of the number of enabled FEs. Performed with YARR released version.

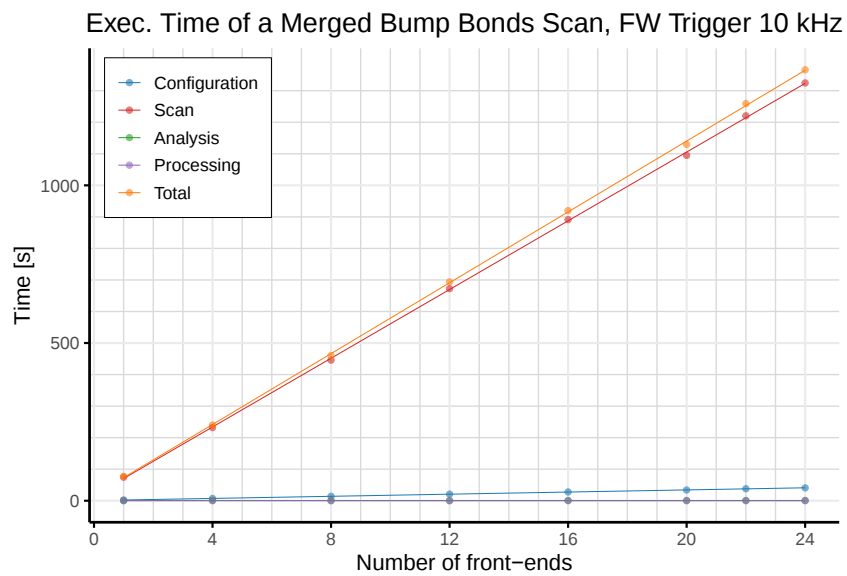


Figure A.3.: The Execution time of a merged bump bonds scan as a function of the number of enabled FEs. Performed with YARR released version.

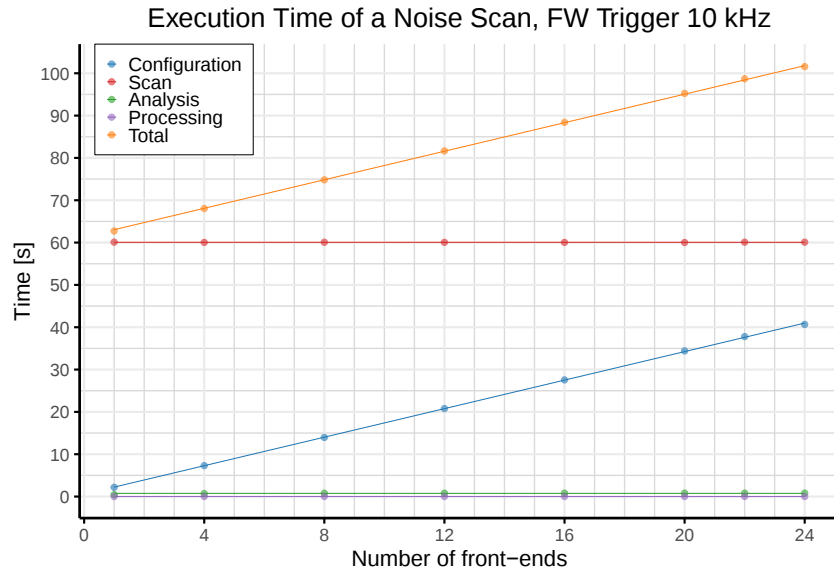


Figure A.4.: The Execution time of a noise scan as a function of the number of enabled FEs. Performed with YARR released version.

