A SciFi tracker for the LHCb experiment

Feo Pereira Rivello de Carvalho, Mauricio

DOI:
10.33612/diss.788179550

university of groningen

A SciFi tracker for the LHCb experiment

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. J.M.A. Scherpen
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

Tuesday 10 October 2023 at 14:30 hours

by

Mauricio Feo Pereira Rivello de Carvalho

born on 22 June 1989
in Petrópolis, Brazil

Supervisors
Prof. A. Pellegrino
Prof. M.H.M. Merk

Assessment committee
Prof. N. Kalantar-Nayestanaki
Prof. S. Hoekstra
Prof. D. Martello

"Almost nothing material is needed for a happy life, for he who has understood existence."

Marcus Aurelius

“p;;3'h
-0 f''''''''''dz.”

Luna the Cat

Contents

Chapter 1

Introduction

Fundamental particles and their interactions are described by the so-called Standard Model of Particle Physics, developed by both theoretical and experimental physicists throughout the 20^{th} century, a quantum-field theory explaining and providing predictions for a vast range of experimental facts. Nevertheless, for all its successes, the Standard Model leaves unexplained phenomena (baryon asymmetry, dark matter, etc.), and it is therefore believed to be an effective theory of a more fundamental theory. Uncovering this underlying theory is the goal of present particle physics experiments.

Lying 100 meters deep underground, in one of the caverns where proton-proton collisions from the Large Hadron Collider (LHC) happen, the Large Hadron Collider Beauty experiment (LHCb) was built with the primary purpose of studying the rare decays of D and B hadrons and measuring the parameters of charge-parity symmetry violation in such decays. This, in turn, could be a step in unveiling hitherto undiscovered physics that could help us, e.g. to understand the matter-antimatter asymmetry observed in our Universe.

Since LHCb started operating with the first beams of the LHC in 2008, the experiment has recorded countless products of collisions and generated valuable data for a variety of physics analyses. These data are produced by many particle detectors with different characteristics and purposes that compose the LHCb experiment. After the first and second runs of the LHC, an LHCb upgrade was undertaken to increase the experiment's performance and allow it to cope with the increased luminosity expected for the run 3 of the LHC. This upgrade, particularly that of the tracking system, is the focus of this dissertation. The old LHCb detector is presented in Chapter 2 and the upgraded detector in Chapter 3.

Many detectors and systems have undergone drastic changes in the upgrade. The detector that is the main subject of this thesis, the Scintillating Fiber Tracker (SciFi), is responsible for providing the trajectory of charged particles. This is achieved by measuring the positions at which particles cross each of the 12 detection layers that compose the SciFi. The reconstructed tracks provide the experiment data that, combined with those of other sub-detectors, lead to the measurement of the momenta of the particles originating from the collision and the reconstruction of the invariant mass of all decaying particles. The SciFi detector is described in Chapter 4.

When such detectors are designed, it is crucial to characterize their behavior with realistic

studies of the performance of prototypes, through many different tests, to validate their use in the intended application and to determine key parameters for different operating settings in order to optimize the detector performance in all running scenarios.

During this dissertation, several activities were carried out concerning the detector design and production, including mechanical and cooling qualification tests, on-detector and readout electronics studies, and control and data-acquisition software developments. This thesis focuses on two crucial contributions to the realization and the test of the SciFi tracker, namely two extensive performance studies to validate the expected SciFi performance in realistic beam conditions:

- **Radiation Hardness Test of the Electronics**: These tests aim at validating the front-end electronics of the SciFi detector for operation in radiation environments such as the one that we expect in the LHCb Experiment.
- **Full-Module Test Beam**: This was the first test of a SciFi detector module using a complete set of front-end electronics boards in a realistic particle beam.

For the radiation tests, two complete sets of electronics were exposed to a cumulative radiation dose equivalent to the one expected over the whole detector's lifetime, during two periods of one week, in an irradiation facility at CERN. A series of tests were conducted during such periods to evaluate single event effects (SEE) and cumulative radiation effects caused by the total ionizing dose (TID). These results are essential to demonstrate that the radiation effects are indeed those expected and that the mitigation measures are sufficient to retain optimal detector performance. The description of the irradiation tests and their results can be found in Chapter 5.

