Development of the detector readout for the ATLAS Phase II ITk pixel detector

Dissertation

zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades
„Doctor rerum naturalium“
der Georg-August-Universität Göttingen

im Promotionsprogramm ProPhys
der Georg-August University School of Science (GAUSS)

vorgelegt von

Wael Alkakhi

aus Hasbaya, Libanon

Göttingen, 2024

Betreuungsausschuss

Prof. Dr. Arnulf Quadt
Prof. Dr. Stan Lai
Prof. Dr. Steffen Schumann

Mitglieder der Prüfungskommission:

Referent: Prof. Dr. Arnulf Quadt
II. Physikalisches Institut, Georg-August-Universität Göttingen

Koreferentin: Prof. Dr. Jochen Dingfelder
Forschungs- und Technologie-Zentrum Detektorphysik, Universität Bonn

Weitere Mitglieder der Prüfungskommission:

Prof. Dr. Ansgar Reiners
Institut für Astrophysik, Georg-August-Universität Göttingen

Prof. Dr. Michael Seibt
IV. Physikalisches Institut, Georg-August-Universität Göttingen

Prof. Dr. Wolfgang Wagner
Fakultät für Mathematik und Naturwissenschaften, Bergische Universität Wuppertal

PD. Dr. Markus Keil
CERN

Tag der mündlichen Prüfung: 18.Oktober.2024

Referenz: II.Physik-UniGö-Diss-2024/06

Development of the detector readout for the ATLAS Phase II ITk pixel detector

Abstract

During the HL-LHC upgrade, a high luminosity upgrade takes place, where the instantaneous luminosity will reach $7.5 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ in Run 4. The current inner detector of the ATLAS experiment will be replaced by an all-silicon detector called the Inner Tracker (ITk). The ITk comprises a pixel detector surrounded by a strip detector. The pixel detector consists of five layers, with the first two layers forming the inner system and the last three layers forming the outer barrel and the endcap.

Due to the large number of components, a database named the ITk production database (ITk PD) has been created to store and track information about ITk elements. The implementation of loaded local support elements in the ITk PD has been made, along with additional work to simplify the handling of these elements within the ITk PD.

The basic read-out chain of the ATLAS ITk-Pixel DAQ system includes the YARR software communicating with the FELIX PCIe board, through an Optoboard, to the frontend chips RD53A and RD53B (ITkPix).

This dissertation presents the implementation of the ITk production database related to loaded local supports. It then summarizes the main contributions to the development of the ITk-Pixel DAQ readout chain, which range from providing support for additional hardware platforms and/or front-end chips in the YARR laboratory readout setup to the validation of the scaled-up YARR/FELIX/optoboard/ITkPix subsystem on a fully loaded local support. This marks an important milestone on the path to developing and testing the full-scale ITk-Pixel DAQ system.

Entwicklung der Detektorauslesung für den ATLAS Phase II ITk-Pixeldetektor

Zusammenfassung

Während des HL-LHC Ausbau findet ein Ausbau auf eine höhere Luminosität statt, bei der die instantane Luminosität im Run 4 $7.5\times10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ erreichen wird. Der derzeitige Inner Detektor des ATLAS-Experiments wird durch einen reiner Siliziumdetektor namens Inner Tracker (ITk) ersetzt. Der ITk besteht aus einem Pixeldetektor, der von einem Streifendetektor umgeben ist. Der Pixeldetektor besteht aus fünf lagen, wobei die ersten beiden Lagen das innere System bilden und die letzten drei Lagen den äußeren Zylinder und die Endkappe bilden.

Aufgrund der großen Anzahl von Komponenten wurde eine Datenbank namens ITk Produktions Datenbank (ITk PD) eingerichtet, in der Informationen über ITk-Elemente abgelegt und verfolgt wurden. Die Implementierung der beladenen lokalen Trägerstrukturen in der ITk PD sowie zusätzliche Arbeiten zur Vereinfachung des Umgangs mit diesen Elementen innerhalb der ITk PD wurden vorgenommen.

Die grundlegende Auslesekette des ATLAS ITk-Pixel DAQ-Systems umfasst die YARR Software, die mit dem FELIX PCIe Board über ein Optoboard mit den Frontend-Chips RD53A und RD53B (ITkPix) kommuniziert.

In dieser Arbeit wird die Implementierung der ITk-Produktionsdatenbank in Bezug auf die beladenen lokalen Trägerstrukturen vorgestellt. Sie fasst dann die wichtigsten Beiträge zur Entwicklung der ITk-Pixel DAQ-Auslesekette zusammen, die von der Unterstützung zusätzlicher Hardwareplattformen und/oder Front-End-Chips zum YARR-Laborauslese-Setup, bis hin zur Validierung des skalierten YARR/FELIX/optoboard/ITkPix-Subsystems auf einem belasteten lokalen Träger. Dies ist ein wichtiger Meilenstein auf dem Weg zur Entwicklung und Erprobung des ITk-Pixel-DAQ-Systems in voller Größe.

Contents

1. Introduction **1**

2. Particle Physics **3**
2.1. The Standard Model (SM) 3
2.2. Limitations of the Standard Model 5

3. The HL-LHC upgrade and ATLAS experiment **9**
3.1. CERN Accelerator system 9
3.2. LHC 11
3.3. ATLAS Detector 13
3.4. HL-LHC and ATLAS upgrade 20

4. Physics of Particle Detection and Semiconductor **23**
4.1. Interactions of Particles with Matter 23
4.2. Physics of Semiconductor detector 28

5. ITk Production Database **33**
5.1. ITk Pixel Module 33
5.2. ITk Outer Barrel 35
5.3. ITk Production Database 37
5.4. OB Bare Local Supports 38
5.5. OB Loaded Local Supports 59
5.6. OB Loaded Module Cell 69
5.7. ITk Component counter 72

6. YARR BDAQ Readout System **77**
6.1. Readout Chips 77
6.2. BDAQ53 81
6.3. YARR 81

6.4. YARR BDAQ readout for RD53A 82
6.5. YARR BDAQ readout for RD53B(ITkPix) 84

7. Netio-next YARR Readout Controller 89
7.1. FELIX 89
7.2. Optoboard 91
7.3. Readout chain 92
7.4. YARR Developments to read out RD53A SCC 93

8. FelixClient YARR Readout Controller 97
8.1. Readout chain 97
8.2. YARR performance with FelixClient 98

9. Conclusion 103
9.1. Outlook 104

Bibliography 105

Appendices 113

A. YARR software developments and results 115
A.1. readout process in YARR bdaq controller for RD53A 115
A.2. readout process in YARR bdaq controller for RD53B 115
A.3. Validation of YARR software using FelixClient results 116

Acknowledgements

I would like to express my heartfelt thanks and appreciation towards Prof. Dr. Arnulf Quadt for his supervision, support and guidance throughout my PhD program. My gratitude also extends to PD Dr. Jörn Große-Knetter, Dr. Ali Skaf, and Dr. Hua Ye for their invaluable guidance and constructive feedback throughout my PhD, including their support in setting and discussing research plans, discussing results, and refining abstracts, presentations, and publications. I am also thankful to all the members of the research group for fostering such a positive working environment, with special thanks to Matthias Drescher for his significant contributions to our work. My thanks also go to the technicians, whose contributions were indispensable. I would like also to thank PD Dr. Susanne Kuehn for her role as a technical supervisor during my qualification task. Additionally, I extend my thanks to Dr. Matthias Hamer, Dr. Florian Hinterkeuser, and Alexandra Wald from FTD University of Bonn for their organizational and technical support during my work at FTD University of Bonn.

I would like to extend my sincere thanks to the ITk ATLAS collaboration for their unwavering support, where numerous experts generously offered assistance to help overcome challenges.

Additionally, I would like to acknowledge the other members of my Thesis Advisory Committee (TAC), Prof. Dr. Stan Lai and Prof. Dr. Steffen Schumann, for their valuable advice and feedback during our meetings. I am grateful to Prof. Dr. Jochen Dingfelder for agreeing to serve as a referee for my PhD thesis, and I extend my thanks to the members of the examination board: Prof. Dr. Ansgar Reiners, Prof. Dr. Wolfgang Wagner, Prof. Dr. Michael Seibt, and PD Dr. Markus Keil.

I also want to declare that I used ChatGPT for proofreading in this thesis.

Finally, I would like to express my deepest appreciation to my wife, Caroline, whose unwavering support and belief in me made this journey much smoother. Without her by my side, this PhD would have been far more challenging. I am also immensely grateful to my parents for their continuous support and for always standing by me. Their encouragement and love have been fundamental to my success.

Contributions by the author

All the work presented in this thesis was conducted for the ATLAS experiment at CERN. The ATLAS experiment is part of a collaboration known as the ATLAS Collaboration, which publishes all work and results under its name. The ATLAS Collaboration consists of physicists, engineers, technicians, students, and support staff, comprising 3,000 scientific authors from 182 institutes across 42 countries.

This thesis focuses on the development of the ITk pixel detector for the ATLAS experiment, which is part of the ATLAS Collaboration. In this section, I will summarize my contributions and acknowledge the contributions of others who supported my work.

My entire PhD research was supervised by Prof. Dr. Arnulf Quadt, and I worked within his research group. In the first year of my Ph.D., I worked on my qualification task, which granted me authorship within the ATLAS Collaboration. The qualification task involved preparing the ITk Production database for the loaded local support, as detailed in Chapter 5. During this task, Dr. Susanne Kuehn served as my technical supervisor, and PD Dr. Jörn Große-Knetter was my local supervisor. They provided significant guidance in planning the work and discussing the results. Additionally, I developed a GUI that was initially created by Eunchong Kim. I also received valuable assistance from ITk Production database experts during discussions at ITk Production database meetings and through the ITk Production database mailing list.

In the second year of my Ph.D., after completing my qualification task, I began developing the YARR software to support the BDAQ53 FPGA board hardware, as explained in Chapter 6. This project was previously developed by Dr. Rafael Goncalves Gama, and I continued its development. Throughout this task, I received support from PD Dr. Jörn Große-Knetter, Dr. Hua Ye and Dr. Ali Skaf in planning and discussing the development work. I also received assistance from experts in YARR and BDAQ53.

After that task, I developed YARR to read out the RD53A frontend chip using FELIX through Felix-star and NetioNext, as discussed in Chapter 7. This project was previously developed by Alexander Paramonov and Daniel John Wilbern. During this work, I received assistance from Alexander Paramonov, PD Dr. Jörn Große-Knetter, and Dr.

Ali Skaf to achieve successful results.

Finally, I worked on testing YARR with FelixClient and the Felix-star application using multiple ITkPix frontend chips, as introduced in Chapter 8. In this work, I successfully read out 10 frontend chips simultaneously. PD Dr. Jörn Große-Knetter and Dr. Ali Skaf supported me in planning and discussing the work. Additionally, Dr. Matthias Hamer, Florian Hinterkeuser, and Alexandra Wald provided significant help and organizational support during my work at FTD at the University of Bonn. After achieving this result, the master's student Paolo Maria Malatesta will take over to test the YARR software with more quad modules and address any problems he encounters during his master's thesis. At the beginning of his thesis, I helped him become familiar with the topic.

CHAPTER 1

Introduction

Particle physics is a branch of physics that studies the fundamental elements of matter and the forces responsible for the interactions between these elements. These studies begin with questions like “What is matter made of?” and “How do the fundamental elements interact?” In the last century, a theory called the Standard Model was developed. It introduces all the fundamental particles and the forces that govern their interactions. The Standard Model has been tested experimentally, and successful results have been achieved. Despite its success in describing elementary particles and their interactions, there are some phenomena that the Standard Model does not explain properly.

Chapter 2 introduces the Standard Model, detailing its elementary particles and their forces of interaction. Additionally, it discusses the limitations of the Standard Model that need to be addressed in the future. The Large Hadron Collider (LHC) at CERN offers the opportunity to provide high-energy beams of particles, as CERN’s accelerator is the largest particle accelerator in the world. The LHC consists of four detectors, one of which is the ATLAS (A Toroidal LHC Apparatus) detector, which studies particles at high energies. The next upgrade of the LHC is the High-Luminosity LHC (HL-LHC), which will also include an upgrade of the ATLAS experiment. In this upgrade, the inner detector of the ATLAS detector will be replaced by an all-silicon detector named the Inner Tracker (ITk). CERN’s accelerator, the LHC, the ATLAS detector, and the HL-LHC upgrade are discussed in Chapter 3. Chapter 4 introduces the physics of particle detection, explaining how particles interact with matter. Additionally, it presents semiconductors and their role in particle detection.

The ITk is composed of two detectors: the pixel detector and the strip detector. One of the components of the pixel detector is the Outer Barrel (OB), which consists of semiconductor modules mounted on supports called loaded local supports. The OB will host more than 4,000 modules. To track the information of these elements during the production phase of the ITk, a database named the ITk Production Database (ITk PD) has been developed. The implementation of the ITk PD regarding loaded local supports is discussed in Chapter 5.

The ITk uses frontend chips developed under the CERN RD53 project. Initially, the RD53A chip served as a prototype readout chip. For ATLAS, the RD53B (ITkPixv1) chip was then used as the pre-production readout chip, leading to the final production of the ITkPix-V2 chip. The frontends of the ITk pixel detector require a readout system to function. The readout system consists of both software and hardware components. Several readout systems were developed for the ITk frontends. In this work, two of these systems are studied: the YARR readout system and the BDAQ53 readout system. The two systems and the development of YARR software to support BDAQ53 hardware are introduced in Chapter 6.

The final readout system of the ITk pixel detector consists of YARR software, which communicates with the FELIX board through an Optoboard to read out the ITk pixel frontends. This chain has been validated by operating FELIX with Felixcore and using the NetIO data messaging protocol. During the operation of the ITk, FELIX will be operated by Felix-star and the NetIONext protocol. The development of YARR to validate the functioning of the YARR/NetIONext/Felix-star/RD53A readout chain has been completed. The FELIX, Optoboard, and the development done for validation are discussed in Chapter 7.

The FelixClient application in FELIX software was developed to simplify the complexity of NetIONext. The YARR software has been developed to read out the RD53B using FELIX through the FelixClient and Felix-star applications. Since a large number of frontend chips must be tested during the production phase of loaded local supports for quality control, there is a need to validate the YARR software for reading out multiple frontend chips. The validation of YARR using FelixClient to read out multiple frontend chips has been achieved and is discussed in Chapter 8.

CHAPTER 2

Particle Physics

Particle physics is concerned with understanding elementary particles and their behavior. These particles are the building blocks of matter in the universe, they are not composed of smaller components. Elementary particles interact with each other through a set of forces. The best theory that describes elementary particles and their forces is the Standard Model.

In this chapter, the Standard Model is introduced in Section 2.1, and its limitations are discussed in Section 2.2.

2.1. The Standard Model (SM)

Figure 2.1 shows all the particles of the Standard Model. The Standard Model consists of 12 fermions and 5 bosons. The fermions are spin $\frac{1}{2}$ particles, while the bosons are particles with integer spin [1].

The 12 fermions are classified into 6 quarks and 6 leptons. These fermions are also classified into generations according to their masses. The first generation, represented by the first column of Figure 2.1, consists of the Up quark (u), Down quark (d), Electron (e^{-}), and Electron neutrino (ν_e) [3]. These particles of the first generation were discovered at low energy scales. As energy increases to high energy scales, the second and third generations of particles are discovered due to the significant mass differences between the particles of the first generation and those of the second and third generations, as shown in Figure 2.1. The second generation is represented by the second column in Figure 2.1 and consists of the Charm quark (c), Strange quark (s), Muon (μ), and Muon neutrino (ν_μ) [3]. The third generation, represented by the third column of Figure 2.1, consists of the Top quark (t), Bottom quark (b), Tau (τ), and Tau neutrino (ν_τ) [3]. Note That the quarks are also classified into two types: up-type quarks, which include the Up quark (u), Charm quark (c), and Top quark (t); and down-type quarks, which

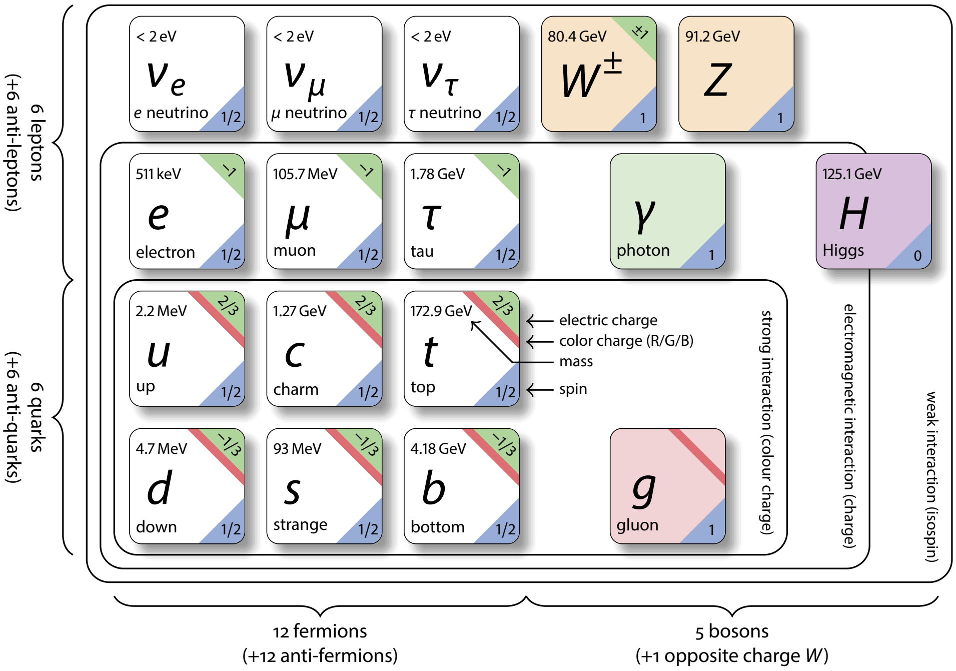


Figure 2.1.: The particles and their force carriers are shown by the Standard Model, where each element is represented with its properties: mass (taken from Ref. [2], electrical charge, color charge, and spin.

include the Down quark (d), Strange quark (s), and Bottom quark (b). According to the Dirac equation, which describes the relativistic quantum mechanics of the Standard Model's fermions, each fermion has an antiparticle with the same mass but opposite charge [4]. This antiparticle is denoted by a bar above the particle symbol for quarks and neutrinos, and by its charge for charged leptons.

In addition to the 12 fermions, there are 6 boson particles. These particles are the Gluon (g), Photon (γ), Z boson (Z), W^+ boson (W^+), W^- boson (W^-), and the Higgs particle (H). The Gluon (g), with spin 1 and zero mass, is the carrier of the strong interaction. The Photon (γ), with spin 1 and zero mass, is the carrier of the electromagnetic interaction. The Z boson (Z), W^+ boson (W^+), and W^- boson (W^-), all with spin 1, are the carriers of the weak interaction. The Higgs particle (H) is a spin 0 particle that provides the mechanism for other particles to obtain their masses. It was discovered in 2012 by the ATLAS [5] and CMS [6] experiments at the LHC. One year later, the Nobel Prize in Physics was awarded to Peter Higgs and François Englert.

As shown in Table2.1, there are four interaction forces that describe the interactions between fundamental particles. The strong interaction occurs through gluon exchange, and only quarks can interact through this force, as shown in Figure 2.1. The quantum chromodynamics theory (QCD) describes the strong interaction [7]. The electromagnetic interaction is mediated by the exchange of photons. Quarks, electrons, muons, and tau

Force	Strength	Boson	Spin	Mass(GeV)
Strong	1	Gluon(g)	1	0
Electromagnetism	10^{-3}	Photon (γ)	1	0
Weak	10^{-8}	W boson ($W^{\pm}$)	1	80.4
	10^{-8}	Z boson (Z)	1	91.2
Gravity	10^{-37}	Graviton	2	0

Table 2.1.: shows the interactions between the fundamental particles and their properties, including strength which is the relative strengths are approximate indicative values for two fundamental particles at a distance of 1 femtometer (1 fm= 10^{-15} meters), carrier, carrier's spin, and carrier's mass [4].

.

particles can undergo electromagnetic interaction, as shown in Figure 2.1. The quantum electromagnetic theory (QED) describes electromagnetic interactions [8]. The weak interaction involves the propagation of one of the W bosons or the Z boson. All fermions of the Standard Model can interact weakly, as shown in Figure 2.1. The gravitational force can be neglected in particle physics because its strength is negligible compared to the other three interaction types, as shown in Figure 2.1 [1].

One question that arises for the reader after the introduction of the Standard Model above is why it is considered the best theory for describing fundamental particles and their interactions. It is regarded as the best theory because of its powerful predictions of cross-sections, as demonstrated in Figure 2.2. This figure illustrates that the expected cross-sections by the Standard Model are comparable to the cross-sections measured by ATLAS.

2.2. Limitations of the Standard Model

Although the Standard Model is considered the best theory for describing fundamental particles and their interactions due to its predictive power, it is not complete and does not explain everything. There are gaps that need to be addressed. In this section, these gaps are discussed.

2.2.1. Neutrino Mass and Neutrino Oscillation

The Homestake experiment in the late 1960s measured the flux of electron neutrinos produced by the Sun [10]. It showed a difference between the measured neutrino flux and the expected neutrino flux from the Sun's luminosity, known as the solar neutrino problem. This difference can be explained by the oscillation of neutrinos between different flavors [11].

[1] In addition, it is also the only fundamental force that we have not been able to quantize: there is no demonstrated quantum gravity, and we have not observed a graviton that would mediate such a force.

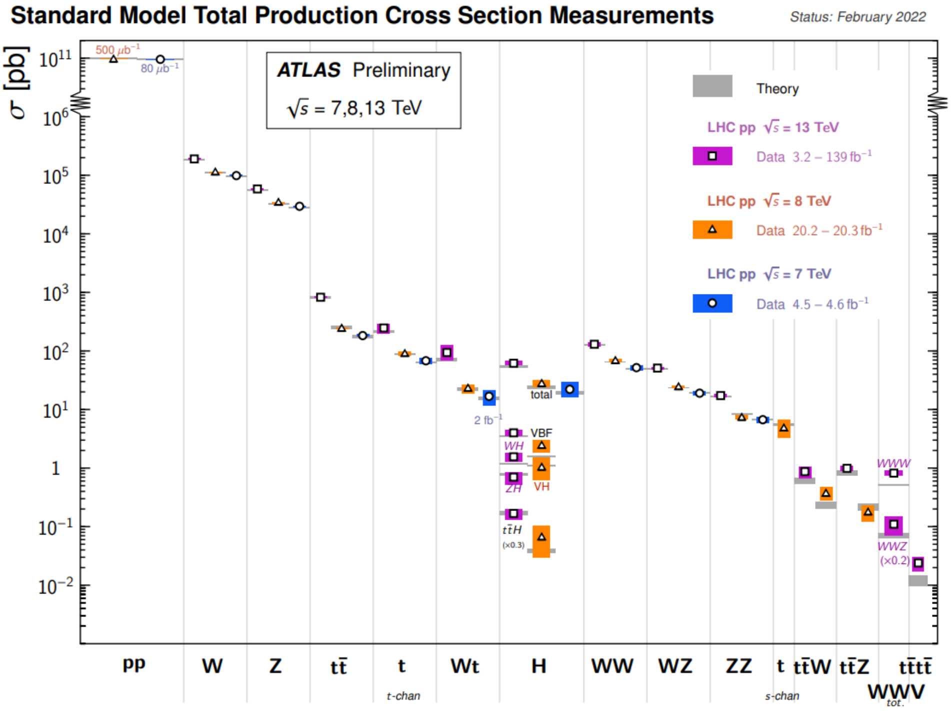


Figure 2.2.: The experimental cross sections measured by ATLAS and the expected cross sections predicted by the Standard Model for certain processes [9].

The Super-Kamiokande experiment [12] in 1998 detected the first evidence of neutrino oscillation after two years of operation. It measured atmospheric neutrinos and found a deficiency of electron neutrinos while muon neutrinos were observed. This result supports the idea that neutrinos must have non-zero mass in order to oscillate.

The Sudbury Neutrino Observatory (SNO) [13] in 2002 detected evidence of electron neutrinos changing types during their travel. This oscillation can occur only if one or more types of neutrinos have small masses, and there must be a difference in their masses.

All of these experimental results contradict the Standard Model, which assumes that all flavors of neutrinos have zero mass. This means the Standard Model must be extended to include neutrino masses in order to explain neutrino oscillations.

2.2.2. Dark Matter

Dark matter [14] is a type of matter that does not interact through electromagnetic interaction, meaning it does not emit any light. This makes it very hard to detect using astronomical detectors. Scientists have deduced the existence of dark matter through its gravitational effects. The evidence for dark matter's existence includes gravitational lensing[2] research of the Bullet Cluster[3]. The universe consists of about 26% dark matter, 69% dark energy, and only about 5% baryonic matter [15].

Until now, no particle that can form dark matter in the universe has been found. This is another gap in the Standard Model that needs to be solved.

2.2.3. CP violation

In particle physics, charge conjugation and parity are two transformations. Both have corresponding symmetries called C-symmetry and P-symmetry. CP violation [16] is a violation of these symmetries. According to the CKM matrix, the Standard Model permits CP violation. However, it is observed that there is no CP violation in the strong interaction [17].

Therefore, the Standard Model must be developed further to solve the issue of CP violation in strong interactions.

[2] Gravitational lensing is the bending of light as it travels from its source to its receiver due to the presence of large gravitational fields (such as those from galaxies, black holes, or perhaps dark matter) along the way.

[3] The Bullet Cluster is composed of two colliding galaxies.

CHAPTER 3

The HL-LHC upgrade and ATLAS experiment

CERN (European Organization for Nuclear Research), considered the world's largest particle physics laboratory, was founded in 1954. Since then, CERN has provided the world with many fascinating particle physics results, starting with the discovery of the weak neutral current in the Gargamelle bubble chamber in 1973 [18]. This was followed by many results as the discovery of the W and Z bosons in 1983 in the UA1 and UA2 experiments [18], the discovery of direct CP violation in the NA48 experiment in 1999 [19], and finally the discovery of the Higgs boson by the ATLAS and CMS (Compact Muon Solenoid) experiments in 2012 [5]. Over the years, CERN and its experiments have undergone numerous developments and upgrades. CERN is home to the LHC (Large Hadron Collider), the world's largest particle collider, which hosts four main experiments, including the ATLAS experiment.

In this chapter, the CERN Accelerator system is discussed in Section 3.1. One of the colliders at CERN, the LHC, is introduced in Section 3.2. Afterwards, the ATLAS experiment and its detector are introduced in Section 3.3. Finally, the upcoming upgrades to the LHC and the ATLAS experiment are discussed in Section 3.4.

3.1. CERN Accelerator system

The Figure 3.1 shows the acceleration system of CERN. It consists of many accelerators, where each one accelerates the particles to reach a certain energy level, then sends the particles to the next accelerator to be accelerated further until they are accelerated by the LHC. At this point, the acceleration process is complete, and the particle beams are ready to collide. These beams are also used for fixed target experiments as NA61 Experiment [21]. CERN has the capability to accelerate both proton beams and heavy ion beams.

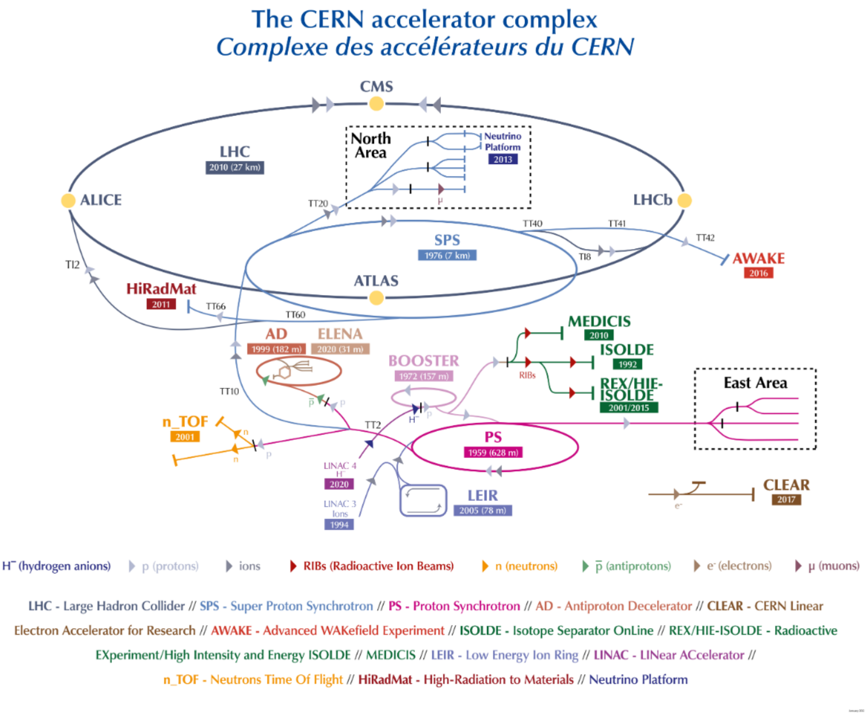

The CERN accelerator complex
Complexe des accélérateurs du CERN
CMS
North Area
Neutrino Platform
2013
LHC
2010 (27 km)
ALICE
LHCb
TT20
TT40
TT41
TT42
AWAKE
2016
SPS
1976 (7 km)
T8
TI2
HiRadMat
2011
TT66
ATLAS
TT60
AD
1999 (182 m)
ELENA
2020 (31 m)
BOOSTER
1972 (157 m)
MEDICIS
2010
ISOLDE
1992
RIBs
REX/HIE-ISOLDE
2001/2015
East Area
TT10
n_TOF
2001
TT2
PS
1959 (628 m)
LINAC 4
H⁻
2020
LINAC 3
Ions
1994
LEIR
2005 (78 m)
CLEAR
2017
H⁻ (hydrogen anions)
p (protons)
ions
RIBs (Radioactive Ion Beams)
n (neutrons)
p̄ (antiprotons)
e⁻ (electrons)
μ (muons)
LHC - Large Hadron Collider // SPS - Super Proton Synchrotron // PS - Proton Synchrotron // AD - Antiproton Decelerator // CLEAR - CERN Linear Electron Accelerator for Research // AWAKE - Advanced WAKefield Experiment // ISOLDE - Isotope Separator OnLine // REX/HIE-ISOLDE - Radioactive EXperiment/High Intensity and Energy ISOLDE // MEDICIS // LEIR - Low Energy Ion Ring // LINAC - LINear ACcelerator // n_TOF - Neutrons Time Of Flight // HiRadMat - High-Radiation to Materials // Neutrino Platform

Figure 3.1.: The CERN accelerator map [20], shows the journey of the particle to be accelerated in order to achieve its desired energy. The acceleration process starts with LINAC3 (for heavy ion beams) and LINAC4 (for negative hydrogen ion beams). The particles in the beam are subsequently accelerated in each ring along the way to the experiment, where the beams are ultimately collided.

The CERN accelerators utilize electromagnetic fields to accelerate the beam using radiofrequency cavities[1] and focus magnets to concentrate and steer the beams.

The starting point is Linac4 (Linear Accelerator 4) [23], which produces negative hydrogen ions and accelerates them to an energy of 160 MeV. Afterwards, the negative hydrogen ions are injected into the PSB (Proton Synchrotron Booster) [24], where the electrons are stripped from the hydrogen ions to form protons. These protons are then accelerated to reach an energy level of 2.0 GeV.

As mentioned above, CERN also accelerates heavy ion beams. The Linac3 (Linear Accelerator 3) [25], which launches and accelerates heavy ions. Then, the beam is sent to LEIR (Low Energy Ion Ring) [26], where the beam is accelerated from 4.2 MeV per nucleon to 72 MeV per nucleon.

Both protons and heavy ions are accelerated by the PS (Proton Synchrotron) [27], which increases the beam energy to 25 GeV. Afterwards, the beam is injected into the

[1] A radiofrequency cavity (RF) is a specially designed metallic chamber that uses electromagnetic fields to accelerate particle beams [22].

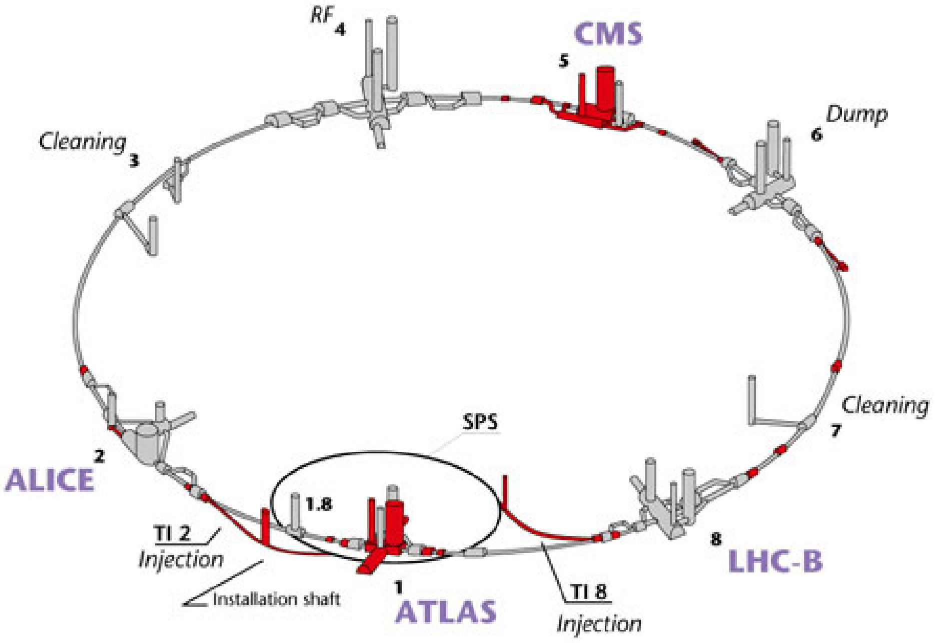


Figure 3.2.: LHC, its experiments, and all its infrastructures [31].

SPS (Super Proton Synchrotron)[2] [28], a 6.9 km circumference tunnel that accelerates the beam up to 450 GeV. The beam takes 21.6 seconds to complete one SPS cycle [29]. The beam is then split into two beams, which are injected into the LHC, a 27 km circumference ring, at two different positions in opposite directions. The beams are accelerated to reach an energy of 6.8 TeV for proton beams and 2.8 TeV for lead ion beams. At this stage, the acceleration process is complete, and the beams are ready to collide at the interaction points of the LHC experiments.

Filling an LHC ring with a beam takes about 4 minutes [29]. The protons then take 20 minutes to reach 6.5 TeV energy [30].

3.2. LHC

The LHC is the biggest particle collider in the world. It is a 27 km circumference tunnel located 100 m underground, shared between France and Switzerland. Before the LHC project, this tunnel was used for the electron-positron collider known as LEP (Large Electron–Positron Collider) [32], which was terminated in 2000. After that, the LHC project started, and its first beam was on September 10^{th} , 2008.

The Figure 3.2 shows the map of the LHC and its components. It hosts four experiments: ATLAS, CMS, LHCb (Large Hadron Collider beauty), and ALICE (A Large

[2] In the past, the SPS hosted the UA1 and UA2 experiments that discovered the W and Z bosons, during which time it was named $SP\bar{P}S$ as it was a proton-antiproton collider.

Ion Collider Experiment). The ATLAS experiment [33], discussed in detail in Section 3.3, and the CMS experiment [34] are general-purpose experiments designed primarily to study Electroweak Symmetry Breaking and the Higgs Mechanism, precisely measure the Standard Model parameters, and search for new physics beyond the SM. The LHCb experiment [35] focuses on studying the bottom quark, estimating the values of CP violation parameters in b-hadron[3] interactions. This study can provide a better understanding of the matter-antimatter asymmetry[4] of the Universe. ALICE [36] is a heavy ion collision experiment interested in understanding strong interaction physics at high energies.

As shown in Figure 3.2, the LHC is composed of eight octants. Each of the four experiments mentioned above is located in one octant: the ATLAS experiment is in octant 1, the CMS experiment is in octant 5, the LHCb experiment is in octant 8, and the ALICE experiment is in octant 2. Octant 6 is used for dumping the beams at the end of each LHC luminosity fill. The Radio Frequency (RF) cavities are located in octant 4. Two cleaning systems are located in octant 3 and octant 7. Additionally, the LHC has two injection points (TI2 and TI8) where the beams are inserted into the LHC once they reach the maximum SPS energy of 450 GeV.

The LHC beam is not a continuous stream of protons; it consists of groups of protons called bunches. The LHC ring can accommodate up to 2808 bunches [29]. Proton bunches collide at the interaction point of each experiment every 25 ns, in an event known as bunch crossing.

The event rate $\frac{dN_P}{dt}$ for certain process P is determined by the following equation:

$$\frac{dN_P}{dt} = L \times \sigma_p \tag{3.1}$$

where:

- L is the luminosity.
- σ_p is the cross section of the process P.

The luminosity L is defined by:

$$L = \frac{N_b^2 n_b f \gamma}{4\pi \beta^* \epsilon_n} \times R \tag{3.2}$$

where:

- N_b is the number of particles per bunch.
- n_b is the number of bunches per beam.
- f is the revolution frequency, indicating how often the bunches circle around.

[3] B-hadrons are hadrons that have a b-quark.

[4] Matter-antimatter asymmetry is indicated by the inequality between the baryonic states (matter) and antimatter in the universe.

- γ is the relativistic gamma factor.
- β^* is the beam beta function at the collision point.
- ϵ_n is the normalised transverse beam emittance.
- R is the geometric reduction factor.

where, the geometric reduction factor R is defined by:

$$R = \frac{1}{\sqrt{1 + \frac{\theta_c \sigma_z}{2\sigma_T}}} \tag{3.3}$$

where:

- θ_c is the crossing angle between two beams.
- σ_z is the RMS of the longitudinal beam size.
- σ_T is the RMS of the transversal beam size.

The integrated luminosity (L_{int}) is the integral of the luminosity with respect to time, it is defined as:

$$L_{int} = \int_{\Delta t} L dt \tag{3.4}$$

During LHC operation, the integrated luminosity increased over the years. By the end of Run 1, the integrated luminosity was 30 fb^{-1} , and by the end of Run 2 in 2018, it had reached 190 fb^{-1} . It is expected to be 450 fb^{-1} by the end of Run 3.

3.3. ATLAS Detector

The ATLAS experiment is a multipurpose particle detector with a forward–backward symmetric cylindrical geometry and a near 4π coverage in solid angle. It is a proton-proton collision experiment where two proton beams, each with an energy of around 7 TeV, collide in opposite directions at the interaction point located at the center of the ATLAS detector, producing a large number of particles. It is a general-purpose experiment and is not specialized in any specific area of particle physics. The ATLAS experiment aims to further understand the Standard Model and address its unresolved issues. Additionally, it seeks to obtain more precise measurements of the properties of discovered particles. One of its main interests is exploring physics beyond the Standard Model, particularly since all particles predicted by the Standard Model have been discovered, raising questions about what might be found at higher energies. The biggest achievement of the ATLAS experiment is the discovery of the Higgs boson in 2012.

The Figure 3.3 shows the ATLAS detector and its components. The detector is 44 m long and has a diameter of 25 m. It is located in octant 1 of the LHC. A detailed description of the ATLAS detector can be found in reference [37]. The ATLAS detector consists of many sub-detectors, as shown in Figure 3.3, and is classified into four systems:

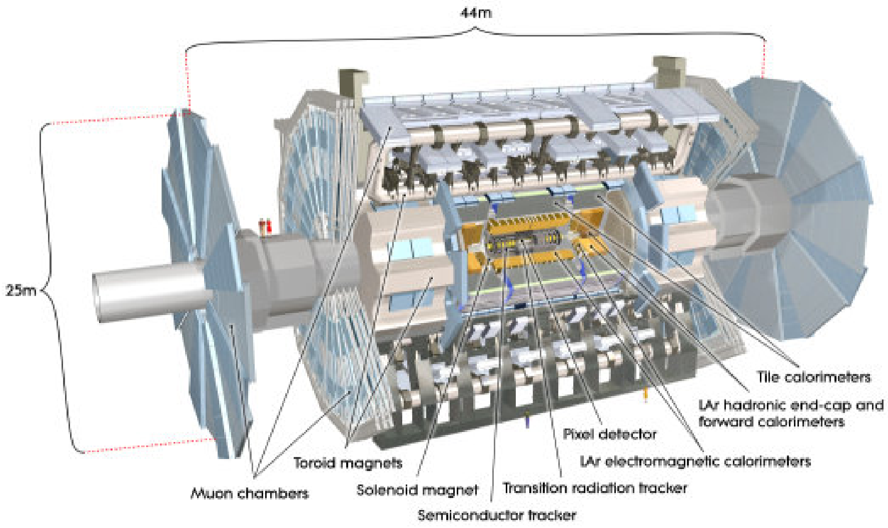

44m
25m
Tile calorimeters
LAr hadronic end-cap and forward calorimeters
Pixel detector
LAr electromagnetic calorimeters
Toroid magnets
Muon chambers
Solenoid magnet
Transition radiation tracker
Semiconductor tracker

Figure 3.3.: ATLAS detector and all its sub-detectors [37].

- Inner Detector.
- Calorimeters.
- Muon Spectrometer.
- Magnet system.

Each system has its own specific function.

The Inner Detector consists of the Pixel detector, SCT (Semi-Conductor Tracker), and TRT (Transition Radiation Tracker). The Calorimeter system is composed of the Electromagnetic Calorimeter and the Hadronic Calorimeter. The Muon Spectrometer system includes the Muon Spectrometer detector. The Magnetic system is composed of superconducting magnets that provide up to 3.5 T magnetic field to bend the trajectories of charged particles in order to measure their momenta and electrical charge. All the sub-detectors of the ATLAS experiment are introduced in this section.

3.3.1. Inner detector

The Inner Detector is composed of the Pixel detector, SCT, and TRT, as shown in Figure 3.5. Its purpose is to reconstruct a precise trajectory of the resulting charged particles created by the collision of the two beams at the interaction point of the ATLAS experiment.

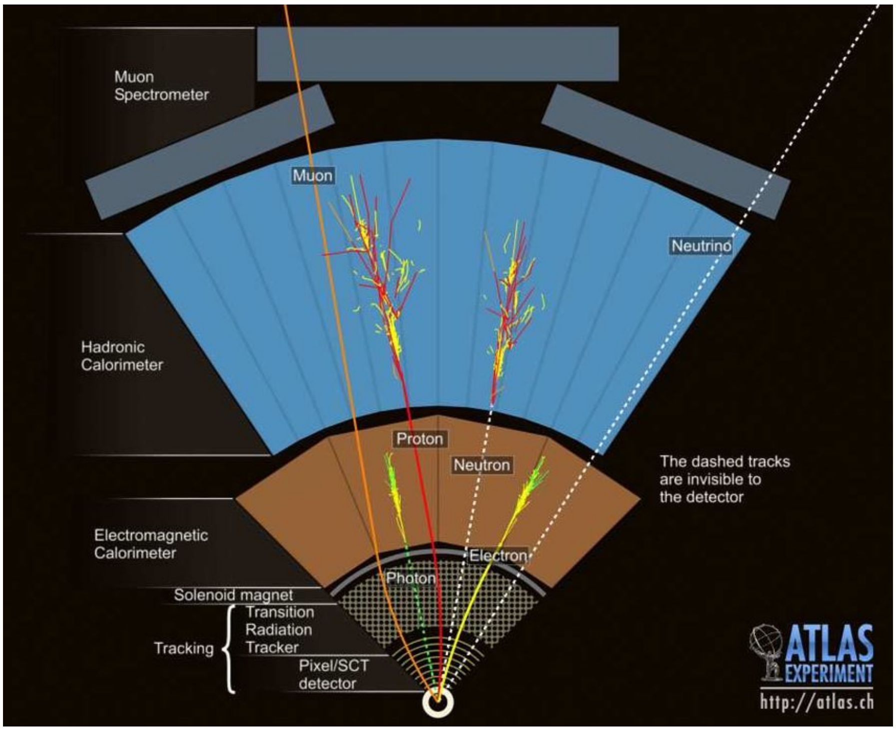


Figure 3.4.: The ATLAS detector cross-section schematic shows the interaction of every type of particle. The Tracking System (Inner Detector) detects the trajectory of charged particles such as electrons, muons, protons, and charged hadrons. The Electromagnetic Calorimeter detects electrons and photons through the showers they create. The Hadron Calorimeter detects protons, neutrons, and hadrons via the showers they produce. The muons are detected by the Muon System, while neutrinos are estimated via missing energy calculations since they cannot be detected by any subsystem and do not leave any tracks in the detector [38].