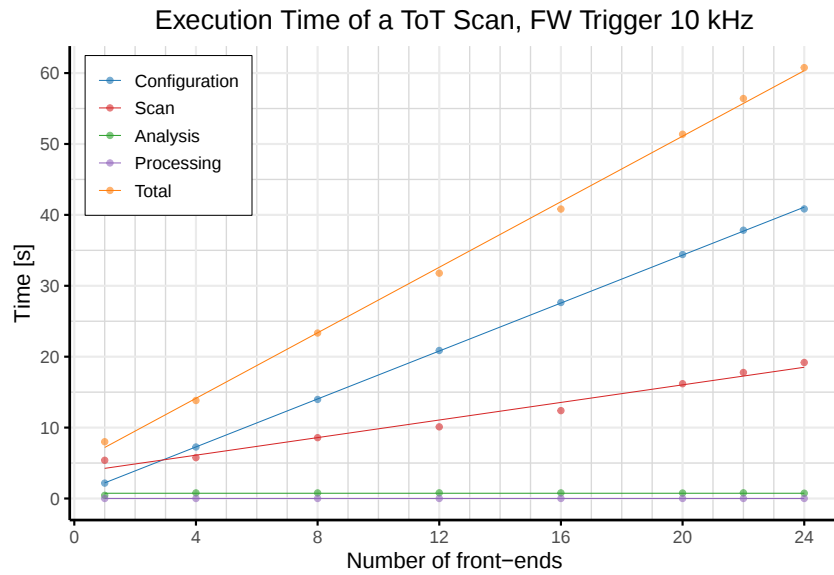


Figure A.5.: The Execution time of a ToT scan as a function of the number of enabled FEs. Performed with YARR released version.

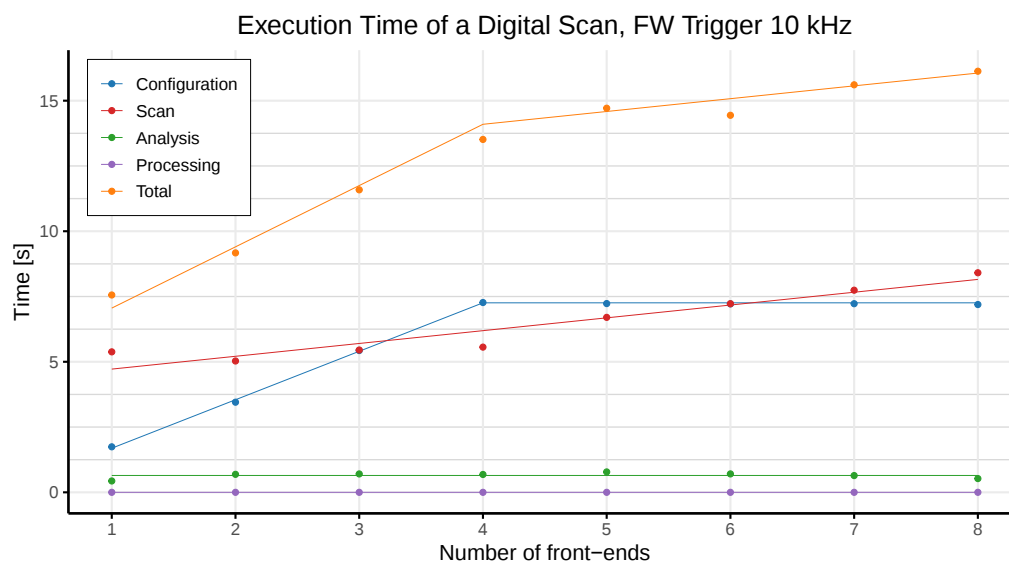


Figure A.6.: Execution time of a digital scan as a function of the number of enabled FEs, using the Multi-Client development.