For the Full-Module Test Beam, two halves of a detector module and their corresponding electronics were brought to one of the test beam lines in the CERN's North Area. A well-characterized telescope was used as a reference for measuring the trajectory of the particles. This test beam campaign had a variety of goals, the primary one being to acquire expertise on setting up the detector and its final electronics together with all required services (cooling, high voltage, control and data acquisition, etc.) and to record the detector performance to ensure that the current version of all components works together as expected. Chapter 6 describes the experimental setup used during the test beam.

The data collected were analyzed to characterize the detector's performance, check that it is within the requirements, and study ways of optimizing the various settings to achieve the best performance. A software framework was developed specifically for this purpose. It provides all the tools for selecting the events recorded by the detector under test, offering the users an easy software toolkit to apply their ideas to the data analysis without dealing with low-level processing. Such a framework is introduced in Chapter 7.

The framework was used for a number of analyses aimed at characterizing the performance of the detector in terms of single-hit efficiency and spatial cluster resolution, as well as optimizing different settings to maximize the performance. The details and results of this analysis are illustrated in Chapter 8.

Chapter 2

The LHCb Experiment

2.1 The Large Hadron Collider

The Large Hadron Collider (LHC) [1] is the main particle accelerator at CERN. Located on the border between Switzerland and France, the LHC is a 27 km long collider that operates in a tunnel at a depth between 50 and 175 meters. The LHC was designed to induce collisions between protons and between heavy ions, at a center-of-mass energy of up to 14 TeV and with a luminosity of up to $10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. Collisions occur at four specific points at which experiments are located. The LHC hosts seven experiments: ATLAS [2], CMS [3], LHCb [4], ALICE [5], TOTEM [6], LHCf [7] and MoEDAL [8].

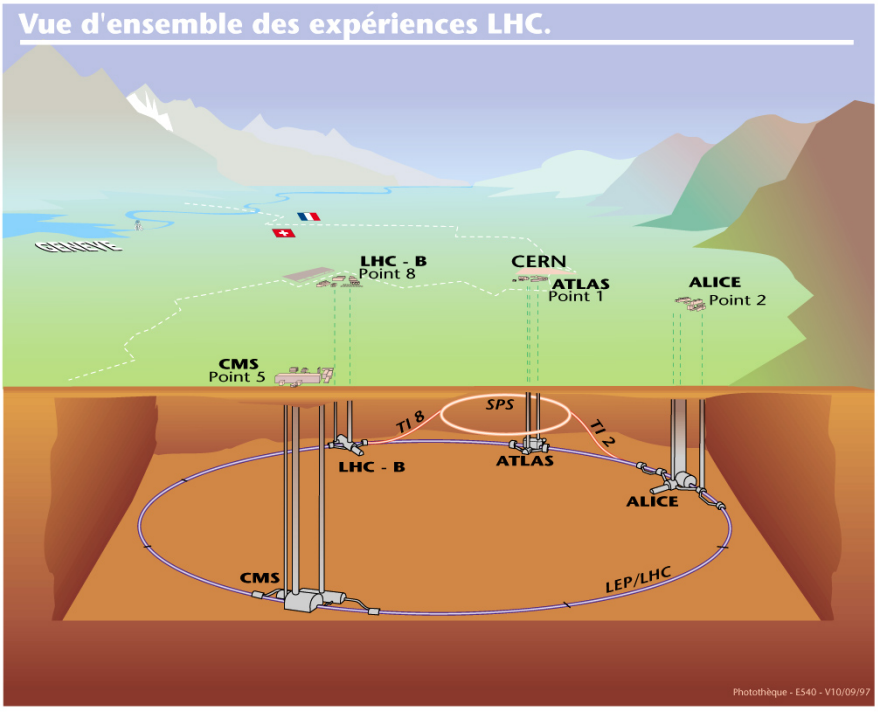


Figure 2.1: Representation of the LHC tunnel and its four main experiments.