Pixel detector

The pixel detector [41] is the closest detector to the beamline and plays an important role in track and vertex reconstruction. It detects the tracks of charged particles, as shown in Figure 3.4. Figure 3.6 presents the pixel detector, which is 1442 mm long and 430 mm wide. It is composed of three barrel layers: the closest layer has a radius of 50.5 mm, the second layer has a radius of 88.5 mm, and the third layer has a radius of 122.5 mm. Additionally, it has two endcaps, each consisting of three discs. This structure allows the pixel detector to cover a pseudorapidity range of $|\eta| < 2.5$ and an active area of about $1.7\,\text{m}^2$. The three layers of the pixel detector can provide three precise positions for the traveling particle. It contains 1744 pixel modules. Each pixel module is constructed of a 250 µm n-on-n silicon sensor and hosts 16 front-end chips with 2880 electronics channels

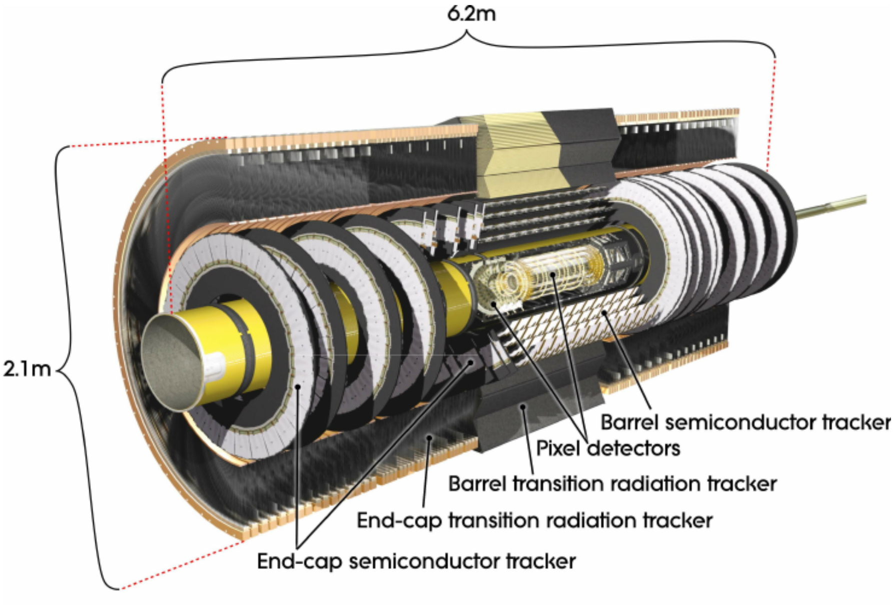

6.2m
2.1m
Barrel semiconductor tracker
Pixel detectors
Barrel transition radiation tracker
End-cap transition radiation tracker
End-cap semiconductor tracker

Figure 3.5.: The Inner detector and its all sub-detectors [39].

per FE chip. The sensor has 47232 pixels, with the majority having a pixel size of 50 µm × 400 µm and about 10 % having a pixel size of 50 µm × 600 µm.

After the first shutdown, an additional layer called the insertable B-Layer (IBL) [42] was inserted between the beamline and the first layer of the pixel detector at a radius of 33.2 mm to enhance tracking performance.

SCT

The SCT [43] is the middle detector in the Inner Detector as shown in Figure 3.5. It consists of 4 barrel layers and 2 endcaps, each endcap having 9 discs. SCT layers are positioned between a radius of 299 mm and 560 mm away from the beamline. The SCT is built of 4088 silicon detector modules with a total of 6.3 million strips (single-sided p-on-n silicon microstrip detectors), where the barrel layers host 2112 modules and each endcap has 988 modules. These modules are double-sided, which means the SCT can provide 8 position measurements per particle track. The SCT covers an area with a pseudorapidity range of $|\eta| < 2.5$.

TRT

The Transition Radiation Tracker (TRT) [44] is the only gaseous detector in the inner detector. It is the outermost detector in the inner detector as shown in the Figure 3.5. It consists of about 298,000 drift tubes named straws. These straws are composed of Kapton reinforced with thin carbon fiber, are 4 mm in diameter, and represent the cathode, supplied with −1.5 kV. Inside each straw is a 31 µm diameter wire of gold-

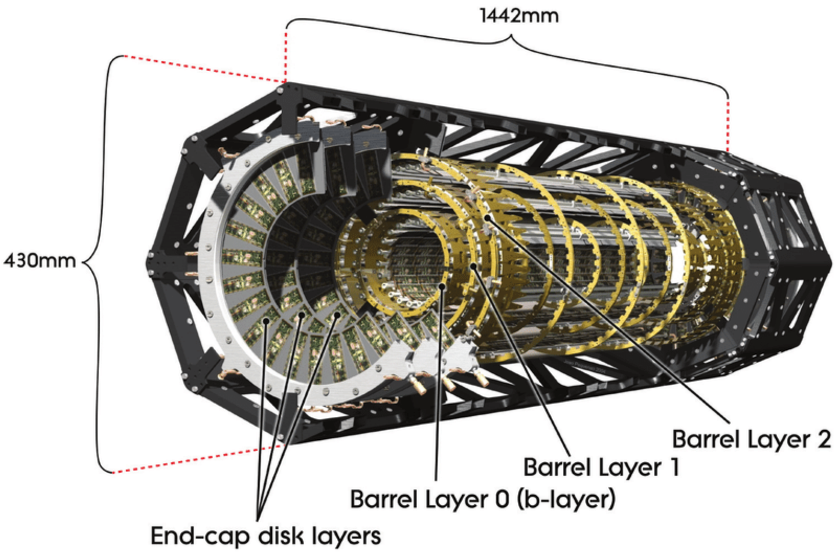


Figure 3.6.: The pixel detector and its components [40].

plated tungsten, considered the anode, which is electrically grounded. The straws are filled with a gas mixture of 70 % Xe, 27 % CO_2, and 10 % O_2. The straws have about 200 µm track position uncertainty.

3.3.2. Calorimeter

The Figure 3.7 presents the calorimeter system that surrounds the inner detector. Its main purpose is to measure the energy of the particles captured by the calorimeter. It is composed of the electromagnetic calorimeter (ECAL) and the hadronic calorimeter (HCAL).

Electromagnetic calorimeter

The electromagnetic calorimeter [46] is the innermost part of the calorimeter system and surrounds the inner detector as shown in Figure 3.7. As shown in Figure 3.4, photons and electrons are detected in the electromagnetic calorimeter by losing their energy through a sequence of consecutive Bremsstrahlung and pair creation reactions, resulting in a shower in the electromagnetic calorimeter. It is a sampling detector consisting of lead absorbers and thin layers of liquid argon(LAr), and it is kept at −184℃ .

It is composed of a Barrel section and two endcaps. The Barrel section is 6.4 m long with an inner radius of 1.15 m and an outer radius of 2.25 m. It has a thickness equivalent to 22 radiation lengths. The endcaps have a radius of 2.077 m and a thickness of 0.632 m, equivalent to 24 radiation lengths.

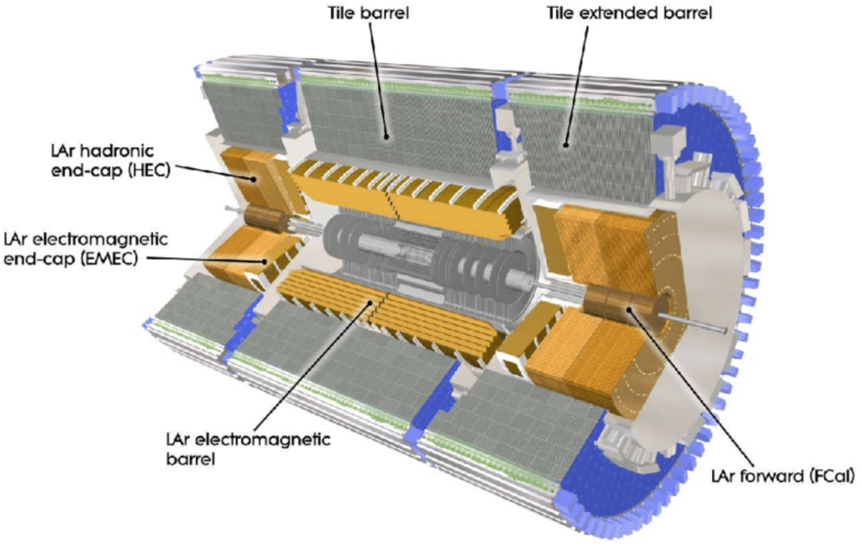

Tile barrel
Tile extended barrel
LAr hadronic end-cap (HEC)
LAr electromagnetic end-cap (EMEC)
LAr electromagnetic barrel
LAr forward (FCal)

Figure 3.7.: The ATLAS calorimeter system and its components, which consist of the electromagnetic calorimeter surrounded by the hadronic calorimeter [45].

Hadronic calorimeter

The Hadronic calorimeter is the outermost sub-detector of the Calorimeter system, as shown in Figure 3.7. It stops all particles except muons and neutrinos, as shown in Figure 3.4. It weighs about 2900 tons and is composed of the following three parts:

- Tile Sampling Calorimeter (TILE).
- LAr hadronic end-cap calorimeter (HEC).
- LAr Forward Calorimeter (FCal).

The Tile calorimeter [47] has a cylindrical shape with an inner radius of 2.280 m and an outer radius of 4.230 m. It is a scintillator tile sampling calorimeter, consisting of 420000 plastic scintillating tiles combined with steel absorber. It consists of a central barrel structure covering the region $|\eta| < 1.0$ and two extended barrels that cover the region $0.8 < |\eta| < 1.7$. The Hadronic endcap is composed of two wheels, each with an outer radius of 2.090 m [46] covering the region $1.5 < |\eta| < 3.2$. The Forward calorimeter [46] consists of three modules in each end-cap; the first uses copper as absorber, for more precise EM energy deposit measurements, while the other two use tungsten and are optimised for hadronic interactions. Each module has a radius of 0.455 m and a thickness of around 0.450 m.

The Hadronic Calorimeter covers a pseudorapidity range of $|\eta| < 4.9$.

3.3.3. Muon system

The Muon system [48] is composed of a muon spectrometer, which is a set of gaseous detectors used to detect muons, as shown in Figure 3.4, in order to provide precise measurements of muons' momenta. It consists of the following gaseous chambers:

- Monitored Drift Tubes (MDT).
- Cathode Strip Chambers (CSC).
- Resistive Plate Chamber (RPC).
- Thin Gap Chamber (TGC).

The MDTs and CSCs provide precise tracking determination of the muon with an accuracy of less than 0.1 mm, resulting in good muon momentum measurements. RPCs and TGCs are used for triggering purposes, with decisions produced by the RPC and TGC within 2.5 µm.

3.3.4. Magnetic System

The magnetic system [49] is used to bend the trajectories of charged particles to allow the detector to measure the charge and momentum of the particles using their curved paths.

The magnetic system is composed of two large superconducting magnet systems:

- Solenoid Magnet: Surrounds the inner detector and provides a 2 T magnetic field.
- Toroid Magnet: Consists of two magnets at the ATLAS detector's ends and one surrounding the ATLAS detector, providing a magnetic field of up to 3.5 T. It assists in the measurement of muon momentum.

3.3.5. Trigger and Data Acquisition system

The ATLAS detector detects more than 1 billion proton-proton collisions per second. Not all the outcomes of these collisions are of interest. The Trigger and Data Acquisition system (TDAQ) [50] selects the interesting collision events via two stages:

- First-level hardware trigger (L1): This is obtained from the Calorimeters and Muon Spectrometers' outcomes, where the decision needs less than 2.5 µs to be taken. Then the event will be stored in storage buffers. If the event is accepted according to certain trigger criteria, it will pass to the second-level trigger (L2), which can keep only 100,000 events per second.
- Second-level software trigger (HLT): This utilizes about 40,000 CPU cores to reduce the selected events from 100,000 per second to around 1000 events per second via collision event analysis and data from certain sub-detectors. The events selected by HLT will be permanently stored.

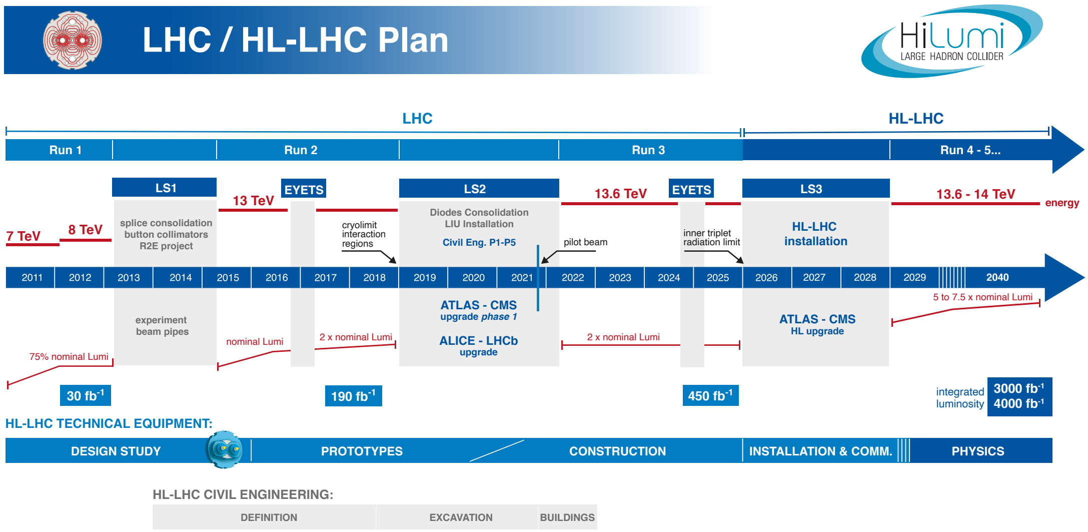


Figure 3.8.: LHC schedule and its future upgrades [51].

3.4. HL-LHC and ATLAS upgrade

The detectors of the LHC collect data in phases known as Runs. After each Run, there is an opportunity to upgrade the detectors to improve performance for the next Run. This is necessary because the next Run is expected to operate with higher energy beams, higher luminosity, or both, to achieve and discover new physics. Between consecutive Runs, there is a Long Shutdown (LS) phase during which these upgrades are implemented.

At the time of writing this thesis, Run 3 has not yet been completed; it is expected to conclude at the end of 2025. The next upgrade of the LHC will be called the High-Luminosity LHC (HL-LHC). Run 3 will be followed by Long Shutdown 3 (LS3), during which the HL-LHC installation will take place. Run 4 is scheduled to start in 2029, as shown in Figure 3.8.

The HL-LHC [52] project was initiated in 2010 and approved in 2016. In the current LHC, the nominal luminosity is about $1 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. In the HL-LHC era, this luminosity will be 5-7 times higher. The expected integrated luminosity at the end of Run 3 will be $450\,\mathrm{fb}^{-1}$, and it is expected to reach $3000\,\mathrm{fb}^{-1}$ after 10 years of operation starting from 2029. The beam energy will reach approximately 14 TeV.

The ATLAS experiment expects the average pile-up[5] to increase to up to 200 at a rate of 40 MHz, resulting in more collision events to detect and higher expected radiation damage.

[5] Pile-up is the number of proton-proton collisions per bunch crossing.

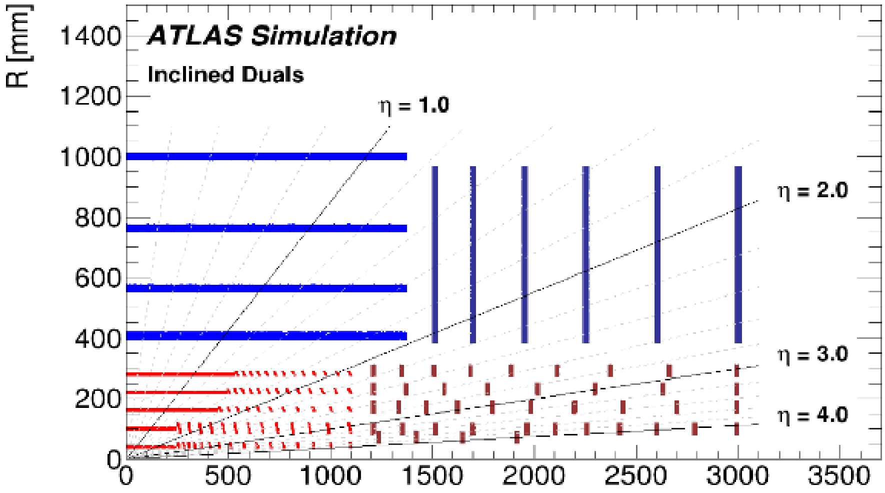


Figure 3.9.: Schematics of a quarter of the Inner Tracker, showing its two sub-detectors: the ITk Pixel detector (red) and the ITk Strips detector (blue). The z-axis represents the beam line, and the point (0,0) represents the interaction point (midpoint) of the ATLAS experiment [54].

3.4.1. Inner Tracker (ITk)

The ATLAS experiment will undergo a significant upgrade to prepare for the HL-LHC era in Run 4. One of the most interesting changes will be in the tracking system, where the entire inner detector (ID), including its components such as the pixel detector, SCT, and TRT, will be replaced by an all-silicon detector named the Inner Tracker (ITk) [53].

As shown in Figure 3.9, the ITk [55] is composed of the ITk pixel detector, which consists of 5 layers, surrounded by the ITk strips detector, which includes one barrel of 4 layers and 2 endcaps, each endcap consisting of 6 layers. The ITk covers an area with pseudorapidity $|\eta| < 4$.

3.4.2. ITk pixel detector

The Figure 3.10 shows the schematics of the ITk Pixel detector, which consists of 5 layers. The ITk Pixel detector [57] covers an area of $13\,\mathrm{m}^2$ and a pseudorapidity range of $|\eta| < 4$. It will host about 9700 modules and around 5 billion readout channels.

The ITk Pixel detector is composed of 3 systems, which are:

- Inner System (IS): This is the first two layers of the ITk Pixel detector, which hosts flat staves and rings. It covers an area of $2.4\,\mathrm{m}^2$ and hosts around 2600 modules. It will be replaced when an integrated luminosity of $2000\,\mathrm{fb}^{-1}$ is reached. It consists of two layers:
  - L0: This is the inner layer of the IS, with a radius of 39 mm. It hosts

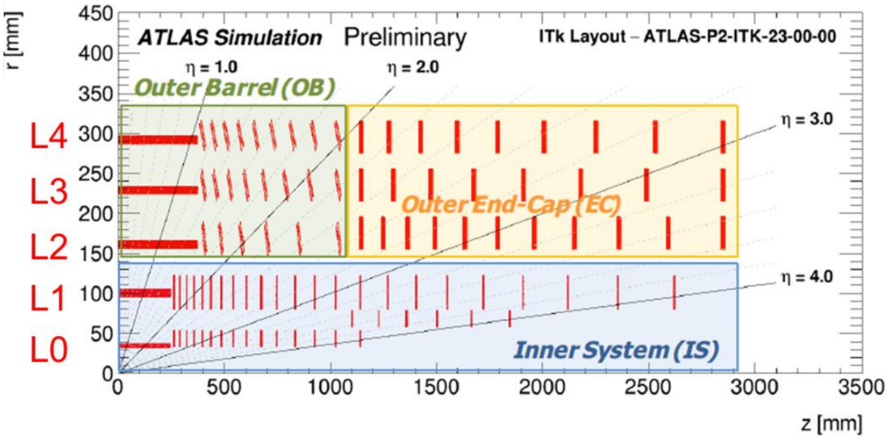


Figure 3.10.: Schematics of a quarter of the ITk Pixel detector, showing its components [56].

3D single modules [58][6] with a pixel size of 25 µm × 100 µm for flat modules and 50 µm × 50 µm for endcap modules. These 3D modules are triples, each having 3 front-end chips.

- L1: This is the second and outermost layer of the IS, with a radius of 85 mm. It will have only n-in-p planar quad modules, each containing 4 front-end chips with a size of 50 µm × 50 µm.

- Outer Barrel (OB): It consists of 3 layers, namely L2, L3, and L4, with flat staves and inclined rings having radii of 155 mm, 213 mm and 271 mm respectively. It holds 4772 n-in-p quad modules and covers an area of 6.94 m^2.

- Outer End-Caps (EC): It also consists of 3 layers, L2, L3, and L4, with vertical rings having the same radii as in the outer barrel. It hosts 2344 n-in-p quad modules and covers an area of 3.64 m^2.

[6] In the 3D modules, the electrodes are columns within the bulk that can be positioned closer to each other. In contrast, in the planar sensor, there are two electrodes of different types separated by a bulk of a different type.

CHAPTER 4

Physics of Particle Detection and Semiconductor

In the HL-LHC upgrade, the ATLAS tracking system will be replaced by an all-silicon detector. To understand how the silicon detector ITk will perform during the Run, it is important to grasp two main particle detection physics concepts. The first concept is understanding how particles interact with matter, and the second is knowing the basics of semiconductors and how they function.

In this chapter, the interaction of particles with matter is discussed in Section 4.1, and the behavior of semiconductor detectors is introduced in Section 4.2.

4.1. Interactions of Particles with Matter

Each particle interacts with matter according to its type and the type of matter. In this context, all particles can be classified into three categories: heavy particles, where their mass is much greater than the mass of an electron ($m \gg m_e$), electrons, and photons.

In this section, the interaction of heavy particles with matter, the interactions of electrons and the interactions of photons are introduced.

4.1.1. Heavy particle

Heavy charged particles are particles with a mass significantly higher than the electron mass ($m \gg m_e$). These particles lose energy through their interaction with electrons in the valence band of atoms. This occurs either via ionization, which liberates a free electron and creates an ion, or via excitation, where the electron is excited by the incoming heavy charged particle and then de-excited by emitting a photon.

These mechanisms are described by Equation 4.1, known as the Bethe-Bloch formula [59]:

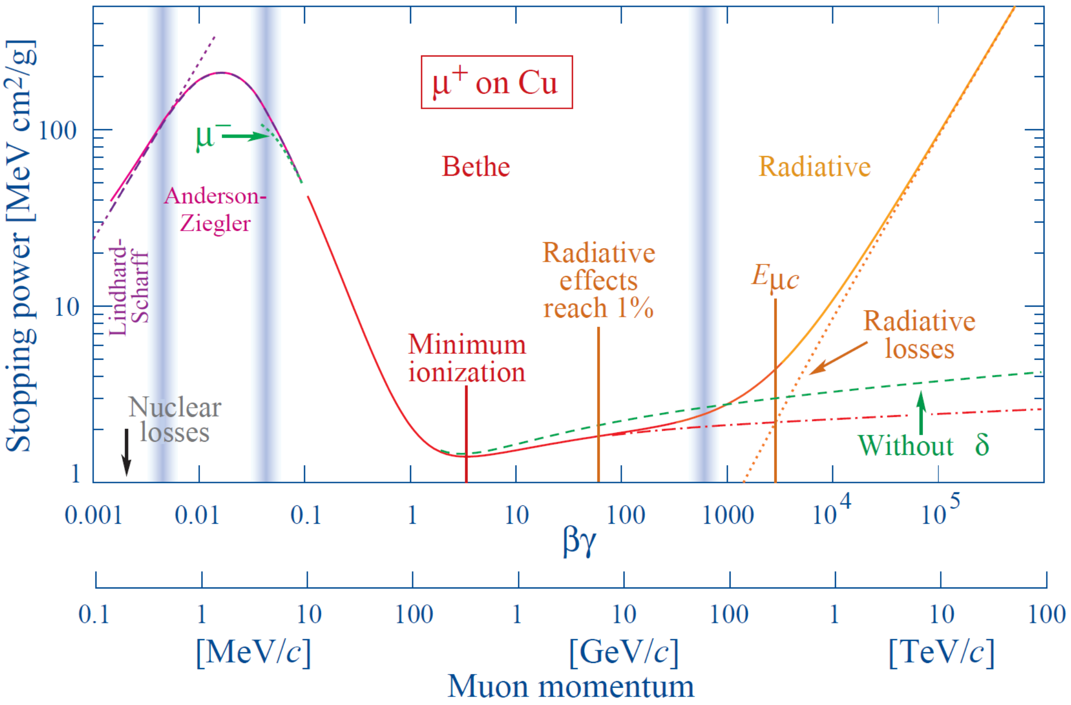


Figure 4.1.: The variation of the energy loss as a variation of $\beta\gamma$ value of the incoming heavy particle muon in Copper [2].

$$\langle -\frac{dE}{dx} \rangle = Kz^2 \frac{Z}{A}\frac{1}{\beta^2}\left[\frac{1}{2}\ln\frac{2m_ec^2\beta^2\gamma^2W_{max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2}\right] \tag{4.1}$$

with $K = 4\pi {r_e}^2 m_e c^2 N_A$
where:

- r_e is the classical electron radius.
- m_e is the electron mass.
- c is the speed of light.
- N_A is the Avogadro's number.
- z is the charge of the incoming particle.
- Z is atomic number of the material.
- A is the atomic weight of the material.
- β and γ are the relativistic factors.

- W_{max} is the maximum energy transfer in a single collision.
- I is the mean excitation potential.

The Bethe-Bloch equation, described in Figure 4.1, is valid in the region of $0.1 < \beta\gamma < 1000$. For $\beta\gamma < 0.1$, the Bethe-Bloch equation is not valid because other processes dominate over ionization. For higher energies ($\beta\gamma > 1000$), energy loss due to radiation becomes more dominant than energy loss due to ionization. Thus, the Bethe-Bloch equation is not valid in the region of $\beta\gamma > 1000$.

The Bethe-Bloch equation is composed of three parts, which are shown in Figure 4.1. In the region of $0.1 < \beta\gamma < 1$, the sharp drop in energy loss is related to the $\frac{1}{\beta^2}$ factor of the Bethe-Bloch equation. This behavior occurs because momentum transfer increases with effective interaction time [60]. Following this, there is a minimum value of energy loss, with coordinates approximately at $\beta\gamma \approx 3 - 4$ and $\langle -\frac{dE}{dx} \rangle = 1.5\,\mathrm{MeVg^{-1}cm^{-2}}$. The particle corresponding to this minimum is called a minimum ionizing particle (*mip*), which is important for determining the smallest signal that a detector can expect. After this minimum, the energy loss increases, which is described by the ln factor in the Bethe-Bloch equation. The last term of the Bethe-Bloch equation is $\delta(\beta\gamma)^2$, where $\delta(\beta\gamma)$ is called the attenuation coefficient. This coefficient decreases the amount of increased energy loss due to polarization effects. As shown in Figure 4.1, the benefit of the attenuation factor is the reduction of the slope of increased energy loss, leading to the consideration of all particles with high $\beta\gamma$ in Bethe-Bloch as mips.

4.1.2. Electron

The electron (and positron) undergoes many types of interactions with matter. The main interactions of electrons are ionization and bremsstrahlung, as shown in Figure 4.2. In the ionization interaction, the incoming electron releases an electron from the valence shell of the atom, changing it into an ion. In bremsstrahlung, the incoming electron is deflected by the effect of another charged particle in the presence of a nucleus, leading to the emission of a photon [61].

There are also many other interactions of the electron with matter, such as Møller interaction [62], Bhabha interaction [62], and positron annihilation interaction [63]. However, their contribution to energy loss is negligible compared to ionization and bremsstrahlung interactions, as shown in Figure 4.2.

As shown in Figure 4.2, the energy loss due to ionization is dominant at low energies, while the energy loss due to bremsstrahlung is dominant at high energies. The intersection point defines the critical energy, where the energy loss due to ionization is equal to the energy loss due to bremsstrahlung.

$$E_c \approx \frac{800}{Z} MeV \tag{4.2}$$

The energy loss is:

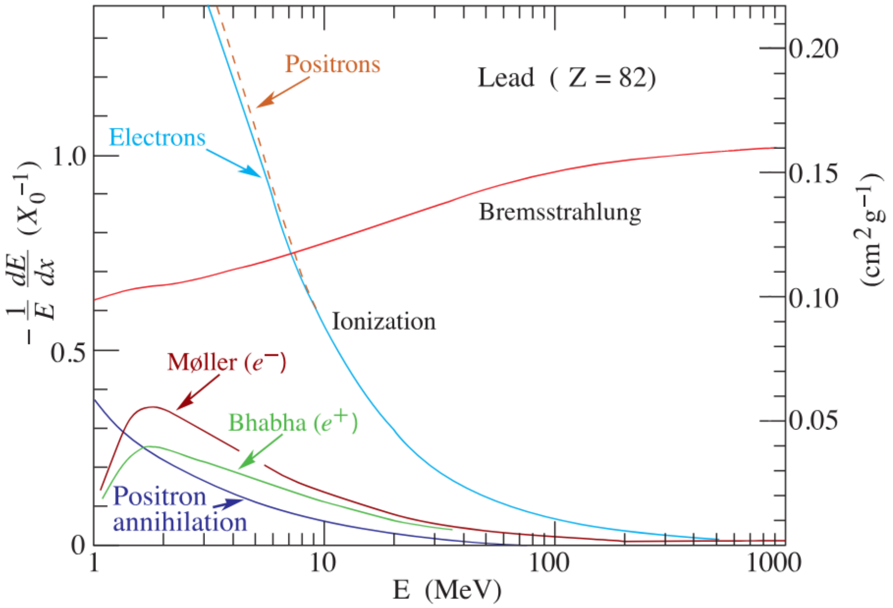


Figure 4.2.: The contribution of all interactions of electrons with matter (Lead) to the energy loss as a function of the electron's energy [59].

$$\langle -\frac{dE}{dx} \rangle = \frac{E}{X_0} \tag{4.3}$$

where X_0 is called the radiation length. It is the distance over which the electron's energy is reduced by $\frac{1}{e}$ due to bremsstrahlung.

The radiation length is material-dependent. for example, the radiation length of silicon is 9.370 cm [2].

4.1.3. Photon

The photons are not detected directly by the detectors; they interact with matter, and the resulting particles are detected. There are three dominant interactions of photons with matter, which are:

- Photoelectric effect.
- Compton scattering.
- Pair creation.

The photoelectric effect occurs when the photon is absorbed by an electron, leading to the electron being ejected from the atom. The ejected electron will have an energy equal to the photon's energy minus the binding energy.

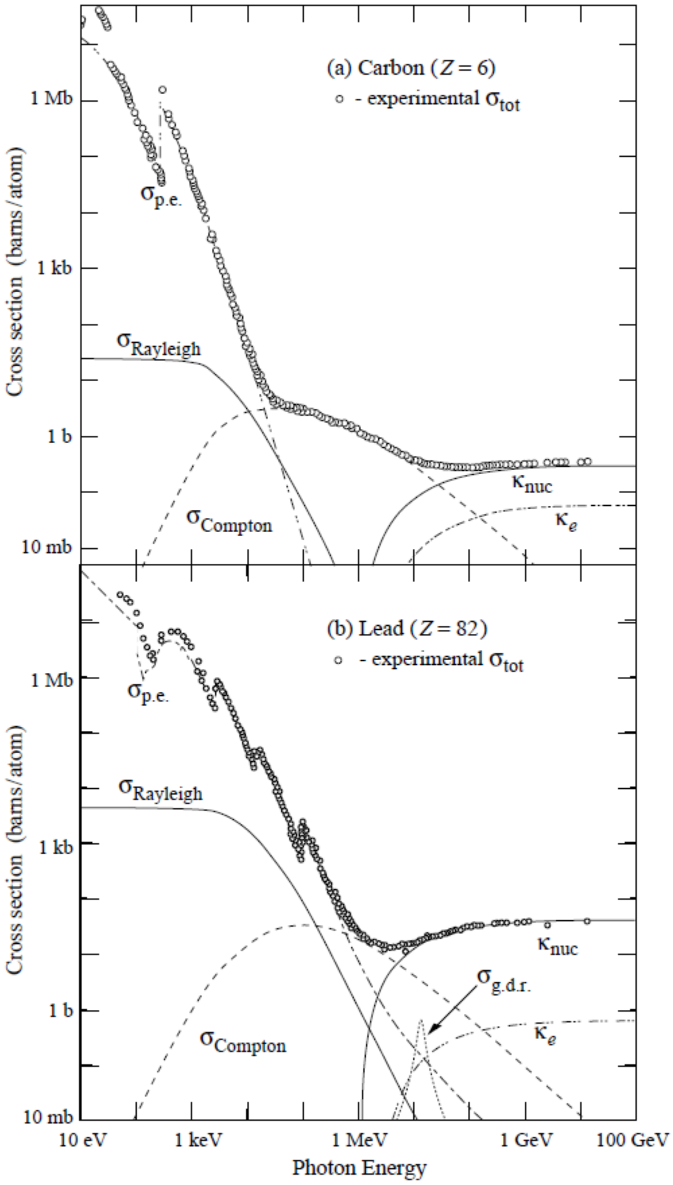


Figure 4.3.: The cross sections of all interactions of photons with Carbon (a) and Lead (b) as a function of the photon's energy. P.e stands for photoelectric effect, κ_e stands for pair production in the electron field, and κ_{nuc} stands for pair production in the nucleus field [2].

Compton scattering is the scattering of the incoming photon by an electron, where the transferred energy from the photon to the electron depends on the scattering angle.

Pair creation happens when the energy of the photon is greater than twice the mass of the electron. In this case, the photon can create an electron-positron pair in the presence of a nucleus.

As shown in Figure 4.3, the photoelectric effect is dominant at low energies in the range of keV. Compton scattering is dominant when the photon energies are in the range of MeV, and at higher energies, pair creation processes are dominant.

Overall, the photon beam intensity decreases exponentially with the penetration depth x

$$I = I_0 e^{-\mu x} \tag{4.4}$$

where:

- I_0 is the initial intensity of the photon beam.
- μ is an attenuation coefficient.

4.2. Physics of Semiconductor detector

All solids are classified into three types: insulators, semiconductors, and conductors. They differ in their energy-band structure, which is composed of a valence band and a conduction band. The valence band is the energy band that contains the valence electrons of the atom's outermost shell. The conduction band is the energy band that contains free electrons.

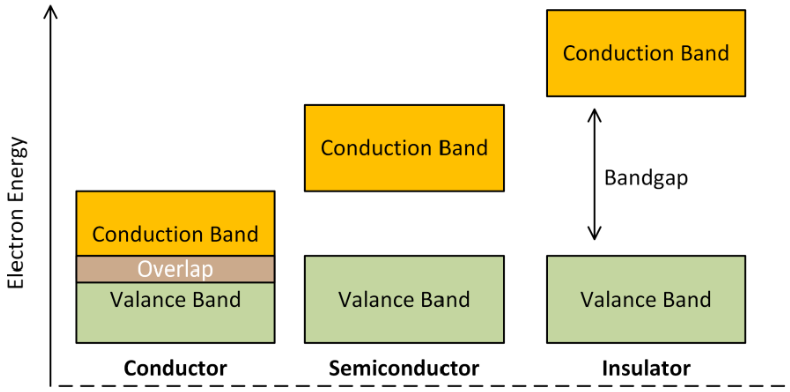


Figure 4.4.: The schematic energy band structure of conductors, insulators, and semiconductors [64].

As shown in Figure 4.4, the valence band and the conduction band overlap in the conductor, allowing electrons to move freely from the valence band to the conduction band. In the case of an insulator, the conduction band and the valence band are separated by a large band gap, which makes the transition of electrons from the valence band to the conduction band difficult. Meanwhile, in the semiconductor, the energy band gap is smaller, allowing for electron transitions from the valence band to the conduction band. The property that differentiates these three types is electrical conductivity, with the electrical conductivity of the semiconductor typically ranging from $10 \times 10^{-8}\,\mathrm{A\,V^{-1}\,cm}$ to $10 \times 10^{3}\,\mathrm{A\,V^{-1}\,cm}$ [65].

The transition of an electron from the valence band to the conduction band leaves a hole in the valence band.

Semiconductors can be elements from Group IV of the periodic table, such as silicon, or compounds from Groups III-V or II-VI. Silicon is the most widely used semiconductor due to its abundance [1]. The energy band gap of silicon is 1.12 eV at 300 K [66].

In a semiconductor detector, incoming particles undergo an ionization process that creates electron-hole pairs. The energy required to create an electron-hole pair in silicon is 3.69 eV [67][2].

4.2.1. PN Junction

There are two categories of semiconductors: intrinsic semiconductors [68] and extrinsic semiconductors [68]. An intrinsic semiconductor is a semiconductor in its pure state without any added impurities, such as silicon. In contrast, an extrinsic semiconductor is a semiconductor that has impurities added through a process called doping. Doping involves adding a small amount of an element to change the semiconductor from an intrinsic semiconductor to an extrinsic semiconductor.

There are two types of extrinsic semiconductors: n-type semiconductors and p-type semiconductors.

An intrinsic semiconductor can be doped by adding a group V element (for example, phosphorus), which results in more free electrons because it has five electrons in its valence shell. The group V element is identified as a donor, and the majority carriers are the electrons. This doped extrinsic semiconductor is considered an n-type semiconductor [69].

An intrinsic semiconductor can also be doped by adding a group III element (for example, aluminum), which creates more holes because it has only three electrons in its valence shell. The group III element is identified as an acceptor, and the majority carriers are the holes. This doped extrinsic semiconductor is considered a p-type semiconductor [69].

The pn-junction is formed when a p-type semiconductor and an n-type semiconductor are in contact. As shown in Figure 4.5, the p-type semiconductor has a majority of holes

[1] Silicon is extracted from desert sand, which typically contains 25 % to 50 % silicon dioxide. The refining process separates silicon from other materials.

[2] Silicon requires 3.69 eV to create an electron-hole pair. This energy is greater than the band gap energy of 1.12 eV because silicon is an indirect semiconductor. This means the minimum of its conduction band and the maximum of its valence band do not occur at the same momentum value. Therefore, electrons must transition in both the momentum and energy axes to become free.

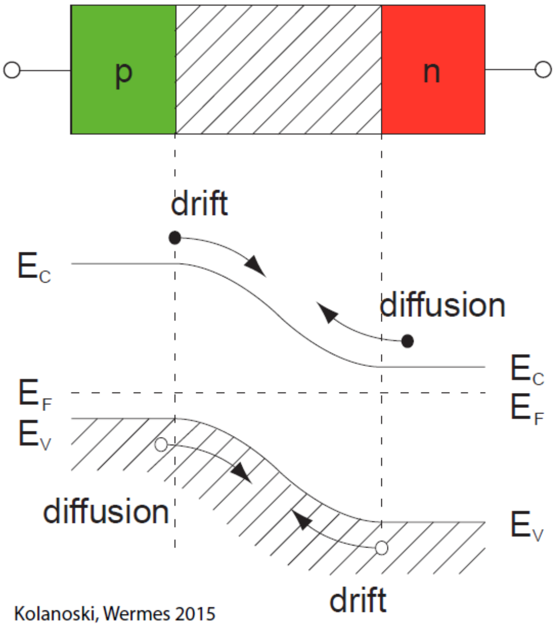


Figure 4.5.: The schematic diagram of a pn-junction and the movement of electrons and holes through the pn-junction [60].

and the valence band is close to the Fermi level. In the n-type semiconductor, the major carriers are the electrons and the conduction band is near the Fermi level. This leads to the diffusion of electrons from the n-type semiconductor to the p-type semiconductor and the diffusion of holes from the p-type semiconductor to the n-type semiconductor. The electrons and holes recombine at the contact between the p-type region and the n-type region, forming a zone called the depletion zone, which is free of charge carriers.

After the formation of the depletion zone and once thermal equilibrium is reached, an electric field with a drift current and a built-in voltage V_{bi} is created due to the distribution of charges.

$$V_{bi} = \frac{e}{2\epsilon\epsilon_0} x_p^2 \frac{N_A}{N_D} (N_A + N_D) \tag{4.5}$$

where:

- ϵ is the absolute permittivity.
- ϵ_0 is the vacuum permittivity.
- x_p is the width of the depletion zone in the p-doped region.
- N_A is the doping concentration for p type.
- N_D is the doping concentration for n type.

Thermal equilibrium can be broken by applying an external voltage. If an external negative voltage potential is applied to the n-doped part and the p-doped part has a positive voltage potential with respect to the n-doped part (forward bias), then the electrons in the n-doped part are pushed towards the depletion zone and the holes in the p-doped part are also pushed towards the depletion zone, leading to a shrinking of the depletion region. In the opposite case, if an external negative voltage potential is applied to the p-doped part and the n-doped part has a positive voltage potential with respect to the p-doped part (reverse bias), then the electrons in the n-doped part are attracted towards the n-doped region and the holes in the p-doped part are attracted towards the p-doped region, leading to an expansion of the depletion region.

The width of the depletion region d with external voltage V_{ext} is :

$$d = \sqrt{(V_{bi} + V_{ext})\frac{2\epsilon\epsilon_0}{e}\frac{(N_A + N_D)}{N_A N_D}} \tag{4.6}$$

4.2.2. Hybrid Pixel Detectors

There are two technologies for pixel detectors: monolithic pixel detectors and hybrid pixel detectors. In a monolithic pixel detector, the sensor and readout parts of the detector are integrated, allowing signals generated in the sensor to pass directly to the readout part.

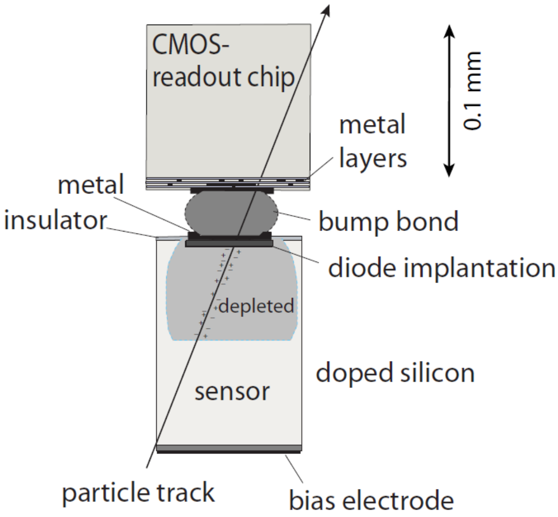


Figure 4.6.: The schematic diagram of the hybrid pixel detector [60].

In a hybrid pixel detector, like the one used in the present ATLAS Inner Detector and Inner Tracker, each pixel sensor is connected to its readout part by a bump bond, a conductive material designed to transfer the signal from the pixel sensor to the readout part, as shown in Figure 4.6. Applying a reverse voltage bias to the sensor pixel depletes it. When incoming particles pass through the depletion region, electron-hole pairs are created. These electrons and holes drift towards the electrodes, generating a current. This current is then directed to the readout chip.

Every pixel sensor has its own readout chip. The electric signal arriving at the readout chip first passes through the analog part, which includes amplifiers to amplify the signal and threshold discriminators to eliminate noise signals. Afterward, the pulse is routed to the digital part to measure its length. This data is then forwarded to a buffer shared among many pixels for later processing and analysis. Due to the high bandwidth requirements of the ATLAS detector, a parallel readout is essential.

CHAPTER 5

ITk Production Database

The ITk detector discussed in Section 3.4.1, comprises numerous elements that require tracking and testing during the production phase. To manage this process, a database called the ITk Production Database (ITk PD) has been created.

This chapter discusses the development and implementation that I did within ITk PD as my ATLAS qualification task for the ITk Outer Barrel (OB) for the OB loaded local supports production phase, where OB loaded local supports are introduced in Section 5.2. First, the ITk Pixel Module and sensors are discussed in Section 5.1, followed by an introduction to the ITk Outer Barrel is provided in Section 5.2, followed by a discussion of the ITk Production Database and its structure in Section 5.3. The OB includes several components such as OB bare local supports, OB loaded local supports, and OB loaded module cell, which are detailed along with their corresponding ITk PD implementations in Section 5.4, Section 5.5 and Section 5.6, respectively. Additionally, Further ITk PD implementations to facilitate dealing with the OB components are introduced in Section 5.7.

5.1. ITk Pixel Module

ITk pixel sensors adopt hybrid pixel technology, where the sensors are connected to the readout chips (front-end chips) via bump bonds to form the so-called bare module.

A printed circuit board (PCB) called a flex is glued to the back side of the sensor, as shown in Figure 5.1. The flex is a PCB hosting Surface Mount Devices (SMDs). It is connected to the readout system on one side and to the front-end chips on the other side. It provides the front-end chips with electrical connections to the readout system, the monitor and control system, and the power supply system. The flex is connected to the front-end chips via wire bonds, as shown in Figure 5.1, through a process called wire bonding.

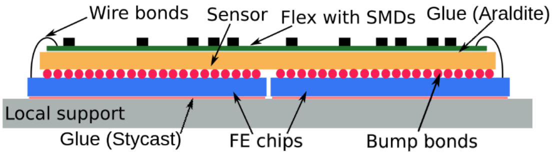


Figure 5.1.: The cross section of ITk pixel module [70].

5.1.1. ITk Pixel sensor

There are two types of sensors used in the ITk pixel detector: planar sensors and 3D sensors.

The planar sensors used are n+ type implants and p- type substrates, as shown in Figure 5.2. They will be used in layer L1 of the ITk pixel detector with an active thickness of 100 µm, and in layers L2, L3, and L4 with an active thickness of 150 µm [57]. The pixel pitch size is $50 \times 50\,\mu m^2$ [57].

The 3D sensors are composed of n+ type and p+ type electrodes placed in a p- type substrate, as shown in Figure 5.2. This technology gives the 3D sensors fast charge collection ability since the drift distance is small. This property makes the 3D sensors suitable for use in high-radiation environments. They will be used in the innermost layer of the inner system of the ITk pixel detector, which is the most irradiated part of the detector. The 3D sensors will be used in Layer L0 with an active thickness of $25 \times 100\,\mu m^2$ in the barrel part and an active thickness of $50 \times 50\,\mu m^2$ in the rest of Layer L0 [57].