# Bibliography

- [1] L. Evans, P. Bryant, *LHC Machine*, JINST **3**, S08001 (2008)
- [2] G. Aad, et al. (ATLAS), *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST **3**, S08003 (2008)
- [3] ATLAS Collaboration, *Technical Design Report for the ATLAS Inner Tracker Strip Detector*, Technical Report CERN-LHCC-2017-005. ATLAS-TDR-025, CERN (2017)
- [4] M. Wu (ATLAS TDAQ), *FELIX: Readout upgrade for the ATLAS Trigger DAQ system in HL-LHC*, Nucl. Instrum. Meth. A **1048**, 167994 (2023)
- [5] G. Aad, et al. (ATLAS), *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, Phys. Lett. B **716**, 1 (2012)
- [6] ALICE Collaboration, *The ALICE experiment at the CERN LHC*, JINST **3**, S08002 (2008)
- [7] CMS Collaboration, *The CMS experiment at the CERN LHC*, JINST **3**, S08004 (2008)
- [8] LHCb Collaboration, *The LHCb Detector at the LHC*, JINST **3**, S08005 (2008)
- [9] G. Aad, et al., *ATLAS Pixel Detector Electronics and Sensors*, JINST **3**, P07007 (2008)
- [10] M. Barbero, et al., *FE-I4 ATLAS pixel chip design*, PoS **VERTEX2009**, 027 (2009)
- [11] A. Ahmad, et al., *The silicon microstrip sensors of the ATLAS semiconductor tracker*, Nucl. Instrum. Meth. A **578**, 98 (2007)
- [12] M. Nicolás Viaux (ATLAS Muon), *The ATLAS ‘New Small Wheel’ Muon detector stations recently commissioned for LHC Run3*, Nucl. Instrum. Meth. A **1045**, 167574 (2023)

## Bibliography

- [13] ATLAS Collaboration, *Technical Design Report for the Phase-II Upgrade of the ATLAS Muon Spectrometer*, Technical Report CERN-LHCC-2017-017, ATLAS-TDR-026, CERN, Geneva (2017)
- [14] G. Aad, et al. (ATLAS), *The ATLAS trigger system for LHC Run 3 and trigger performance in 2022*, JINST **19(06)**, P06029 (2024)
- [15] O. Brüning, L. Rossi, editors, *The High Luminosity Large Hadron Collider*, World Scientific (2024)
- [16] O. Aberle, et al., *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report*, Technical Report CERN-2020-010, Geneva (2020)
- [17] ATLAS Collaboration, *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*, Technical Report CERN-LHCC-2017-021, ATLAS-TDR-030, CERN, Geneva (2017)
- [18] S. Möbius (ATLAS ITk), *Module development for the ATLAS ITk pixel detector*, JINST **17(03)**, C03042 (2022)
- [19] J. Chistiansen, M. Garcia-Sciveres, *RD Collaboration Proposal: Development of pixel readout integrated circuits for extreme rate and radiation*, Technical Report CERN-LHCC-2013-008, LHCC-P-006, CERN, Geneva (2013)
- [20] F. Loddo, et al., *RD53 pixel chips for the ATLAS and CMS Phase-2 upgrades at HL-LHC*, Nucl. Instrum. Meth. A **1067**, 169682 (2024)
- [21] Xilinx, Inc., *Aurora 64B/66B Protocol Specification* (2014), SP011 (v1.3)
- [22] J. Kampkötter, et al., *Characterization and verification of the Shunt-LDO regulator and its protection circuits for serial powering of the ATLAS and CMS pixel detectors*, J. Phys. Conf. Ser. **2374(1)**, 012071 (2022)
- [23] F. Munoz Sanchez (ATLAS ITk pixel), *Carbon Based Local Supports for the ATLAS ITk Pixel Detector*, PoS **Pixel2022**, 077 (2023)
- [24] S. Diez (ATLAS), *The ATLAS ITk strip local support structures*, Nucl. Instrum. Meth. A **1066**, 169552 (2024)
- [25] G. Chiodini, *Loading of ATLAS ITk pixel module on multi flavour local supports*, PoS **TIPP2023**, 122 (2025)