The objective of the LHC is to provide particle collisions with sufficient energy and lu-

minosity to enable the experiments to investigate fundamental questions of nature that have not yet been understood. Among these are the theory of supersymmetry, the nature of dark matter, the violation of symmetry between matter and anti-matter, and the existence of extra dimensions.

The Experiments

The ATLAS (A Toroidal LHC Apparatus) and CMS (Compact Muon Solenoid) experiments are general-purpose experiments. They discovered the Higgs Boson in 2012 by analyzing the data accumulated in the first *run* of the LHC and are currently searching for phenomena inconsistent with the Standard Model.

The main objective of the ALICE experiment (A Large Ion Collider Experiment) is to study the physics of a state of matter called the quark-gluon plasma, in which the universe is believed to have been right after the *Big Bang*, and that the LHC tries to reproduce by colliding lead ions. ALICE also studies proton-proton collisions.

The LHCb experiment (Large Hadron Collider Beauty) is specially designed to study the violation of charge-parity symmetry (CP) in the interactions of B hadrons. Such a study could help explain the asymmetry of matter and anti-matter found in the Universe. The LHCb experiment is discussed in more detail in Section 2.2.

The TOTEM experiment (TOTal Elastic and diffractive cross-section Measurement) aims to study the proton structure through measurements of the cross-section of proton-proton interactions and the study of diffractive processes.

The LHCf (LHC forward) experiment performs a precise measurement of neutral particles originating from collisions in a direction very close to the beam to better understand the interaction of cosmic rays with the Earth's atmosphere.

The MoEDAL experiment (the Monopole and Exotics Detector At the LHC) is dedicated to the search for very specific, stable, and exotic particles, such as magnetic monopoles, that could arise in proton-proton collisions.

The Collider

The LHC consists of two parallel tubes (See Figure 2.2) in which proton beams circulate in opposite directions, close to the speed of light, with energy of up to 7 TeV. At four specific points, the two beams intersect, causing collisions with a total center-of-mass energy of up to 14 TeV. Each beam consists of 2808 *bunches*. The formation and acceleration of the bunches are performed by the radio frequency cavities, whose alternating frequency is synchronized with the bunches in order to always provide acceleration in the region where the bunches should be and an opposite force in the intermediate regions, in this way squeezing protons toward the central region of the bunches.

Along the circular tube of the LHC, bunches are kept on a nearly circular trajectory by different magnetic modules. The magnetic dipoles are responsible for maintaining the beam curvature. Each dipole magnet consists of a 15-meter superconducting magnet,

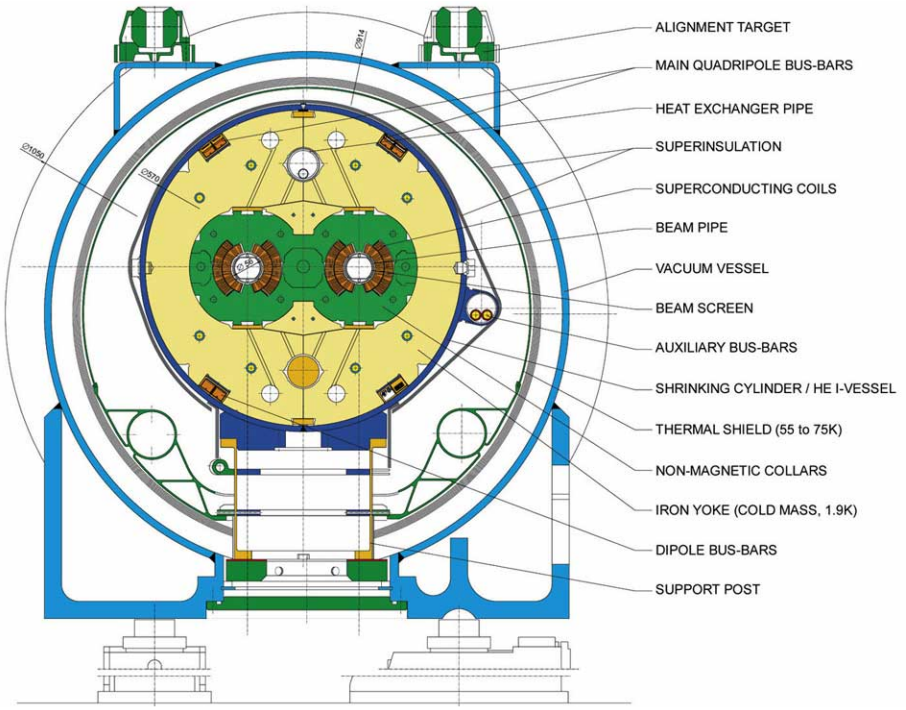


Figure 2.2: Transverse cut of the frontal plane of a dipole module.