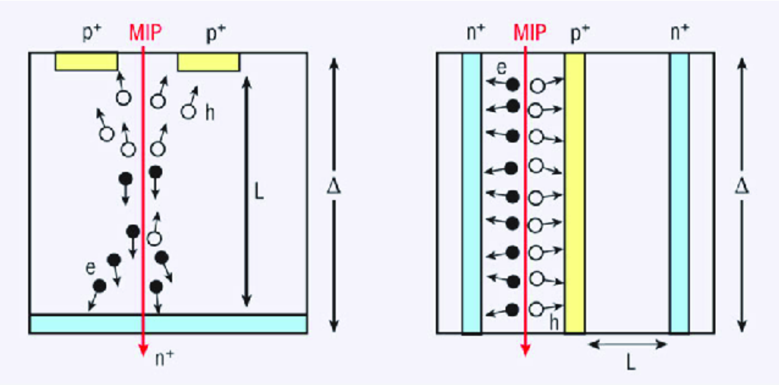


Figure 5.2.: The planar sensor structure (left) and the 3D sensor structure (right) and their particle detection behaviors [71].

5.1.2. ITk Pixel Module Type

In the ITk pixel detector, two types of modules are used: Triplets and Quad modules.

The Triplets consist of three front-end chips, each serving one sensor of size $2 \times 2\,\mathrm{cm}^2$.

They will be used for 3D sensors in layer L0 of the ITk pixel detector. The Triplet used in the barrel part of layer L0 is shown in Figure 5.3, and the Triplet used in the ring part of layer L0 is shown in Figure 5.4.

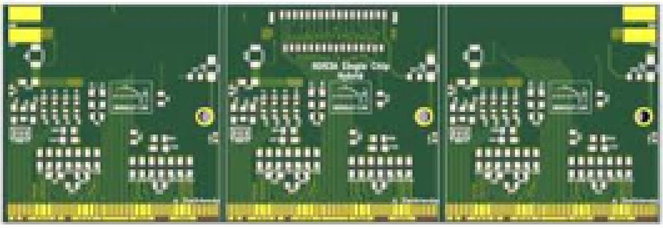


Figure 5.3.: The Triplet used in the barrel part of layer L0 in the ITk pixel detector [72].

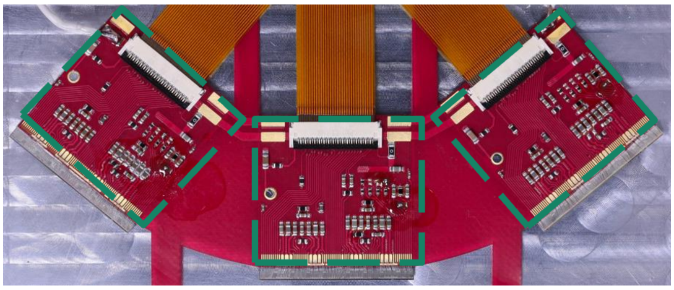


Figure 5.4.: The Triplet used in the ring part of layer L0 in the ITk pixel detector [72].

The Quad modules are formed of four front-end chips, each of size $2 \times 2\,\mathrm{cm}^2$. All four front-end chips serve one large sensor of size $4 \times 4\,\mathrm{cm}^2$. The Quad modules will be used in layers L1, L2, L3, and L4 of the ITk pixel detector. The Quad module type is shown in Figure 5.5.

5.2. ITk Outer Barrel

The ITk Outer Barrel is composed of three layers (L2, L3, and L4) and hosts 4772 Quad modules, covering an active area of $6.94\,\mathrm{m}^2$. Each layer of the Outer Barrel has a flat region in the middle and two inclined regions, one on each side. The modules in the flat central region are parallel to the beamline, while the modules in the inclined regions are placed at an angle relative to the beamline.

When the flex is glued to the bare module (sensors and front-end chips), the Module is formed. The Module is then glued to the OB Bare Module cell to form the OB Loaded Module cell. The OB Bare Module cell enables the module to be loaded into the Outer Barrel.

In the ITk Outer Barrel, there are two types of supports to host the OB Loaded Module cells: OB functional longerons and OB functional inclined half-rings. When these supports are loaded with OB Loaded Module cells, they form the OB Loaded

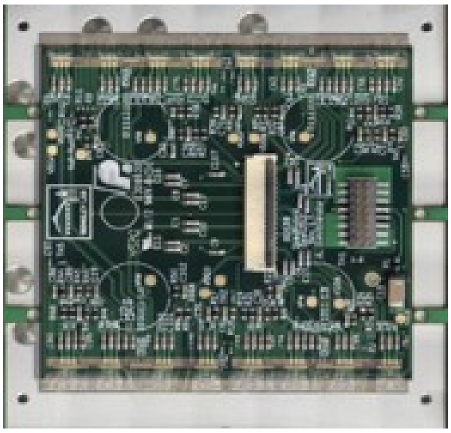


Figure 5.5.: The Quad module used in the ITk pixel detector [72].

Local Supports, where the OB functional longeron becomes the OB Loaded longeron and the OB functional inclined half-ring becomes the OB Loaded inclined half-ring. The OB Loaded longeron hosts the flat OB Loaded Module cells, and the OB Loaded inclined half-ring hosts the inclined OB Loaded Module cells, as shown in Figure 5.6.

The OB Loaded longerons and OB Loaded inclined half-rings are assembled together to form OB half-layers, as shown in Figure 5.6. There are six OB half-layers, with two OB half-layers for each layer. Finally, all six OB half-layers are mounted to form the ITk Outer Barrel.

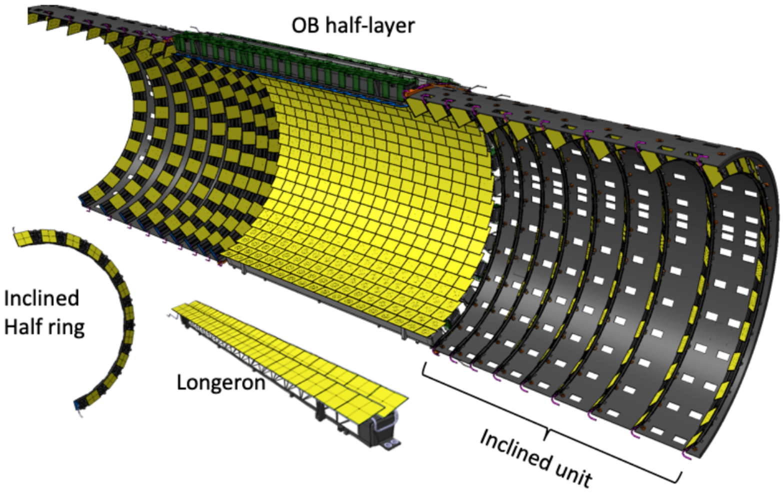


Figure 5.6.: The two types of loaded local supports in the ITk Outer Barrel are the OB Loaded Local Longeron and the OB Loaded Inclined Half-Ring, along with their location when mounted to form an OB half-layer [73].

Table 5.1 shows all properties of the ITk Outer Barrel for each layer. These properties include the radius, which is the distance between the layer and the beamline; the number of longerons; the number of modules per longeron; the number of inclined half-rings; the number of modules per inclined half-ring; and the angle of the inclined modules with respect to the beamline.

Layer	Radius (mm)	Number of Longerons per Layer	Number of Module per Longeron	Number of Inclined Half Rings per Layer	Number of Module per Inclined Half Ring	Tilt Angle in the inclined region (°)
2	160	16	36	24	16	67
3	228	22	36	32	22	58
4	288	28	36	36	28	55

Table 5.1.: The properties of the ITk Outer Barrel.

5.3. ITk Production Database

During the assembly and production phase of the ITk, a large number of elements are used. These elements need to be tracked during production. The ITk Production Database (ITk PD) is created to track and store information about the ITk components.

The reasons for creating the ITk PD are as follows [74]:

- To record details about the used ITk components (e.g., electrical details, performance details).
- To record details about the quality assurance process performed at multiple stages during the production process.
- To record details about the construction of the elements.
- To record details about the location of the elements and their movement history (records of deliveries between sites).
- To record details about the production situation, such as how many tests are passed, which sites are facing difficulties during production, and the performance of ITk components.
- To investigate the collected details to ensure good performance in upcoming production challenges, thereby avoiding wasting time and effort.

The ITk PD is built and designed by Unicorn College, Prague. It covers all four areas of the ITk: Strips Database, Pixel Database, Common Electronics, and Common Mechanics.

All the components of the ITk are defined as component types, and each component has sub-components that form it, called children. Each component type can have a set of types that must be determined, which differ by the type of children they have and their quantity.

During production, each component type has to follow a set of stages. At each stage, a set of required tests is assigned. These tests are called test types, and each test type has properties and parameters. The test type's properties store information about the test, and the test type's parameters present the results of the test.

The ITk PD provides users with the ability to register a component, allowing them to set the component's stage, add tests to the component, and assemble the component's children.

5.4. OB Bare Local Supports

The Figure 5.7 presents the three types of OB Bare local supports. These types are OB Bare Module Cell, OB Functional Longeron, and OB Functional Inclined Half Ring. The OB Bare Module Cell is used to host the OB Module to form the OB Loaded Module Cell. The OB Functional Longeron and OB Functional Inclined Half Ring are used to host the OB Loaded Module Cell to form the OB Loaded Longeron and OB Loaded Inclined Half Ring, respectively.

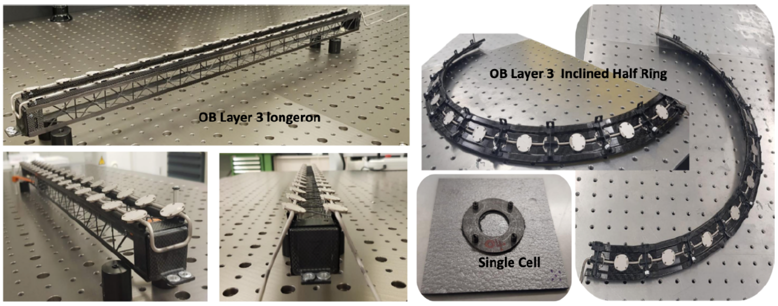


Figure 5.7.: The three types of OB Bare local supports in the ITk Outer Barrel are the OB Functional Longeron, the OB Functional Inclined Half-Ring, and the OB Bare Module Cell [75].

5.4.1. OB Bare Module Cell

The OB Bare Module Cell is the holder of the module. It consists of a PG tile that will be glued to the module and a cooling block glued to the other side of the PG tile. The cooling block connects the module to the LLS.

The OB Bare Module Cell component type has one type named Dummy. Its children are shown in Figure 5.8. They include:

- One OB Cooling Block of type Dummy.
- One OB PG Tile of type Dummy.

- One OB Bare LS Adhesive of type Dummy.

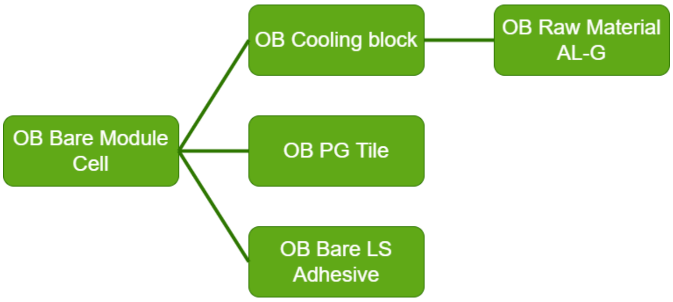


Figure 5.8.: OB Bare Module cell component type and its children.

The stages of the OB Bare Module Cell component type are shown in Figure 5.9. They are:

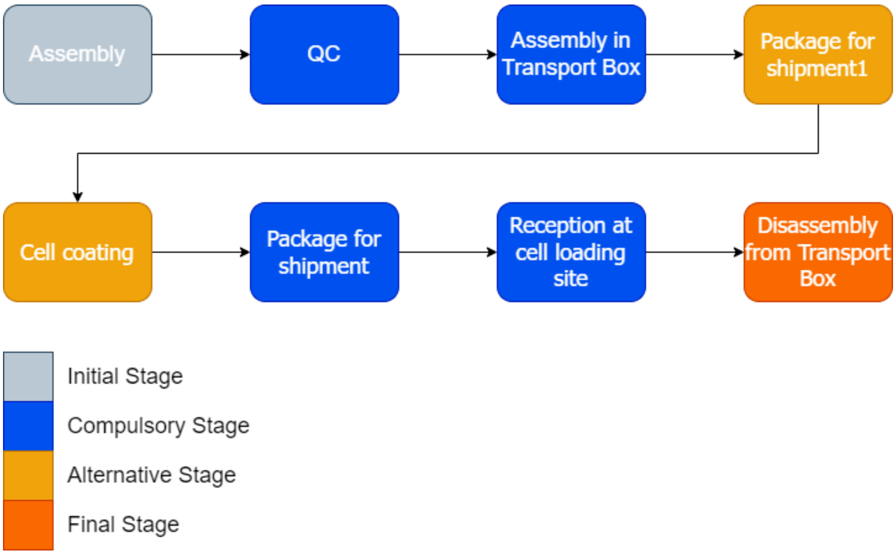


Figure 5.9.: OB Bare Module Cell component type's stages.

1. **Assembly:** The OB Bare Module Cell is assembled in this stage.
2. **QC:** This is the stage where the quality control (QC) tests are done. The associated tests are:
   a) **Visual Inspection:** Checks the homogeneity of the glue layer and identifies damaged areas.
   b) **Thermal Cycling:** Involves a single thermal cycle between −45℃ and 40℃.

    c) **Dimensional Control:** Uses metrology of the bare cell using an MMT or optical system.
    d) **Mass Control:** Checks the mass using a scale.
    e) **Thermal Performance Check:** A non-destructive test of the tile to determine the apparent thermal impedance using the modified tower setup.

3. **Assembly in Transport Box:** The OB Bare Module Cell is mounted in the OB Bare Cell Transport Box for shipping.
4. **Packaging for Shipment:** In this stage, the OB Bare Module Cell is packaged and shipped to the location where cell coating will be done. Note that this is an alternative stage; if cell coating is excluded, this stage is also excluded.
5. **Cell Coating:** In this stage, the cell is coated for dust protection. Note that this stage is alternative, as it is not officially approved whether the coating will be done.
6. **Packaging for Shipment:** The OB Bare Module Cell is packaged and shipped to the cell loading site. This stage is compulsory because the QC stage location is different from the cell loading site.
7. **Reception at Cell Loading Site:** This is the stage where the OB Bare Module Cell is received at the cell loading site. The associated test is:
    a) **Visual Inspection:** Same as in the QC stage.
8. **Disassembly in Transport Box:** This is the final stage where the OB Module Cell is dismounted from the OB Bare Cell Transport Box after being received at the cell loading site.

All the component types, properties of the component types, component type stages, tests done at each stage with their properties, and parameters of the OB Bare Module Cell and its children are implemented in ITk PD and summarized in Table 5.2.

Table 5.2.: OB Bare Module Cell and its children characteristics.

Component	**Types**	**Properties**	**Stage**	**Test**	**Test's properties**	**Test's parameters**
OB Raw Material AL-G	Dummy	Batch Number	Reception			
		Delivery Date	QC	Density Check		Density
		Number of Rods				

		Supplier Reference				
		Material Certificate Link				
OB Cooling Block	Dummy	Part Number	Machining			
		Machining Date	QC	Mass Control		Mass
				Dimensional control		Metrology Report
						Cooling Block Thickness
						Pass/No Pass
				Thread Check		Failure Load
						Failure Mode
				Visual Inspection		Cooling Block Pictures
OB PG Tile	Dummy	Part Number	Reception			
		Delivery Date	QC	Dimensional Control		Thickness
		Supplier Batch Reference				Lenght
						Width
						Pass/No pass
				Mass Control		Mass
			Package for shipment (alternative)			

			Reception at Cell Assembly Site	Visual Inspection	Operator Name	Pass/No Pass
OB Bare LS Adhesive	Dummy	Delivery Date	Reception			
		Site Name				
		Quantity	In Use			
		Supplier Reference				
		Manufacturer Production Date	Expired (alternative)			
		Expiration Date				
		Opening Date	Empty			
		Empty Date				
OB Bare Module Cell	Dummy	Part Number	Assembly			
		Assembly Date	QC	Visual Inspection	Operate Name	Pass/No pass
		Assembly Tool Identifier				Pictures of the Bare cell
		Package Date		Thermal Cycling	Maximum Temperature	Done/Not Done
		Assembly Tool Position			Minimum Temperature	
					Number of Cycles	
				Dimensional Control	Machine Name	Metrology Report
						Glue + Cooling Block Thickness
						Pass/No pass
				Mass Control		Cell Mass

						Estimated Glue Mass
				Thermal Performance Check	Test Date	Apparent Thermal Impedance
					Operate Name	
					Test Temperature	
					Test Power	
			Assembly in Transport Box			
			Package for Shipment 1 (alternative)			
			Cell Coating (alternative)			
			Package for Shipment			
			Reception at cell Loading Site	Visual Inspection	same as in QC	Same as in QC
			Disassembly from Transport Box			

5.4.2. OB Functional Longeron

The OB Functional Longeron component type in ITk PD has four types: FL2 A Side, FL2 C Side, FL34 A Side, and FL34 C Side. These types share the same child components but differ in type. The Figure 5.10 shows the OB Functional Longeron children. These children are OB Longeron Composite Structure, OB Longeron Inserts, OB Functional Pipe for Longeron, OB Bare LS Adhesive and OB Longeron HF[1].

Each OB Functional Longeron type consists of:

- One OB Longeron Composite Structure of type Layer 2 if the OB Functional

[1] OB Longeron HF is a handling frame for the OB Functional Longeron and OB Loaded Longeron.

Longeron is of type FL2 A Side or FL2 C Side, or Layer 34 if the OB Functional Longeron is of type FL34 A Side or FL34 C Side.

- One OB Longeron HF of type Dummy.
- number OB Longeron Inserts of different types.
- One OB Functional Pipe for Longeron of type FCP_L2 A Side if the OB Functional Longeron is of type FL2 A Side, FCP_L2 C Side if it is of type FL2 C Side, FCP_L34 A Side if it is of type FL34 A Side, or FCP_L34 C Side if it is of type FL34 C Side.
- One OB Bare LS Adhesive of type Dummy.

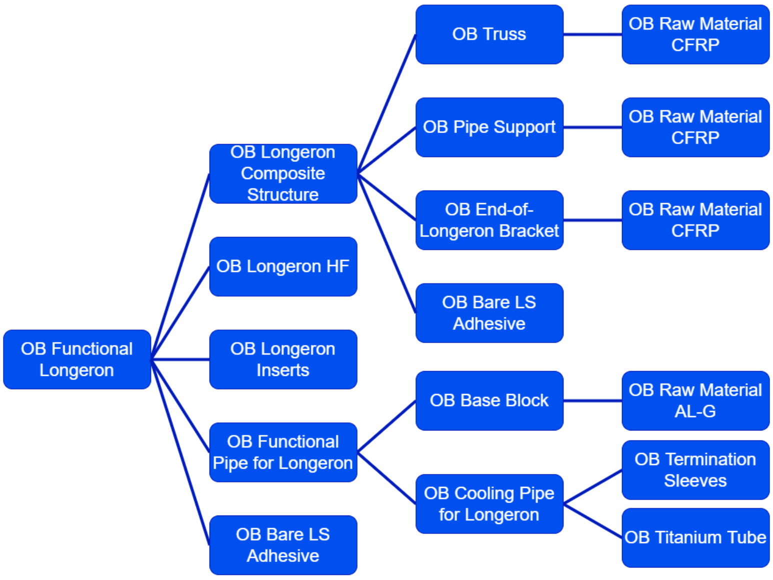


Figure 5.10.: OB Functional Longeron component type and its children.

The stages of the OB Functional Longeron type are shown in Figure 5.11. They are:

1. **Assembly:** The OB Functional Longeron is assembled in this stage.
2. **Packaging for Shipment:** This is an alternative stage where the OB Functional Longeron is packaged and shipped to the location where the QC testing will be done. Note that if the QC tests are performed at the same location as the assembly, this stage is excluded.
3. **QC:** This stage involves quality control (QC) tests. The associated tests are:
    a) **Visual Inspection:** Checks soldered and bent joints to identify potential damaged areas.

   b) **Thermal Cycling:** Involves a single thermal cycle between −45°C and 40°C.

   c) **Pressure Check:** A static pressure test using Argon at 162 bar for 30 minutes.

   d) **Leak Check:** A leak-rate test using a Helium leak checker.

   e) **Mass Control:** Measures the mass using a scale.

   f) **Thermal Performance Check:** A non-destructive test to determine the apparent TFM of the base block-to-pipe joints.

   g) **Grounding Check:** Measures DC resistance between various base blocks and the reference locators.

   h) **Dimensional Control:** Uses metrology to survey the position of the precise and slotted holes of the base blocks with respect to the reference inserts, and also involves envelope control using a 3D laser scanning system.

4. **Packaging for Shipment1:** This is another alternative stage where the OB Functional Longeron is packaged and shipped to the cell integration site. Note that if QC tests are performed at the same location as cell integration, this stage is excluded.

5. **Reception at Cell Integration Site:** The final stage where the OB Functional Longeron and other components are received to assemble the OB Loaded Longeron.

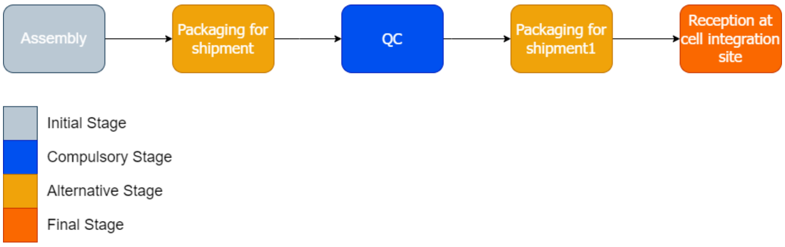


Figure 5.11.: OB Functional Longeron component type's stage.

In addition, the properties of the OB Functional Longeron component type are implemented, including Assembly Date, Operator Name, and Part Number. Table 5.3 shows the types, properties, stages, and tests in each stage of the OB Functional Longeron and its children. All the children of the OB Functional Longeron are implemented in the ITk PD.

Table 5.3.: OB Functional Longeron and its children characteristics.

Component	Types	Properties	Stage	Test	Test's properties	Test's parameters
OB Raw Material AL-G	Same as in OB Bare Module Cell					
OB Base Block	Dummy	Machining Date	Machining			
		Part Number	QC before Ni-coating	Visual Inspection		Pass/No pass
						Base Block pictures
				Dimensional Control	Machine Name	Metrology Report
						Pass/No pass
				Mass Control		Base Block Mass
						Estimated Ni Mass(set it to 0 in this stage)
			Package for Shipment for Ni-coating			
			QC after Ni-coating	Coating Check	Ni Coating Batch	Supplier's coating Report
					Coating Date	
				Visual inspection	same as in QC before Ni-coating	same as in QC before Ni-coating
				Mass Control	same as in QC before Ni-coating	same as in QC before Ni-coating
				Magnetic Check	Operate Name	Pass/No pass
OB Titanium Tube	Dummy	Delivery Date	Reception			
		Number of Pipes	QC	Dimensional Control	Test Date	Inner Diameter

		Batch Number			Operate Name	Outer Diameter
		Supplier Reference				Pass/No pass
		Material Certificate Link		Metallography	Test Date	Metallography Report
					Operate Name	Pass/No pass
				Mechanical Test	Test Date	Test Report
					Operate Name	Pass/No pass
OB Termination Sleeves	Dummy	Delivery Date	Reception			
		Number of Sleeves				
		Batch Number				
		Supplier Reference				
		Material Certificate Link				
OB Cooling Pipe for longeron	L2	Part Number	Pipe Bending	Pipe Bending Check	Pipe Bending Date	
	L34				Pipe Bending Operator	
			Pipe Cleaning	Pipe Cleaning Check	Pipe Cleaning Date	
					Pipe Cleaning Operator	
			QC after Bending	Visual Inspection	Test Date	Pass/No pass
					Operator Name	Pipe Pictures
				Dimensional Control	Test Date	Pass/No pass
					Operator Name	Metrology Report
				Mass Control		Pipe Mass

						Estimated Ni Mass
			Sleeve Brazing	Brazing Check	Brazing Date	Mass sleeve 1
					Brazing Operator	Mass Sleeve 2
					Metal Filler	
			QC after Sleeve Brazing	Visual Inspection after Brazing	Operator Name	Pass/No pass
					Test Date	Pictures of Brazed Sleeves
						Pipe Pictures after brazing
				Dimensional Control	same as in QC after Bending	same as in QC after Bending
				Mass Control	same as in QC after Bending	same as in QC after Bending
				Pressure Check	Test Date	Pass/No pass
					Operator Name	Test Report with Raw Data
					Machine Name	
				Leak Check	Test Date	Pass/No pass
					Operator Name	Leak Rate
					Machine Name	
			Package for Shipment for Ni-Coating			
			QC after Ni-Coating	Coating Check	Ni Coating Batch	supplier's Coating Report

					Coating Date	
				Visual Inspection	same as in QC after Bending	same as in QC after Bending
				Mass Control	same as in QC after Bending	same as in QC after Bending
				Magnetic Check	Operator Name	Pass/No pass
OB Functional Pipe for Longeron	FCP_L2 A Side	Assembly Date	Assembly	Solder Mass Check		Mass Solder Pre-forms
	FCP_L2 C Side	Operator Name	QC	Visual Inspection	Operator Name	Pass/No pass
	FCP_L34 A Side	Part Number				Pictures of Soldered Blocks
	FCP_L34 C Side			Soldering Check	Test date	Pass/No pass
					Operator Name	Test Report
OB Raw Material CFRP	Dummy	Delivery Date	Reception			
		Quantity				
		Supplier Reference	In Use			
		Manufacturer Production Date				
		Expiration Date	Expired (alternative)			
		Opening Date				
		Empty Date	Empty			
		Material Certificate Link				
OB Truss	Dummy	Manufacturing Date	Manufacturing			
		Operator Name	Machining			
		Mould Identifier	QC	Visual Inspection	Operator Name	Pass/No pass

		Part Number				Truss Pictures
				Dimensional Control	Test Date	Pass/No pass
					Operator Name	Metrology Report
					Machine Name	
				Mass Control		Mass
				Flexion Test	Test Date	Compliance 1
					Operator Name	Compliance 2
					Applied Mass	Pass/No pass
OB Pipe support	L2 X+	Manufacturing Date	Manufacturing			
	L2 X-	Operator Name	QC	Visual Inspection	Operator Name	Pass/No pass
	L34 X+	Mould Identifier				Cooling Pipe Support Pictures
	L34 X-	Part Number		Dimensional Control	Operator Name	Pass/No pass
						Metrology Report
				Mass Control		Mass
OB End-of-Longeron Bracket	A side	Manufacturing Date	Manufacturing			
	C side	Operator Name	Machining			
		Mould Identifier	QC	Visual Inspection	Operator Name	Pass/No pass
		Part Number				EoL Bracket Pictures
				Dimensional Control	Operator Name	Pass/No pass
						Metrology Report

				Mass Control		Mass
OB Longeron Composite Structure	Layer 2	Assembly Date	Assembly			
	Layer 34	Operator name	QC	Visual Inspection	Operator Name	Pass/No pass
		Part Number				Longeron Structure Pictures
				Dimensional Control	Test Name	Pass/No pass
					Operator Name	Metrology Report
					Machine Name	
				Mass Control		Mass
						Estimated Glue Mass
				Mechanical Test	Test Date	Compliance 1
					Operator Name	Compliance 2
					Applied Mass	Pass/No pass
OB Longeron Inserts	Type-0 Inserts	Delivery Date	Reception			
	Handling Inserts	Number of Inserts				
	Metrology Inserts	Batch Number				
	Positioning Inserts	Supplier Reference				
		Material certificate Link				
OB Functional Longeron	FL2 A Side	Assembly Date	Assembly			
	FL2 C Side	Part Number	Packaging for shipment (alternative)			

	FL34 A Side	Operator Name	QC	Visual Inspection	Part Number	Pass/No pass
	FL34 C Side					Pictures of Functional Longeron
				Thermal Cycling	Maximum Temperature	Done/Not Done
					Minimum Temperature	
					Number of cycling	
				Pressure Check	Test Date	Pass/No pass
					Operator Name	Test Report with Raw Data
					Machine Name	
				Leak Check	Test Date	Pass/No pass
					Operator Name	Leak Rate
					Machine Name	
				Mass Control		Mass
						Estimated Glue Mass
				Thermal Performance Test	Test Date	Pass/No pass
					Operator Name	Grade(A, B, C)
						Test Report
				Grounding Check	Test date	Pass/No pass
					Operator Name	Maximum DC Resistance
				Dimensional Control	Test Date	Pass/No pass
					Operator Name	Metrology Raw Data

					Machine Name	Metrology Report
			Package for Shipment (alternative)			
			Reception at Cell Integration Site	Visual Inspection	same as in QC	same as in QC

5.4.3. OB Functional Inclined Half Ring

The OB Functional Inclined Half Ring component type has three types: FL2, FL3, and FL4. These types have the same children but with different types and amounts. Figure 5.12 shows the children of the OB Inclined Half Ring, which are OB Half Ring Shell, OB Gusset, OB Pipe Strain Relief, OB Base Block, OB Cooling Pipe for IHR, OB Bare LS Adhesive, and OB Inclined Half Ring HF[2].

Each OB Functional Inclined Half Ring type consists of:

- One OB Half Ring Shell of type HRS L2 if the OB Functional Inclined Half Ring is of type FL2, HRS L3 if it is of type FL3, or HRS L4 if it is of type FL4.
- Eight OB Gussets of type Gusset L2 if the OB Functional Inclined Half Ring is of type FL2, eleven OB Gussets of type Gusset L3 if it is of type FL3, or fourteen OB Gussets of type Gusset L4 if it is of type FL4.
- Two OB Pipe Strain Reliefs of type Dummy.
- Sixteen OB Base Blocks of type Dummy if the OB Functional Inclined Half Ring is of type FL2, twenty-two OB Base Blocks of type Dummy if it is of type FL3, or twenty-eight OB Base Blocks of type Dummy if it is of type FL4.
- One OB Cooling Pipe for IHR of type L2 if the OB Functional Inclined Half Ring is of type FL2

  , L3 if it is of type FL3, or L4 if it is of type FL4.
- One OB Bare LS Adhesive of type Dummy.
- One OB Inclined Half Ring HF of type L2 if the OB Functional Inclined Half Ring is of type FL2, L3 if it is of type FL3, or L4 if it is of type FL4.

[2] OB Inclined Half Ring HF is a handling frame for the OB Functional Inclined Half Ring and the OB Loaded Inclined Half Ring.

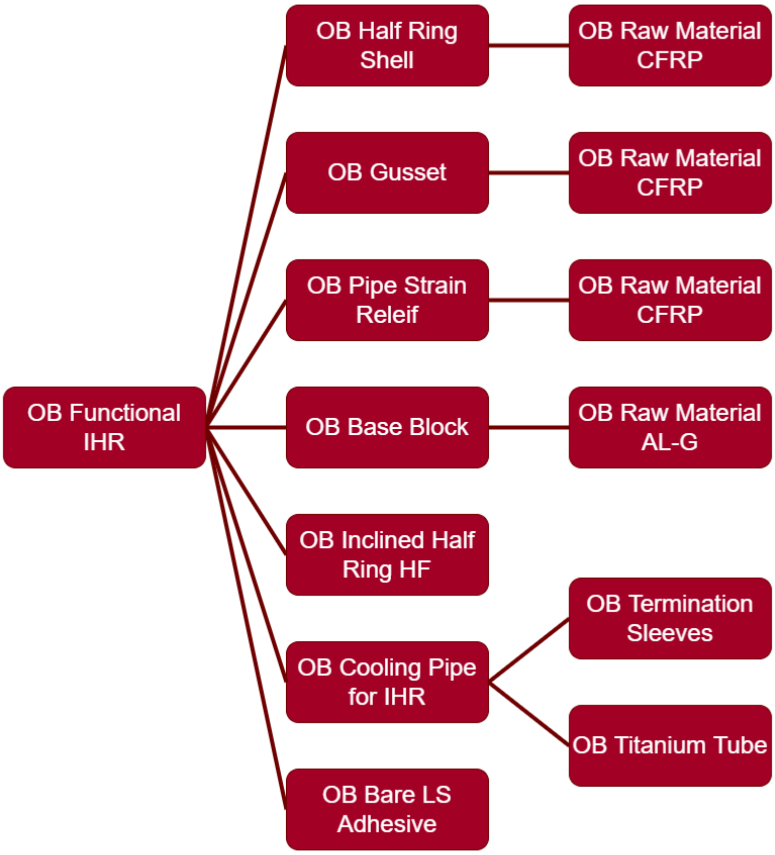


Figure 5.12.: OB Functional Inclined Half Ring component type and its children.

The stages of the OB Functional Inclined Half Ring are the same as those of the OB Functional Longeron, with an additional test in the QC stage named Soldering Check.

Table 5.4 shows the types, properties, stages, and tests in each stage of the OB Functional Inclined Half-Ring and its children. All the children of the OB Functional Inclined Half-Ring are implemented in the ITk PD.

Table 5.4.: OB Functional Inclined Half ring and its children characteristics.

Component	Types	Properties	Stage	Test	Test's properties	Test's parameters
OB Raw Material AL-G	Same as in OB Bare Module Cell					
OB Base Block	Same as in OB Functional Longeron					

OB Titanium Tube	Same as in OB Functional Longeron					
OB Termination Sleeves	Same as in OB Functional Longeron					
OB Cooling Pipe for IHR	L2	Part Number	Pipe Bending 1	Pipe Bending Check	Pipe Bending Date	
	L3				Pipe Bending Operator	
	L4		Pipe Cleaning 1	Pipe Cleaning Check	Pipe Cleaning Date	
					Pipe Cleaning Operator	
			QC after Bending 1	Visual Inspection	Test Date	Pass/No pass
					Operator Name	Pipe Pictures
				Dimensional Control	Test Date	Pass/No pass
					Operator Name	Metrology Report
				Mass Control		Mass
						Estimated Ni Mass
			Sleeve Brazing	Brazing Check	Brazing Date	Mass Sleeve 1
					Brazing Operator	Mass Sleeve 2
					Metal Filler	
			Pipe Bending 2	Pipe Bending Check	Same as in Pipe Bending 1	Same as in Pipe Bending 1
			Pipe Cleaning 2	Pipe Cleaning Check	Same as in Pipe Cleaning 1	Same as in Pipe Cleaning 1
			QC after Sleeve Brazing and Bending 2	Visual Inspection after Brazing	Test Date	Pass/No pass

					Operator Name	Pictures of Brazed Sleeves
						Pipe Pictures after brazing
				Dimensional Control	same as in QC after Bending 1	same as in QC after Bending 1
				Mass Control	same as in QC after Bending 1	same as in QC after Bending 1
				Pressure Check	Test Date	Pass/No pass
					Operator Name	Test Report with Raw Data
					Machine Name	
				Leak Check	Test Date	Pass/No pass
					Operator Name	Leak Rate
					Machine Name	
			Package for Shipment for Ni-Coating			
			QC after Ni-Coating	Coating Check	Ni Coating Date	Supplier's Coating Report
					Coating Date	
				Visual Inspection	same as in QC after Bending 1	same as in QC after Bending 1
				Mass Control	same as in QC after Bending 1	same as in QC after Bending 1
				Magnetic Check	Operator Name	Pass/No pass

OB Raw Material CFRP	Same as in OB Functional Longeron					
OB Half Ring Shell	HRS L2	Manufacturing Date	Manufacturing			
	HRS L3	Operator Name	Machining			
	HRS L4	Mould Identifier	QC	Visual Inspection	Operator Name	Pass/No pass
		Part Number				IHR Shell Pictures
				Dimensional Control	Test Date	Pass/No pass
					Operator Name	Metrology Report
					Machine Name	
				Mass Control		Mass
				Mechanical Test	Test Date	Compliance 1
					Operator Name	Compliance 2
					Applied Force	Pass/No pass
OB Gusset	Gusset L2	Manufacturing Date	Manufacturing			
	Gusset L3	Mould Identifier	QC	Visual Inspection	Operator Name	Pass/No pass
	Gusset L4	Part Number				Gusset Pictures
		Operator Name		Dimensional Control	Operator Name	Pass/No pass
						Metrology Report
				Mass Control		Mass
				Thread Check		Pass/No pass
OB IHR Pipe Strain Relief	Dummy	Manufacturing Date	Manufacturing			

		Mould Identifier	QC	Visual Inspection	Operator Name	Pass/No pass
		Part Number				Strain Relief Pictures
				Dimensional Control	Operator Name	Pass/No pass
				Mass Control		Mass
OB Functional IHR	FL2	Assembly Date	Assembly	Solder Mass Check		Mass Solder Pre-forms
	FL3	Operator Name	Package for Shipment			
	FL4	Part Number	QC	Visual Inspection	Operator Name	Pass/No pass
						Pictures of Functional IHR
				Thermal Cycling	Maximum Temperature	Done/Not Done
					Minimum Temperature	
					Number of Cycles	
				Pressure Check	Test Date	Pass/No pass
					Operator Name	Test Report with Raw Data
					Machine Name	
				Leak Check	Test Date	Pass/No pass
					Operator Name	Leak Rate
					Machine Name	
				Mass control		Mass
						Estimated Glue Mass

				Thermal Performance Test	Test Date	Pass/No pass
					Operator Name	Grade(A, B, C)
						Test Report
				Grounding Check	Test Date	Pass/No pass
					Operator Name	Maximum DC Resistance
				Dimensional Control	Test Date	Pass/No pass
					Operator Name	Metrology Raw Data
					Machine name	Metrology Report
				Soldering Check	Test Date	Pass/No pass
					Operator Name	Test Report
			Package for Shipment1			
			Reception at Cell Integration Site	Visual Inspection	Same as in QC	Same as in QC

5.5. OB Loaded Local Supports

When the OB loaded module cells are mounted on the OB functional local support, the OB functional local support becomes the OB loaded local support. The OB functional longeron and OB functional inclined half ring become the OB loaded longeron and OB loaded inclined half ring, respectively, after mounting the OB loaded module cells.

Two types of OB-loaded local supports are implemented in ITk PD. Additionally, a code for uploading electrical test results has been written to ease the introduction of these results into ITk PD. Moreover, a graphical user interface (GUI) for managing OB-loaded local supports has been developed.

5.5.1. OB Loaded Longeron

The OB Loaded Longeron component type in ITk PD has six types: LL2A, LL2C, LL3A, LL3C, LL4A, and LL4C. For each layer, there are two types, A and C, corresponding to the two different cooling types for the longeron. These types share the same children component types but differ in their specific types. Figure 5.13 shows the children of the OB Loaded Longeron, which include the OB Functional Longeron, OB Pigtails, OB Loaded Module Cell, and OB PP0 Flat Longeron.

Each OB Loaded Longeron component type consists of:

- One OB Functional Longeron of type FL2 A Side if the OB Loaded Longeron is of type LL2A, or type FL2 C Side if it is type LL2C, or type FL34 A Side if it is of type LL3A or LL4A, or type FL34 C Side if it is type LL3C or LL4C.
- Eighteen OB Pigtails of type PG-Longeron-Top-L2L3L4.
- Eighteen OB Pigtails of type PG-Longeron-Bottom-L2L3L4.
- Thirty-six OB Loaded Module Cells of type Dummy.
- Two OB PP0 Flat Longerons of type PP0 Flat Longeron M6-SP1 L2 if the OB Loaded Longeron is of type LL2A or LL2C, or type PP0 Flat Longeron M6-SP1 L3L4 if it is type LL3A, LL3C, LL4A, or LL4C.
- Two OB PP0 Flat Longerons of type PP0 Flat Longeron M12-SP2 L2 if the OB Loaded Longeron is of type LL2A or LL2C, or type PP0 Flat Longeron M12-SP2 L3L4 if it is of type LL3A, LL3C, LL4A, or LL4C.

The stages of the OB Loaded Longeron component type are shown in Figure 5.14. The stages are:

1. **Assembly :** The OB Loaded Longeron is assembled in this stage.
2. **Thermal Cycle :** This is considered an alternative stage since its test can be done in one of the electrical stages, and it consists of the following test:
    a) **Thermal Cycle :** Checks if the OB Loaded Longeron can withstand temperature variation.
3. **Connectivity Checkout Electrical :** This is the first stage for electrical tests, where the temperature is set to warm, and it consists of the following tests:
    a) **Tilock and MOPS Check :** Checks if the Tilock[3] and the MOPS[4] are working properly.
    b) **LpGBT Phase Tuning :** Determines optimal phase values to sample FE uplink data in the LpGBT[5]-based detector read-out system.

[3] Tilock (Temperature interlock): It is used to protect the detector elements against heat ups.

[4] MOPS (Monitoring Of Pixel System): It is an ASIC used in the control and feedback path of the DCS.

[5] LpGBT (Low power Giga Bit Transceiver): It is a radiation tolerant ASIC that can be used to implement multipurpose high speed bidirectional optical links for high-energy physics experiments.

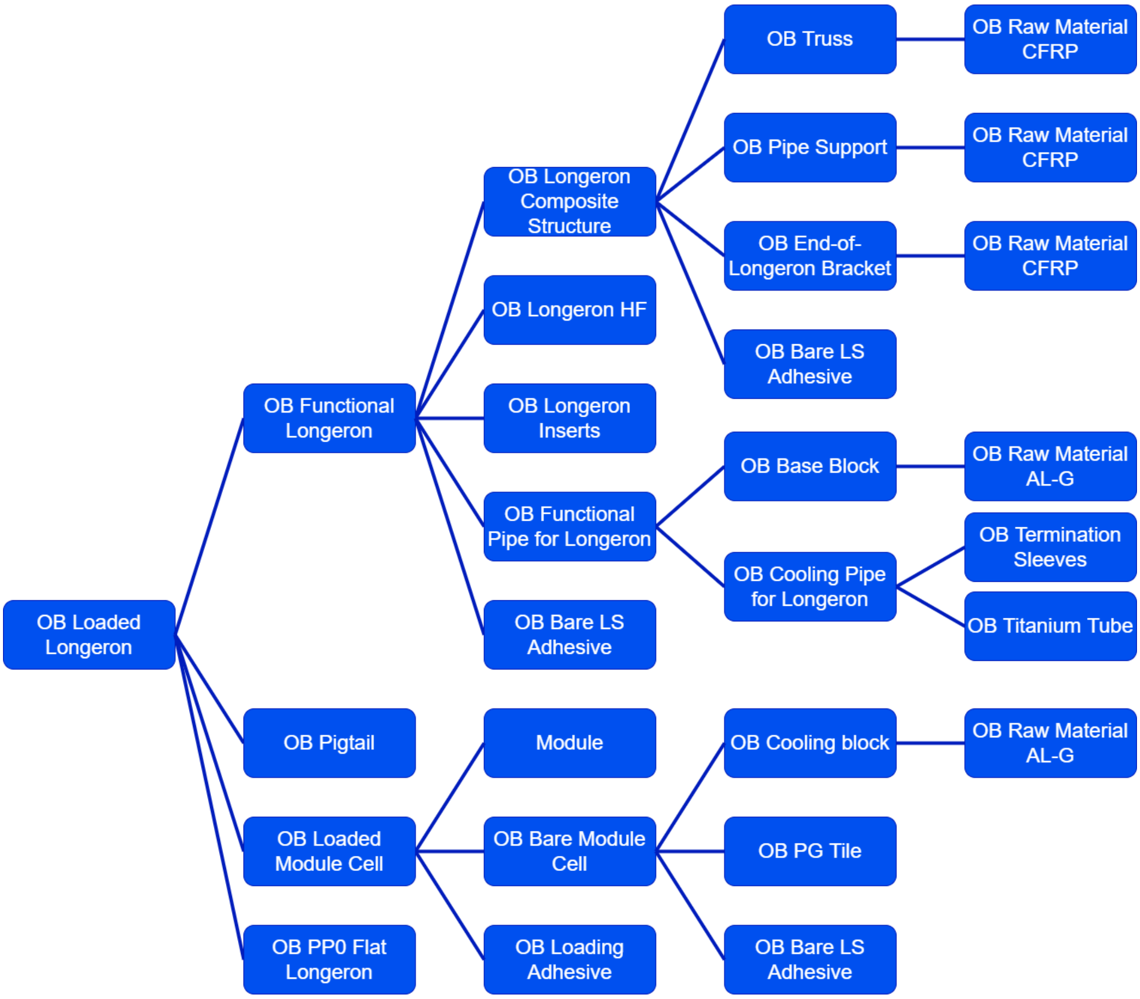


Figure 5.13.: OB Loaded Longeron component type and its children.

c) **Frontend Temperature Readback :** Reads the temperature values of each Frontend.

d) **Tilock and MOPS Readback :** Reads the temperature values and voltage drop of each module.

4. **Performance QC Electrical :** This is the second stage for electrical tests, where the temperature is set to cold, and it consists of the following tests:

a) **LpGBT Phase Tuning :** Same as in the Connectivity Checkout Electrical stage but at a cold temperature.

b) **IV Scan :** Determines the variation of the current with respect to voltage for each module.

c) **Digital Scan :** Tests the digital part of the chip.

   d) **Analog Scan :** Tests the analog part of the chip.
   e) **Threshold/Noise Measurements :** Determines the threshold and noise of every pixel.
   f) **Disconnected Bump Bonds Scan :** Identifies disconnected bump bonds of each module.
   g) **Frontend Temperature Readback :** Same as in the Connectivity Checkout Electrical stage but at a cold temperature.
   h) **Tilock and MOPS Readback :** Same as in the Connectivity Checkout Electrical stage but at a cold temperature.