- [26] Z. Chubinidze (ATLAS), *ITk pixel system test of the ATLAS experiment*, JINST **20(02)**, C02038 (2025)
- [27] L. Franconi (ATLAS ITK), *The Opto-electrical conversion system for the data transmission chain of the ATLAS ITk Pixel detector upgrade for the HL-LHC*, J. Phys. Conf. Ser. **2374(1)**, 012105 (2022)
- [28] P. Moreira, et al., *lpGBT documentation: release* (2024), cds/2809058
- [29] C. Chen, et al., *1.28 and 5.12 Gbps multi-channel twinax cable receiver ASICs for the ATLAS Inner Tracker Pixel Detector upgrade*, Nucl. Instrum. Meth. A **981**, 164439 (2020)
- [30] W. Zhang, et al., *Characterization and quality control test of a gigabit cable receiver ASIC (GBCR2) for the ATLAS Inner Tracker Detector upgrade*, JINST **16(08)**, P08013 (2021)
- [31] L. Amaral, et al. (CMS), *The versatile link, a common project for super-LHC*, JINST **4**, P12003 (2009)
- [32] C. Soós, et al., *Versatile Link PLUS transceiver development*, JINST **12(03)**, C03068 (2017)
- [33] Z. Zeng, et al., *LDQ10: a compact ultra low-power radiation-hard  $4 \times 10$  Gb/s driver array*, JINST **12(02)**, P02020 (2017)
- [34] T. Zhang, et al., *GBLD10+: a compact low-power 10 Gb/s VCSEL driver*, JINST **11(01)**, C01015 (2016)
- [35] M. Menouni, et al., *The GBTIA, a 5 Gbit/s Radiation-Hard Optical Receiver for the SLHC Upgrades*, in *Topical Workshop on Electronics for Particle Physics*, CERN (2009)
- [36] A. Paramonov (ATLAS TDAQ), *FELIX: the Detector Interface for the ATLAS Experiment at CERN*, EPJ Web Conf. **251**, 04006 (2021)
- [37] ATLAS Collaboration, *Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System*, Technical Report CERN-LHCC-2017-020, ATLAS-TDR-029, CERN, Geneva (2017)
- [38] P. Moreira, *The GBT, a proposed architecture for multi-Gbps data transmission in high energy physics*, in *Topical Workshop on Electronics for Particle Physics*, page 71 (2007)

## Bibliography

- [39] E. David, T. Kiss, *Integrating lpGBT links into the Common Readout Units (CRU) of the ALICE Experiment* (2024), 2403.11723
- [40] ATLAS FELIX Group, *FELIX user manual* (2024), RM 5.1.x
- [41] T. Heim, *YARR - A PCIe based readout concept for current and future ATLAS Pixel modules*, J. Phys. Conf. Ser. **898**(3), 032053 (2017)
- [42] H. Kolanoski, N. Wermes, *Teilchendetektoren. Grundlagen und Anwendungen*, Springer (2016)
- [43] M. P. Drescher, *ITk-Pixel FELIX/RD53A Read-out Chain Stress Test Preparations*, Master's thesis, Göttingen U. (2023)
- [44] W. Alkakh (ATLAS ITK), *ATLAS ITk-Pixel DAQ system*, JINST **19**(11), C11013 (2024)

# Danksagung

First of all, I'd like to thank Arnulf Quadt and Jörn Große-Knetter for the great opportunity to work on this project. I'd also like to thank Jörn, Ali Skaf and Matthias Drescher for the day-to-day help on my project. Thanks to Matthias and Ali also for proofreading my thesis and for the help on the emulators and the electronics, respectively. I'd also like to thank the ATLAS members with whom I worked, whose expertise was key in the preparation of this thesis, especially Juliette Martin. In addition, I'd like to thank the members of the II. Physikalisches Institut for the fun chats at the coffee break. I cannot thank my parents enough for their continuous support, also in my decision to move abroad. I'm also deeply thankful to Marlene for her irreplaceable support and her ability to always show me the bright side of things. Lastly, I'd like to thank all my friends, old and new. It is truly wonderful to be around you, and I am grateful to have met you.

**Erklärung**

nach §17(9) der Prüfungsordnung für den Bachelor-Studiengang Physik und den Master-Studiengang Physik an der Universität Göttingen: Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe.

Darüberhinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, im Rahmen einer nichtbestandenenen Prüfung an dieser oder einer anderen Hochschule eingereicht wurde.

Göttingen, den 15. Juli 2025

(Paolo Maria Malatesta)