providing an 8.33 T magnetic field. To supply this field, the magnet draws a current of 13,000 A and is maintained at a temperature of 1.9 K by the cryogenic system. More than 60 % of the LHC extension consists of dipoles (1232 in total). Figure 2.2 shows the cross-section of a dipole module. The module responsible for the focus and collimation of the beam is the quadripole module.

The Accelerator Chain

The proton beams circulating in the LHC are initially extracted from a cylinder of hydrogen gas. The protons are separated using a so-called Duoplasmatron, which ionizes the gas through the application of a strong electric field. The protons leave the Duoplasmatron at a speed of 1.4 % of the speed of light and are sent to a radio frequency quadrupole, which focuses the beam and accelerates it up to 750 KeV. This beam is injected in a linear accelerator called LINAC2, in which the protons are accelerated to an energy of 50 MeV and then injected in the Proton Synchrotron Booster (PSB), a 4-ring circular accelerator, accelerating the protons up to 1.4 GeV. From the PSB, the beam is inserted in the Proton Synchrotron (PS), which forms the beam into 81 spaced bunches and accelerates them up to 26 GeV. The bunches are then injected into the Super Proton Synchrotron (SPS), which accelerates them to 450 GeV. Protons take less than 18 seconds from the source to the SPS output and are then injected into the LHC. This injection occurs at two different points, into two tubes of the LHC, in opposite directions. At the LHC, the beams can be accelerated up to 7 TeV. Until now, the LHC operated up to 6.8 TeV per beam. Figure 2.3 illustrates the acceleration steps described, showing on the left the energy reached by the beam at the end of each step.

After accelerating the bunches to the desired energy, the ones of opposite directions then

start to collide at the four crossing points previously mentioned.