5. **Low Power QC Electrical :** This is the last stage for electrical tests, where the temperature is set to warm, and it consists of the following tests:
   a) **Tilock and MOPS Readback :** Same as in the Connectivity Checkout Electrical stage where low power mode is enabled.
   b) **Digital Scan :** Same as in the Performance QC Electrical stage.
   c) **Analog Scan :** Same as in the Performance QC Electrical stage.

6. **Packaging for Shipment1 :** This is an alternative stage where the OB Loaded Longeron is packaged and shipped to the institute where the geometry and mass measurements will be done. Note that if the geometry and mass measurement tests are done at the same place where the electrical stages are conducted, then this stage is excluded.

7. **QC Geometry and Mass :** This stage involves geometry and mass measurements, consisting of the following tests:
   a) **Mass Measurement :** Measures the mass of the OB Loaded Longeron.
   b) **Envelope Control :** Checks the envelope of the component.
   c) **Metrology Survey :** Measures the geometry of the OB Loaded Longeron.

8. **Packaging for Shipment :** This is an alternative stage where the OB Loaded Longeron is packaged and shipped to the Integration site. Note that if the geometry and mass measurements are done at the same place where the integration of the outer barrel is done, then this stage is excluded.

9. **Reception at the Integration Site :** This is the final stage. The OB Loaded Longeron is received, and the integration process of the outer barrel is done. It consists of the following tests:
   a) **Tilock and MOPS Readback :** Same as in the Connectivity Checkout Electrical stage.
   b) **Digital Scan :** Same as in the Performance QC Electrical stage.

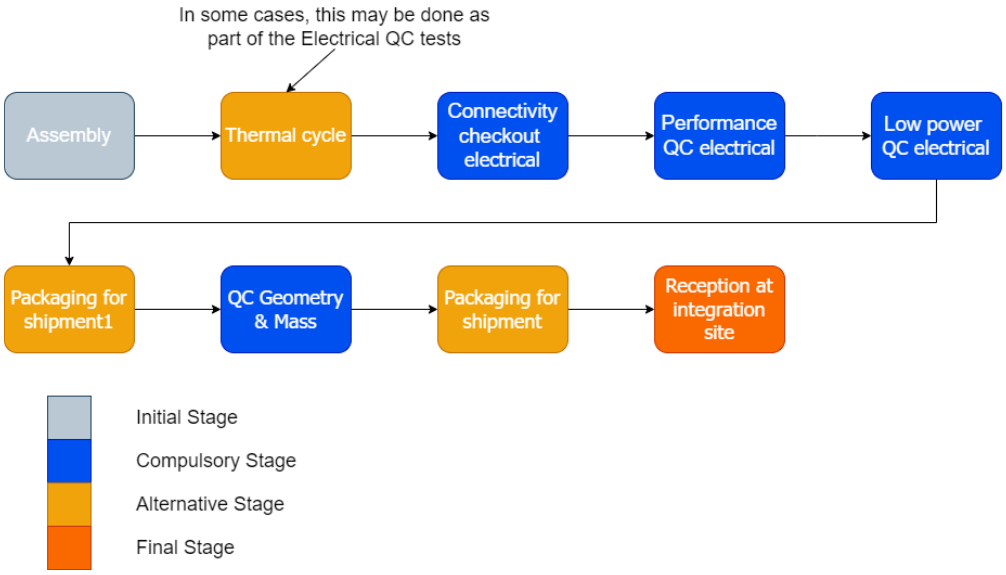


Figure 5.14.: OB Loaded Longeron component type's stages.

c) **Analog Scan :** Same as in the Performance QC Electrical stage.

All the ITk PD implementations of test properties and parameters of the OB Loaded Longeron are introduced in Table 5.5.

Table 5.5.: OB loaded longeron tests' properties and parameters.

Tests	Properties	Parameters
Thermal Cycle	Minimum temperature, Maximum temperature, Number of cycles, Thermal cycling speed, Machine Name	Temperature Log, Humidity Log
Tilock and MOPS check	Warm Temperature, LLS powered OFF, MOPShub voltage, Software version, Firmware version	Tilock and MOPS check result
LpGBT Phase tuning	Temperature, LV ON, HV OFF, Software version, Firmware version	Phase tuning result, Phase tuning value, Decoding efficiency
Frontend temperature readback	Temperature, LV ON, HV OFF, Software version, Firmware version	Temperature values
Tilock and MOPS readback	Temperature, LV ON, HV OFF, LP mode8 enabled, Software version, Firmware version	NTC temperature values, Voltage drop across each module

IV scan	Cold Temperature, LV ON, HV maximum voltage, Software version, Firmware version	IV data
Digital scan	Cold Temperature, LV ON, HV ON, Software version, Firmware version	Digital scan data
Analog scan	Cold Temperature, LV ON, HV ON, Software version, Firmware version	Analog scan data
Threshold/noise measurements	Cold Temperature, LV ON, HV ON, Software version, Frimware version	Threshold scan data
Disconnected bump bonds scan	Cold Temperature, LV ON, HV ON, Software version, Firmware version	Disconnected bond bumps scan data
Mass measurement	Tool	Mass
Envelope control	Tool	Pass or fail, Envelope file
Metrology Survey	MMT details	Metrology survey

5.5.2. OB Loaded Inclined Half Ring

The OB Loaded Inclined Half Ring component type has six types in ITk PD: L2, L3, L4, Last L2, Last L3, and Last L4. For each layer, there is an associated type since the number of modules held by the OB Loaded Inclined Half Ring depends on the related layer. Additionally, there are three types labeled ”Last” since the associated type-0 service depends on the position of the layer. These types have the same children component types, but they differ in the number and type of the children component types. Figure 5.15 shows the children of the OB Loaded Inclined Half Ring, which include the OB Functional Inclined Half Ring, OB Pigtails, OB Loaded Module Cell, and OB PP0 Flat Inclined IHR.

Each OB Loaded Inclined Half Ring type consists of:

- One OB Functional Inclined Half Ring of type FL2 if the OB Loaded Inclined Half Ring is of type L2 or Last L2, or type FL3 if it is of type L3 or Last L3, or type FL4 if it is of type L4 or Last L4.

- N/2 OB Pigtails of type PG-IHR-Front-L2 if the OB Loaded Inclined Half Ring is of type L2, or type PG-IHR-Front-LastRing-L2 if it is of type Last L2, or type PG-IHR-Front-L3L4 if it is of type L3 or L4, or type PG-IHR-Front-LastRing-L3L4 if it is of type Last L3 or Last L4.

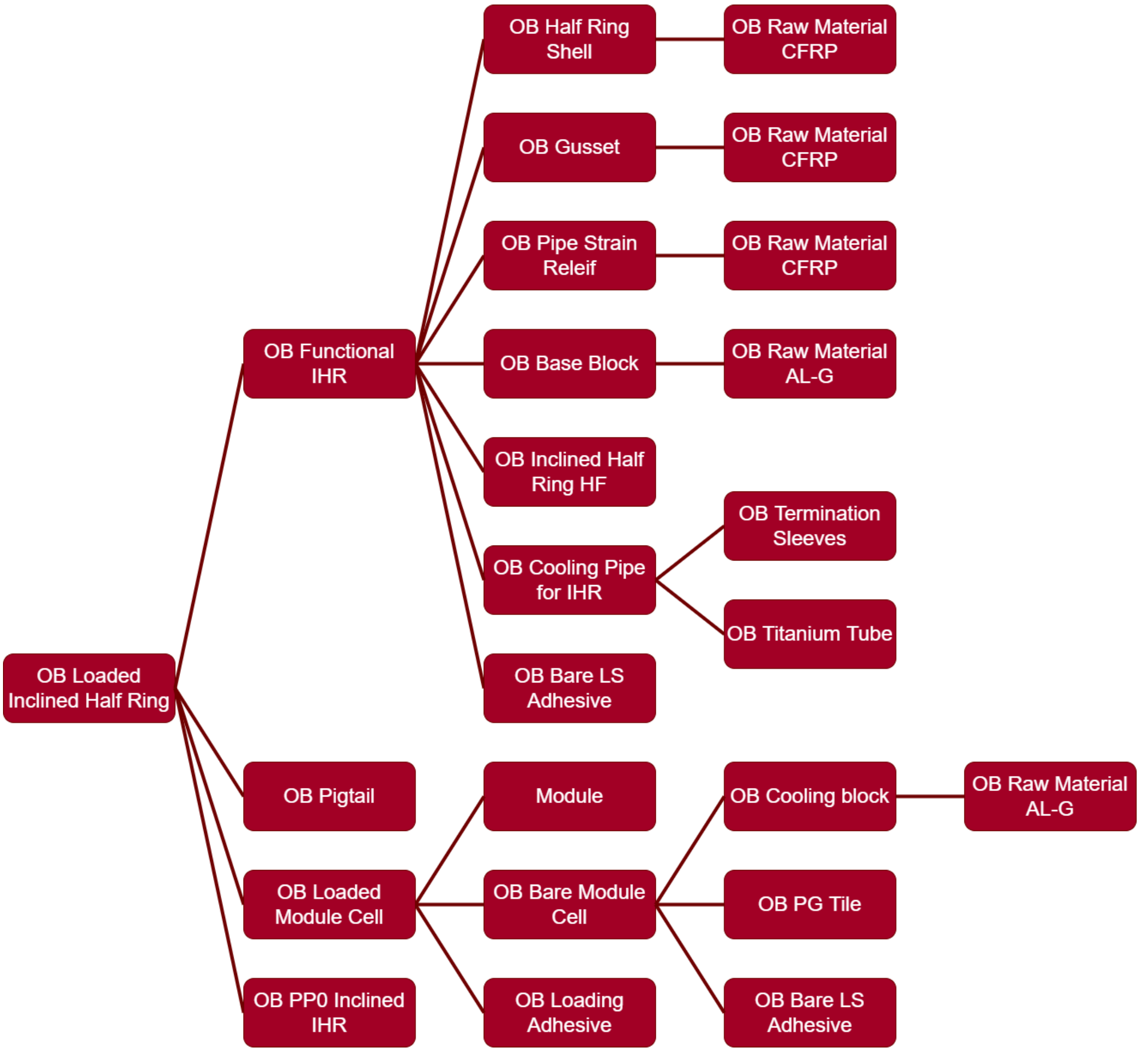


Figure 5.15.: OB Loaded Inclined Half Ring component type and its children.

- N/2 OB Pigtails of type PG-IHR-Back-L2 if the OB Loaded Inclined Half Ring is of type L2, or type PG-IHR-Back-LastRing-L2 if it is of type Last L2, or type PG-IHR-Back-L3L4 if it is of type L3 or L4, or type PG-IHR-Back-LastRing-L3L4 if it is of type Last L3 or Last L4.

- N OB Loaded Module Cells of type Dummy.

- One OB PP0 Inclined IHR of type OB PP0 Inclined IHR-SP1-L2 if the OB Loaded Inclined Half Ring is of type L2 or Last L2, or type OB PP0 Inclined IHR-SP1-L3 if it is of type L3 or Last L3, or type OB PP0 Inclined IHR-SP1-L4 if it is of type L4 or Last L4.

- One OB PP0 Inclined IHR of type OB PP0 Inclined IHR-SP2-L2 if the OB Loaded Inclined Half Ring is of type L2 or Last L2, or type OB PP0 Inclined IHR-SP2-L3 if it is of type L3 or Last L3, or type OB PP0 Inclined IHR-SP2-L4 if it is of type L4 or Last L4.

Where N is equal to 16 if the OB Loaded Inclined Half Ring is of type L2 or Last L2, equal to 22 if it is of type L3 or Last L3, or equal to 28 if it is of type L4 or Last L4.

The OB Loaded Inclined Half Ring follows the same stages and tests per stage as the OB Loaded Longeron.

5.5.3. Electrical tests results uploader

After assembling the OB-loaded local support components, electrical tests must be uploaded during the subsequent electrical quality control stages, which involve uploading a significant number of data files. The YARR[6] output from the electrical tests provides results for each frontend chip. Each OB-loaded local support contains numerous frontend chips; for instance, a longeron holds 36 modules, each with 4 frontend chips, resulting in the need to upload data for 144 frontend chips. Therefore, a code is required to facilitate the upload of all this data to ITk PD.

The code's algorithm is shown in Figure 5.16. The code prompts the user for their ITk PD credentials and the test result outputs, sets the stage for the LLS and the FE chips, zips the corresponding data files, and uploads them to the relevant LLS and FE chips.

The code is available at the following link: `https://gitlab.cern.ch/atlas-ITk/sw/db/pixels/ob-local-support` in the directory named electrical_test_uploader, with detailed documentation.

Figure 5.17 displays the result of uploading data to the FE chip into ITk PD, while Figure 5.18 shows the result of uploading data to LLS into ITk PD.

[6] YARR (Yet Another Rapid Readout) is a data acquisition (DAQ) system designed to read out ATLAS ITk Pixel frontend chips.

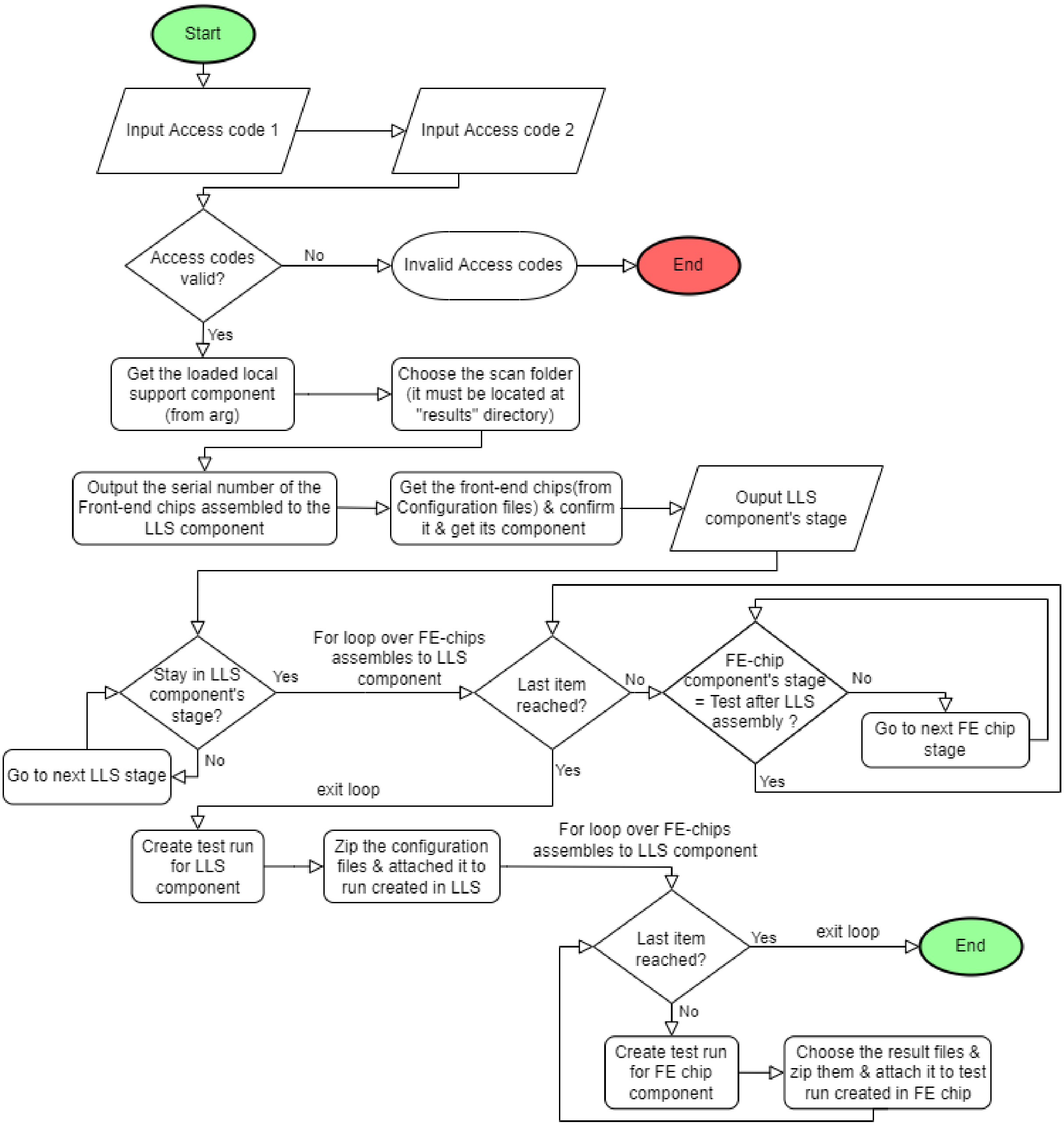
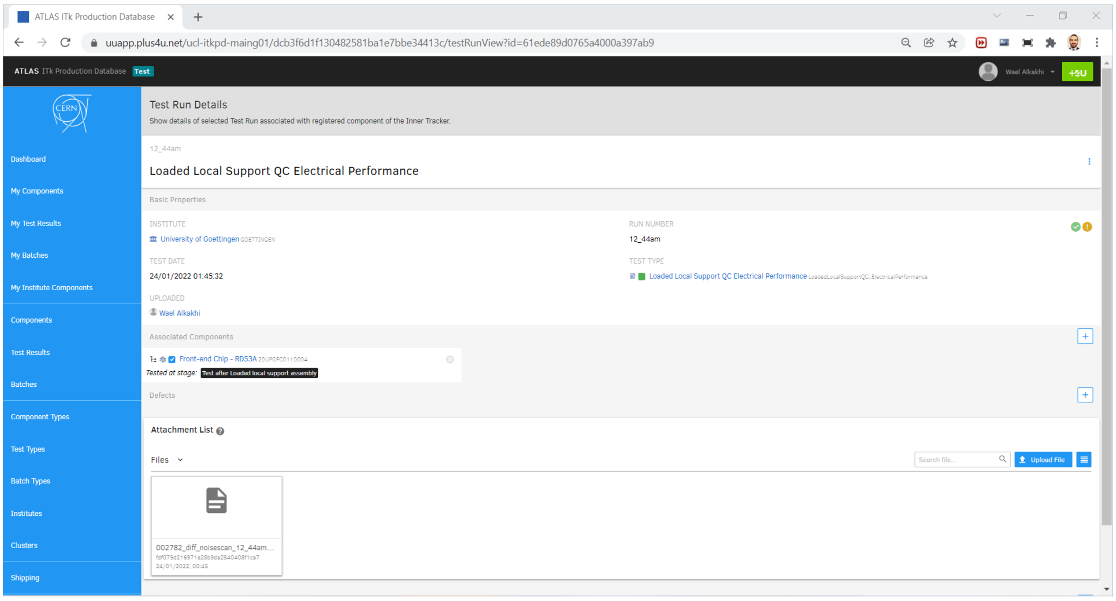

Start
Input Access code 1
Input Access code 2
Access codes valid?
No
Invalid Access codes
End
Yes
Get the loaded local support component (from arg)
Choose the scan folder (it must be located at "results" directory)
Output the serial number of the Front-end chips assembled to the LLS component
Get the front-end chips(from Configuration files) & confirm it & get its component
Ouput LLS component's stage
Stay in LLS component's stage?
Yes
For loop over FE-chips assembles to LLS component
Last item reached?
No
FE-chip component's stage = Test after LLS assembly ?
No
Go to next FE chip stage
Yes
Yes
No
Go to next LLS stage
exit loop
Create test run for LLS component
Zip the configuration files & attached it to run created in LLS
For loop over FE-chips assembles to LLS component
Last item reached?
Yes
exit loop
End
No
Create test run for FE chip component
Choose the result files & zip them & attach it to test run created in FE chip

Figure 5.16.: The uploader code Algorithm.



Test Run Details

Show details of selected Test Run associated with registered component of the Inner Tracker.

12_44am

Loaded Local Support QC Electrical Performance

Basic Properties

INSTITUTE: University of Goettingen GOETTINGEN

RUN NUMBER: 12_44am

TEST DATE: 24/01/2022 01:45:32

TEST TYPE: Loaded Local Support QC Electrical Performance LoadedLocalSupportQC_ElectricalPerformance

UPLOADED: Wael Alkakhi

Associated Components

Front-end Chip - RD53A 20UPGFC0110004

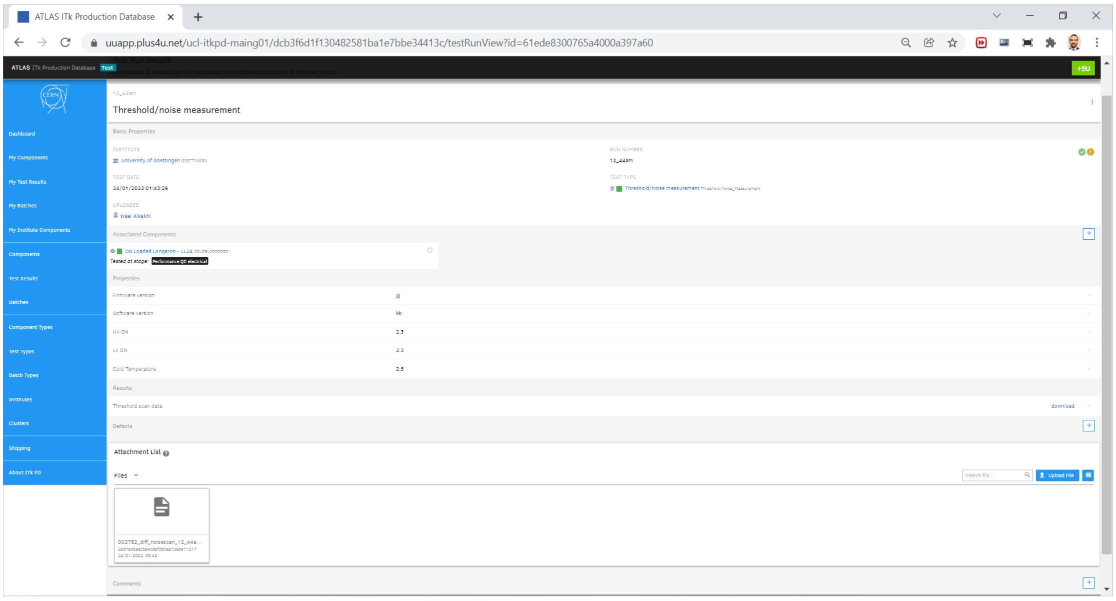
Tested at stage: Test after Loaded local support assembly

Defects

Attachment List

002782_diff_noisescan_12_44am...
fdf079d216971e28b9da2840408f1ca7
24/01/2022, 00:45

Figure 5.17.: Uploading electrical test of FE chip component results to ITk PD, where it is the zip file attached to Attachment List.



12_44am

Threshold/noise measurement

Basic Properties

INSTITUTE: University of Goettingen GOETTINGEN

RUN NUMBER: 12_44am

TEST DATE: 24/01/2022 01:43:29

TEST TYPE: Threshold/noise measurement Threshold/Noise_Measurement

UPLOADED: Wael Alkakhi

Associated Components

OB Loaded Longeron - LL2A 20UPBL0000001

Tested at stage: Performance QC electrical

Properties

Firmware version	jjj
Software version	kk
HV ON	2.3
LV ON	2.3
Cold Temperature	2.5

Results

Threshold scan data — download

Defects

Attachment List

002782_diff_noisescan_12_44a...
25d7a4baecba4cd5f0b0ea736be71217
24/01/2022, 00:43

Comments

Figure 5.18.: Uploading electrical test of LLS component results to ITk PD, where it is the zip file attached to Attachment List.

5.5.4. Graphical User Interface(GUI)

Dealing with the ITk production database during the assembly and production of the ITk pixel outer barrel is challenging. To address this, a graphical user interface (GUI) has been created to interface with the ITk PD. The GUI allows users to store all relevant information and track the stages of component types in the ITk PD for loaded local support during the production process. The GUI can be accessed via this link `https://gitlab.cern.ch/atlas-ITk/sw/db/pixels/ob-local-support`. Additionally, detailed GUI documentation is available in the given link. The GUI provides users with many features such as component registration, test registration, stage setting, and more.

The GUI layout consists of two main tabs: the Component tab and the User tab. The User tab contains information about the user and is used for signing in the first time the GUI is used.

The Component tab of the GUI consists of two sections, as shown in Figure 5.19. In the left section, the user can search for a registered component by typing its name in part number or serial number and clicking “Find”. The component and its children will then appear in the hierarchy. If no component is registered, or if the user wants to register a new component, the user can do so by clicking “Register”.

When the user selects a component in the hierarchy, the component’s information is shown in the right section. This section consists of three tabs. The first tab, named Component Info and Tests, displays information about the component, its properties, and the tests performed on it, as shown in Figure 5.19. The user can change the stage of the component by clicking the “Change Stage” button and create a new test run by clicking the “Add New Test Run” button.

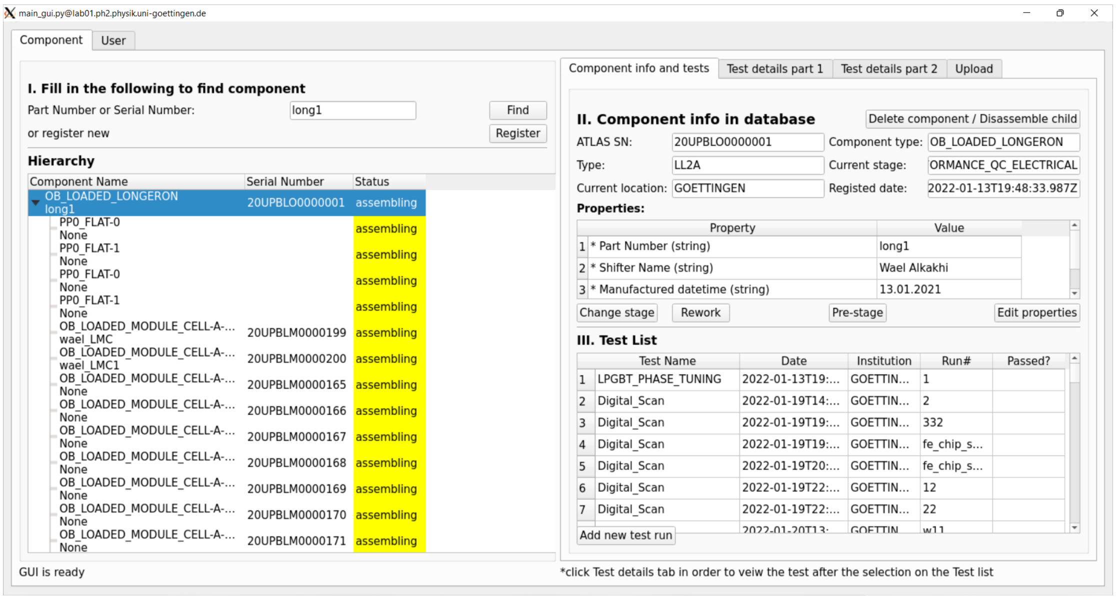
When the user selects a test in the Test List within the Component Info and Tests tab, the test details are shown in the other two tabs. The Test Details Part 1 tab displays the properties, results, and comments of the test, as shown in Figure 5.20. The user can delete the test run, and the test’s properties and results can be edited. The user can also add, delete, and edit comments.

The Test Details Part 2 tab shows the remaining test details, including attachments and defects, as shown in Figure 5.21. The user can open attachment files or images and add, update, and delete attachments. Additionally, the user can add, update, and delete defects.

The Upload tab shows information about uploading electrical tests to the loaded local support and provides the user with a link to the uploader code introduced in Section 5.5.3.

5.6. OB Loaded Module Cell

One child of the OB loaded local supports is the OB Loaded Module Cell, which represents the entire module. The OB Loaded Module Cell component type consists of the Module, OB Bare Module Cell, and OB Loading Adhesive component types as children.



main_gui.py@lab01.ph2.physik.uni-goettingen.de

Component | User

I. Fill in the following to find component

Part Number or Serial Number: long1 [Find]

or register new [Register]

Hierarchy

Component Name	Serial Number	Status
OB_LOADED_LONGERON long1	20UPBLO0000001	assembling
PP0_FLAT-0 None		assembling
PP0_FLAT-1 None		assembling
PP0_FLAT-0 None		assembling
PP0_FLAT-1 None		assembling
OB_LOADED_MODULE_CELL-A-... wael_LMC	20UPBLM0000199	assembling
OB_LOADED_MODULE_CELL-A-... wael_LMC1	20UPBLM0000200	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000165	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000166	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000167	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000168	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000169	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000170	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000171	assembling

GUI is ready

Component info and tests | Test details part 1 | Test details part 2 | Upload

II. Component info in database [Delete component / Disassemble child]

ATLAS SN: 20UPBLO0000001 Component type: OB_LOADED_LONGERON

Type: LL2A Current stage: ORMANCE_QC_ELECTRICAL

Current location: GOETTINGEN Registed date: 2022-01-13T19:48:33.987Z

Properties:

	Property	Value
1	* Part Number (string)	long1
2	* Shifter Name (string)	Wael Alkakhi
3	* Manufactured datetime (string)	13.01.2021

[Change stage] [Rework] [Pre-stage] [Edit properties]

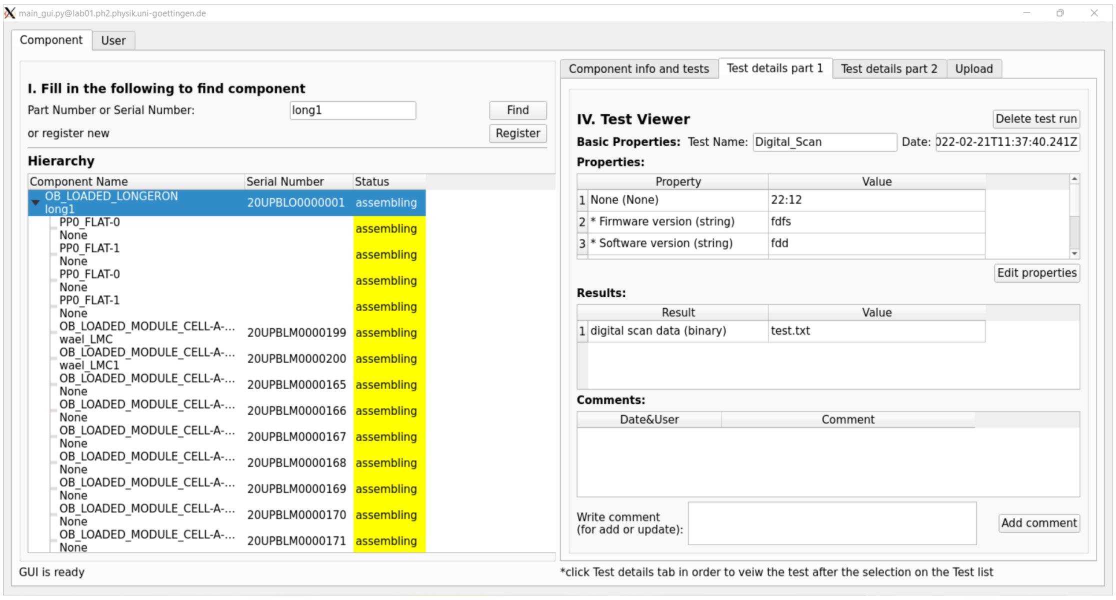
III. Test List

	Test Name	Date	Institution	Run#	Passed?
1	LPGBT_PHASE_TUNING	2022-01-13T19:...	GOETTIN...	1	
2	Digital_Scan	2022-01-19T14:...	GOETTIN...	2	
3	Digital_Scan	2022-01-19T19:...	GOETTIN...	332	
4	Digital_Scan	2022-01-19T19:...	GOETTIN...	fe_chip_s...	
5	Digital_Scan	2022-01-19T20:...	GOETTIN...	fe_chip_s...	
6	Digital_Scan	2022-01-19T22:...	GOETTIN...	12	
7	Digital_Scan	2022-01-19T22:...	GOETTIN...	22	
		2022-01-20T13:	GOETTIN	w11	

[Add new test run]

*click Test details tab in order to veiw the test after the selection on the Test list

Figure 5.19.: GUI layout.



main_gui.py@lab01.ph2.physik.uni-goettingen.de

Component | User

I. Fill in the following to find component

Part Number or Serial Number: long1 [Find]

or register new [Register]

Hierarchy

Component Name	Serial Number	Status
OB_LOADED_LONGERON long1	20UPBLO0000001	assembling
PP0_FLAT-0 None		assembling
PP0_FLAT-1 None		assembling
PP0_FLAT-0 None		assembling
PP0_FLAT-1 None		assembling
OB_LOADED_MODULE_CELL-A-... wael_LMC	20UPBLM0000199	assembling
OB_LOADED_MODULE_CELL-A-... wael_LMC1	20UPBLM0000200	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000165	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000166	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000167	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000168	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000169	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000170	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000171	assembling

GUI is ready

Component info and tests | Test details part 1 | Test details part 2 | Upload

IV. Test Viewer [Delete test run]

Basic Properties: Test Name: Digital_Scan Date: 022-02-21T11:37:40.241Z

Properties:

	Property	Value
1	None (None)	22:12
2	* Firmware version (string)	fdfs
3	* Software version (string)	fdd

[Edit properties]

Results:

	Result	Value
1	digital scan data (binary)	test.txt

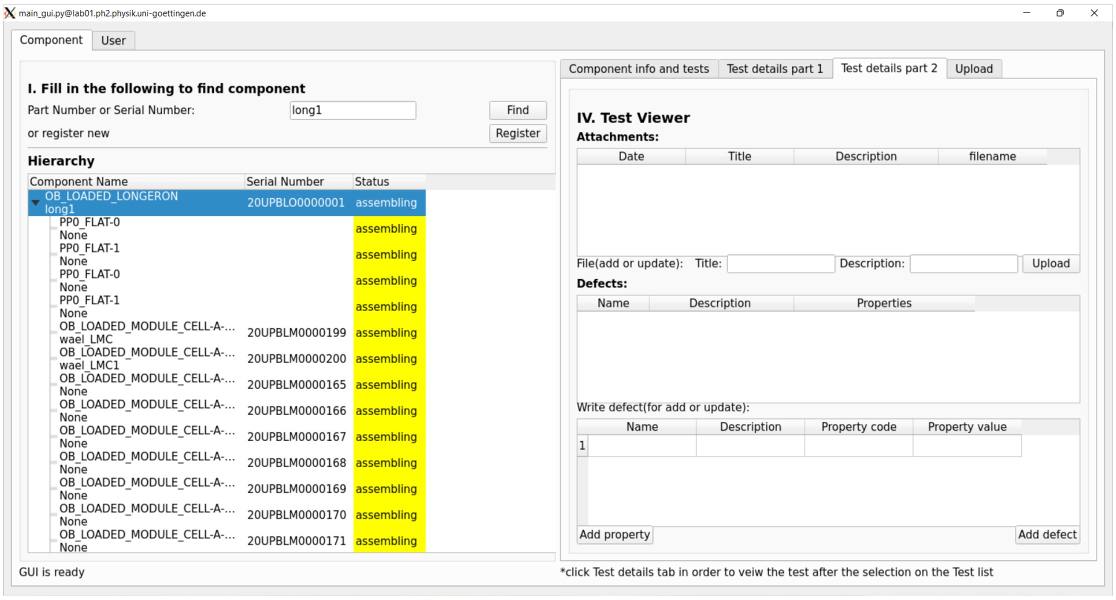
Comments:

Date&User	Comment

Write comment (for add or update): [Add comment]

*click Test details tab in order to veiw the test after the selection on the Test list

Figure 5.20.: Test details part1 GUI layout.



I. Fill in the following to find component

Part Number or Serial Number: long1

or register new

Hierarchy

Component Name	Serial Number	Status
OB_LOADED_LONGERON long1	20UPBLO0000001	assembling
PP0_FLAT-0 None		assembling
PP0_FLAT-1 None		assembling
PP0_FLAT-0 None		assembling
PP0_FLAT-1 None		assembling
OB_LOADED_MODULE_CELL-A-... wael_LMC	20UPBLM0000199	assembling
OB_LOADED_MODULE_CELL-A-... wael_LMC1	20UPBLM0000200	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000165	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000166	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000167	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000168	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000169	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000170	assembling
OB_LOADED_MODULE_CELL-A-... None	20UPBLM0000171	assembling

GUI is ready

IV. Test Viewer

Attachments:

Date	Title	Description	filename

File(add or update): Title: Description:

Defects:

Name	Description	Properties

Write defect(for add or update):

	Name	Description	Property code	Property value
1				

*click Test details tab in order to veiw the test after the selection on the Test list

Figure 5.21.: Test details part2 GUI layout.

Since there are two cell loading methods[7] being used in parallel for production, two types of OB Loaded Module Cells are implemented in ITk PD.

The first type is called Standard, where the Module component type is a child of the OB Loaded Module Cell, and the Module component type includes Bare Module and Module PCB component types as children. The second type is called Combined, where the Bare Module and Module PCB component types are directly considered children of the OB Loaded Module Cell, as shown in Figure 5.22.

According to the structure of the OB Loaded Module Cell component type, the test types of the Module component type need to be cloned to the OB Loaded Module Cell. A comparison tool is necessary to protect against future changes to one of the test types. For this reason, a code was written to compare a test type with its corresponding clone in another component. This code is generic and can be used to compare any two test types of any two component types. It is not limited to comparing Module test types with the corresponding OB Loaded Module Cell test types. The code is uploaded to GitLab and can be found at the following link: `https://gitlab.cern.ch/atlas-ITk/sw/db/pixels/compare-test-types`. The repository includes the comparison code compare.py and a documentation file that explains the installation procedure and describes the running process of the code.

The code's algorithm is shown in Figure 5.23, where the code prompts the user for their ITk PD credentials, checks the existence of the provided test types and component types, and then outputs all the differences between the two test types. Figure 5.24 shows the output of the code, highlighting the differences between the test type and its

[7] The cell loading method involves gluing the module onto the OB Bare Module Cell.

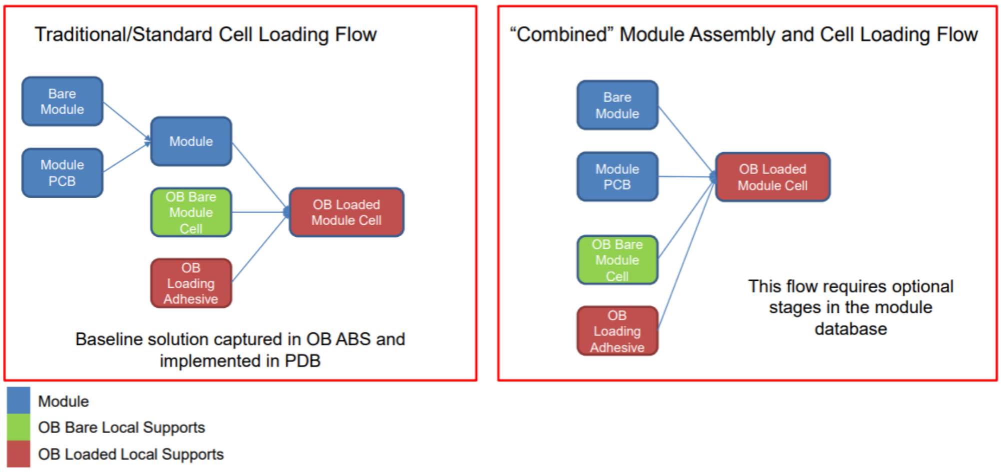


Figure 5.22.: OB Loaded Module Cell component type.

corresponding cloned test type.

5.7. ITk Component counter

During the assembly and production of ITk, managing numerous components is essential. To facilitate this, there exists a need for a code that counts the number of registered components in ITk PD and identifies the number of unused (not yet assembled) components of a specified type. Additionally, it must output details such as serial numbers, assembly status, grade, and current location for each registered component.

A code fulfilling these requirements can be found at the following link `https://gitlab.cern.ch/atlas-ITk/sw/db/pixels/ob-local-support/-/tree/master/component_counter`, named component_counter.py. Users execute the code by providing the component type code as an argument, and the results are written to a file.

The algorithm of the code is shown in Figure 5.25, where the code prompts the user for their ITk PD credentials, checks the existence of the provided component types, and then writes the result to a file. Figure 5.26 displays the output written to the file. This output includes the count of components of the specified type, the number of unused components, and detailed information for each component, including its serial number, properties, and location.

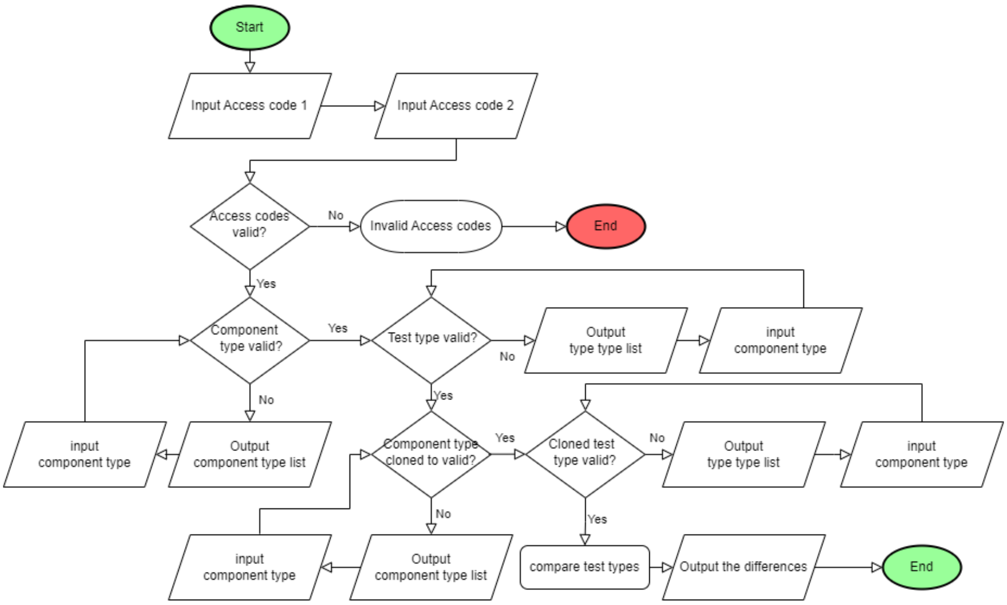
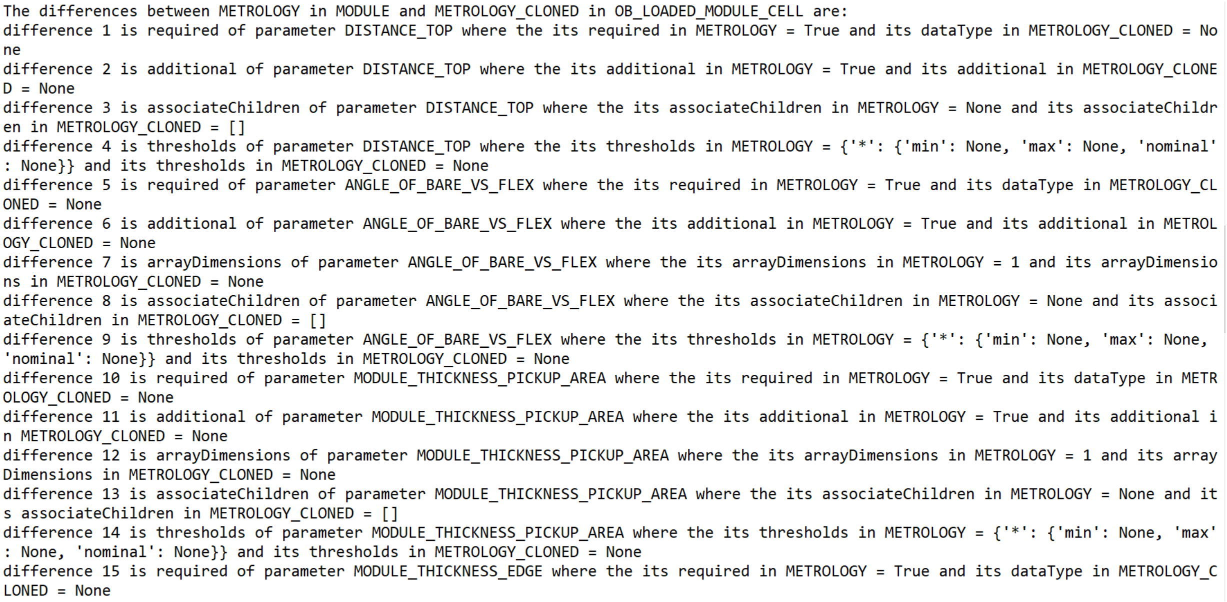


Figure 5.23.: Compare test type code algorithm.



```
The differences between METROLOGY in MODULE and METROLOGY_CLONED in OB_LOADED_MODULE_CELL are:
difference 1 is required of parameter DISTANCE_TOP where the its required in METROLOGY = True and its dataType in METROLOGY_CLONED = No
ne
difference 2 is additional of parameter DISTANCE_TOP where the its additional in METROLOGY = True and its additional in METROLOGY_CLONE
D = None
difference 3 is associateChildren of parameter DISTANCE_TOP where the its associateChildren in METROLOGY = None and its associateChildr
en in METROLOGY_CLONED = []
difference 4 is thresholds of parameter DISTANCE_TOP where the its thresholds in METROLOGY = {'*': {'min': None, 'max': None, 'nominal'
: None}} and its thresholds in METROLOGY_CLONED = None
difference 5 is required of parameter ANGLE_OF_BARE_VS_FLEX where the its required in METROLOGY = True and its dataType in METROLOGY_CL
ONED = None
difference 6 is additional of parameter ANGLE_OF_BARE_VS_FLEX where the its additional in METROLOGY = True and its additional in METROL
OGY_CLONED = None
difference 7 is arrayDimensions of parameter ANGLE_OF_BARE_VS_FLEX where the its arrayDimensions in METROLOGY = 1 and its arrayDimensio
ns in METROLOGY_CLONED = None
difference 8 is associateChildren of parameter ANGLE_OF_BARE_VS_FLEX where the its associateChildren in METROLOGY = None and its associ
ateChildren in METROLOGY_CLONED = []
difference 9 is thresholds of parameter ANGLE_OF_BARE_VS_FLEX where the its thresholds in METROLOGY = {'*': {'min': None, 'max': None,
'nominal': None}} and its thresholds in METROLOGY_CLONED = None
difference 10 is required of parameter MODULE_THICKNESS_PICKUP_AREA where the its required in METROLOGY = True and its dataType in METR
OLOGY_CLONED = None
difference 11 is additional of parameter MODULE_THICKNESS_PICKUP_AREA where the its additional in METROLOGY = True and its additional i
n METROLOGY_CLONED = None
difference 12 is arrayDimensions of parameter MODULE_THICKNESS_PICKUP_AREA where the its arrayDimensions in METROLOGY = 1 and its array
Dimensions in METROLOGY_CLONED = None
difference 13 is associateChildren of parameter MODULE_THICKNESS_PICKUP_AREA where the its associateChildren in METROLOGY = None and it
s associateChildren in METROLOGY_CLONED = []
difference 14 is thresholds of parameter MODULE_THICKNESS_PICKUP_AREA where the its thresholds in METROLOGY = {'*': {'min': None, 'max'
: None, 'nominal': None}} and its thresholds in METROLOGY_CLONED = None
difference 15 is required of parameter MODULE_THICKNESS_EDGE where the its required in METROLOGY = True and its dataType in METROLOGY_C
LONED = None
```

Figure 5.24.: Output of compare test types code.

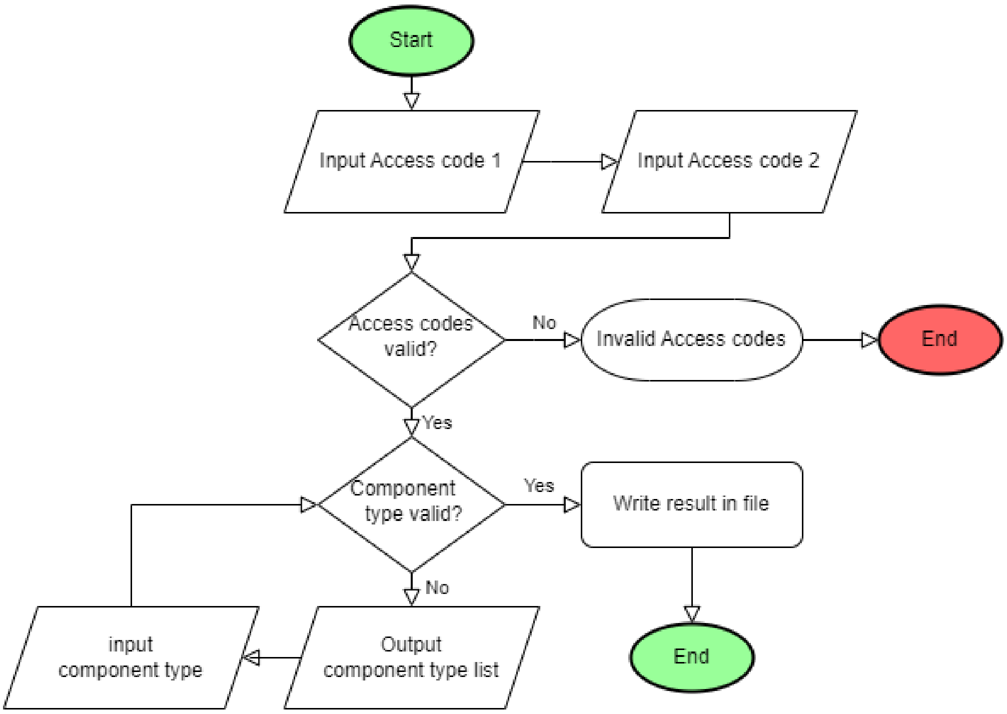
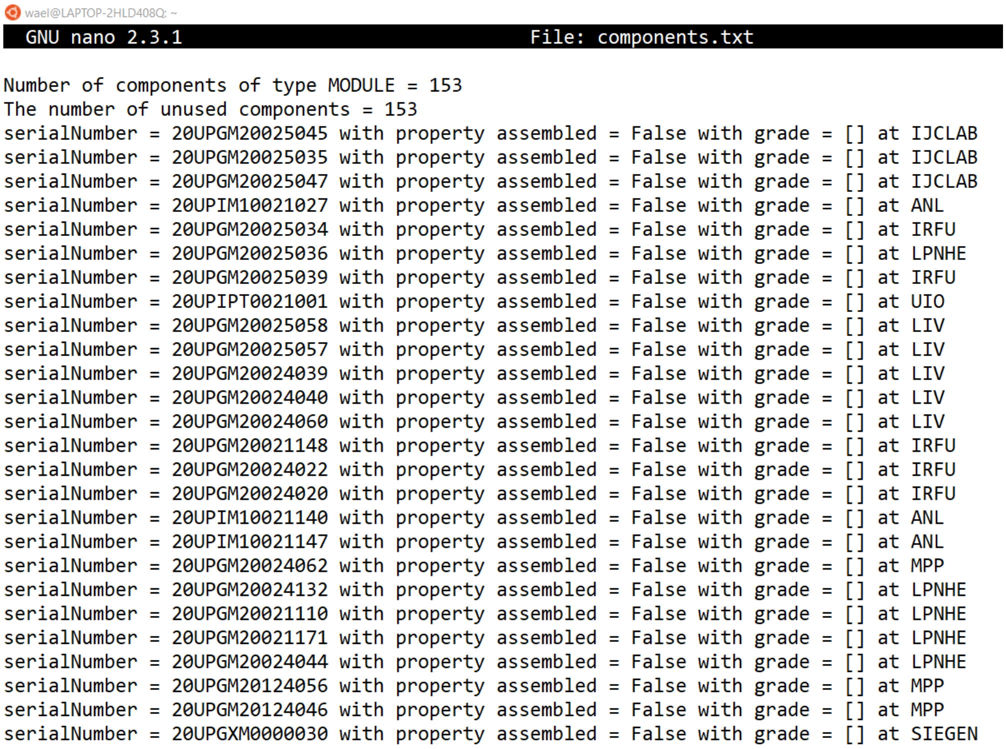


Figure 5.25.: ITk component counter code algorithm.



```
wael@LAPTOP-2HLD408Q: ~
  GNU nano 2.3.1                           File: components.txt

Number of components of type MODULE = 153
The number of unused components = 153
serialNumber = 20UPGM20025045 with property assembled = False with grade = [] at IJCLAB
serialNumber = 20UPGM20025035 with property assembled = False with grade = [] at IJCLAB
serialNumber = 20UPGM20025047 with property assembled = False with grade = [] at IJCLAB
serialNumber = 20UPIM10021027 with property assembled = False with grade = [] at ANL
serialNumber = 20UPGM20025034 with property assembled = False with grade = [] at IRFU
serialNumber = 20UPGM20025036 with property assembled = False with grade = [] at LPNHE
serialNumber = 20UPGM20025039 with property assembled = False with grade = [] at IRFU
serialNumber = 20UPIPT0021001 with property assembled = False with grade = [] at UIO
serialNumber = 20UPGM20025058 with property assembled = False with grade = [] at LIV
serialNumber = 20UPGM20025057 with property assembled = False with grade = [] at LIV
serialNumber = 20UPGM20024039 with property assembled = False with grade = [] at LIV
serialNumber = 20UPGM20024040 with property assembled = False with grade = [] at LIV
serialNumber = 20UPGM20024060 with property assembled = False with grade = [] at LIV
serialNumber = 20UPGM20021148 with property assembled = False with grade = [] at IRFU
serialNumber = 20UPGM20024022 with property assembled = False with grade = [] at IRFU
serialNumber = 20UPGM20024020 with property assembled = False with grade = [] at IRFU
serialNumber = 20UPIM10021140 with property assembled = False with grade = [] at ANL
serialNumber = 20UPIM10021147 with property assembled = False with grade = [] at ANL
serialNumber = 20UPGM20024062 with property assembled = False with grade = [] at MPP
serialNumber = 20UPGM20024132 with property assembled = False with grade = [] at LPNHE
serialNumber = 20UPGM20021110 with property assembled = False with grade = [] at LPNHE
serialNumber = 20UPGM20021171 with property assembled = False with grade = [] at LPNHE
serialNumber = 20UPGM20024044 with property assembled = False with grade = [] at LPNHE
serialNumber = 20UPGM20124056 with property assembled = False with grade = [] at MPP
serialNumber = 20UPGM20124046 with property assembled = False with grade = [] at MPP
serialNumber = 20UPGXM0000030 with property assembled = False with grade = [] at SIEGEN
```

Figure 5.26.: Output of component counter code.

CHAPTER 6

YARR BDAQ Readout System

The current readout chips of the pixel detector in the ATLAS inner detector are not suitable for the high luminosity upgrade of the LHC due to new HL-LHC requirements. Thus, new readout chips for pixel detectors in ATLAS have been developed by the RD53 collaboration. The RD53 collaboration is a research and development collaboration between the ATLAS and CMS experiments. Its goal is to develop new-generation readout ASICs for hybrid silicon pixel detectors to meet the requirements of the HL-LHC. The first prototype readout chip is RD53A, while the first production readout chip is named RD53B, which comes in two types: ITkPix for the ATLAS experiment and CROC_V1 for the CMS experiment. These readout chips require a readout system comprising both readout hardware and readout software. For this reason, there are two readout systems designed to read out these chips: the BDAQ53 readout system and the YARR readout system.

The work presented in this chapter aims to develop the YARR software to communicate with the BDAQ53 hardware to read out the RD53A and RD53B (ITkPix) readout chips.

In the following chapter, the readout chips are introduced in Section 6.1. The BDAQ53 readout system and the YARR readout system are introduced in Section 6.2 and Section 6.3, respectively. The development of YARR software using BDAQ hardware for RD53A and RD53B (ITkPix) is discussed in Section 6.4 and Section 6.5, respectively.

6.1. Readout Chips

As mentioned above, the current readout chips of the pixel detector are not suitable for the HL-LHC ATLAS experiment. For this reason, new readout chips have been developed. In the following sections, the RD53A readout chip and the RD53B (ITkPix) readout chip are introduced.

6.1.1. RD53A Readout Chip

The RD53A readout chip [76] [77] is designed to meet several challenging conditions, such as high granularity with small pixels of $50 \times 50\,\mu\text{m}^2$ or $25 \times 100\,\mu\text{m}^2$, a high hit rate of $3\,\text{GHz}\,\text{cm}^{-2}$, a chip size of $2 \times 2\,\text{cm}^2$, handling a long latency of the trigger signal of $\approx 12.5\,\mu\text{s}$ and operating with serial powering[1].

The RD53A chip is developed using 65 nm CMOS technology [78]. It measures 20 mm in width and 11.6 mm in length, containing 400 columns and 192 rows of $50 \times 50\,\mu\text{m}^2$ pixels. Each pixel includes an Analog Front-End (AFE), which is the first stage of the analog circuitry responsible for amplifying and filtering signals from the elements. Every four pixels are grouped into an analog island, which is placed within a sea of synthesized digital logic, following the analog islands in a digital sea design, as shown in Figure 6.1. The pixel core consists of 4×4 analog islands (8×8 pixels), forming the pixel matrix. Thus, the entire pixel matrix comprises 1,200 pixel cores.

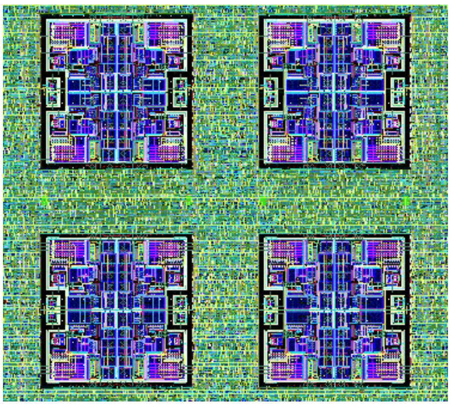


Figure 6.1.: The RD53A layout features four analog islands, each consisting of 4 pixels, embedded within a synthesized digital "sea" [77].

Figure 6.2 shows the three types of analog front-ends used by the RD53A chip: Synchronous FE, Linear FE, and Differential FE. The Synchronous FE [79] uses a baseline auto-zeroing scheme that requires periodic acquisition of a baseline instead of pixel-by-pixel threshold trimming. The Linear FE [79] uses linear pulse amplification to compare the pulse to the threshold voltage. The Differential FE [79] uses a differential gain stage in front of the discriminator and implements a threshold by unbalancing the two branches.

The RD53A chip can be operated and controlled with a single link at a speed of 160 Mbit/s, and its data can be transmitted through one to four links at speeds up to 1280 Mbit/s. The RD53A data output is encoded with Aurora 64b/66b on one to four lanes at a speed of 1.28 Gbit/s. Figure 6.4 shows the data output format, where RD53A sends blocks of N data frames (N is programmable) separated by one register frame.

[1] Serial powering is a method of powering readout chips that reduces power consumption and the number of cables used.

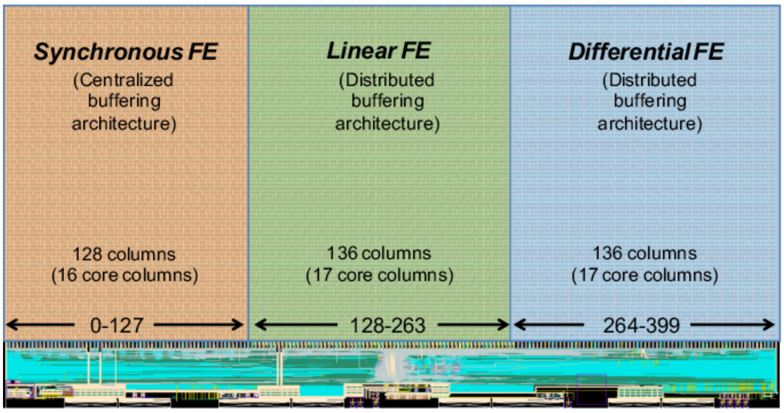

Synchronous FE
(Centralized buffering architecture)
128 columns (16 core columns)
0-127
Linear FE
(Distributed buffering architecture)
136 columns (17 core columns)
128-263
Differential FE
(Distributed buffering architecture)
136 columns (17 core columns)
264-399

Figure 6.2.: The types and locations of the analog front-ends in the RD53A chip [78].

The register frame carries information of two registers. The data frame is composed of a 64-bit data word, consisting of a 2-bit header and 0 to 2 32-bit hit data words.

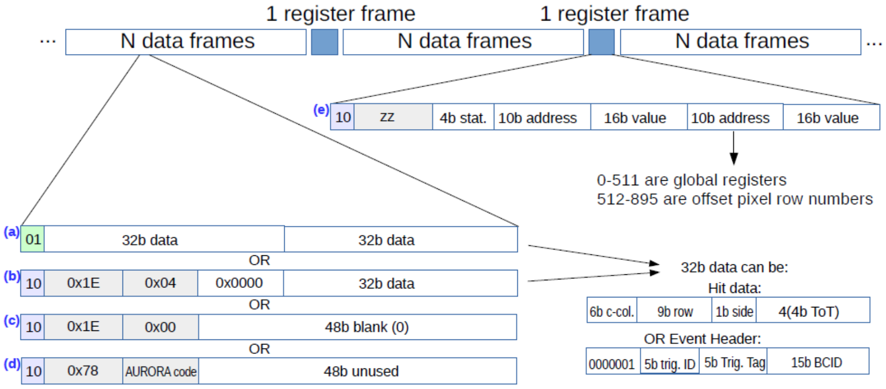

1 register frame
1 register frame
... N data frames
N data frames
N data frames ...
(e) 10 | zz | 4b stat. | 10b address | 16b value | 10b address | 16b value
0-511 are global registers
512-895 are offset pixel row numbers
(a) 01 | 32b data | 32b data
OR
(b) 10 | 0x1E | 0x04 | 0x0000 | 32b data
OR
(c) 10 | 0x1E | 0x00 | 48b blank (0)
OR
(d) 10 | 0x78 | AURORA code | 48b unused
32b data can be:
Hit data:
6b c-col. | 9b row | 1b side | 4(4b ToT)
OR Event Header:
0000001 | 5b trig. ID | 5b Trig. Tag | 15b BCID

Figure 6.3.: The RD53A data output format [77].

The RD53A chip also offers the option to operate with serial powering, which reduces power loss and decreases the number of cables used, thereby lowering the material budget.

6.1.2. RD53B ITkPix Readout Chip

After producing the RD53A readout chip as a prototype version, the RD53 collaboration produced the RD53B production version in two types: ITkPix for the ATLAS experiment and CROC_V1 for the CMS experiment.

The RD53B(ITkPix) readout chip [80] features only a differential FE type for AFE. It also uses 65 nm CMOS technology and includes a serial powering option. All specifications for the ITkPix readout chip are detailed in Table 6.1.

Chip size	$20 \times 21\,\mathrm{mm}^2$
Pixel size	$50 \times 50\,\mathrm{\mu m}^2$
Hit rate	$3\,\mathrm{GHz\,cm}^{-2}$
Trigger rate	1 MHz
Latency	$12.8\,\mathrm{\mu s}$
Data rate	5.12 Gbit/s
Low threshold	600e
Radiation tolerance	500 Mrad
Low power	$4\,\mathrm{\mu A/pixel}$

Table 6.1.: RD53B ITkPix specifications [81].

There are many versions of the ITkPix readout chip. The first version of the ITkPix readout chip is ITkPixV1. Different improved versions of the first ITkPixV1 chip have been produced to fix issues found during testing. For instance, the ITkPixV1.1 version is an improved version that addresses the issue with the time-over-threshold (ToT) latch. The ITkPixV2 version is the final and production version.

The RD53B data output stream is composed of packages of N data blocks separated by a service block that contains some register information. The hit output data of RD53B is formed by a set of stream bits with a header indicating the beginning of the stream. The stream ends with a set of zero-value bits called orphan bits to fill the 66-bit data block. The 66-bit data block, shown in Figure 6.4, consists of a 2-bit header, a 1-bit new stream indicator (which is 1 if the data block holds a new stream and 0 if the data block belongs to the current stream), and the remaining bits are data hits.

Additionally, the ITkPix RD53B readout chip offers a data merging feature called link-sharing, which allows data from four FE chips in a quad module to be merged and sent via one or two lanes instead of four lanes.

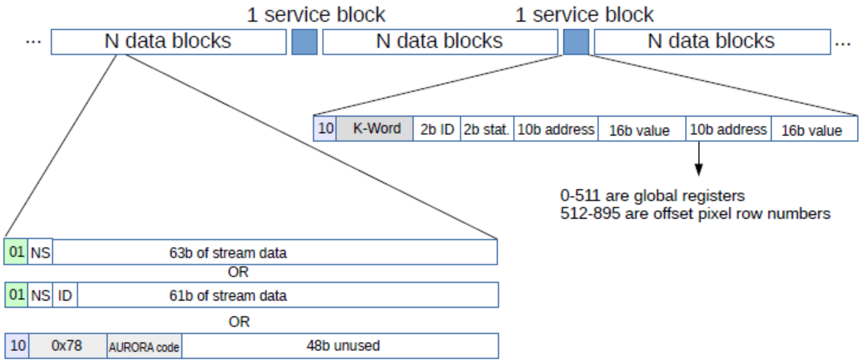


Figure 6.4.: The RD53B data output format [81].

6.2. BDAQ53

To operate the readout chips, there is a need to develop readout systems, generaly making use of an FPGA platform. One example is the Bdaq53 [82] readout system, developed at Rheinische Friedrich-Wilhelms-Universität Bonn, capable of reading out both RD53A and RD53B readout chips. It consists of hardware and software components.

The hardware is a Xilinx FPGA-based board, named the Bdaq53 board, shown in Figure 6.5. It can be connected to readout chips via DP cable and has 4 DP ports. One port has one command lane and 4 readout data lanes, providing the ability to read out a QM or an SCC with 4 lanes. The other 3 DP ports each have one command lane and one readout data lane. The data transmission speed can be either 1.28 Gbit/s or 640 Mbit/s. The Bdaq53 board is connected to a DAQ PC, which hosts the Bdaq53 software, via an Ethernet connection. Additionally, it has a JTAG connection for firmware flashing purposes.

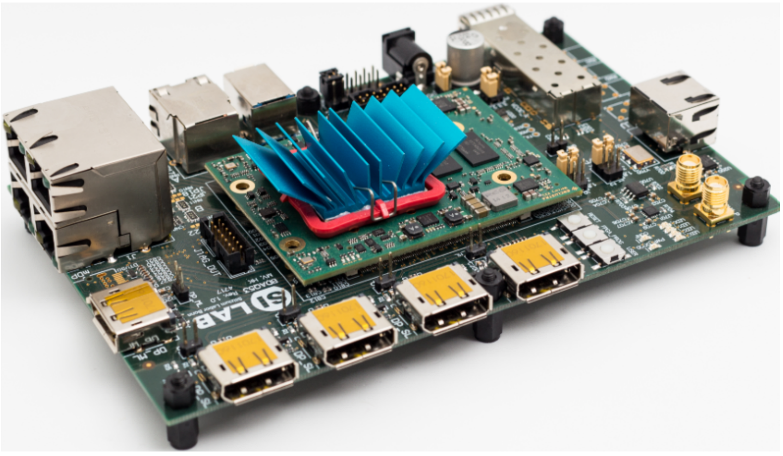


Figure 6.5.: The bdaq53 board [82].

The Bdaq53 software is Python-based. It is built on data acquisition source and the Basil software, where Basil supplies the software with FPGA firmware modules, such as FIFO and drivers. The Bdaq53 can read and write readout chip configurations, operate scans, and control the readout chip. Additionally, it processes, stores, and analyzes the incoming readout data from the chips.

6.3. YARR

YARR (Yet Another Rapid Readout) [83] [84] is another DAQ readout system that can read out the RD53A and RD53B readout chips. It uses FPGA hardware with a PCIe connection to the DAQ PC to interact with the YARR software, utilizing commercial FPGA boards with PCIe connections like the Xilinx-KC705 development board.

The YARR software is C++-based and consists of two types of libraries. The first type is the chip libraries, where each readout chip type has a corresponding chip library. These libraries are responsible for preparing commands to be sent to the readout chip, processing and analyzing the readout data from the chip, and storing this data in databases. Additionally, the chip libraries provide the capability to create histograms for visualizing the results.

The second type is the controller libraries, which provide the interface between the chip libraries and the hardware. Each hardware type has a corresponding controller in the YARR software. The controller prepares the commands created by the chip library to the hardware and sends them to the hardware. It also receives the readout data from the hardware and prepares it for the chip library. Within the controller library, there are TxCore and RxCore codes responsible for sending commands to the readout chip and receiving readout data from the chip, respectively.

In this work, the Bdaq controller is developed within the YARR software, as shown in Figure 6.6. The readout chip (RD53A or RD53B) is connected to the Bdaq53 board, which in turn is connected to the YARR software. The YARR software includes libRd53a and libRd53b libraries. The libRd53a library creates commands for the RD53A chip and processes the readout data received from the RD53A chip, while the libRd53b library performs the same functions for the RD53B chip. Both libRd53A and libRd53b interact with the libBdaq controller. The libBdaq contains BdaqTxCore, which prepares commands to send to the Bdaq53 board, and BdaqRxCore, which processes the data received from the Bdaq53 board and sends it to either libRd53a or libRd53b.

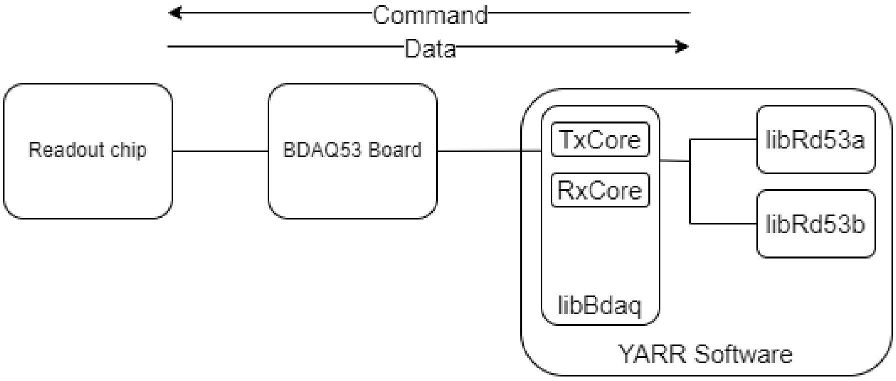


Figure 6.6.: The readout chain consists of the readout chip, the Bdaq53 board, and the YARR software. The direction of the commands sent and the readout data received are also shown.

6.4. YARR BDAQ readout for RD53A

Figure 6.7 shows the lab setup of the project for the RD53A readout chips. Two RD53A SCCs are connected to the Bdaq53 board via a DP-DP cable, and the Bdaq53 board is connected to the PC via an Ethernet cable. The PC hosts the YARR software. The

RD53A SCC chips can be enabled and disabled by the YARR software during operation.

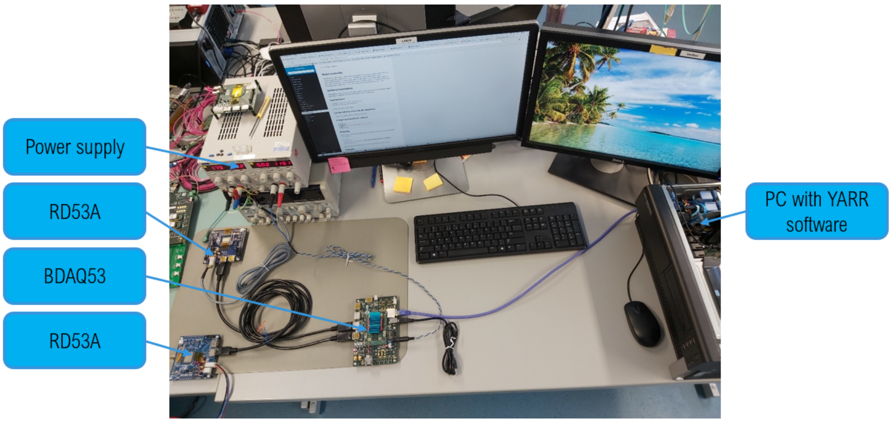


Figure 6.7.: The lab setup for developing the YARR software to read out RD53A readout chips using the Bdaq53 board.

The BDAQ53 support in YARR was developped few years ago in the Goettingen group for reading out RD53A SCC using the Bdaq board. However, after certain developments in parts of the YARR software[2], the YARR Bdaq controller was broken due to changes in some C++ YARR components' types. This required significant development work to adapt the YARR Bdaq controller to the new YARR algorithm. Additionally, it required developing the readout data process of the Bdaq controller in the YARR software to receive data from the Bdaq, prepare it, and send it to libRd53a for processing. The new algorithm for the readout process in the Bdaq controller is explained in Section A.1.

In the digital scan, a digital signal (hit) is injected into every enabled pixel to test the digital part of the chip. In the YARR digital scan, 100 hits are injected into each pixel. To validate the readout, a YARR digital scan must be performed, and it must output an occupancy map with 100 hits at each pixel, as shown in Figure 6.8.

After these developments, the YARR Bdaq controller was fixed and validated by running a YARR digital scan with one RD53A SCC. The scan was successful, as shown in Figure 6.9. It shows zero failed pixels, where a failed pixel is defined as a pixel that does not have a value of 100 hits. The scan took 22.373 s seconds to complete.

Moreover, the YARR software was validated by running a digital scan with 2 RD53A SCCs as shown in Figure 6.10. The figure shows a successful scan with 2 RD53A SCCs having zero failed pixels, indicating that all the loops over the readout channels were executed properly. The scan took 264.631 s to complete.

[2] YARR was initially collecting data from all the data lanes and then processing it together, while in the new version, YARR processes the data from each data lane separately.

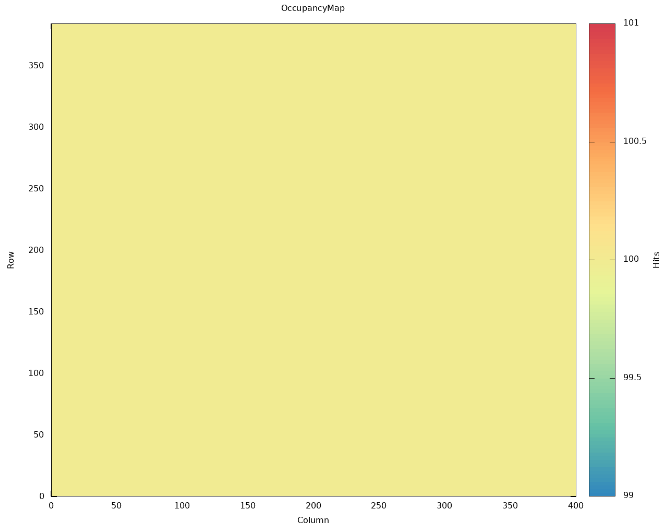
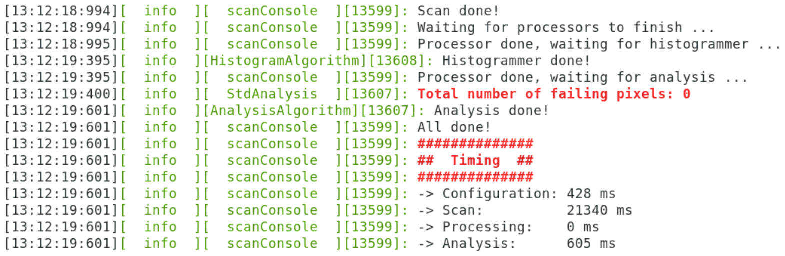


Figure 6.8.: An occupancy map with 100 hits at every pixel.



```
[13:12:18:994][  info  ][  scanConsole  ][13599]: Scan done!
[13:12:18:994][  info  ][  scanConsole  ][13599]: Waiting for processors to finish ...
[13:12:18:995][  info  ][  scanConsole  ][13599]: Processor done, waiting for histogrammer ...
[13:12:19:395][  info  ][HistogramAlgorithm][13608]: Histogrammer done!
[13:12:19:395][  info  ][  scanConsole  ][13599]: Processor done, waiting for analysis ...
[13:12:19:400][  info  ][  StdAnalysis  ][13607]: Total number of failing pixels: 0
[13:12:19:601][  info  ][AnalysisAlgorithm][13607]: Analysis done!
[13:12:19:601][  info  ][  scanConsole  ][13599]: All done!
[13:12:19:601][  info  ][  scanConsole  ][13599]: ##############
[13:12:19:601][  info  ][  scanConsole  ][13599]: ##  Timing  ##
[13:12:19:601][  info  ][  scanConsole  ][13599]: ##############
[13:12:19:601][  info  ][  scanConsole  ][13599]: -> Configuration: 428 ms
[13:12:19:601][  info  ][  scanConsole  ][13599]: -> Scan:          21340 ms
[13:12:19:601][  info  ][  scanConsole  ][13599]: -> Processing:    0 ms
[13:12:19:601][  info  ][  scanConsole  ][13599]: -> Analysis:      605 ms
```

Figure 6.9.: Output of a successful YARR digital scan of the RD53A SCC using the Bdaq board.

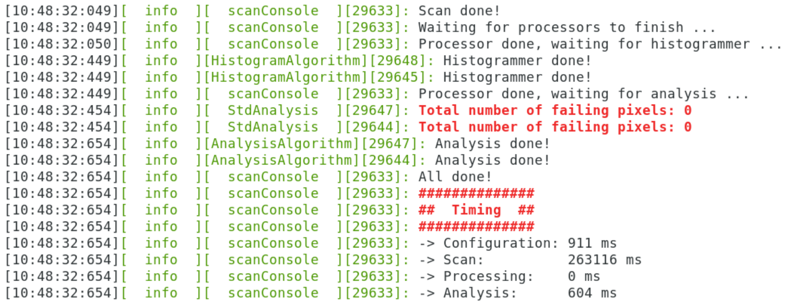
6.5. YARR BDAQ readout for RD53B(ITkPix)

After the successful development of the YARR bdaq controller to read out RD53A readout chips, additional YARR bdaq controller developments were done to read out RD53B (ITkPix) readout chips. The YARR bdaq controller was initially developed to read out ITkPix SCC and was subsequently further developed to read out an ITkPix quad module.

6.5.1. Readout development using RD53B(ITkPix) SCC

Figure 6.11 shows the used setup to read out the ITkPix single chip card. It is the same as the setup for the RD53A SCC, but with the RD53A SCCs replaced by an ITkPix SCC. Additionally, a fan is added to cool down the ITkPix SCC.

The first step was to avoid disrupting the successful BDAQ controller developments for reading out RD53A. For this reason, a parameter specifying the readout chip type



```
[10:48:32:049][  info  ][  scanConsole  ][29633]: Scan done!
[10:48:32:049][  info  ][  scanConsole  ][29633]: Waiting for processors to finish ...
[10:48:32:050][  info  ][  scanConsole  ][29633]: Processor done, waiting for histogrammer ...
[10:48:32:449][  info  ][HistogramAlgorithm][29648]: Histogrammer done!
[10:48:32:449][  info  ][HistogramAlgorithm][29645]: Histogrammer done!
[10:48:32:449][  info  ][  scanConsole  ][29633]: Processor done, waiting for analysis ...
[10:48:32:454][  info  ][  StdAnalysis  ][29647]: Total number of failing pixels: 0
[10:48:32:454][  info  ][  StdAnalysis  ][29644]: Total number of failing pixels: 0
[10:48:32:654][  info  ][AnalysisAlgorithm][29647]: Analysis done!
[10:48:32:654][  info  ][AnalysisAlgorithm][29644]: Analysis done!
[10:48:32:654][  info  ][  scanConsole  ][29633]: All done!
[10:48:32:654][  info  ][  scanConsole  ][29633]: ##############
[10:48:32:654][  info  ][  scanConsole  ][29633]: ##  Timing  ##
[10:48:32:654][  info  ][  scanConsole  ][29633]: ##############
[10:48:32:654][  info  ][  scanConsole  ][29633]: -> Configuration: 911 ms
[10:48:32:654][  info  ][  scanConsole  ][29633]: -> Scan:          263116 ms
[10:48:32:654][  info  ][  scanConsole  ][29633]: -> Processing:    0 ms
[10:48:32:654][  info  ][  scanConsole  ][29633]: -> Analysis:      604 ms
```

Figure 6.10.: Output of a successful YARR digital scan of 2 RD53A SCCs using the Bdaq board.

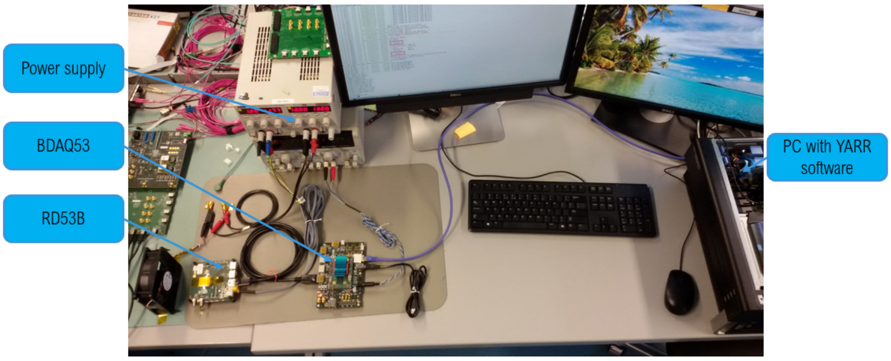


Figure 6.11.: The lab setup for developing the YARR software to read out the RD53B SCC using the Bdaq53 board.

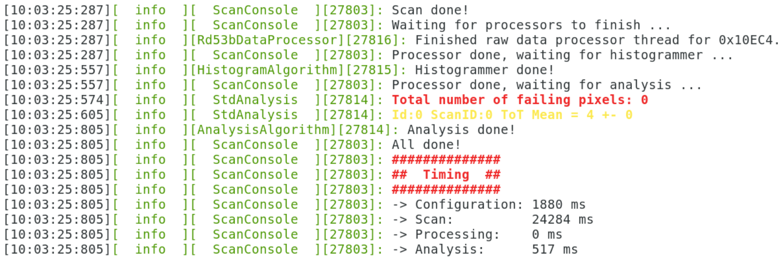
was added. Then, a process for reading out the received data words from BDAQ was created to prepare the data for the libRd53b library for later analysis. The readout process is discussed in Section A.2. These developments led to a uniform occupancy map of the YARR digital scan with three hit values at each pixel, indicating that part of the readout data was lost.

Inspired by the BDAQ53 software and understanding the trigger sent to the readout chip, the solution to this issue was to add some waiting times to allow the BDAQ controller to process all the received readout data and send synchronization commands to the readout chip simultaneously to avoid losing synchronization.

These developments led to a successful YARR BDAQ controller capable of reading out the ITkPix SCC. Following this development, the YARR digital scan of the ITkPix SCC performed properly with zero failed pixels, as shown in Figure 6.12, where the scan took 26.681 s.

6.5.2. Readout development using RD53B(ITkPix) Quad Module

After receiving a digital quad module, there was an opportunity to test the developed YARR bdaq controller with the QM. Figure 6.13 shows the setup for reading out the



```
[10:03:25:287][  info  ][  ScanConsole  ][27803]: Scan done!
[10:03:25:287][  info  ][  ScanConsole  ][27803]: Waiting for processors to finish ...
[10:03:25:287][  info  ][Rd53bDataProcessor][27816]: Finished raw data processor thread for 0x10EC4.
[10:03:25:287][  info  ][  ScanConsole  ][27803]: Processor done, waiting for histogrammer ...
[10:03:25:557][  info  ][HistogramAlgorithm][27815]: Histogrammer done!
[10:03:25:557][  info  ][  ScanConsole  ][27803]: Processor done, waiting for analysis ...
[10:03:25:574][  info  ][  StdAnalysis  ][27814]: Total number of failing pixels: 0
[10:03:25:605][  info  ][  StdAnalysis  ][27814]: Id:0 ScanID:0 ToT Mean = 4 +- 0
[10:03:25:805][  info  ][AnalysisAlgorithm][27814]: Analysis done!
[10:03:25:805][  info  ][  ScanConsole  ][27803]: All done!
[10:03:25:805][  info  ][  ScanConsole  ][27803]: ##############
[10:03:25:805][  info  ][  ScanConsole  ][27803]: ##  Timing  ##
[10:03:25:805][  info  ][  ScanConsole  ][27803]: ##############
[10:03:25:805][  info  ][  ScanConsole  ][27803]: -> Configuration: 1880 ms
[10:03:25:805][  info  ][  ScanConsole  ][27803]: -> Scan:          24284 ms
[10:03:25:805][  info  ][  ScanConsole  ][27803]: -> Processing:    0 ms
[10:03:25:805][  info  ][  ScanConsole  ][27803]: -> Analysis:      517 ms
```

Figure 6.12.: Output of a successful YARR digital scan of ITkPix SCC using the Bdaq board.

ITkPix QM.

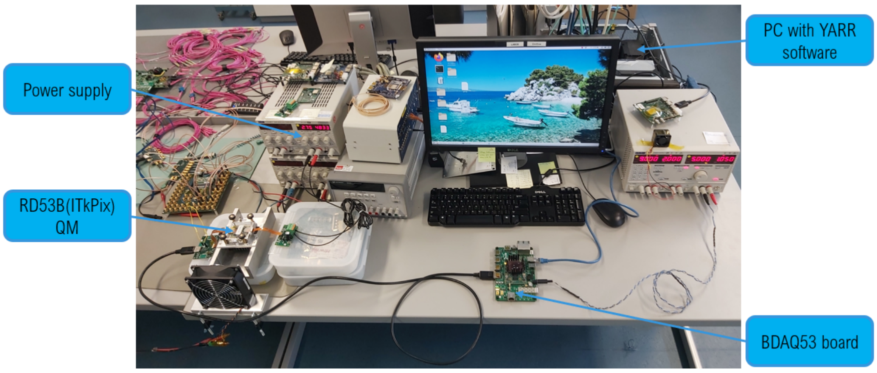
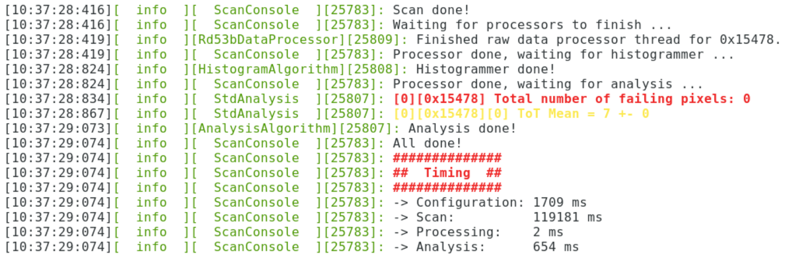


Figure 6.13.: The lab setup used to develop the YARR software for reading out the RD53B QM using the Bdaq53 board.

The YARR bdaq controller was validated by running a digital scan for one readout chip of the ITkPix QM, and then scaling up to validate the YARR bdaq controller for the entire QM.

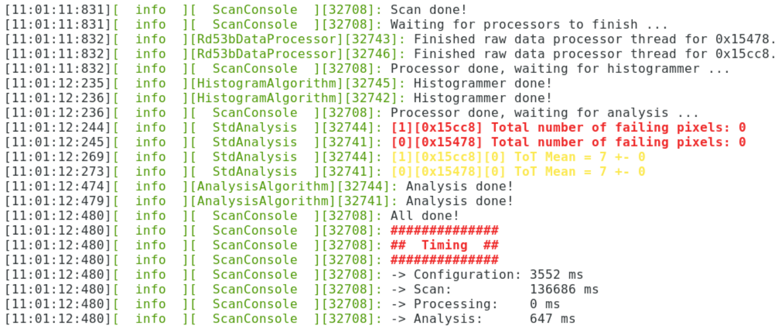
The YARR bdaq controller was successfully validated with one readout chip enabled. It was then validated successfully with two readout chips of the ITkPix QM enabled. Subsequently, it was validated properly with three enabled readout chips of the ITkPix QM. However, when the entire QM was operated, the scan failed due to data processing errors caused by overwriting the incoming data buffer. The solution to avoid overwriting the buffer was to maximize the data buffer size and process the trigger command repetitions in the software rather than in the firmware, since the firmware processes data faster than the software. After this modification, the YARR bdaq controller was properly validated for the entire ITkPix QM.

Figure 6.14, Figure 6.15, Figure 6.16, and Figure 6.17 show the output of proper YARR digital scan with zero failed pixels for one enabled ITkPix readout chip, two enabled ITkPix readout chips, three enabled ITkPix readout chips, and all four enabled ITkPix readout chips of the QM, respectively.



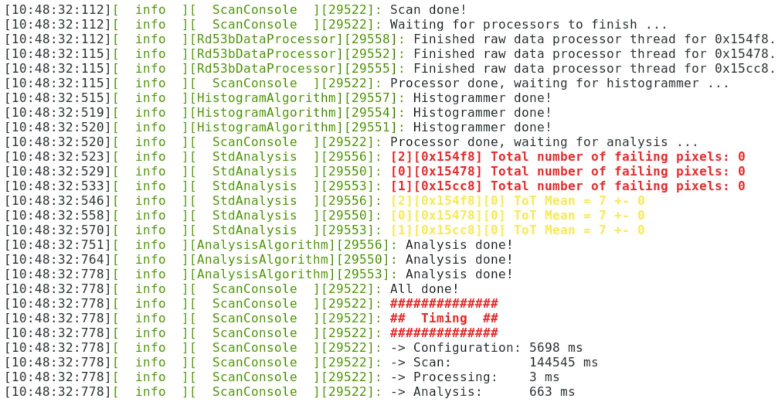
```
[10:37:28:416][  info  ][  ScanConsole  ][25783]: Scan done!
[10:37:28:416][  info  ][  ScanConsole  ][25783]: Waiting for processors to finish ...
[10:37:28:419][  info  ][Rd53bDataProcessor][25809]: Finished raw data processor thread for 0x15478.
[10:37:28:419][  info  ][  ScanConsole  ][25783]: Processor done, waiting for histogrammer ...
[10:37:28:824][  info  ][HistogramAlgorithm][25808]: Histogrammer done!
[10:37:28:824][  info  ][  ScanConsole  ][25783]: Processor done, waiting for analysis ...
[10:37:28:834][  info  ][  StdAnalysis  ][25807]: [0][0x15478] Total number of failing pixels: 0
[10:37:28:867][  info  ][  StdAnalysis  ][25807]: [0][0x15478][0] ToT Mean = 7 +- 0
[10:37:29:073][  info  ][AnalysisAlgorithm][25807]: Analysis done!
[10:37:29:074][  info  ][  ScanConsole  ][25783]: All done!
[10:37:29:074][  info  ][  ScanConsole  ][25783]: ##############
[10:37:29:074][  info  ][  ScanConsole  ][25783]: ##  Timing  ##
[10:37:29:074][  info  ][  ScanConsole  ][25783]: ##############
[10:37:29:074][  info  ][  ScanConsole  ][25783]: -> Configuration: 1709 ms
[10:37:29:074][  info  ][  ScanConsole  ][25783]: -> Scan:          119181 ms
[10:37:29:074][  info  ][  ScanConsole  ][25783]: -> Processing:    2 ms
[10:37:29:074][  info  ][  ScanConsole  ][25783]: -> Analysis:      654 ms
```

Figure 6.14.: Output of a successful YARR digital scan of one readout chip of the ITkPix QM using the Bdaq board. The scan took 121.546 s.