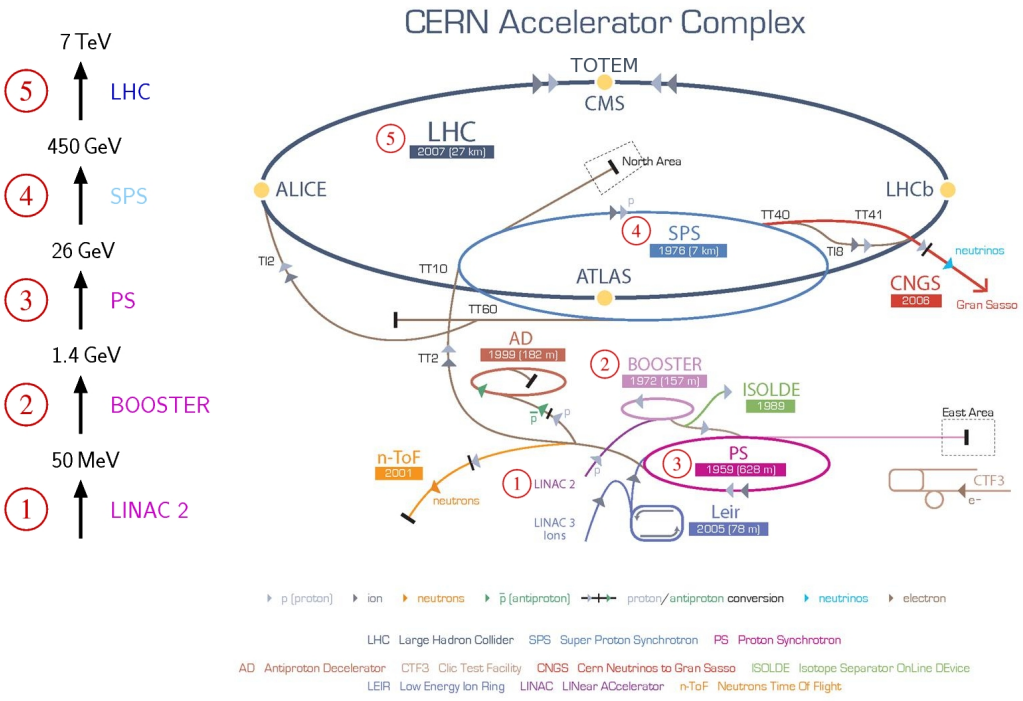


Figure 2.3: LHC acceleration chain.

2.2 The LHCb Detector

The Large Hadron Collider beauty (LHCb) experiment is a single-arm spectrometer that covers the angular region from 10 mrad to 300 mrad in the horizontal plane and 10 mrad to 250 mrad in the vertical plane. The LHCb detector is located at the site known as "Point 8" of the LHC in an experimental hall excavated around one of the four collision points. It consists of a set of detectors and elements that allow the identification of particles and the reconstruction of their trajectory and momentum. In this section, it will be described the layout of the previous LHCb detector, while the upgraded detector, which is the focus of this dissertation, will be presented in the next chapter. The layout of the old LHCb experiment is shown in Figure 2.4. From the interaction point, one can distinguish the LHCb subdetectors: the Vertex Locator (VeLo), the Tracking System, the Muons System, the Magnet, the Hadronic and Electromagnetic Calorimeters, and the Ring-Imaging Cherenkov (RICH) detectors.

The objective of the LHCb is to study rare decays of D and B hadrons, as well as the violation of charge-parity symmetries in these decays.

The LHCb detector is 21 m long, 10 m high, and 13 m wide. It has a forward geometry due to the fact that the production of light hadrons occurs preferentially forward and backward. To make optimal use of the existing experimental cavern, the experiment was set up in only one direction. The typical acceptance of the LHCb detector is $1.6 < \eta < 4.9$ [4].