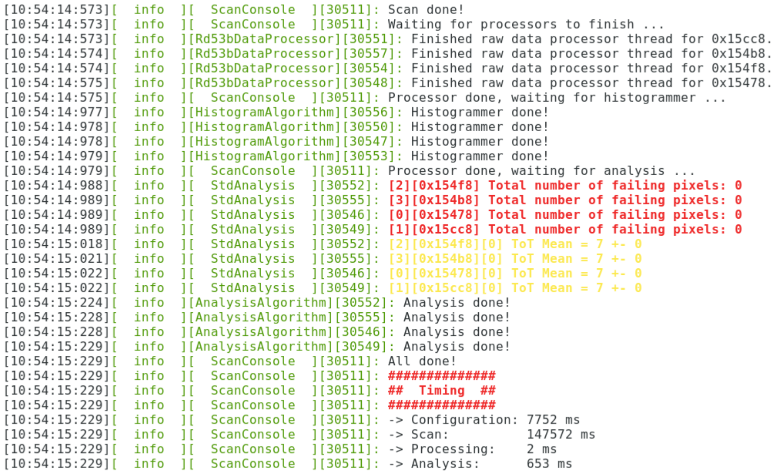
```
[11:01:11:831][  info  ][  ScanConsole  ][32708]: Scan done!
[11:01:11:831][  info  ][  ScanConsole  ][32708]: Waiting for processors to finish ...
[11:01:11:832][  info  ][Rd53bDataProcessor][32743]: Finished raw data processor thread for 0x15478.
[11:01:11:832][  info  ][Rd53bDataProcessor][32746]: Finished raw data processor thread for 0x15cc8.
[11:01:11:832][  info  ][  ScanConsole  ][32708]: Processor done, waiting for histogrammer ...
[11:01:12:235][  info  ][HistogramAlgorithm][32745]: Histogrammer done!
[11:01:12:236][  info  ][HistogramAlgorithm][32742]: Histogrammer done!
[11:01:12:236][  info  ][  ScanConsole  ][32708]: Processor done, waiting for analysis ...
[11:01:12:244][  info  ][  StdAnalysis  ][32744]: [1][0x15cc8] Total number of failing pixels: 0
[11:01:12:245][  info  ][  StdAnalysis  ][32741]: [0][0x15478] Total number of failing pixels: 0
[11:01:12:269][  info  ][  StdAnalysis  ][32744]: [1][0x15cc8][0] ToT Mean = 7 +- 0
[11:01:12:273][  info  ][  StdAnalysis  ][32741]: [0][0x15478][0] ToT Mean = 7 +- 0
[11:01:12:474][  info  ][AnalysisAlgorithm][32744]: Analysis done!
[11:01:12:479][  info  ][AnalysisAlgorithm][32741]: Analysis done!
[11:01:12:480][  info  ][  ScanConsole  ][32708]: All done!
[11:01:12:480][  info  ][  ScanConsole  ][32708]: ##############
[11:01:12:480][  info  ][  ScanConsole  ][32708]: ##  Timing  ##
[11:01:12:480][  info  ][  ScanConsole  ][32708]: ##############
[11:01:12:480][  info  ][  ScanConsole  ][32708]: -> Configuration: 3552 ms
[11:01:12:480][  info  ][  ScanConsole  ][32708]: -> Scan:          136686 ms
[11:01:12:480][  info  ][  ScanConsole  ][32708]: -> Processing:    0 ms
[11:01:12:480][  info  ][  ScanConsole  ][32708]: -> Analysis:      647 ms
```

Figure 6.15.: Output of a successful YARR digital scan of two readout chips of the ITkPix QM using the Bdaq board. The scan took 140.885 s.



```
[10:48:32:112][  info  ][  ScanConsole  ][29522]: Scan done!
[10:48:32:112][  info  ][  ScanConsole  ][29522]: Waiting for processors to finish ...
[10:48:32:112][  info  ][Rd53bDataProcessor][29558]: Finished raw data processor thread for 0x154f8.
[10:48:32:115][  info  ][Rd53bDataProcessor][29552]: Finished raw data processor thread for 0x15478.
[10:48:32:115][  info  ][Rd53bDataProcessor][29555]: Finished raw data processor thread for 0x15cc8.
[10:48:32:115][  info  ][  ScanConsole  ][29522]: Processor done, waiting for histogrammer ...
[10:48:32:515][  info  ][HistogramAlgorithm][29557]: Histogrammer done!
[10:48:32:519][  info  ][HistogramAlgorithm][29554]: Histogrammer done!
[10:48:32:520][  info  ][HistogramAlgorithm][29551]: Histogrammer done!
[10:48:32:520][  info  ][  ScanConsole  ][29522]: Processor done, waiting for analysis ...
[10:48:32:523][  info  ][  StdAnalysis  ][29556]: [2][0x154f8] Total number of failing pixels: 0
[10:48:32:529][  info  ][  StdAnalysis  ][29550]: [0][0x15478] Total number of failing pixels: 0
[10:48:32:533][  info  ][  StdAnalysis  ][29553]: [1][0x15cc8] Total number of failing pixels: 0
[10:48:32:546][  info  ][  StdAnalysis  ][29556]: [2][0x154f8][0] ToT Mean = 7 +- 0
[10:48:32:558][  info  ][  StdAnalysis  ][29550]: [0][0x15478][0] ToT Mean = 7 +- 0
[10:48:32:570][  info  ][  StdAnalysis  ][29553]: [1][0x15cc8][0] ToT Mean = 7 +- 0
[10:48:32:751][  info  ][AnalysisAlgorithm][29556]: Analysis done!
[10:48:32:764][  info  ][AnalysisAlgorithm][29550]: Analysis done!
[10:48:32:778][  info  ][AnalysisAlgorithm][29553]: Analysis done!
[10:48:32:778][  info  ][  ScanConsole  ][29522]: All done!
[10:48:32:778][  info  ][  ScanConsole  ][29522]: ##############
[10:48:32:778][  info  ][  ScanConsole  ][29522]: ##  Timing  ##
[10:48:32:778][  info  ][  ScanConsole  ][29522]: ##############
[10:48:32:778][  info  ][  ScanConsole  ][29522]: -> Configuration: 5698 ms
[10:48:32:778][  info  ][  ScanConsole  ][29522]: -> Scan:          144545 ms
[10:48:32:778][  info  ][  ScanConsole  ][29522]: -> Processing:    3 ms
[10:48:32:778][  info  ][  ScanConsole  ][29522]: -> Analysis:      663 ms
```

Figure 6.16.: Output of a successful YARR digital scan of three readout chips of the ITkPix QM using the Bdaq board. The scan took 150.909 s.



```
[10:54:14:573][  info  ][  ScanConsole  ][30511]: Scan done!
[10:54:14:573][  info  ][  ScanConsole  ][30511]: Waiting for processors to finish ...
[10:54:14:573][  info  ][Rd53bDataProcessor][30551]: Finished raw data processor thread for 0x15cc8.
[10:54:14:574][  info  ][Rd53bDataProcessor][30557]: Finished raw data processor thread for 0x154b8.
[10:54:14:574][  info  ][Rd53bDataProcessor][30554]: Finished raw data processor thread for 0x154f8.
[10:54:14:575][  info  ][Rd53bDataProcessor][30548]: Finished raw data processor thread for 0x15478.
[10:54:14:575][  info  ][  ScanConsole  ][30511]: Processor done, waiting for histogrammer ...
[10:54:14:977][  info  ][HistogramAlgorithm][30556]: Histogrammer done!
[10:54:14:978][  info  ][HistogramAlgorithm][30550]: Histogrammer done!
[10:54:14:978][  info  ][HistogramAlgorithm][30547]: Histogrammer done!
[10:54:14:979][  info  ][HistogramAlgorithm][30553]: Histogrammer done!
[10:54:14:979][  info  ][  ScanConsole  ][30511]: Processor done, waiting for analysis ...
[10:54:14:988][  info  ][  StdAnalysis  ][30552]: [2][0x154f8] Total number of failing pixels: 0
[10:54:14:989][  info  ][  StdAnalysis  ][30555]: [3][0x154b8] Total number of failing pixels: 0
[10:54:14:989][  info  ][  StdAnalysis  ][30546]: [0][0x15478] Total number of failing pixels: 0
[10:54:14:989][  info  ][  StdAnalysis  ][30549]: [1][0x15cc8] Total number of failing pixels: 0
[10:54:15:018][  info  ][  StdAnalysis  ][30552]: [2][0x154f8][0] ToT Mean = 7 +- 0
[10:54:15:021][  info  ][  StdAnalysis  ][30555]: [3][0x154b8][0] ToT Mean = 7 +- 0
[10:54:15:022][  info  ][  StdAnalysis  ][30546]: [0][0x15478][0] ToT Mean = 7 +- 0
[10:54:15:022][  info  ][  StdAnalysis  ][30549]: [1][0x15cc8][0] ToT Mean = 7 +- 0
[10:54:15:224][  info  ][AnalysisAlgorithm][30552]: Analysis done!
[10:54:15:228][  info  ][AnalysisAlgorithm][30555]: Analysis done!
[10:54:15:228][  info  ][AnalysisAlgorithm][30546]: Analysis done!
[10:54:15:229][  info  ][AnalysisAlgorithm][30549]: Analysis done!
[10:54:15:229][  info  ][  ScanConsole  ][30511]: All done!
[10:54:15:229][  info  ][  ScanConsole  ][30511]: ##############
[10:54:15:229][  info  ][  ScanConsole  ][30511]: ##  Timing  ##
[10:54:15:229][  info  ][  ScanConsole  ][30511]: ##############
[10:54:15:229][  info  ][  ScanConsole  ][30511]: -> Configuration: 7752 ms
[10:54:15:229][  info  ][  ScanConsole  ][30511]: -> Scan:          147572 ms
[10:54:15:229][  info  ][  ScanConsole  ][30511]: -> Processing:    2 ms
[10:54:15:229][  info  ][  ScanConsole  ][30511]: -> Analysis:      653 ms
```

Figure 6.17.: Output of a successful YARR digital scan of all readout chips of the ITkPix QM using the Bdaq board. The scan took 155.979 s.

CHAPTER 7

Netio-next YARR Readout Controller

During the production of the ITk loaded local support, the ITk software, YARR, will interact with hardware called FELIX to read out the ITk quad modules. This hardware does not connect directly to the frontend chips; instead, it is connected to a hardware component named Optoboard via an optical connection. The Optoboard, in turn, is connected to the frontend chips via an electrical connection.

In the current readout system, YARR reads out the frontend via FELIX using the FELIX application, felixcore, which is the central application of FELIX operation, and the corresponding protocol, Netio, which is a communication protocol for messaging between FELIX and YARR. The aim of the work presented in this chapter is to upgrade YARR to read out RD53A readout chips with the felix-star application, the successor of felixcore, and its corresponding protocol, netio-next.

In the following chapter, FELIX is introduced in Section 7.1, followed by the introduction of the Optoboard in Section 7.2. Next, the readout chain is introduced in Section 7.3. Finally, the development work for reading out the RD53A single chip card with YARR and NetioNext is discussed in Section 7.4.

7.1. FELIX

FELIX (Front-End Link eXchange) [85] is a system that communicates with the DAQ software on one side and the detector frontend components on the other side. It is used in some detectors in ATLAS and will be used in the ITk detector. It is situated on the off-detector side in a radiation-free area where it will be connected to frontend components via optical cables in ITk. FELIX acts as a router, arranging and preparing the incoming data from the frontend side and sending it to the ITk software (YARR). It also organizes the detector control commands and triggers and sends them to the frontend side. Additionally, it is capable of communicating with other FELIX systems.

The FELIX system consists of hardware and software. The hardware can be custom-designed, such as the FLX-712 [85], or a commercial board like the VC709[1]. The FLX-712 FELIX hardware board, shown in Figure 7.1, hosts a Xilinx Kintex Ultrascale XCKU115 FPGA. It has 2 optical devices that can serve 48 bidirectional optical links, supported by 8 Mini transceivers (MiniPODs) with a data transmission speed of 9.6 Gbit/s for input links and 4.8 Gbit/s for output links. It is connected to a PC via a 16-lane PCIe Gen3 interface with a throughput of up to 128 Gbit/s. It also has 2 GB of memory for storing firmware and a port for the ATLAS TTC (Timing, Trigger, and Control) signal.

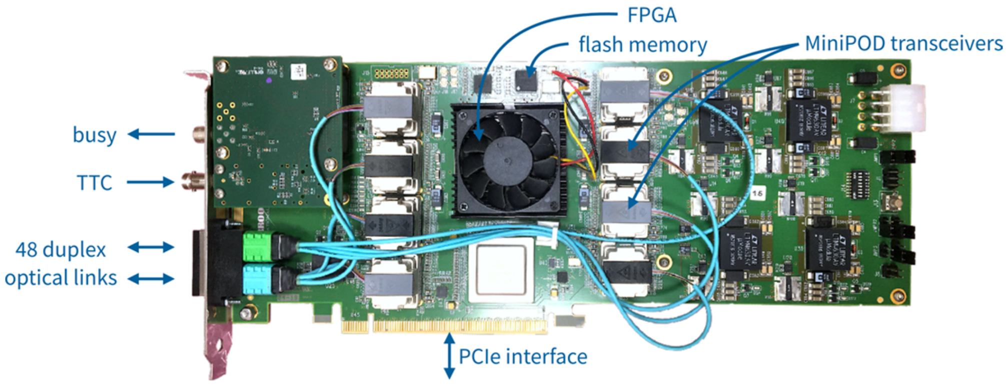


Figure 7.1.: Picture of a FLX-712 hardware board.

In addition to the hardware board, FELIX has software to run the hardware. This software consists of many applications, such as **flx-init** to initialize the FELIX card after powering up, and **flx-info** to output information about FELIX and more. The central process for running FELIX to communicate with YARR is **felixcore** [86], which communicates with YARR with the help of the Netio protocol. However, felixcore is a multi-threaded application.

In ITk, FELIX will be operated with **felix-star** [87], the successor of felixcore. felix-star is a single-threaded application and can communicate with multiple FELIX cards. felix-star has three main sub-applications:

- felix-tohost: Responsible for incoming readout data from the FLX card to the DAQ software.
- felix-toflx: Responsible for Timing, Trigger, and Control commands (TTC) from the DAQ software to the FLX card.
- felix-register: Provides remote access to FLX registers, allowing for writing and reading FELIX registers.

[1] The VC709 is a development board that can host the FELIX firmware and has an SFP connection for providing optical connections.

Moreover, felix-star communicates with YARR using netio-next [87], a fast communication protocol for data messaging between FELIX Star and FELIX Clients. netio-next is the successor of Netio.

7.2. Optoboard

The FELIX card is not a radiation-hard system, and it is placed on the off-detector side, about 65 m away from the on-detector side. Due to this large distance, the connection between the FELIX and the frontend is optical, necessitating hardware that performs the electrical/optical conversion. This hardware is called the Optoboard. The Optoboard is connected to FELIX via an optical connection and to the frontend via an electrical connection.

The Optoboard [88] is a PCB (printed circuit board) shown in Figure 7.2. It hosts three main types of ASICs [89]:

- 4 GBCR (Giga-Bit Cable Receiver) [90] units are used for the equalization and recovery of signals. For the readout data coming from the ITk frontend, this is done at 1.28 Gb/s, and for the commands sent to the ITk frontend, it is done at 160 Mb/s.
- 4 lpGBT (Low Power Giga Bit Transceiver) [91] units are used for the multiplexing of uplink lines (readout data lines) and demultiplexing of downlink lines (command and trigger lines). For uplinks, each optoboard receives six uplinks from GBCRs, which are serialized into one uplink at 10.24 Gb/s using Forward Error Correction FEC12[2], capable of correcting 12 consecutive bits to ensure correct serialization. For downlinks, the signal is demultiplexed into two links. In total, there are eight downlinks in the optoboard, each transmitted at 2.56 Gb/s. One of the four lpGBT units is a master lpGBT that hosts all eight downlinks and is also used to configure all the lpGBTs. The other three lpGBTs are slaves that only host uplinks. The lpGBT master is configured using information transmitted through the IC field in the downlink data frame. The other components, including the lpGBT slaves, GBCRs, and VTRx+, are configured via I2C control of the lpGBT master.
- A VTRx+ (Versatile Link+ Transceiver) [92] is located on the backside of the Optoboard and is used for the electrical-to-optical conversion and vice versa.

Figure 7.3 shows the speed of data transmission for data links and command links. The command is sent to the VTRx+ via optical at 2.56 Gb/s and then converted to an electrical signal at 160 Mb/s. The incoming data from the frontend arrives at 1.28 Gb/s, is then converted to optical signal, and sent via optical fiber to FELIX at 10.24 Gb/s.

[2] Forward error correction (FEC) is a technique used for error control in data transmission. In this method, the transmitter sends redundant data, allowing the receiver to identify and use only the error-free portions of the data.

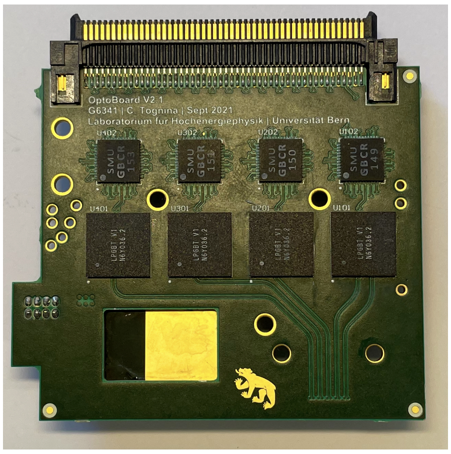

OptoBoard V2.1
G6341 | C. Tognina | Sept 2021
Laboratorium fur Hochenergiephysik | Universitat Bern

Figure 7.2.: Optoboard board [93].

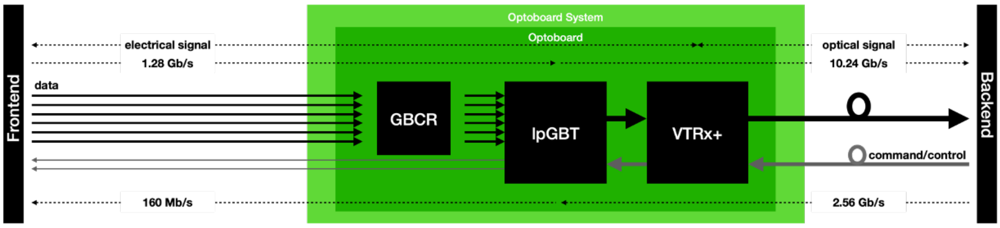


Figure 7.3.: data transmission through optoboard board [93].

7.3. Readout chain

The current readout chain used to validate the YARR software is shown in Figure 7.4. In this setup, the RD53A quad module is connected to the optobox, a unit that hosts up to 8 optoboards. The command signals coming from FELIX are converted to electrical signals by VTRx+ and then demultiplexed by LpGBT. Following this, the signals are enhanced by the GBCR, sent to the frontend, and the readout data coming from the frontend is also enhanced by the GBCR. The data is then aggregated by LpGBT, converted back to optical signals by VTRx+, and sent to FELIX. The optobox is connected to the FELIX card hosted in the DAQ PC via optical fibers. The DAQ PC has the FELIX software installed to operate and control FELIX, as well as the YARR software. For this work, FELIX will be operated with the felix-star application instead of felixcore and will use the NetioNext protocol instead of the Netio protocol.

Figure 7.5 shows the readout chain used in this work, where the RD53A SCC is utilized. The YARR software is introduced in Section 6.3. In the YARR software, there is the chip library libRd53a that creates commands and triggers to be sent to the RD53A SCC, and it also analyzes and processes the incoming data from the RD53A SCC. Additionally, there is a controller library named libNetioNext that provides communication with FELIX through NetioNext. libNetioNext includes TxCore, which is responsible

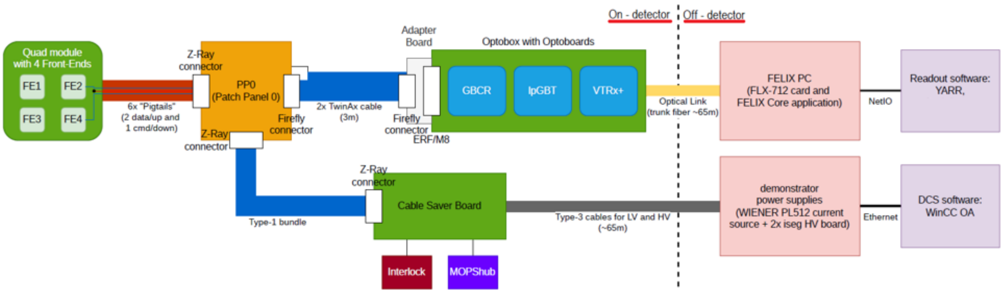


Figure 7.4.: The current readout chain [94], where the upper path represents the DAQ part and the lower path presents the DCS (Data Control System) path, monitors the frontend and intervenes in case of unusual behaviors.

for preparing the commands and triggers created by libRd53a to send them to FELIX. It also includes RxCore, which is responsible for receiving the readout data from the RD53A SCC, preparing it, and sending it to libRd53a to be analyzed and processed later.

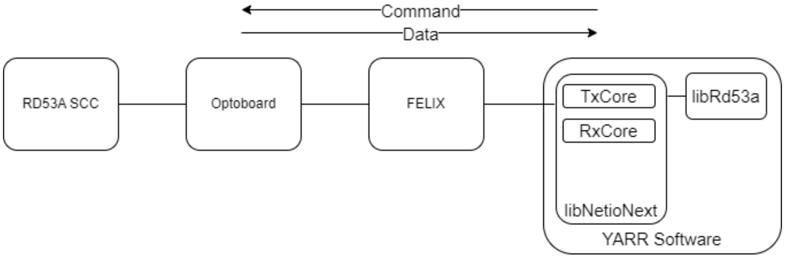


Figure 7.5.: Schematics of the readout chain used in this work, showing the direction of transmission for commands and readout data.

7.4. YARR Developments to read out RD53A SCC

The aim of this work is to validate the YARR/FELIX/Optoboard/RD53A readout chain using felix-star and netio-next. Prior to this work, YARR was developed and validated to read out RD53A SCC using felix-star and netio-next through the use of the FELIX VC709 board and Versatile Link Demonstrator Board(VLDB) [95] which is an evaluation kit for the radiation-hard Optical Link ecosystem. It provides a 4.8 Gbps data transfer link for communication between the front-end chip and DAQ system, it has a GBT ASIC[3]. However, after many developments of the netio-next protocol and FELIX software, YARR is no longer able to read out RD53A SCC with the recent versions of

[3] The GBT ASIC [95] is a radiation-hard chip that implements a high-speed link. It integrates Data Acquisition (DAQ), Timing, Trigger and Control (TTC), and Slow Control (SC) information into a pair of optical links.

netio-next and FELIX software.

After receiving the optoboard, there is a possibility to upgrade YARR to read out RD53A SCC using the optoboard instead of the VLDB board, and the FLX-712 instead of the VC709 board, with the recent versions of felix-star and netio-next.

The lab setup used is shown in Figure 7.6, where the RD53A SCC is connected to the optoboard through a DP-SMA adapter board and an ERF-SMA adapter board. The optoboard is connected to the FELIX PC via optical fibers. In addition, two power supplies are used: one to operate the RD53A SCC and one to operate the optoboard.

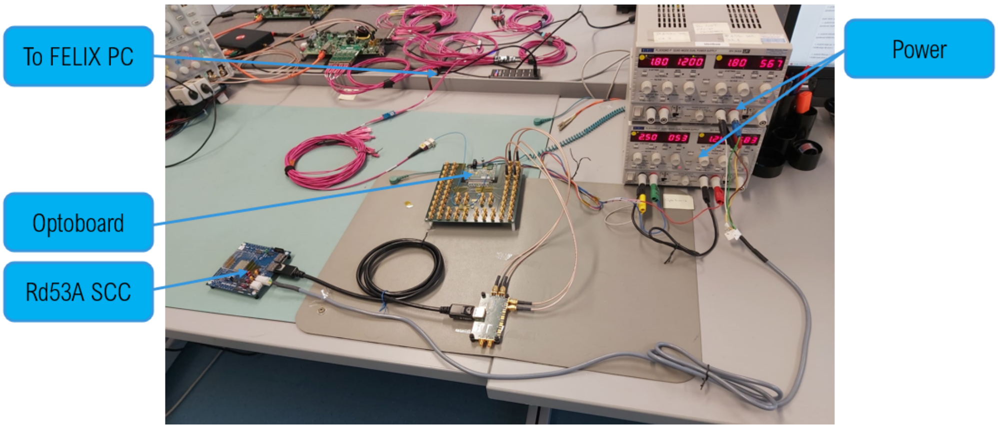
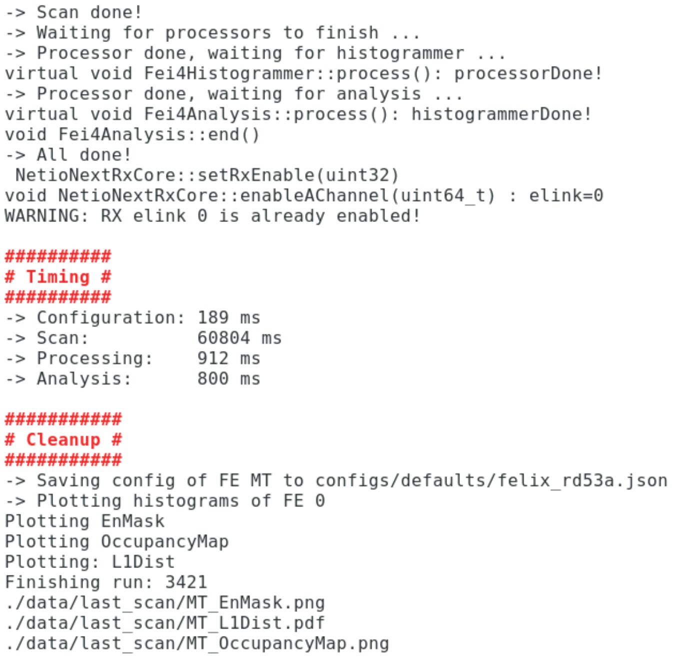


Figure 7.6.: Lab setup where the FELIX server is hidden (under the table).

First, the installation of the YARR netio-next controller was broken and was fixed by re-initializing some parameters to fit the new version of netio-next. Then, during a digital scan, the connection was refused due to incorrect loading configurations. This was solved by updating the version of a software module that YARR uses. After that, some data package truncation errors appeared in one of the YARR modules, which were solved by increasing the buffer size used in YARR. Then, the headers of the data packages were extracted in the RxCore of libNetioNext, and the addresses of the data packages were identified. The data packages were then sent to the libRd53a library using the obtained addresses.

After these developments, YARR successfully read out RD53A SCC using felix-star and netio-next with the optoboard and FLX-712 board. Figure 7.7 shows that the YARR digital scan performs properly. Figure 7.8 and Figure 7.9 show the mask map and the occupancy map of the scan, respectively, concluding that each enabled pixel provides a value of 100 hits in the occupancy map. The result indicates that the YARR netio-next controller works properly[4].

[4] For clarification, part of the RD53A chip is disabled to demonstrate that the developed YARR software is functioning correctly, as it should produce no output when the pixel is disabled.



```
-> Scan done!
-> Waiting for processors to finish ...
-> Processor done, waiting for histogrammer ...
virtual void Fei4Histogrammer::process(): processorDone!
-> Processor done, waiting for analysis ...
virtual void Fei4Analysis::process(): histogrammerDone!
void Fei4Analysis::end()
-> All done!
 NetioNextRxCore::setRxEnable(uint32)
void NetioNextRxCore::enableAChannel(uint64_t) : elink=0
WARNING: RX elink 0 is already enabled!

##########
# Timing #
##########
-> Configuration: 189 ms
-> Scan:          60804 ms
-> Processing:    912 ms
-> Analysis:      800 ms

###########
# Cleanup #
###########
-> Saving config of FE MT to configs/defaults/felix_rd53a.json
-> Plotting histograms of FE 0
Plotting EnMask
Plotting OccupancyMap
Plotting: L1Dist
Finishing run: 3421
./data/last_scan/MT_EnMask.png
./data/last_scan/MT_L1Dist.pdf
./data/last_scan/MT_OccupancyMap.png
```

Figure 7.7.: Output of a successful YARR digital scan

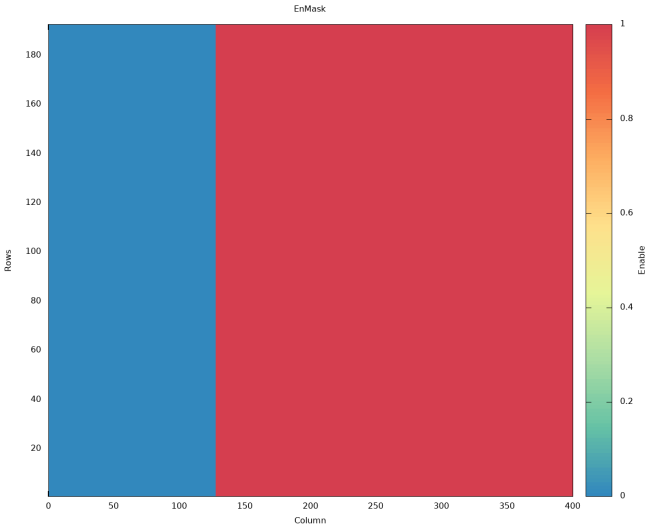


Figure 7.8.: The mask map of the YARR digital scan of RD53A SCC.

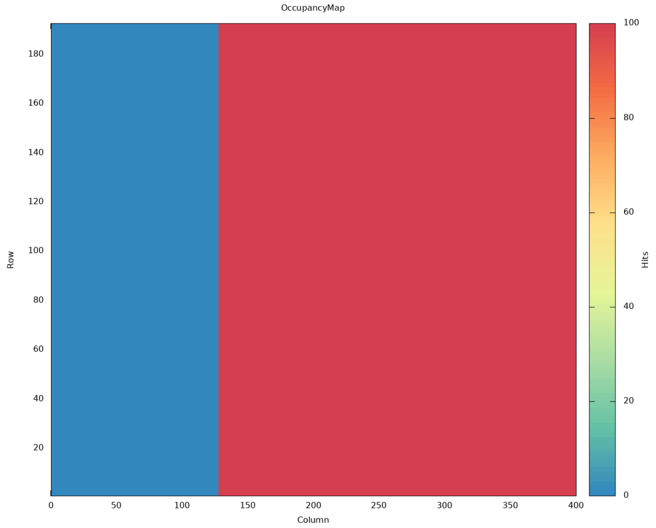


Figure 7.9.: The occupancy map of the YARR digital scan of RD53A SCC.

CHAPTER 8

FelixClient YARR Readout Controller

As discussed in previous chapters, the modules will be mounted onto local supports during the production phase. Afterward, all modules must undergo quality control (QC) tests to ensure they are in good condition and perform properly once mounted. Some of these tests include electrical tests, such as digital tests to verify the proper functionality of the digital components of the pixel, and analog scans to check the analog components.

Loaded local supports hold different number of modules; for example, each longeron will hold 36 QMs, requiring the readout of approximately 144 frontend chips. This situation necessitates an ITk DAQ system capable of reading out this large number of QMs. Starting with an ITk DAQ chain validated for reading out one QM, the natural next step would be to validate the ITk DAQ for multiple QMs simultaneously.

In this chapter, the ITk DAQ system is explained in Section 8.1, followed by the validation of the ITk software YARR using the ITk Readout system, which is discussed in Section 8.2.

8.1. Readout chain

The ITk DAQ readout chain is similar to the one described in Section 7.3, where the ITkPix QM is connected to the optoboard via electrical connections, and the optoboard is connected to the DAQ PC via optical connections as shown in Figure 8.1. Inside the DAQ PC, there is a FELIX card and the ITk DAQ software YARR.

Commands for the ITKpix QM are created in YARR, sent to FELIX, and then transmitted to the optoboard. In the optoboard, these commands are converted from optical to electrical signals by VTRx+, distributed via LpGBT, enhanced by GBCR, and finally sent to the ITKpix QMs. In the opposite direction, readout data from the ITKpix QM is enhanced by GBCR, aggregated by the LpGBT, and transmitted to the FELIX card in the DAQ PC via optical fibers. The readout data is then sent to YARR for analysis and processing.

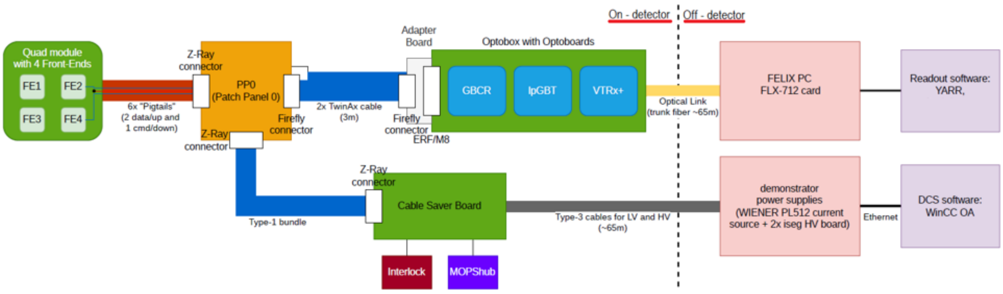


Figure 8.1.: The ITk readout chain [94], where the upper path represents the DAQ part and the lower path presents the DCS (Data Control System) path, monitors the frontend and intervenes in case of unusual behaviors.

In the ITk DAQ readout system, FELIX is operated via the central application Felixstar, which was introduced in Section 7.1, using the netio-next data messaging protocol. The FELIX software provides an application named FelixClient, which abstracts the complexity of netio-next. As shown in Figure 8.2, the YARR software includes a chip library called libRd53b that generates commands and analyzes data. It also has a corresponding controller named libFelixClient, which interacts with the FelixClient application of FELIX. The libFelixClient contains TxCore, responsible for sending commands and triggers, and RxCore, responsible for preparing readout data from the frontend to be sent to the libRd53b library.

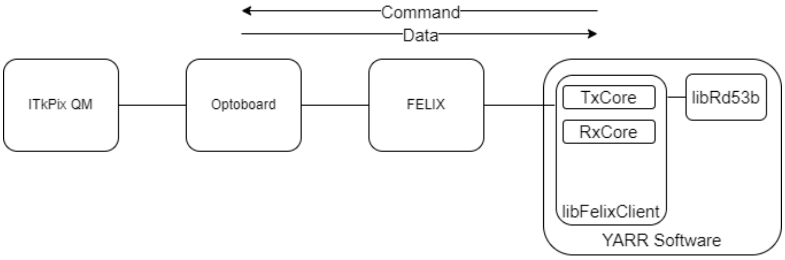


Figure 8.2.: Schematics of the ITk DAQ readout chain, showing the direction of transmission for commands and readout data.

8.2. YARR performance with FelixClient

The plan is to validate the ITk software YARR for a single frontend chip and then scale it to validate it for multiple frontend chips running simultaneously. The validation is done by running a successful YARR digital scan.

The used lab setup is shown in Figure 8.3. It consists of three ITkPix QMs, each containing four ITkPix frontend chips connected to an optoboard through a DP-ERF adapter board developed by INFN group (Italy) [96]. The optoboard is connected to a FELIX server, which hosts a FELIX card and YARR software, via optical fibers. Additionally, there are power supplies to power the ITkPix QMs and the optoboard. The FELIX is operated via Felix-star, and YARR communicates with the FELIX through FelixClient.

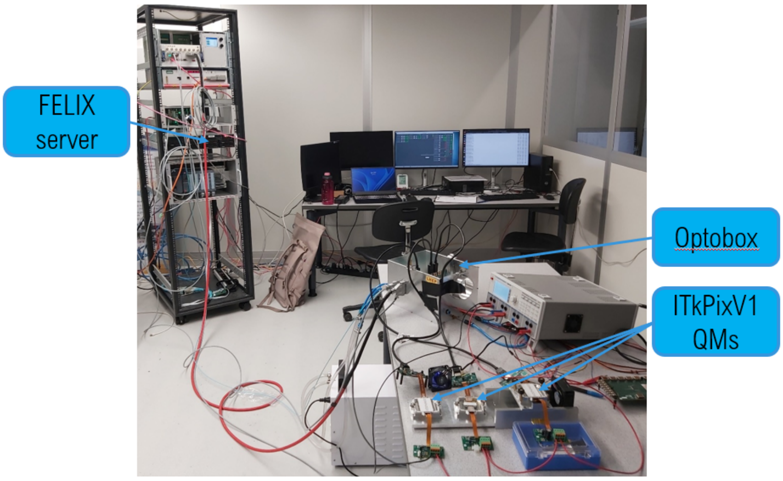


Figure 8.3.: Lab Setup for ITk Software YARR Validation.

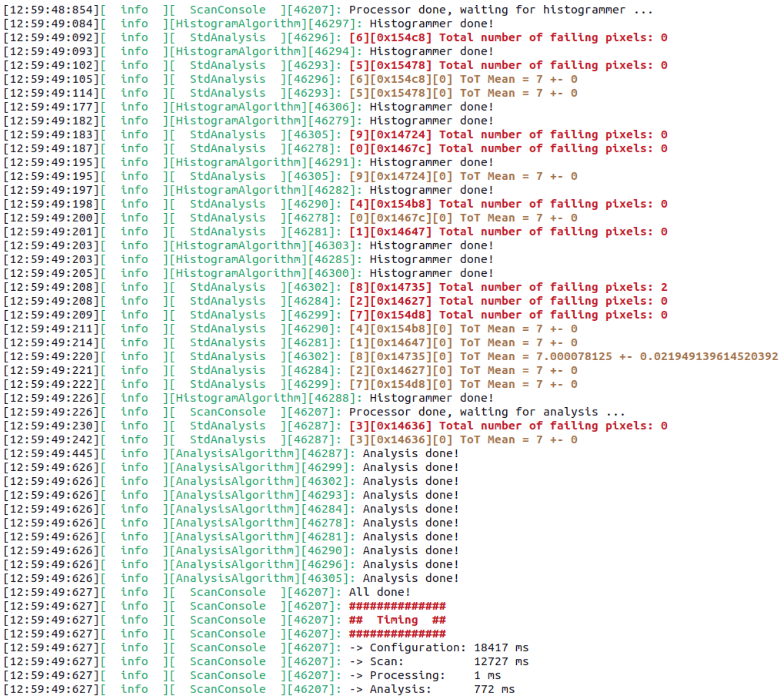
After successfully performing a digital scan for one frontend chip, the validation process was extended to include two frontend chips. This was achieved by conducting another proper digital scan. This procedure continued step by step, adding extra frontend chips until the YARR software was validated for 10 ITkPix frontend chips without any data transmission errors, as shown in Figure 8.4.

The output of the YARR digital scans during the scaling up process are shown in Section A.3.

During the scaling-up procedure, the time consumed by the YARR processes during the scan was recorded. The YARR processes include Configuration, Scan, Processing, and Analysis. In the configuration process, YARR sends commands to configure the chips. Then, YARR sends the scan command during the Scan process. After the scan, YARR checks the incoming data in the Processing stage and analyzes them in the Analysis process.

Figure 8.5 shows the time consumed by each YARR process and the total time consumed by all the processes. As the number of scanned chips increases, the time consumed by YARR configuration increases, as YARR configures the read-out chips one by one, which is a sequential process. The scan time consumed by YARR increases slightly as the number of chips increases, which positively indicates the scalability of the YARR software since the related scan commands are sent to all frontend chips simultaneously, not chip by chip.

This achieved work and the results are being written in a proceeding, which is currently



```
[12:59:48:854][  info  ][  ScanConsole  ][46207]: Processor done, waiting for histogrammer ...
[12:59:49:084][  info  ][HistogramAlgorithm][46297]: Histogrammer done!
[12:59:49:092][  info  ][  StdAnalysis  ][46296]: [6][0x154c8] Total number of failing pixels: 0
[12:59:49:093][  info  ][HistogramAlgorithm][46294]: Histogrammer done!
[12:59:49:102][  info  ][  StdAnalysis  ][46293]: [5][0x15478] Total number of failing pixels: 0
[12:59:49:105][  info  ][  StdAnalysis  ][46296]: [6][0x154c8][0] ToT Mean = 7 +- 0
[12:59:49:114][  info  ][  StdAnalysis  ][46293]: [5][0x15478][0] ToT Mean = 7 +- 0
[12:59:49:177][  info  ][HistogramAlgorithm][46306]: Histogrammer done!
[12:59:49:182][  info  ][HistogramAlgorithm][46279]: Histogrammer done!
[12:59:49:183][  info  ][  StdAnalysis  ][46305]: [9][0x14724] Total number of failing pixels: 0
[12:59:49:187][  info  ][  StdAnalysis  ][46278]: [0][0x1467c] Total number of failing pixels: 0
[12:59:49:195][  info  ][HistogramAlgorithm][46291]: Histogrammer done!
[12:59:49:195][  info  ][  StdAnalysis  ][46305]: [9][0x14724][0] ToT Mean = 7 +- 0
[12:59:49:197][  info  ][HistogramAlgorithm][46282]: Histogrammer done!
[12:59:49:198][  info  ][  StdAnalysis  ][46290]: [4][0x154b8] Total number of failing pixels: 0
[12:59:49:200][  info  ][  StdAnalysis  ][46278]: [0][0x1467c][0] ToT Mean = 7 +- 0
[12:59:49:201][  info  ][  StdAnalysis  ][46281]: [1][0x14647] Total number of failing pixels: 0
[12:59:49:203][  info  ][HistogramAlgorithm][46303]: Histogrammer done!
[12:59:49:203][  info  ][HistogramAlgorithm][46285]: Histogrammer done!
[12:59:49:205][  info  ][HistogramAlgorithm][46300]: Histogrammer done!
[12:59:49:208][  info  ][  StdAnalysis  ][46302]: [8][0x14735] Total number of failing pixels: 2
[12:59:49:208][  info  ][  StdAnalysis  ][46284]: [2][0x14627] Total number of failing pixels: 0
[12:59:49:209][  info  ][  StdAnalysis  ][46299]: [7][0x154d8] Total number of failing pixels: 0
[12:59:49:211][  info  ][  StdAnalysis  ][46290]: [4][0x154b8][0] ToT Mean = 7 +- 0
[12:59:49:214][  info  ][  StdAnalysis  ][46281]: [1][0x14647][0] ToT Mean = 7 +- 0
[12:59:49:220][  info  ][  StdAnalysis  ][46302]: [8][0x14735][0] ToT Mean = 7.000078125 +- 0.021949139614520392
[12:59:49:221][  info  ][  StdAnalysis  ][46284]: [2][0x14627][0] ToT Mean = 7 +- 0
[12:59:49:222][  info  ][  StdAnalysis  ][46299]: [7][0x154d8][0] ToT Mean = 7 +- 0
[12:59:49:226][  info  ][HistogramAlgorithm][46288]: Histogrammer done!
[12:59:49:226][  info  ][  ScanConsole  ][46207]: Processor done, waiting for analysis ...
[12:59:49:230][  info  ][  StdAnalysis  ][46287]: [3][0x14636] Total number of failing pixels: 0
[12:59:49:242][  info  ][  StdAnalysis  ][46287]: [3][0x14636][0] ToT Mean = 7 +- 0
[12:59:49:445][  info  ][AnalysisAlgorithm][46287]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46299]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46302]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46293]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46284]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46278]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46281]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46290]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46296]: Analysis done!
[12:59:49:626][  info  ][AnalysisAlgorithm][46305]: Analysis done!
[12:59:49:627][  info  ][  ScanConsole  ][46207]: All done!
[12:59:49:627][  info  ][  ScanConsole  ][46207]: ##############
[12:59:49:627][  info  ][  ScanConsole  ][46207]: ##  Timing  ##
[12:59:49:627][  info  ][  ScanConsole  ][46207]: ##############
[12:59:49:627][  info  ][  ScanConsole  ][46207]: -> Configuration: 18417 ms
[12:59:49:627][  info  ][  ScanConsole  ][46207]: -> Scan:          12727 ms
[12:59:49:627][  info  ][  ScanConsole  ][46207]: -> Processing:    1 ms
[12:59:49:627][  info  ][  ScanConsole  ][46207]: -> Analysis:      772 ms
```

Figure 8.4.: Output of YARR digital scan for 10 ITkPix frontend chips.

in the circulation phase at the time of writing this thesis. The next step will be to extend these tests to a full-featured LLS readout, which could not be done because of the lack of available QMs.

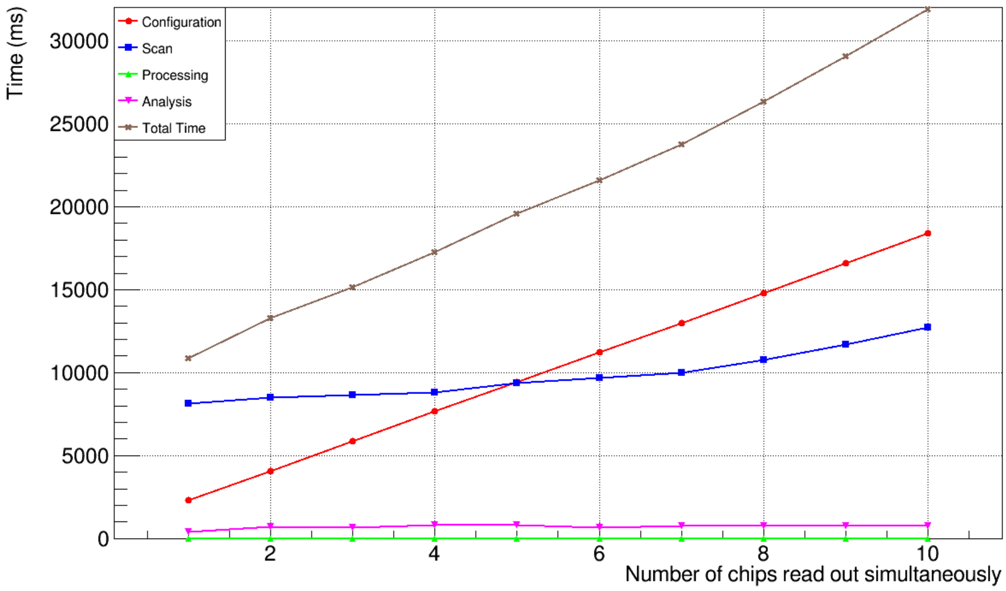


Figure 8.5.: Time consumed by the YARR processes as a function of the number of enabled frontend chips scanned at the same time. The configuration of the frontend chips is done sequentially, one chip at a time, while the scan process is done simultaneously, with scan commands sent to all frontend chips at once. Note that the digital YARR scan with 11 and 12 ITkPix frontend chips operating simultaneously failed due to data buffering issues related to the FELIX hardware.

CHAPTER 9

Conclusion

In this dissertation, significant developments related to the ITk were made in various fields. Specifically, there was development work in the ITk Production Database (ITk PD) related to loaded local support, as well as work on the ITk readout DAQ system.

Initially, extensive work was done on the ITk PD, focusing on the component types of the Outer Barrel (OB) functional local support, which includes the OB functional longeron, the OB functional inclined half-ring, and the OB Bare Module Cell. The production stages for these components were defined, along with the tests for each stage, including their properties and parameters. Additionally, the component types for their children, as well as the corresponding stages and tests, were also implemented.

As previously mentioned, when the modules are mounted onto the OB functional local support, they form the OB loaded local support, which consists of the OB loaded longeron and the OB loaded inclined half-ring. The component types of the OB loaded local supports, along with their stages and the tests for each stage, were implemented as well. To simplify interaction with the OB loaded local supports, a GUI was developed. This GUI provides users with all relevant information about the OB loaded local supports and allows them to control these supports without directly interacting with the ITk PD. Moreover, during the electrical testing of the OB loaded local supports, the results from a large number of frontends need to be uploaded to the ITk PD. To facilitate this, a code named "Uploader" was created to upload these electrical test results to the ITk PD.

Within the ITk PD, the OB Loaded Module Cell has two types: standard and combined. Since some tests are cloned from one type to another, a script was developed to highlight the differences between tests after cloning to prevent the loss of information. Additionally, a script was written to demonstrate the status of components, including their type, serial number, and current location.

After the development and preparation of the ITk PD for the LLS production phase, developments were made in the readout of the frontend chips. The ITk project uses the RD53A frontend chip as a prototype readout chip and the RD53B (ITkPixV1) frontend chip as the pre-production readout chip, the ITkPix-V2 will be used in the final prduction phase. These frontend chips can be read out using various readout systems, such as the BDAQ53 and YARR readout systems. The BDAQ53 readout system which is composed of BDAQ53 software and BDAQ53 FPGA board hardware, the YARR readout system consists of YARR software and YARR FPGA firmware that can be hosted on a commercial FPGA development board. The YARR software was successfully developed to support the BDAQ53 FPGA board for reading out RD53A and RD53B (ITkPix) frontend chips. As a result, YARR software is now capable of properly reading out RD53A SCC, RD53B (ITkPix) SCC, and RD53B (ITkPix) QM. This development allows institutes that have a BDAQ53 board to use it with YARR software during ITk module production for electrical testing, eliminating the need to purchase an additional FPGA board.

During the LLS production phase, electrical testing will be conducted using YARR software, along with the FELIX board and optoboard. The FELIX will be operated with the Felix-star application as the central application, replacing the Felixcore application. The Felix-star application communicates with YARR through the NetIONext data messaging protocol instead of the NetIO protocol. A YARR software branch was developed to read out RD53A frontend chips with Felix-star using GBT ASIC. This YARR software was successfully developed to read out RD53A SCC with Felix-star and the latest version of NetIONext through the use of an optoboard equipped with lpGBT.

The FELIX software provides users with an application named FelixClient, which simplifies the complexity of NetIONext. FelixClient will be used alongside the Felix-star application during the electrical testing of LLS in the production phase. The YARR software supports FELIX by integrating the FelixClient application. However, during the electrical testing of LLS, a large number of ITkPix frontend chips will be tested simultaneously, which requires software capable of handling a significant number of frontend chip readouts. YARR software was tested with the FELIX board and optoboard to read out multiple frontend chips. Successful results were achieved by properly reading out 10 ITkPix frontend chips using Felix-star and FelixClient.

9.1. Outlook

In the last development work, 10 frontend chips were read out simultaneously due to the limited availability of QMs. Only 3 QMs were available in the lab setup, which meant that there were 12 frontend chips in total. The work needs to continue once more QMs are available to explore the performance of the YARR software with a larger number of QMs. This will also help identify and resolve any issues that arise when multiple QMs are used together.

Another issue that needs to be addressed is the data transmission errors during the YARR digital scan when all the links (6 uplinks) passing through an optoboard are used. This problem requires a solution, which could involve reconfiguring the optoboard or making adjustments at the FELIX hardware and software levels.

Bibliography

[1] J. Burzynski, pp. , 3–32. Springer International Publishing, Cham, 2023. `https://doi.org/10.1007/978-3-031-30466-8_1`.

[2] Particle Data Group Collaboration, S. Navas, et al., *Review of Particle Physics*, Phys. Rev. D **110** (2024) 030001.

[3] *The standard model of particle physics*, Nature **448** (2007) 270–270.

[4] M. Thomson, *Modern Particle Physics*. Cambridge University Press, 2013.

[5] ATLAS Collaboration, G. Aad et al., *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** (2012) 1–29.

[6] CMS Collaboration, S. Chatrchyan et al., *Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC*, Phys. Lett. B **716** (2012) 30–61.

[7] G. Altarelli and J. Wells, pp. , 27–96. Springer International Publishing, Cham, 2017. `https://doi.org/10.1007/978-3-319-51920-3_2`.

[8] P. Paganini, *Fundamentals of Particle Physics: Understanding the Standard Model*. 08, 2023.

[9] ATLAS Collaboration, *Standard Model Summary Plots February 2022*, tech. rep., ATL-PHYS-PUB-2022-009, CERN, Geneva, 2022.

[10] R. Davis, D. S. Harmer, and K. C. Hoffman, *Search for Neutrinos from the Sun*, Phys. Rev. Lett. **20** (1968) 1205–1209.

[11] S. Bilenky, pp. , 175–207. Springer International Publishing, Cham, 2018. `https://doi.org/10.1007/978-3-319-74802-3_11`.

[12] C. W. Walter, p. , 19–43. WORLD SCIENTIFIC, Mar., 2008. `http://dx.doi.org/10.1142/9789812771971_0002`.

[13] SNO Collaboration, A. Bellerive, et al., *The Sudbury Neutrino Observatory*, Nucl. Phys. B **908** (2016) 30–51.

[14] A. Arbey and F. Mahmoudi, *Dark matter and the early Universe: a review*, Prog. Part. Nucl. Phys. **119** (2021) 103865, 72 pages, 35 figures.

[15] M. M. Waldrop, pp. , 101–134. Springer International Publishing, Cham, 2022. `https://doi.org/10.1007/978-3-030-98214-0_5`.

[16] A. Pich, *CP violation. CHARGE PARITY; 1993 ed.*, ICTP Ser. Theor. Phys. **10** (1993) 14–42.

[17] C. A. Baker, et al., *Improved Experimental Limit on the Electric Dipole Moment of the Neutron*, Phys. Rev. Lett. **97** (2006) 131801.

[18] R. Cashmore, L. Maiani, and J. Revol, *Prestigious Discoveries at CERN: 1973 Neutral Currents 1983 W & Z Bosons.* Springer Berlin Heidelberg, 2010. `https://books.google.de/books?id=o511cgAACAAJ`.

[19] NA48 Collaboration, V. Fanti et al., *A New measurement of direct CP violation in two pion decays of the neutral kaon*, Phys. Lett. B **465** (1999) 335–348.

[20] E. Lopienska, *The CERN accelerator complex, layout in 2022. Complexe des accélérateurs du CERN en janvier 2022*, `https://cds.cern.ch/record/2800984`, General Photo.

[21] NA61 Collaboration, N. Abgrall et al., *NA61/SHINE facility at the CERN SPS: beams and detector system*, JINST **9** (2014) P06005.

[22] *Radiofrequency cavities*, `https://cds.cern.ch/record/1997424`.

[23] M. Vretenar, et al., *Linac4 design report*, vol. 6 of *CERN Yellow Reports: Monographs.* CERN, Geneva, 2020. `https://cds.cern.ch/record/2736208`.

[24] K. H. Reich, *The CERN proton synchrotron booster*, `https://cds.cern.ch/record/349912`.

[25] N. Angert, et al., *CERN heavy-ion facility design report.* CERN Yellow Reports: Monographs. CERN, Geneva, 1993. `https://cds.cern.ch/record/249000`.

[26] M. Chanel, *LEIR: the low energy ion ring at CERN*, Nucl. Instrum. Meth. A **532** (2004) 137–143.

[27] J. B. ADAMS, *The Cern Proton Synchrotron*, Nature **185** (1960) 568–572.

[28] *The Super Proton Synchrotron*, `https://cds.cern.ch/record/1997188`.

[29] R. Bailey and P. Collier, *Standard Filling Schemes for Various LHC Operation Modes*, tech. rep., CERN, Geneva, 2003. `https://cds.cern.ch/record/691782`.

[30] *The accelerator complex*, `https://cds.cern.ch/record/1997193`.

[31] A. Ayan, *The CMS forward calorimeter prototype design studies and omega(c)0 search at E781 experiment at FermiLab*, Theses and Dissertations (2004) .

[32] A. Straessner, pp. , 45–54. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. `https://doi.org/10.1007/978-3-642-05169-2_2`.

[33] ATLAS Collaboration, G. Aad, et al., *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST **3** (2008) S08003, Also published by CERN Geneva in 2010.

[34] CMS Collaboration, S. Chatrchyan, et al., *The CMS experiment at the CERN LHC. The Compact Muon Solenoid experiment*, JINST **3** (2008) S08004, Also published by CERN Geneva in 2010.

[35] LHCb Collaboration, A. A. Alves, et al., *The LHCb Detector at the LHC*, JINST **3** (2008) S08005, Also published by CERN Geneva in 2010.

[36] ALICE Collaboration, K. Aamodt, et al., *The ALICE experiment at the CERN LHC. A Large Ion Collider Experiment*, JINST **3** (2008) S08002, Also published by CERN Geneva in 2010.

[37] J. T. Shank, *The ATLAS Detector: Status and Results from Cosmic Rays*, tech. rep., 2009. `arXiv:0910.3081`. `https://cds.cern.ch/record/1213385`. Comments: To be published in the proceedings of DPF-2009, Detroit, MI, July 2009, eConf C090726.

[38] J. Pequenao, *Event Cross Section in a computer generated image of the ATLAS detector.*, 2008.

[39] A. Abdesselam et al., *The detector control system of the ATLAS SemiConductor tracker during macro-assembly and integration*, JINST **3** (2008) P02007.

[40] M. Keil, *Operational Experience with the ATLAS Pixel Detector at the LHC*, `https://cds.cern.ch/record/1357042`.

[41] *Technical Design Report of the ATLAS Pixel Detector*, tech. rep., CERN, 1998.

[42] ATLAS Collaboration, M. Capeans, et al., *ATLAS Insertable B-Layer Technical Design Report*, tech. rep., 2010.

[43] ATLAS Collaboration, G. Aad et al., *Operation and performance of the ATLAS semiconductor tracker*, JINST **9** (2014) P08009.

[44] A. Vogel, *ATLAS Transition Radiation Tracker (TRT): Straw Tube Gaseous Detectors at High Rates*, tech. rep., CERN, Geneva, 2013. https://cds.cern.ch/record/1537991.

[45] B. Guo, *Measurement of the Top Quark Pair Production Cross Section and an in-situ B-tagging efficiency Calibration with ATLAS in pp collisions at sqrts = 7 TeV in Dilepton Final States*, 2011. https://cds.cern.ch/record/1390481. Presented 19 Jul 2011.

[46] ATLAS Collaboration, *ATLAS liquid-argon calorimeter: Technical Design Report.* Technical design report. ATLAS. CERN, Geneva, 1996. https://cds.cern.ch/record/331061.

[47] ATLAS Collaboration, *ATLAS tile calorimeter: Technical Design Report.* Technical design report. ATLAS. CERN, Geneva, 1996. https://cds.cern.ch/record/331062.

[48] S. Palestini, *The muon spectrometer of the ATLAS experiment*, Nucl. Phys. B Proc. Suppl. **125** (2003) 337–345.

[49] ATLAS Collaboration, *ATLAS magnet system: Technical Design Report, 1.* Technical design report. ATLAS. CERN, Geneva, 1997. https://cds.cern.ch/record/338080.

[50] ATLAS Collaboration, T. Colombo, *Data-flow Performance Optimisation on Unreliable Networks: the ATLAS Data-Acquisition Case*, J. Phys. Conf. Ser. **608** (2015) 012005.

[51] O. Aberle, et al., *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report.* CERN Yellow Reports: Monographs. CERN, Geneva, 2020. https://cds.cern.ch/record/2749422.

[52] O. Brüning and L. Rossi, *The High-Luminosity Large Hadron Collider*, Nature **1** (2019) .

[53] L. Gonella, *The ATLAS ITk detector system for the Phase-II LHC upgrade*, Nucl. Instrum. Meth. A **1045** (2023) 167597.

[54] ATLAS Collaboration, V. Izzo, *ATLAS upgrades*, PoS **LHCP2020** (2021) 094.

[55] ATLAS Collaboration, *Technical Design Report for the ATLAS Inner Tracker Strip Detector*, tech. rep., CERN, Geneva, 2017. https://cds.cern.ch/record/2257755.

[56] ATLAS Collaboration, *ATLAS ITK Pixel Detector Overview*, https://cds.cern.ch/record/2860701.

[57] ATLAS Collaboration, *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*, tech. rep., CERN, Geneva, 2017. `https://cds.cern.ch/record/2285585`.

[58] ATLAS Collaboration, F. J. Munoz Sanchez, *Module and electronics developments for the ATLAS ITk pixel system*, JINST **13** (2018) C03045.

[59] Particle Data Group Collaboration, R. L, et al., *Review of Particle Physics*, PTEP **2022** (2022) 083C01.

[60] H. Kolanoski and N. Wermes, *Particle Detectors: Fundamentals and Applications.* OUP Oxford, 2020. `https://books.google.de/books?id=QyrtDwAAQBAJ`.

[61] R. Meyers, *Encyclopedia of Physical Science and Technology.* No. v. 12 in Encyclopedia of Physical Science and Technology. Academic Press, 2002. `https://books.google.de/books?id=lo1UAAAAMAAJ`.

[62] A. Pankov and N. Paver, *Bhabha versus Møller scattering as a contact-interaction analyzer at a polarized Linear Collider*, Eur. Phys. J. C **29** (2003) 313–323.

[63] P. Hautojärvi and A. Vehanen, pp. , 1–23. Springer Berlin Heidelberg, Berlin, Heidelberg, 1979. `https://doi.org/10.1007/978-3-642-81316-0_1`.

[64] R. A. Siddique, *Totem-pole power factor correction rectifier with Gallium-Nitride devices for telecom power supply*, 2016. `https://api.semanticscholar.org/CorpusID:113619594`.

[65] S. Sze, *Semiconductor Devices: Physics and Technology.* John Wiley & Sons Singapore Pte. Limited, 2012. `https://books.google.de/books?id=gmmscQAACAAJ`.

[66] R. Paschotta, *Band Gap*, Rp photonics encyclopedia. `https://www.rp-photonics.com/band_gap.html`. Available online at `https://www.rp-photonics.com/band_gap.html`.

[67] J. Fang, et al., *Understanding the Average Electron–Hole Pair-Creation Energy in Silicon and Germanium Based on Full-Band Monte Carlo Simulations*, IEEE **66** (2019) 444–451.

[68] K. Böer and U. Pohl, *Semiconductor Physics.* Springer International Publishing, 2023. `https://books.google.de/books?id=V8KrEAAAQBAJ`.

[69] R. Sedha, *Materials Science.* S. Chand Limited, 2008. `https://books.google.de/books?id=_UVLAgAAQBAJ`.

[70] ATLAS ITK Collaboration, J. Grosse-Knetter, *ATLAS ITk Pixel Module Bump Bond Stress Analysis*, PoS **Pixel2022** (2023) 056.

[71] A. Cervelli, *Status and Perspectives of Silicon Detectors*, EPJ Web of Conferences **290** (2023) 10007.

[72] A. Skaf, *ATLAS ITk Pixel Detector Overview*, tech. rep., CERN, Geneva, 2024. `https://cds.cern.ch/record/2893857`. DIS2024 - Grenoble 08/04/2024.

[73] ATLAS Collaboration, F. Munoz Sanchez, *Carbon based local supports for the ATLAS ITk-pixel detector*, tech. rep., CERN, Geneva, 2023. `https://cds.cern.ch/record/2847861`.

[74] ATLAS Collaboration, M. Wielers, *ATLAS ITk Production Database use and tools*, `https://cds.cern.ch/record/2914825`.

[75] J. Francisca and M. Sánchez, *ATLAS ITk Pixel Detector Overview*, J. Phys.: Conf. Ser. **2374** (2022) 012061.

[76] RD53 Collaboration, M. Garcia-Sciveres, *RD53A Integrated Circuit Specifications*, tech. rep., CERN, Geneva, 2015. `https://cds.cern.ch/record/2113263`.

[77] RD53 Collaboration, M. Garcia-Sciveres, *The RD53A Integrated Circuit*, tech. rep., CERN, Geneva, 2017. `https://cds.cern.ch/record/2287593`.

[78] S. Marconi, et al., *Design implementation and test results of the RD53A, a 65 nm large scale chip for next generation pixel detectors at the HL-LHC*, 2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC) (2018) 1–4.

[79] RD53 Collaboration, E. Conti et al., *Development of a Large Pixel Chip Demonstrator in RD53 for ATLAS and CMS Upgrades*, PoS **TWEPP-17** (2017) 005.

[80] ATLAS Collaboration, L. Meng, *ATLAS ITk Pixel Detector Overview*, in *International Workshop on Future Linear Colliders*. 5, 2021. `arXiv:2105.10367 [physics.ins-det]`.

[81] RD53 Collaboration, M. Garcia-Sciveres, F. Loddo, and J. Christiansen, *RD53B Manual*, tech. rep., CERN, Geneva, 2019. `https://cds.cern.ch/record/2665301`.

[82] M. Daas, et al., *BDAQ53, a versatile pixel detector readout and test system for the ATLAS and CMS HL-LHC upgrades*, Nucl. Instrum. Meth. A **986** (2021) 164721.

[83] N. L. Whallon, et al., *Upgrade of the YARR DAQ system for the ATLAS Phase-II pixel detector readout chip*, PoS **TWEPP-17** (2018) 076.

[84] T. Heim, *YARR - A PCIe based readout concept for current and future ATLAS Pixel modules*, J. Phys.: Conf. Ser. **898** (2017) 032053.

[85] ATLAS Collaboration, J. Hoya, *FELIX: first operational experience with the new ATLAS readout system and perspectives for HL-LHC*, tech. rep., CERN, Geneva, 2023. `https://cds.cern.ch/record/2871991`.

[86] N. Ilic, J. Vermeulen, and S. Kolos, *FELIX: the new detector interface for the ATLAS experiment*, in *EPJ Web of Conferences*, p. , 01023, EDP Sciences. 2019.

[87] A. Paramonov, *FELIX: the Detector Interface for the ATLAS Experiment at CERN*, EPJ Web Conf. **251** (2021) 04006.

[88] D. Santo, et al., *Test of the Optosystem for the ATLAS ITk data transmission chain*, JINST **18** (2023) C03021.

[89] ATLAS Collaboration, L. Franconi, *The Opto-electrical conversion system for the data transmission chain of the ATLAS ITk Pixel detector upgrade for the HL-LHC*, Tech. Rep. 1, CERN, Geneva, 2022. `https://cds.cern.ch/record/2773360`.

[90] C. Chen, et al., *A gigabit transceiver for the ATLAS inner tracker pixel detector readout upgrade*, JINST **14** (2019) C07005.

[91] P. Moreira, et al., *lpGBT documentation: release.* 2022. `https://cds.cern.ch/record/2809058`.

[92] J. Troska, et al., *The VTRx+, an optical link module for data transmission at HL-LHC*, PoS **TWEPP-17** (2017) 048.

[93] ATLAS ITK Collaboration, S. Möbius, *The Optosystem: validation and testing of the high-speed optical-to-electrical conversion system for the readout of the ATLAS ITk Pixel upgrade*, `arXiv:2310.19637`, `https://cds.cern.ch/record/2878710`, Topical Workshop on Electronics for Particle Physics 2023, 1 October 2023 to 6 October 2023, Geremeas, Sardinia.

[94] ATLAS TDAQ Collaboration, J. Hoya, *FELIX: First operational experience with the new ATLAS readout system and perspectives for HL-LHC*, EPJ Web Conf. **295** (2024) 02012.

[95] R. Lesma, et al., *The Versatile Link Demo Board (VLDB)*, JINST **12** (2017) C02020–C02020.

[96] *First time setup — atlaswiki.lbl.gov*, `https://atlaswiki.lbl.gov/pixels/felixdaq/setup#new-6-dp-zaza-adapter-board`. [Accessed 07-08-2024].

[97] ATLAS Collaboration, G. Aad et al., *Accuracy versus precision in boosted top tagging with the ATLAS detector*, JINST **19** (2024) P08018.

Appendices

APPENDIX A

YARR software developments and results

In this appendix, all the YARR developments are presented.

A.1. readout process in YARR bdaq controller for RD53A

Figure A.1 shows the algorithm of the readout process in the Bdaq controller. The readout process receives a buffer of 32-bit data words, each consisting of 16 bits of hit data, a header, and the remaining bits set to zero. It first checks if the buffer is empty. Then, it checks all data words and writes them to storage according to the readout channel they come from. Next, it loops over all the created maps, checking each data word to determine if it is a register data word or a hit data word. It forms 32-bit data words from every two consecutive data words by removing the header and the zero-value bits from each data word. These created data words are stored in another map with an address related to their readout channel. Finally, these maps are packaged and sent to the libRd53a library.

A.2. readout process in YARR bdaq controller for RD53B

It is almost the same readout process used for RD53A, with the difference being in preparing the hit data words. The libRd53b processor expects to receive the data words as the ITkPix readout chip outputs them (64-bit data words representing its stream, discussed in Section 6.1.2). The readout process of YARR bdaq controller forms a pair of 32-bit data words to present the data as it is output from the RD53B readout chip to libRd53b.

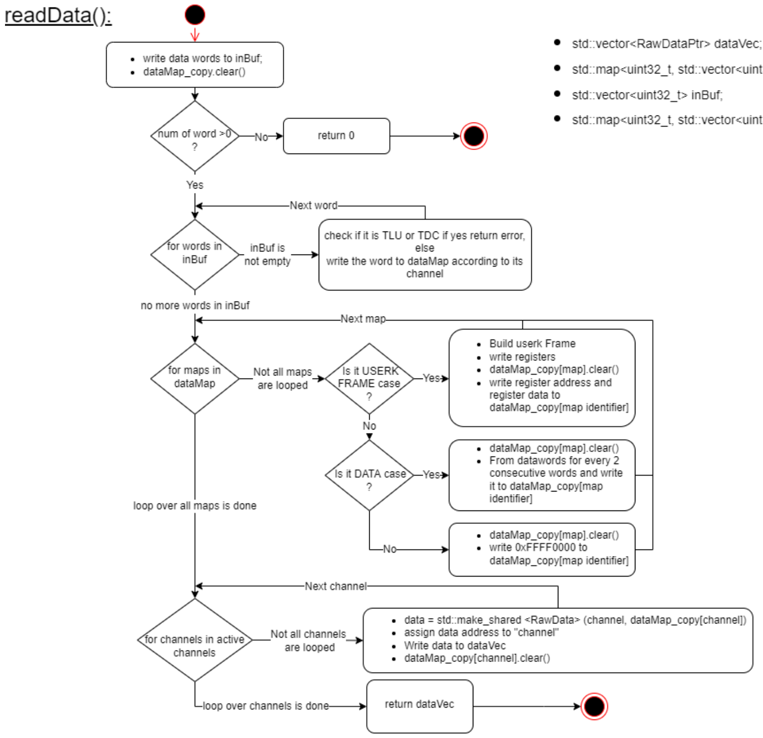

readData():
write data words to inBuf;
dataMap_copy.clear()
std::vector<RawDataPtr> dataVec;
std::map<uint32_t, std::vector<uint32_t>> dataMap_copy;
std::vector<uint32_t> inBuf;
std::map<uint32_t, std::vector<uint32_t>> dataMap;
num of word >0 ?
No
return 0
Yes
Next word
for words in inBuf
inBuf is not empty
check if it is TLU or TDC if yes return error, else write the word to dataMap according to its channel
no more words in inBuf
Next map
for maps in dataMap
Not all maps are looped
Is it USERK FRAME case ?
Yes
Build userk Frame
write registers
dataMap_copy[map].clear()
write register address and register data to dataMap_copy[map identifier]
No
Is it DATA case ?
Yes
dataMap_copy[map].clear()
From datawords for every 2 consecutive words and write it to dataMap_copy[map identifier]
No
dataMap_copy[map].clear()
write 0xFFFF0000 to dataMap_copy[map identifier]
loop over all maps is done
Next channel
for channels in active channels
Not all channels are looped
data = std::make_shared <RawData> (channel, dataMap_copy[channel])
assign data address to "channel"
Write data to dataVec
dataMap_copy[channel].clear()
loop over channels is done
return dataVec

Figure A.1.: The algorithm of the readout process in the YARR Bdaq controller.

A.3. Validation of YARR software using FelixClient results

The output YARR digital scans for ITkPix frontend chips are shown in this section. Figure A.3, Figure A.4, Figure A.5, Figure A.6, Figure A.7, Figure A.8, Figure A.9, Figure A.10 and Figure A.11 display the results of successful digital scans for one, two, three, four, five, six, seven, eight, and nine frontend chips running simultaneously, respectively.

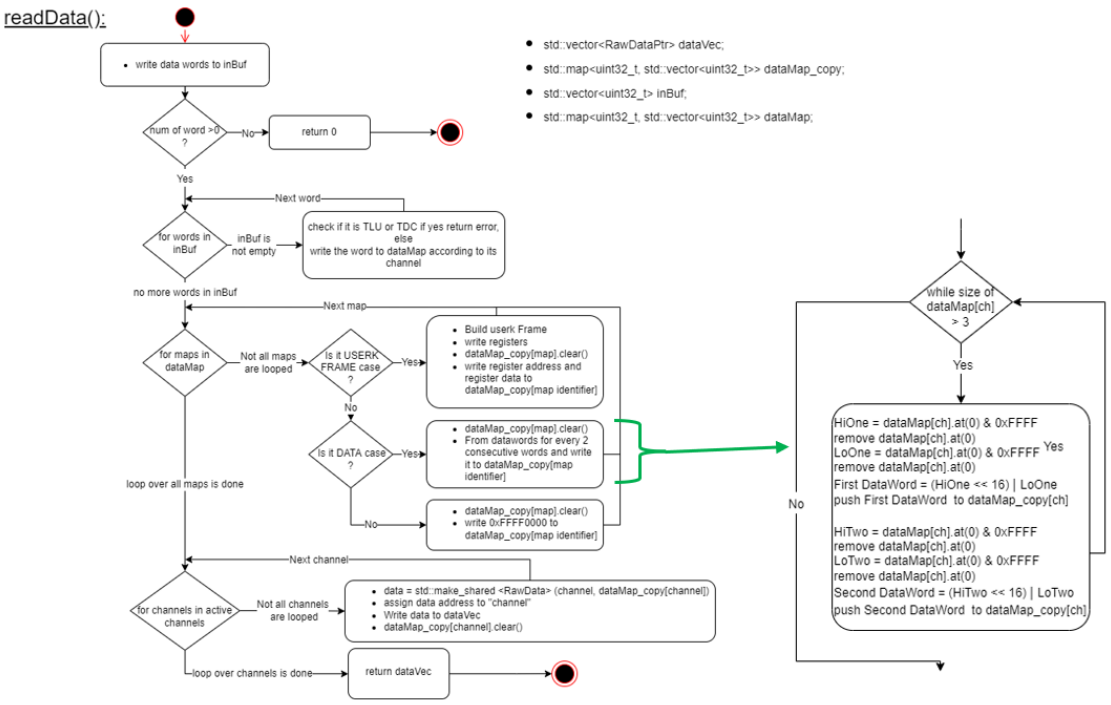
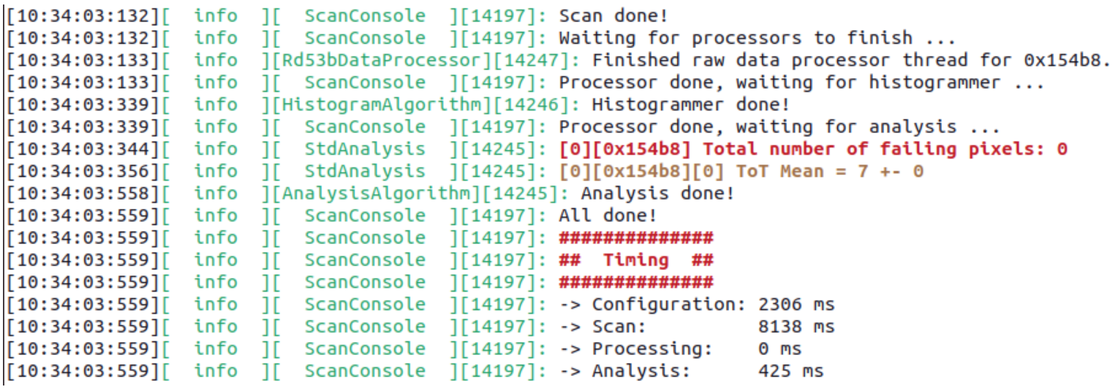
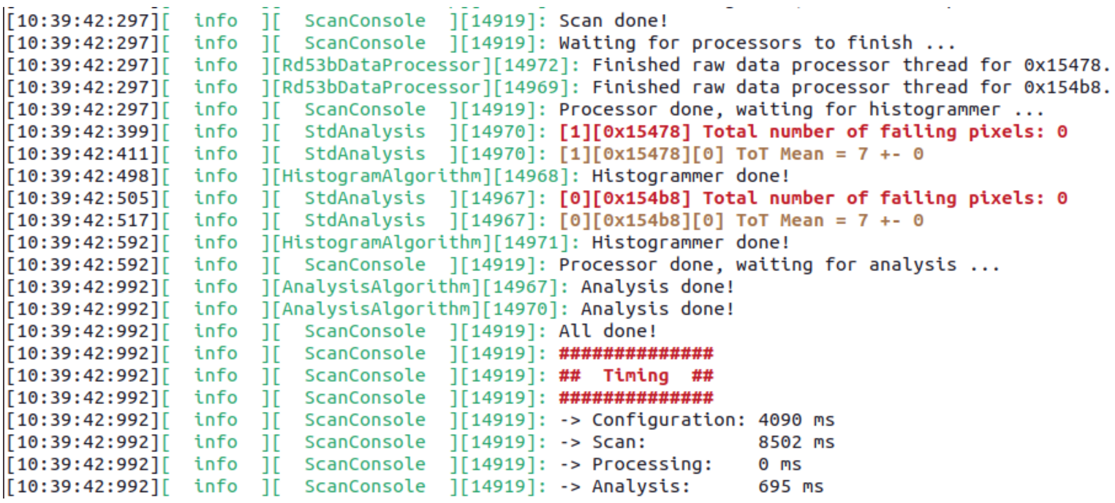

readData():
write data words to inBuf
num of word >0 ?
No
return 0
Yes
Next word
for words in inBuf
inBuf is not empty
check if it is TLU or TDC if yes return error, else write the word to dataMap according to its channel
no more words in inBuf
Next map
for maps in dataMap
Not all maps are looped
Is it USERK FRAME case ?
Yes
Build userk Frame
write registers
dataMap_copy[map].clear()
write register address and register data to dataMap_copy[map identifier]
No
Is it DATA case ?
Yes
dataMap_copy[map].clear()
From datawords for every 2 consecutive words and write it to dataMap_copy[map identifier]
No
dataMap_copy[map].clear()
write 0xFFFF0000 to dataMap_copy[map identifier]
loop over all maps is done
Next channel
for channels in active channels
Not all channels are looped
data = std::make_shared <RawData> (channel, dataMap_copy[channel])
assign data address to "channel"
Write data to dataVec
dataMap_copy[channel].clear()
loop over channels is done
return dataVec
std::vector<RawDataPtr> dataVec;
std::map<uint32_t, std::vector<uint32_t>> dataMap_copy;
std::vector<uint32_t> inBuf;
std::map<uint32_t, std::vector<uint32_t>> dataMap;
while size of dataMap[ch] > 3
Yes
No
HiOne = dataMap[ch].at(0) & 0xFFFF
remove dataMap[ch].at(0)
LoOne = dataMap[ch].at(0) & 0xFFFF
remove dataMap[ch].at(0)
First DataWord = (HiOne << 16) | LoOne
push First DataWord to dataMap_copy[ch]
HiTwo = dataMap[ch].at(0) & 0xFFFF
remove dataMap[ch].at(0)
LoTwo = dataMap[ch].at(0) & 0xFFFF
remove dataMap[ch].at(0)
Second DataWord = (HiTwo << 16) | LoTwo
push Second DataWord to dataMap_copy[ch]
Yes

Figure A.2.: The algorithm for the readout process in the YARR BDAQ controller for RD53B (ITkPix).



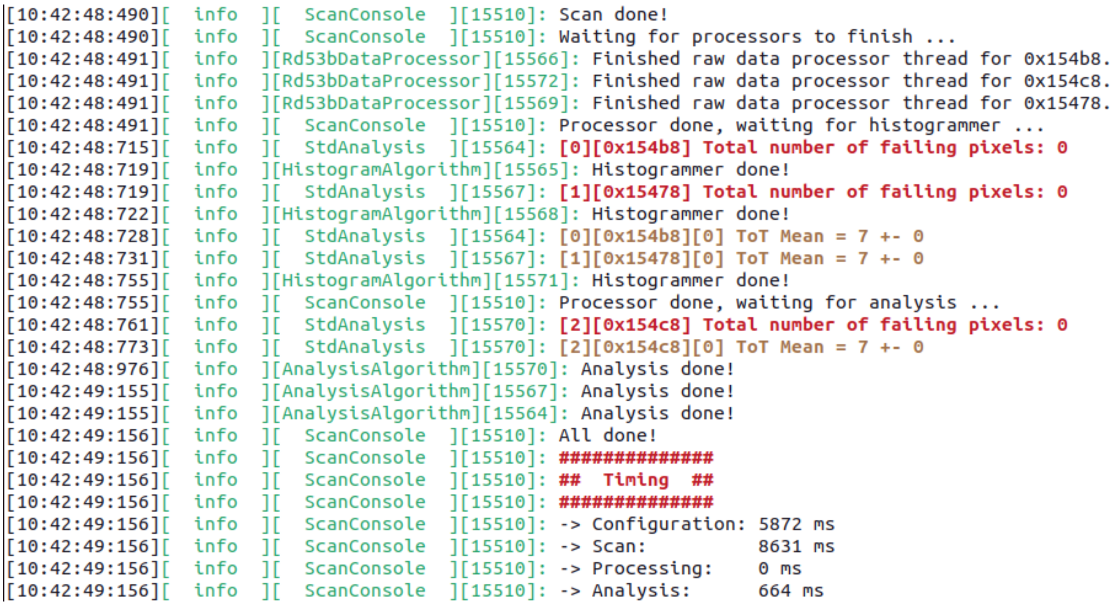
```
[10:34:03:132][  info  ][  ScanConsole  ][14197]: Scan done!
[10:34:03:132][  info  ][  ScanConsole  ][14197]: Waiting for processors to finish ...
[10:34:03:133][  info  ][Rd53bDataProcessor][14247]: Finished raw data processor thread for 0x154b8.
[10:34:03:133][  info  ][  ScanConsole  ][14197]: Processor done, waiting for histogrammer ...
[10:34:03:339][  info  ][HistogramAlgorithm][14246]: Histogrammer done!
[10:34:03:339][  info  ][  ScanConsole  ][14197]: Processor done, waiting for analysis ...
[10:34:03:344][  info  ][  StdAnalysis  ][14245]: [0][0x154b8] Total number of failing pixels: 0
[10:34:03:356][  info  ][  StdAnalysis  ][14245]: [0][0x154b8][0] ToT Mean = 7 +- 0
[10:34:03:558][  info  ][AnalysisAlgorithm][14245]: Analysis done!
[10:34:03:559][  info  ][  ScanConsole  ][14197]: All done!
[10:34:03:559][  info  ][  ScanConsole  ][14197]: ##############
[10:34:03:559][  info  ][  ScanConsole  ][14197]: ##   Timing  ##
[10:34:03:559][  info  ][  ScanConsole  ][14197]: ##############
[10:34:03:559][  info  ][  ScanConsole  ][14197]: -> Configuration: 2306 ms
[10:34:03:559][  info  ][  ScanConsole  ][14197]: -> Scan:          8138 ms
[10:34:03:559][  info  ][  ScanConsole  ][14197]: -> Processing:    0 ms
[10:34:03:559][  info  ][  ScanConsole  ][14197]: -> Analysis:      425 ms
```

Figure A.3.: A successful YARR digital scan with one frontend chip was completed in 10.869 s.



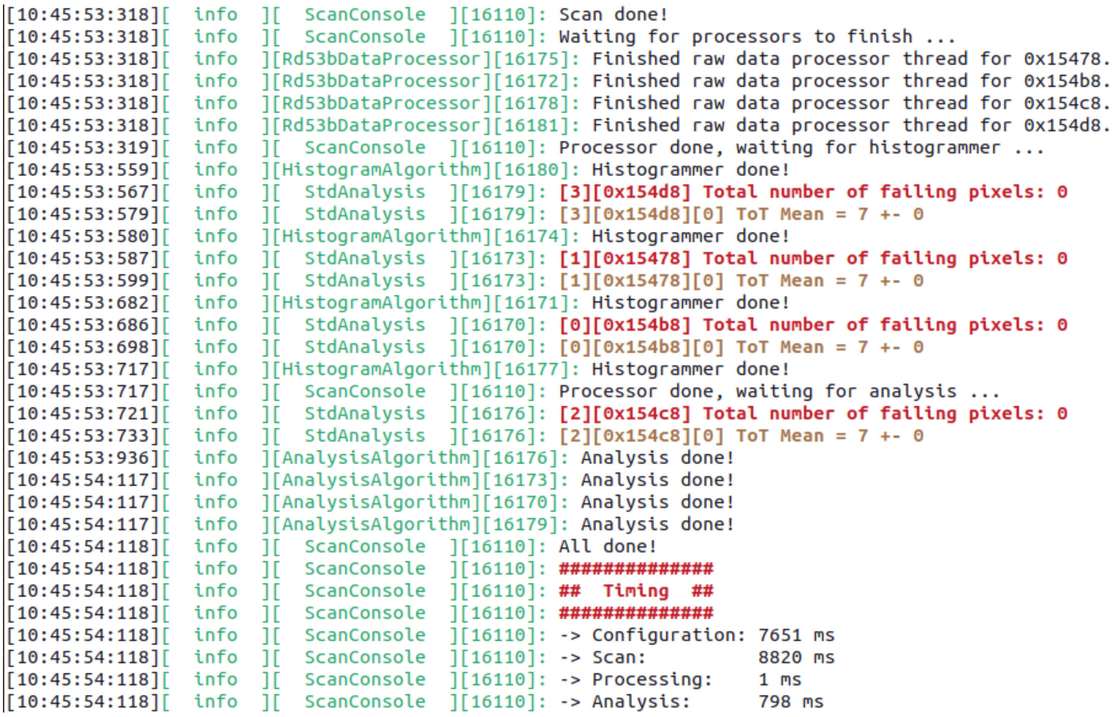
```
[10:39:42:297][   info   ][   ScanConsole   ][14919]: Scan done!
[10:39:42:297][   info   ][   ScanConsole   ][14919]: Waiting for processors to finish ...
[10:39:42:297][   info   ][Rd53bDataProcessor][14972]: Finished raw data processor thread for 0x15478.
[10:39:42:297][   info   ][Rd53bDataProcessor][14969]: Finished raw data processor thread for 0x154b8.
[10:39:42:297][   info   ][   ScanConsole   ][14919]: Processor done, waiting for histogrammer ...
[10:39:42:399][   info   ][   StdAnalysis   ][14970]: [1][0x15478] Total number of failing pixels: 0
[10:39:42:411][   info   ][   StdAnalysis   ][14970]: [1][0x15478][0] ToT Mean = 7 +- 0
[10:39:42:498][   info   ][HistogramAlgorithm][14968]: Histogrammer done!
[10:39:42:505][   info   ][   StdAnalysis   ][14967]: [0][0x154b8] Total number of failing pixels: 0
[10:39:42:517][   info   ][   StdAnalysis   ][14967]: [0][0x154b8][0] ToT Mean = 7 +- 0
[10:39:42:592][   info   ][HistogramAlgorithm][14971]: Histogrammer done!
[10:39:42:592][   info   ][   ScanConsole   ][14919]: Processor done, waiting for analysis ...
[10:39:42:992][   info   ][AnalysisAlgorithm][14967]: Analysis done!
[10:39:42:992][   info   ][AnalysisAlgorithm][14970]: Analysis done!
[10:39:42:992][   info   ][   ScanConsole   ][14919]: All done!
[10:39:42:992][   info   ][   ScanConsole   ][14919]: ##############
[10:39:42:992][   info   ][   ScanConsole   ][14919]: ##  Timing  ##
[10:39:42:992][   info   ][   ScanConsole   ][14919]: ##############
[10:39:42:992][   info   ][   ScanConsole   ][14919]: -> Configuration: 4090 ms
[10:39:42:992][   info   ][   ScanConsole   ][14919]: -> Scan:          8502 ms
[10:39:42:992][   info   ][   ScanConsole   ][14919]: -> Processing:    0 ms
[10:39:42:992][   info   ][   ScanConsole   ][14919]: -> Analysis:      695 ms
```

Figure A.4.: A successful YARR digital scan with two frontend chips operated simultaneously took 13.287 s.



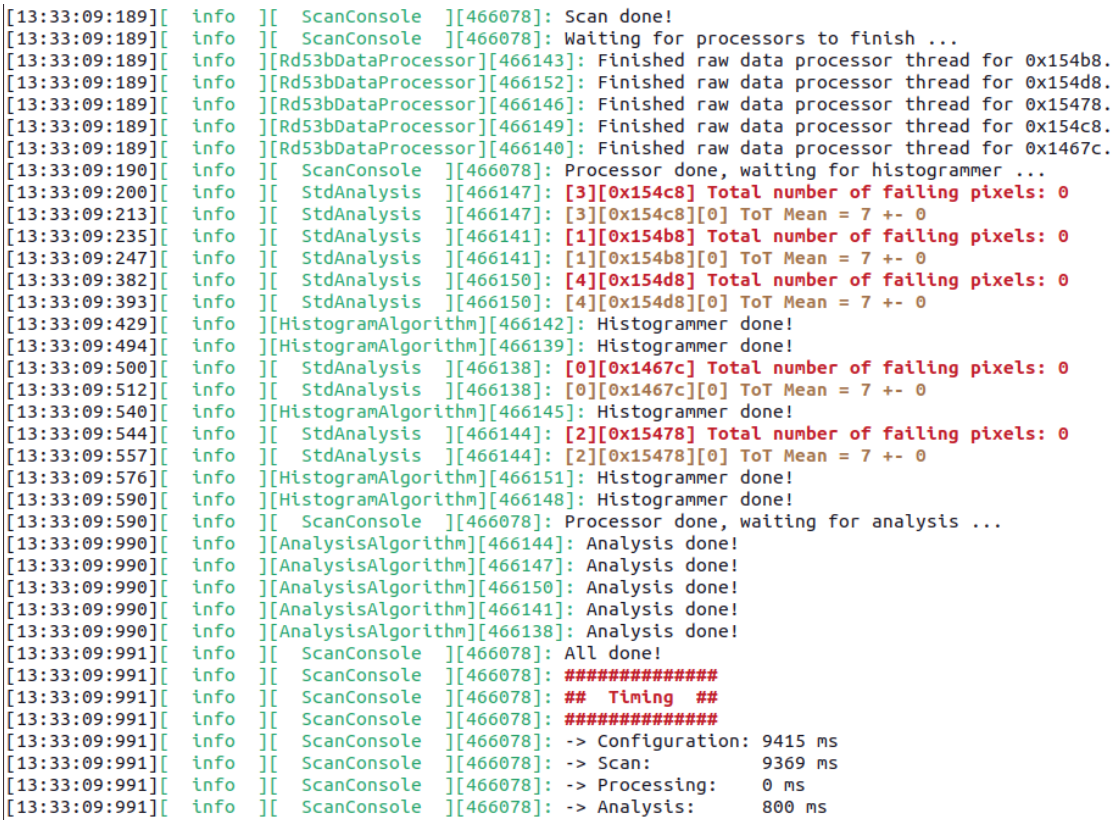
```
[10:42:48:490][   info   ][   ScanConsole   ][15510]: Scan done!
[10:42:48:490][   info   ][   ScanConsole   ][15510]: Waiting for processors to finish ...
[10:42:48:491][   info   ][Rd53bDataProcessor][15566]: Finished raw data processor thread for 0x154b8.
[10:42:48:491][   info   ][Rd53bDataProcessor][15572]: Finished raw data processor thread for 0x154c8.
[10:42:48:491][   info   ][Rd53bDataProcessor][15569]: Finished raw data processor thread for 0x15478.
[10:42:48:491][   info   ][   ScanConsole   ][15510]: Processor done, waiting for histogrammer ...
[10:42:48:715][   info   ][   StdAnalysis   ][15564]: [0][0x154b8] Total number of failing pixels: 0
[10:42:48:719][   info   ][HistogramAlgorithm][15565]: Histogrammer done!
[10:42:48:719][   info   ][   StdAnalysis   ][15567]: [1][0x15478] Total number of failing pixels: 0
[10:42:48:722][   info   ][HistogramAlgorithm][15568]: Histogrammer done!
[10:42:48:728][   info   ][   StdAnalysis   ][15564]: [0][0x154b8][0] ToT Mean = 7 +- 0
[10:42:48:731][   info   ][   StdAnalysis   ][15567]: [1][0x15478][0] ToT Mean = 7 +- 0
[10:42:48:755][   info   ][HistogramAlgorithm][15571]: Histogrammer done!
[10:42:48:755][   info   ][   ScanConsole   ][15510]: Processor done, waiting for analysis ...
[10:42:48:761][   info   ][   StdAnalysis   ][15570]: [2][0x154c8] Total number of failing pixels: 0
[10:42:48:773][   info   ][   StdAnalysis   ][15570]: [2][0x154c8][0] ToT Mean = 7 +- 0
[10:42:48:976][   info   ][AnalysisAlgorithm][15570]: Analysis done!
[10:42:49:155][   info   ][AnalysisAlgorithm][15567]: Analysis done!
[10:42:49:155][   info   ][AnalysisAlgorithm][15564]: Analysis done!
[10:42:49:156][   info   ][   ScanConsole   ][15510]: All done!
[10:42:49:156][   info   ][   ScanConsole   ][15510]: ##############
[10:42:49:156][   info   ][   ScanConsole   ][15510]: ##  Timing  ##
[10:42:49:156][   info   ][   ScanConsole   ][15510]: ##############
[10:42:49:156][   info   ][   ScanConsole   ][15510]: -> Configuration: 5872 ms
[10:42:49:156][   info   ][   ScanConsole   ][15510]: -> Scan:          8631 ms
[10:42:49:156][   info   ][   ScanConsole   ][15510]: -> Processing:    0 ms
[10:42:49:156][   info   ][   ScanConsole   ][15510]: -> Analysis:      664 ms
```

Figure A.5.: A successful YARR digital scan with three frontend chips operated simultaneously took 15.167 s.



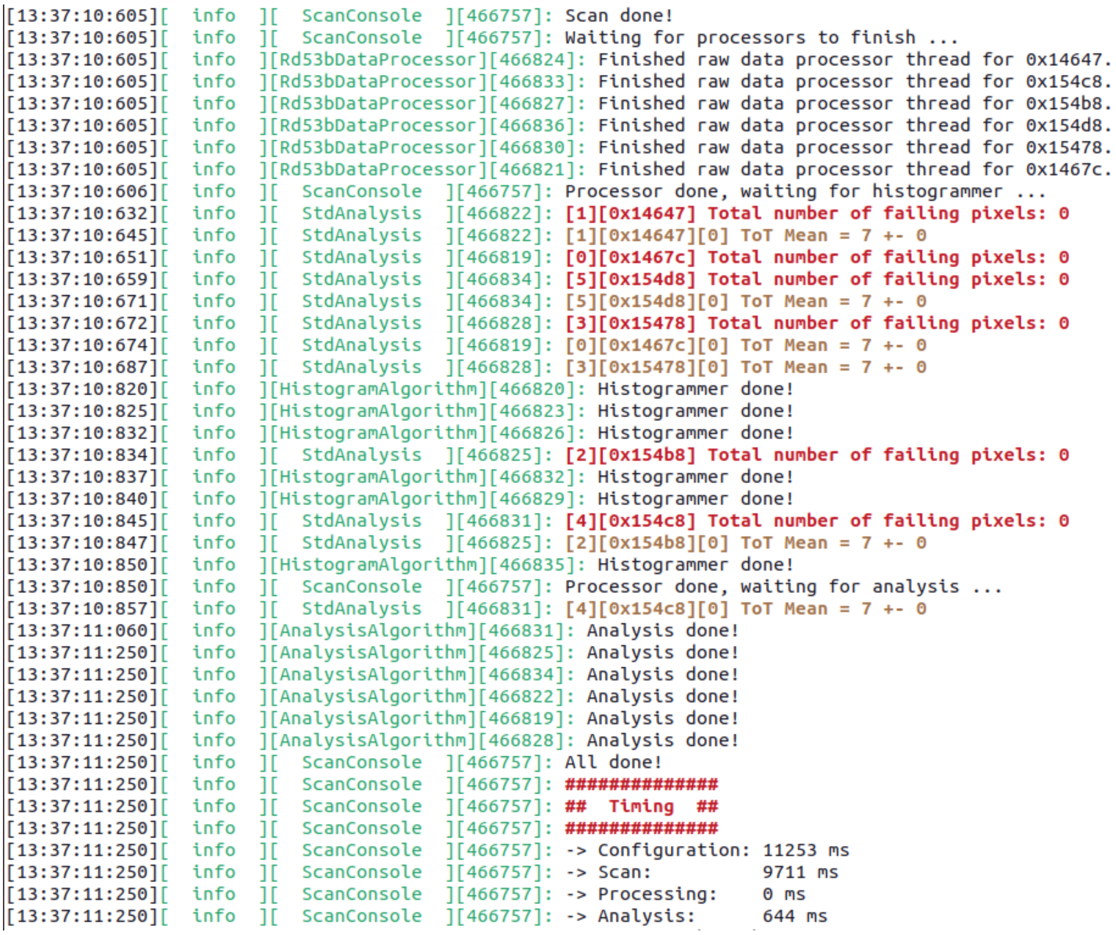
```
[10:45:53:318][  info  ][  ScanConsole  ][16110]: Scan done!
[10:45:53:318][  info  ][  ScanConsole  ][16110]: Waiting for processors to finish ...
[10:45:53:318][  info  ][Rd53bDataProcessor][16175]: Finished raw data processor thread for 0x15478.
[10:45:53:318][  info  ][Rd53bDataProcessor][16172]: Finished raw data processor thread for 0x154b8.
[10:45:53:318][  info  ][Rd53bDataProcessor][16178]: Finished raw data processor thread for 0x154c8.
[10:45:53:318][  info  ][Rd53bDataProcessor][16181]: Finished raw data processor thread for 0x154d8.
[10:45:53:319][  info  ][  ScanConsole  ][16110]: Processor done, waiting for histogrammer ...
[10:45:53:559][  info  ][HistogramAlgorithm][16180]: Histogrammer done!
[10:45:53:567][  info  ][  StdAnalysis  ][16179]: [3][0x154d8] Total number of failing pixels: 0
[10:45:53:579][  info  ][  StdAnalysis  ][16179]: [3][0x154d8][0] ToT Mean = 7 +- 0
[10:45:53:580][  info  ][HistogramAlgorithm][16174]: Histogrammer done!
[10:45:53:587][  info  ][  StdAnalysis  ][16173]: [1][0x15478] Total number of failing pixels: 0
[10:45:53:599][  info  ][  StdAnalysis  ][16173]: [1][0x15478][0] ToT Mean = 7 +- 0
[10:45:53:682][  info  ][HistogramAlgorithm][16171]: Histogrammer done!
[10:45:53:686][  info  ][  StdAnalysis  ][16170]: [0][0x154b8] Total number of failing pixels: 0
[10:45:53:698][  info  ][  StdAnalysis  ][16170]: [0][0x154b8][0] ToT Mean = 7 +- 0
[10:45:53:717][  info  ][HistogramAlgorithm][16177]: Histogrammer done!
[10:45:53:717][  info  ][  ScanConsole  ][16110]: Processor done, waiting for analysis ...
[10:45:53:721][  info  ][  StdAnalysis  ][16176]: [2][0x154c8] Total number of failing pixels: 0
[10:45:53:733][  info  ][  StdAnalysis  ][16176]: [2][0x154c8][0] ToT Mean = 7 +- 0
[10:45:53:936][  info  ][AnalysisAlgorithm][16176]: Analysis done!
[10:45:54:117][  info  ][AnalysisAlgorithm][16173]: Analysis done!
[10:45:54:117][  info  ][AnalysisAlgorithm][16170]: Analysis done!
[10:45:54:117][  info  ][AnalysisAlgorithm][16179]: Analysis done!
[10:45:54:118][  info  ][  ScanConsole  ][16110]: All done!
[10:45:54:118][  info  ][  ScanConsole  ][16110]: ##############
[10:45:54:118][  info  ][  ScanConsole  ][16110]: ##  Timing  ##
[10:45:54:118][  info  ][  ScanConsole  ][16110]: ##############
[10:45:54:118][  info  ][  ScanConsole  ][16110]: -> Configuration: 7651 ms
[10:45:54:118][  info  ][  ScanConsole  ][16110]: -> Scan:          8820 ms
[10:45:54:118][  info  ][  ScanConsole  ][16110]: -> Processing:    1 ms
[10:45:54:118][  info  ][  ScanConsole  ][16110]: -> Analysis:      798 ms
```

Figure A.6.: A successful YARR digital scan with four frontend chips operated simultaneously took 17.270 s.



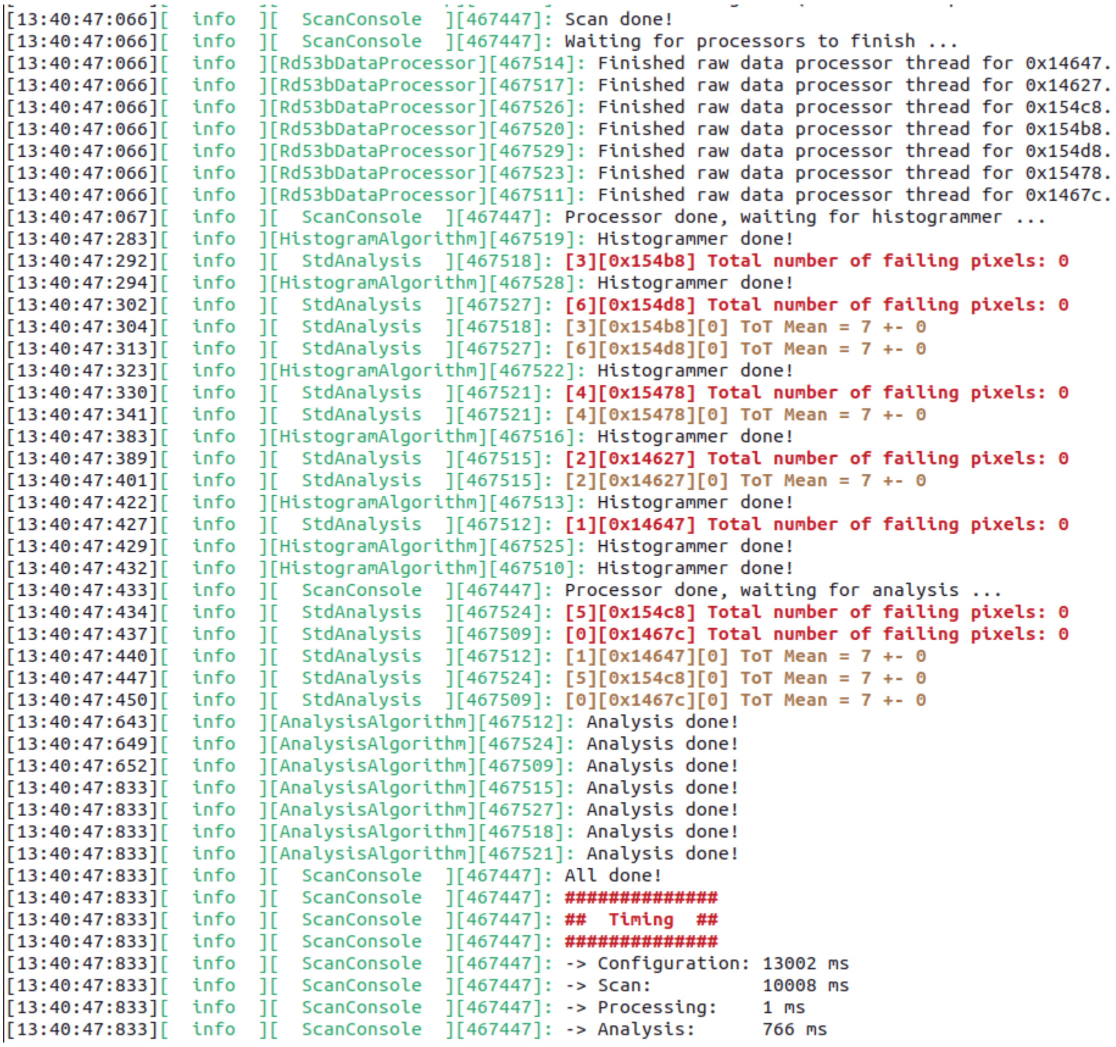
```
[13:33:09:189][  info  ][  ScanConsole  ][466078]: Scan done!
[13:33:09:189][  info  ][  ScanConsole  ][466078]: Waiting for processors to finish ...
[13:33:09:189][  info  ][Rd53bDataProcessor][466143]: Finished raw data processor thread for 0x154b8.
[13:33:09:189][  info  ][Rd53bDataProcessor][466152]: Finished raw data processor thread for 0x154d8.
[13:33:09:189][  info  ][Rd53bDataProcessor][466146]: Finished raw data processor thread for 0x15478.
[13:33:09:189][  info  ][Rd53bDataProcessor][466149]: Finished raw data processor thread for 0x154c8.
[13:33:09:189][  info  ][Rd53bDataProcessor][466140]: Finished raw data processor thread for 0x1467c.
[13:33:09:190][  info  ][  ScanConsole  ][466078]: Processor done, waiting for histogrammer ...
[13:33:09:200][  info  ][  StdAnalysis  ][466147]: [3][0x154c8] Total number of failing pixels: 0
[13:33:09:213][  info  ][  StdAnalysis  ][466147]: [3][0x154c8][0] ToT Mean = 7 +- 0
[13:33:09:235][  info  ][  StdAnalysis  ][466141]: [1][0x154b8] Total number of failing pixels: 0
[13:33:09:247][  info  ][  StdAnalysis  ][466141]: [1][0x154b8][0] ToT Mean = 7 +- 0
[13:33:09:382][  info  ][  StdAnalysis  ][466150]: [4][0x154d8] Total number of failing pixels: 0
[13:33:09:393][  info  ][  StdAnalysis  ][466150]: [4][0x154d8][0] ToT Mean = 7 +- 0
[13:33:09:429][  info  ][HistogramAlgorithm][466142]: Histogrammer done!
[13:33:09:494][  info  ][HistogramAlgorithm][466139]: Histogrammer done!
[13:33:09:500][  info  ][  StdAnalysis  ][466138]: [0][0x1467c] Total number of failing pixels: 0
[13:33:09:512][  info  ][  StdAnalysis  ][466138]: [0][0x1467c][0] ToT Mean = 7 +- 0
[13:33:09:540][  info  ][HistogramAlgorithm][466145]: Histogrammer done!
[13:33:09:544][  info  ][  StdAnalysis  ][466144]: [2][0x15478] Total number of failing pixels: 0
[13:33:09:557][  info  ][  StdAnalysis  ][466144]: [2][0x15478][0] ToT Mean = 7 +- 0
[13:33:09:576][  info  ][HistogramAlgorithm][466151]: Histogrammer done!
[13:33:09:590][  info  ][HistogramAlgorithm][466148]: Histogrammer done!
[13:33:09:590][  info  ][  ScanConsole  ][466078]: Processor done, waiting for analysis ...
[13:33:09:990][  info  ][AnalysisAlgorithm][466144]: Analysis done!
[13:33:09:990][  info  ][AnalysisAlgorithm][466147]: Analysis done!
[13:33:09:990][  info  ][AnalysisAlgorithm][466150]: Analysis done!
[13:33:09:990][  info  ][AnalysisAlgorithm][466141]: Analysis done!
[13:33:09:990][  info  ][AnalysisAlgorithm][466138]: Analysis done!
[13:33:09:991][  info  ][  ScanConsole  ][466078]: All done!
[13:33:09:991][  info  ][  ScanConsole  ][466078]: ##############
[13:33:09:991][  info  ][  ScanConsole  ][466078]: ##  Timing  ##
[13:33:09:991][  info  ][  ScanConsole  ][466078]: ##############
[13:33:09:991][  info  ][  ScanConsole  ][466078]: -> Configuration: 9415 ms
[13:33:09:991][  info  ][  ScanConsole  ][466078]: -> Scan:          9369 ms
[13:33:09:991][  info  ][  ScanConsole  ][466078]: -> Processing:    0 ms
[13:33:09:991][  info  ][  ScanConsole  ][466078]: -> Analysis:      800 ms
```

Figure A.7.: A successful YARR digital scan with five frontend chips operated simultaneously took 19.584 s.



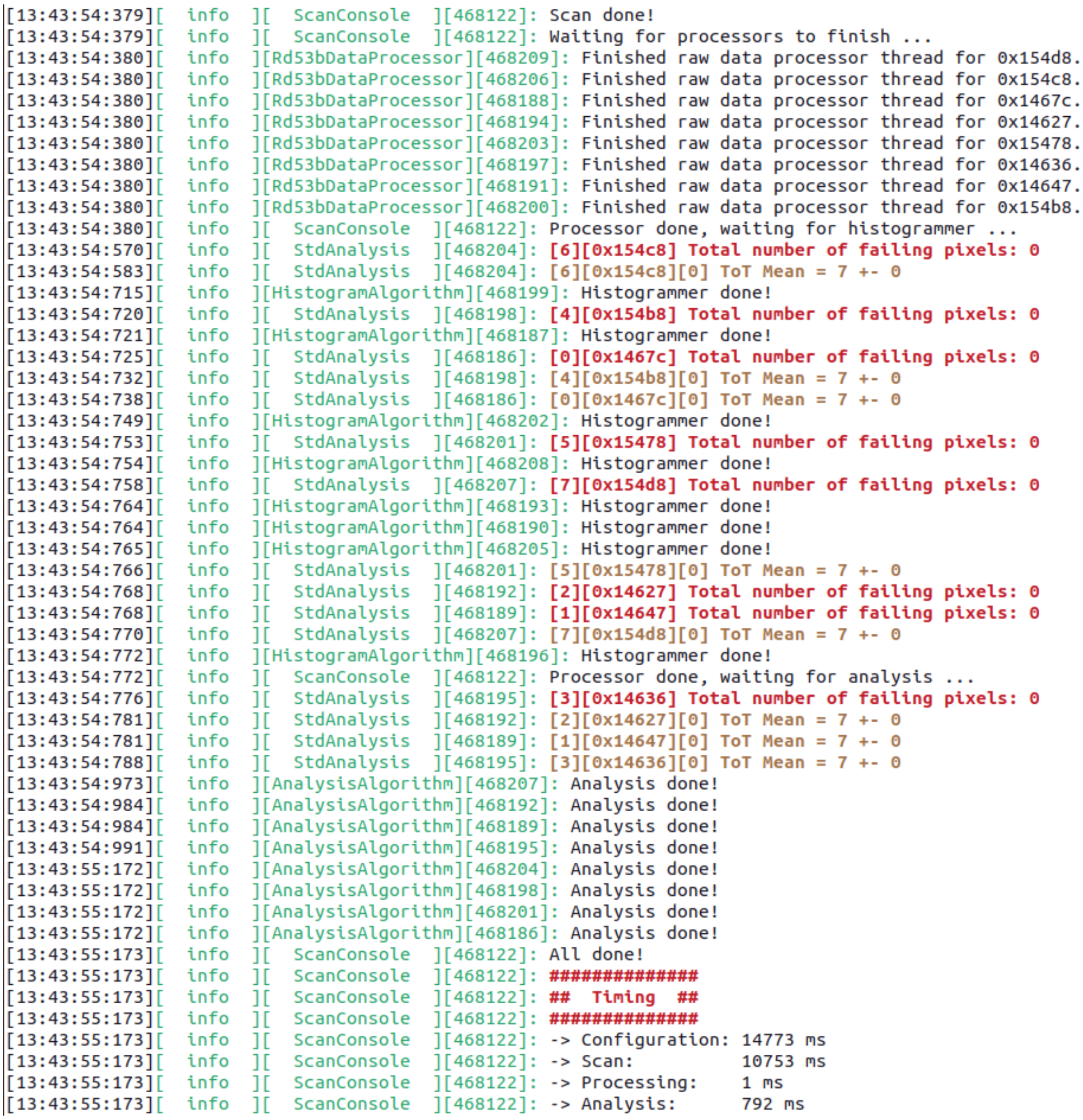
```
[13:37:10:605][  info  ][  ScanConsole  ][466757]: Scan done!
[13:37:10:605][  info  ][  ScanConsole  ][466757]: Waiting for processors to finish ...
[13:37:10:605][  info  ][Rd53bDataProcessor][466824]: Finished raw data processor thread for 0x14647.
[13:37:10:605][  info  ][Rd53bDataProcessor][466833]: Finished raw data processor thread for 0x154c8.
[13:37:10:605][  info  ][Rd53bDataProcessor][466827]: Finished raw data processor thread for 0x154b8.
[13:37:10:605][  info  ][Rd53bDataProcessor][466836]: Finished raw data processor thread for 0x154d8.
[13:37:10:605][  info  ][Rd53bDataProcessor][466830]: Finished raw data processor thread for 0x15478.
[13:37:10:605][  info  ][Rd53bDataProcessor][466821]: Finished raw data processor thread for 0x1467c.
[13:37:10:606][  info  ][  ScanConsole  ][466757]: Processor done, waiting for histogrammer ...
[13:37:10:632][  info  ][  StdAnalysis  ][466822]: [1][0x14647] Total number of failing pixels: 0
[13:37:10:645][  info  ][  StdAnalysis  ][466822]: [1][0x14647][0] ToT Mean = 7 +- 0
[13:37:10:651][  info  ][  StdAnalysis  ][466819]: [0][0x1467c] Total number of failing pixels: 0
[13:37:10:659][  info  ][  StdAnalysis  ][466834]: [5][0x154d8] Total number of failing pixels: 0
[13:37:10:671][  info  ][  StdAnalysis  ][466834]: [5][0x154d8][0] ToT Mean = 7 +- 0
[13:37:10:672][  info  ][  StdAnalysis  ][466828]: [3][0x15478] Total number of failing pixels: 0
[13:37:10:674][  info  ][  StdAnalysis  ][466819]: [0][0x1467c][0] ToT Mean = 7 +- 0
[13:37:10:687][  info  ][  StdAnalysis  ][466828]: [3][0x15478][0] ToT Mean = 7 +- 0
[13:37:10:820][  info  ][HistogramAlgorithm][466820]: Histogrammer done!
[13:37:10:825][  info  ][HistogramAlgorithm][466823]: Histogrammer done!
[13:37:10:832][  info  ][HistogramAlgorithm][466826]: Histogrammer done!
[13:37:10:834][  info  ][  StdAnalysis  ][466825]: [2][0x154b8] Total number of failing pixels: 0
[13:37:10:837][  info  ][HistogramAlgorithm][466832]: Histogrammer done!
[13:37:10:840][  info  ][HistogramAlgorithm][466829]: Histogrammer done!
[13:37:10:845][  info  ][  StdAnalysis  ][466831]: [4][0x154c8] Total number of failing pixels: 0
[13:37:10:847][  info  ][  StdAnalysis  ][466825]: [2][0x154b8][0] ToT Mean = 7 +- 0
[13:37:10:850][  info  ][HistogramAlgorithm][466835]: Histogrammer done!
[13:37:10:850][  info  ][  ScanConsole  ][466757]: Processor done, waiting for analysis ...
[13:37:10:857][  info  ][  StdAnalysis  ][466831]: [4][0x154c8][0] ToT Mean = 7 +- 0
[13:37:11:060][  info  ][AnalysisAlgorithm][466831]: Analysis done!
[13:37:11:250][  info  ][AnalysisAlgorithm][466825]: Analysis done!
[13:37:11:250][  info  ][AnalysisAlgorithm][466834]: Analysis done!
[13:37:11:250][  info  ][AnalysisAlgorithm][466822]: Analysis done!
[13:37:11:250][  info  ][AnalysisAlgorithm][466819]: Analysis done!
[13:37:11:250][  info  ][AnalysisAlgorithm][466828]: Analysis done!
[13:37:11:250][  info  ][  ScanConsole  ][466757]: All done!
[13:37:11:250][  info  ][  ScanConsole  ][466757]: ##############
[13:37:11:250][  info  ][  ScanConsole  ][466757]: ##  Timing  ##
[13:37:11:250][  info  ][  ScanConsole  ][466757]: ##############
[13:37:11:250][  info  ][  ScanConsole  ][466757]: -> Configuration: 11253 ms
[13:37:11:250][  info  ][  ScanConsole  ][466757]: -> Scan:          9711 ms
[13:37:11:250][  info  ][  ScanConsole  ][466757]: -> Processing:    0 ms
[13:37:11:250][  info  ][  ScanConsole  ][466757]: -> Analysis:      644 ms
```

Figure A.8.: A successful YARR digital scan with six frontend chips operated simultaneously took 21.608 s.



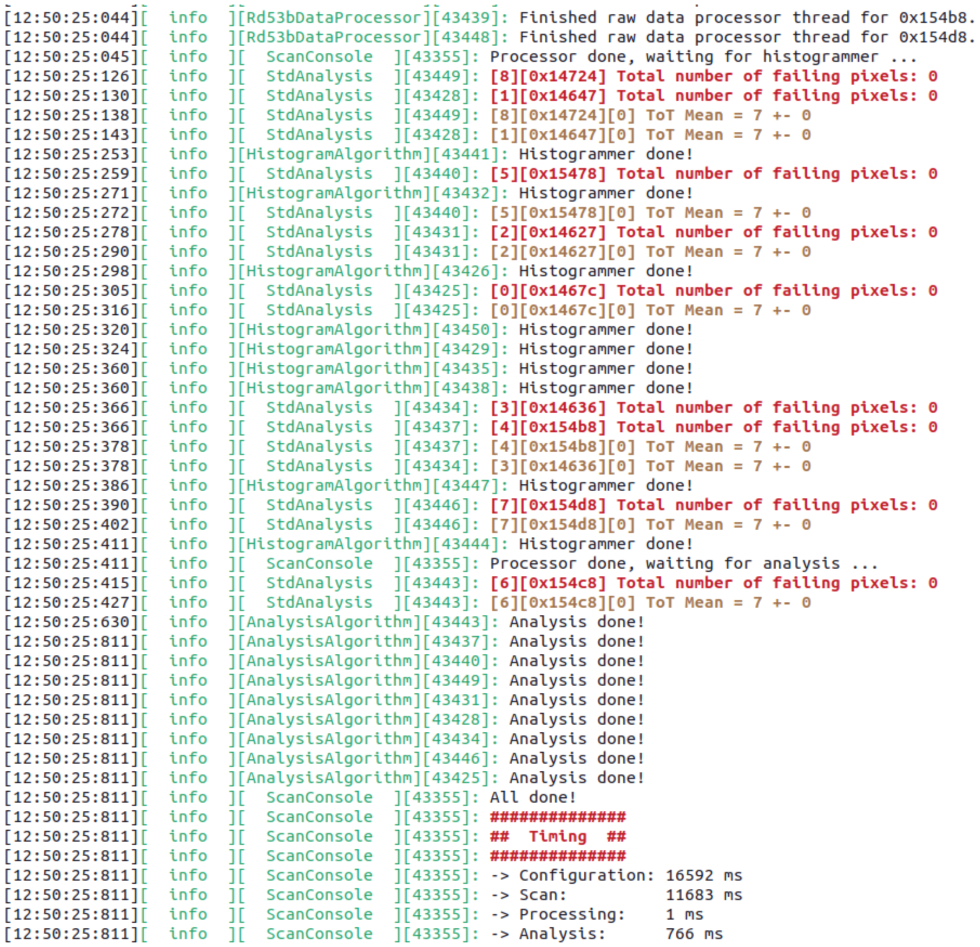
```
[13:40:47:066][   info   ][  ScanConsole   ][467447]: Scan done!
[13:40:47:066][   info   ][  ScanConsole   ][467447]: Waiting for processors to finish ...
[13:40:47:066][   info   ][Rd53bDataProcessor][467514]: Finished raw data processor thread for 0x14647.
[13:40:47:066][   info   ][Rd53bDataProcessor][467517]: Finished raw data processor thread for 0x14627.
[13:40:47:066][   info   ][Rd53bDataProcessor][467526]: Finished raw data processor thread for 0x154c8.
[13:40:47:066][   info   ][Rd53bDataProcessor][467520]: Finished raw data processor thread for 0x154b8.
[13:40:47:066][   info   ][Rd53bDataProcessor][467529]: Finished raw data processor thread for 0x154d8.
[13:40:47:066][   info   ][Rd53bDataProcessor][467523]: Finished raw data processor thread for 0x15478.
[13:40:47:066][   info   ][Rd53bDataProcessor][467511]: Finished raw data processor thread for 0x1467c.
[13:40:47:067][   info   ][  ScanConsole   ][467447]: Processor done, waiting for histogrammer ...
[13:40:47:283][   info   ][HistogramAlgorithm][467519]: Histogrammer done!
[13:40:47:292][   info   ][  StdAnalysis   ][467518]: [3][0x154b8] Total number of failing pixels: 0
[13:40:47:294][   info   ][HistogramAlgorithm][467528]: Histogrammer done!
[13:40:47:302][   info   ][  StdAnalysis   ][467527]: [6][0x154d8] Total number of failing pixels: 0
[13:40:47:304][   info   ][  StdAnalysis   ][467518]: [3][0x154b8][0] ToT Mean = 7 +- 0
[13:40:47:313][   info   ][  StdAnalysis   ][467527]: [6][0x154d8][0] ToT Mean = 7 +- 0
[13:40:47:323][   info   ][HistogramAlgorithm][467522]: Histogrammer done!
[13:40:47:330][   info   ][  StdAnalysis   ][467521]: [4][0x15478] Total number of failing pixels: 0
[13:40:47:341][   info   ][  StdAnalysis   ][467521]: [4][0x15478][0] ToT Mean = 7 +- 0
[13:40:47:383][   info   ][HistogramAlgorithm][467516]: Histogrammer done!
[13:40:47:389][   info   ][  StdAnalysis   ][467515]: [2][0x14627] Total number of failing pixels: 0
[13:40:47:401][   info   ][  StdAnalysis   ][467515]: [2][0x14627][0] ToT Mean = 7 +- 0
[13:40:47:422][   info   ][HistogramAlgorithm][467513]: Histogrammer done!
[13:40:47:427][   info   ][  StdAnalysis   ][467512]: [1][0x14647] Total number of failing pixels: 0
[13:40:47:429][   info   ][HistogramAlgorithm][467525]: Histogrammer done!
[13:40:47:432][   info   ][HistogramAlgorithm][467510]: Histogrammer done!
[13:40:47:433][   info   ][  ScanConsole   ][467447]: Processor done, waiting for analysis ...
[13:40:47:434][   info   ][  StdAnalysis   ][467524]: [5][0x154c8] Total number of failing pixels: 0
[13:40:47:437][   info   ][  StdAnalysis   ][467509]: [0][0x1467c] Total number of failing pixels: 0
[13:40:47:440][   info   ][  StdAnalysis   ][467512]: [1][0x14647][0] ToT Mean = 7 +- 0
[13:40:47:447][   info   ][  StdAnalysis   ][467524]: [5][0x154c8][0] ToT Mean = 7 +- 0
[13:40:47:450][   info   ][  StdAnalysis   ][467509]: [0][0x1467c][0] ToT Mean = 7 +- 0
[13:40:47:643][   info   ][AnalysisAlgorithm][467512]: Analysis done!
[13:40:47:649][   info   ][AnalysisAlgorithm][467524]: Analysis done!
[13:40:47:652][   info   ][AnalysisAlgorithm][467509]: Analysis done!
[13:40:47:833][   info   ][AnalysisAlgorithm][467515]: Analysis done!
[13:40:47:833][   info   ][AnalysisAlgorithm][467527]: Analysis done!
[13:40:47:833][   info   ][AnalysisAlgorithm][467518]: Analysis done!
[13:40:47:833][   info   ][AnalysisAlgorithm][467521]: Analysis done!
[13:40:47:833][   info   ][  ScanConsole   ][467447]: All done!
[13:40:47:833][   info   ][  ScanConsole   ][467447]: ##############
[13:40:47:833][   info   ][  ScanConsole   ][467447]: ##  Timing  ##
[13:40:47:833][   info   ][  ScanConsole   ][467447]: ##############
[13:40:47:833][   info   ][  ScanConsole   ][467447]: -> Configuration: 13002 ms
[13:40:47:833][   info   ][  ScanConsole   ][467447]: -> Scan:          10008 ms
[13:40:47:833][   info   ][  ScanConsole   ][467447]: -> Processing:    1 ms
[13:40:47:833][   info   ][  ScanConsole   ][467447]: -> Analysis:      766 ms
```

Figure A.9.: A successful YARR digital scan with seven frontend chips operated simultaneously took 23.777 s.



```
[13:43:54:379][  info  ][  ScanConsole  ][468122]: Scan done!
[13:43:54:379][  info  ][  ScanConsole  ][468122]: Waiting for processors to finish ...
[13:43:54:380][  info  ][Rd53bDataProcessor][468209]: Finished raw data processor thread for 0x154d8.
[13:43:54:380][  info  ][Rd53bDataProcessor][468206]: Finished raw data processor thread for 0x154c8.
[13:43:54:380][  info  ][Rd53bDataProcessor][468188]: Finished raw data processor thread for 0x1467c.
[13:43:54:380][  info  ][Rd53bDataProcessor][468194]: Finished raw data processor thread for 0x14627.
[13:43:54:380][  info  ][Rd53bDataProcessor][468203]: Finished raw data processor thread for 0x15478.
[13:43:54:380][  info  ][Rd53bDataProcessor][468197]: Finished raw data processor thread for 0x14636.
[13:43:54:380][  info  ][Rd53bDataProcessor][468191]: Finished raw data processor thread for 0x14647.
[13:43:54:380][  info  ][Rd53bDataProcessor][468200]: Finished raw data processor thread for 0x154b8.
[13:43:54:380][  info  ][  ScanConsole  ][468122]: Processor done, waiting for histogrammer ...
[13:43:54:570][  info  ][  StdAnalysis  ][468204]: [6][0x154c8] Total number of failing pixels: 0
[13:43:54:583][  info  ][  StdAnalysis  ][468204]: [6][0x154c8][0] ToT Mean = 7 +- 0
[13:43:54:715][  info  ][HistogramAlgorithm][468199]: Histogrammer done!
[13:43:54:720][  info  ][  StdAnalysis  ][468198]: [4][0x154b8] Total number of failing pixels: 0
[13:43:54:721][  info  ][HistogramAlgorithm][468187]: Histogrammer done!
[13:43:54:725][  info  ][  StdAnalysis  ][468186]: [0][0x1467c] Total number of failing pixels: 0
[13:43:54:732][  info  ][  StdAnalysis  ][468198]: [4][0x154b8][0] ToT Mean = 7 +- 0
[13:43:54:738][  info  ][  StdAnalysis  ][468186]: [0][0x1467c][0] ToT Mean = 7 +- 0
[13:43:54:749][  info  ][HistogramAlgorithm][468202]: Histogrammer done!
[13:43:54:753][  info  ][  StdAnalysis  ][468201]: [5][0x15478] Total number of failing pixels: 0
[13:43:54:754][  info  ][HistogramAlgorithm][468208]: Histogrammer done!
[13:43:54:758][  info  ][  StdAnalysis  ][468207]: [7][0x154d8] Total number of failing pixels: 0
[13:43:54:764][  info  ][HistogramAlgorithm][468193]: Histogrammer done!
[13:43:54:764][  info  ][HistogramAlgorithm][468190]: Histogrammer done!
[13:43:54:765][  info  ][HistogramAlgorithm][468205]: Histogrammer done!
[13:43:54:766][  info  ][  StdAnalysis  ][468201]: [5][0x15478][0] ToT Mean = 7 +- 0
[13:43:54:768][  info  ][  StdAnalysis  ][468192]: [2][0x14627] Total number of failing pixels: 0
[13:43:54:768][  info  ][  StdAnalysis  ][468189]: [1][0x14647] Total number of failing pixels: 0
[13:43:54:770][  info  ][  StdAnalysis  ][468207]: [7][0x154d8][0] ToT Mean = 7 +- 0
[13:43:54:772][  info  ][HistogramAlgorithm][468196]: Histogrammer done!
[13:43:54:772][  info  ][  ScanConsole  ][468122]: Processor done, waiting for analysis ...
[13:43:54:776][  info  ][  StdAnalysis  ][468195]: [3][0x14636] Total number of failing pixels: 0
[13:43:54:781][  info  ][  StdAnalysis  ][468192]: [2][0x14627][0] ToT Mean = 7 +- 0
[13:43:54:781][  info  ][  StdAnalysis  ][468189]: [1][0x14647][0] ToT Mean = 7 +- 0
[13:43:54:788][  info  ][  StdAnalysis  ][468195]: [3][0x14636][0] ToT Mean = 7 +- 0
[13:43:54:973][  info  ][AnalysisAlgorithm][468207]: Analysis done!
[13:43:54:984][  info  ][AnalysisAlgorithm][468192]: Analysis done!
[13:43:54:984][  info  ][AnalysisAlgorithm][468189]: Analysis done!
[13:43:54:991][  info  ][AnalysisAlgorithm][468195]: Analysis done!
[13:43:55:172][  info  ][AnalysisAlgorithm][468204]: Analysis done!
[13:43:55:172][  info  ][AnalysisAlgorithm][468198]: Analysis done!
[13:43:55:172][  info  ][AnalysisAlgorithm][468201]: Analysis done!
[13:43:55:172][  info  ][AnalysisAlgorithm][468186]: Analysis done!
[13:43:55:173][  info  ][  ScanConsole  ][468122]: All done!
[13:43:55:173][  info  ][  ScanConsole  ][468122]: ##############
[13:43:55:173][  info  ][  ScanConsole  ][468122]: ##  Timing  ##
[13:43:55:173][  info  ][  ScanConsole  ][468122]: ##############
[13:43:55:173][  info  ][  ScanConsole  ][468122]: -> Configuration: 14773 ms
[13:43:55:173][  info  ][  ScanConsole  ][468122]: -> Scan:          10753 ms
[13:43:55:173][  info  ][  ScanConsole  ][468122]: -> Processing:    1 ms
[13:43:55:173][  info  ][  ScanConsole  ][468122]: -> Analysis:      792 ms
```

Figure A.10.: A successful YARR digital scan with eight frontend chips operated simultaneously took 26.319 s.



```
[12:50:25:044][  info  ][Rd53bDataProcessor][43439]: Finished raw data processor thread for 0x154b8.
[12:50:25:044][  info  ][Rd53bDataProcessor][43448]: Finished raw data processor thread for 0x154d8.
[12:50:25:045][  info  ][  ScanConsole  ][43355]: Processor done, waiting for histogrammer ...
[12:50:25:126][  info  ][  StdAnalysis  ][43449]: [8][0x14724] Total number of failing pixels: 0
[12:50:25:130][  info  ][  StdAnalysis  ][43428]: [1][0x14647] Total number of failing pixels: 0
[12:50:25:138][  info  ][  StdAnalysis  ][43449]: [8][0x14724][0] ToT Mean = 7 +- 0
[12:50:25:143][  info  ][  StdAnalysis  ][43428]: [1][0x14647][0] ToT Mean = 7 +- 0
[12:50:25:253][  info  ][HistogramAlgorithm][43441]: Histogrammer done!
[12:50:25:259][  info  ][  StdAnalysis  ][43440]: [5][0x15478] Total number of failing pixels: 0
[12:50:25:271][  info  ][HistogramAlgorithm][43432]: Histogrammer done!
[12:50:25:272][  info  ][  StdAnalysis  ][43440]: [5][0x15478][0] ToT Mean = 7 +- 0
[12:50:25:278][  info  ][  StdAnalysis  ][43431]: [2][0x14627] Total number of failing pixels: 0
[12:50:25:290][  info  ][  StdAnalysis  ][43431]: [2][0x14627][0] ToT Mean = 7 +- 0
[12:50:25:298][  info  ][HistogramAlgorithm][43426]: Histogrammer done!
[12:50:25:305][  info  ][  StdAnalysis  ][43425]: [0][0x1467c] Total number of failing pixels: 0
[12:50:25:316][  info  ][  StdAnalysis  ][43425]: [0][0x1467c][0] ToT Mean = 7 +- 0
[12:50:25:320][  info  ][HistogramAlgorithm][43450]: Histogrammer done!
[12:50:25:324][  info  ][HistogramAlgorithm][43429]: Histogrammer done!
[12:50:25:360][  info  ][HistogramAlgorithm][43435]: Histogrammer done!
[12:50:25:360][  info  ][HistogramAlgorithm][43438]: Histogrammer done!
[12:50:25:366][  info  ][  StdAnalysis  ][43434]: [3][0x14636] Total number of failing pixels: 0
[12:50:25:366][  info  ][  StdAnalysis  ][43437]: [4][0x154b8] Total number of failing pixels: 0
[12:50:25:378][  info  ][  StdAnalysis  ][43437]: [4][0x154b8][0] ToT Mean = 7 +- 0
[12:50:25:378][  info  ][  StdAnalysis  ][43434]: [3][0x14636][0] ToT Mean = 7 +- 0
[12:50:25:386][  info  ][HistogramAlgorithm][43447]: Histogrammer done!
[12:50:25:390][  info  ][  StdAnalysis  ][43446]: [7][0x154d8] Total number of failing pixels: 0
[12:50:25:402][  info  ][  StdAnalysis  ][43446]: [7][0x154d8][0] ToT Mean = 7 +- 0
[12:50:25:411][  info  ][HistogramAlgorithm][43444]: Histogrammer done!
[12:50:25:411][  info  ][  ScanConsole  ][43355]: Processor done, waiting for analysis ...
[12:50:25:415][  info  ][  StdAnalysis  ][43443]: [6][0x154c8] Total number of failing pixels: 0
[12:50:25:427][  info  ][  StdAnalysis  ][43443]: [6][0x154c8][0] ToT Mean = 7 +- 0
[12:50:25:630][  info  ][AnalysisAlgorithm][43443]: Analysis done!
[12:50:25:811][  info  ][AnalysisAlgorithm][43437]: Analysis done!
[12:50:25:811][  info  ][AnalysisAlgorithm][43440]: Analysis done!
[12:50:25:811][  info  ][AnalysisAlgorithm][43449]: Analysis done!
[12:50:25:811][  info  ][AnalysisAlgorithm][43431]: Analysis done!
[12:50:25:811][  info  ][AnalysisAlgorithm][43428]: Analysis done!
[12:50:25:811][  info  ][AnalysisAlgorithm][43434]: Analysis done!
[12:50:25:811][  info  ][AnalysisAlgorithm][43446]: Analysis done!
[12:50:25:811][  info  ][AnalysisAlgorithm][43425]: Analysis done!
[12:50:25:811][  info  ][  ScanConsole  ][43355]: All done!
[12:50:25:811][  info  ][  ScanConsole  ][43355]: ##############
[12:50:25:811][  info  ][  ScanConsole  ][43355]: ##  Timing  ##
[12:50:25:811][  info  ][  ScanConsole  ][43355]: ##############
[12:50:25:811][  info  ][  ScanConsole  ][43355]: -> Configuration: 16592 ms
[12:50:25:811][  info  ][  ScanConsole  ][43355]: -> Scan:          11683 ms
[12:50:25:811][  info  ][  ScanConsole  ][43355]: -> Processing:    1 ms
[12:50:25:811][  info  ][  ScanConsole  ][43355]: -> Analysis:      766 ms
```

Figure A.11.: A successful YARR digital scan with nine frontend chips operated simultaneously took 29.042 s.