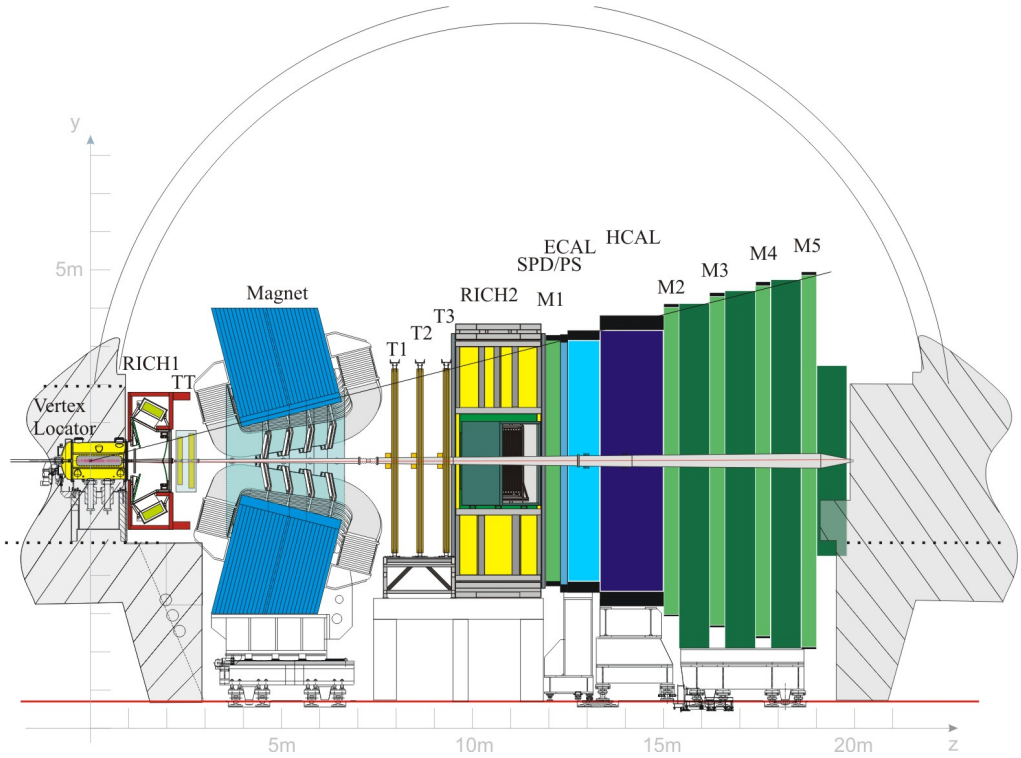


Figure 2.4: Side view of the LHCb detector.

Tracking System

The Tracking System is the system responsible for providing the data necessary to reconstruct the trajectory of charged particles that pass through the experiment. In combination with the LHCb magnet, it provides a highly accurate estimate of the momenta of charged particles. The trajectory information from the tracking system also helps the RICH detector to seed its particle identification algorithm.

The Tracking System comprises the VeLo detector, the Tracking Stations, and the Magnet. As shown in Figure 2.4, there are four tracking stations in the system: TT (Trigger Tracker), and the tracking stations T (T1, T2, and T3), in turn subdivided into Inner Tracker (IT) and Outer Tracker (OT).

The tracking stations consisted of layers of detectors arranged vertically or slightly inclined for pattern recognition purposes at an angle of $\pm 5°$. This was done to optimize the spatial resolution in the horizontal coordinate, at the detriment of the vertical resolution, since the horizontal plane is the bending plane of the magnet. This allows the precise measurement of the deviations of the particle trajectories caused by the magnetic field and, consequently, the precise measurement of the particle momentum and the decays invariant mass.

Trigger Tracker The TT station was located upstream of the magnet, and its main objective was to reconstruct low momentum particles that, when they enter the magnet region, are deflected out of the LHCb's acceptance and, therefore, do not reach the tracking stations downstream. Due to the proximity to the magnet, the existence of a magnetic

field in the TT region allowed (albeit less precise) momentum measurement. The layout of the TT detector is shown in Figure 2.5.

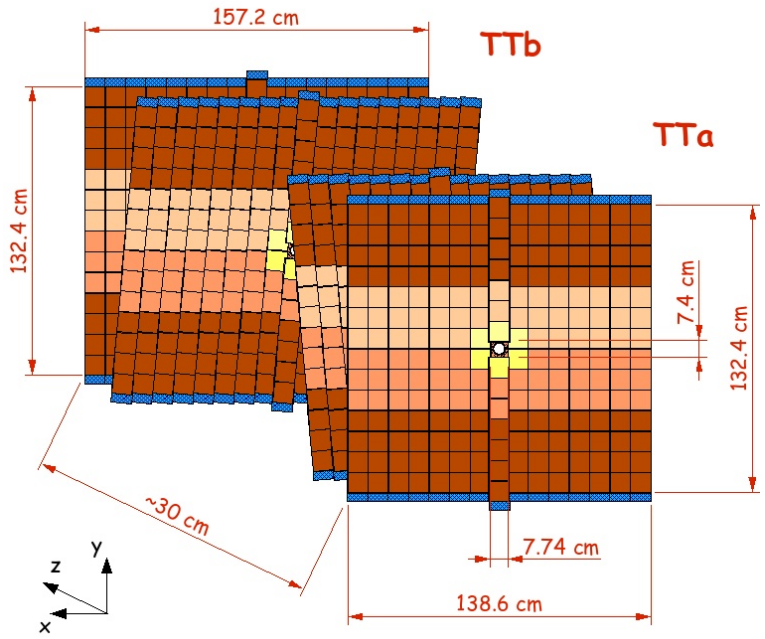


Figure 2.5: Layout of the TT micro-strips detector.

The TT had a spatial resolution of approximately 50 µm, and its detection planes consisted of silicon micro-strips 11 cm long, 7.8 cm wide, and 198 µm thick. The TT was comprised of 4 detection planes, of which the two outer ones had strips positioned vertically and the two inner ones had strips positioned at angles of $\pm 5°$, respectively.

Inner and Outer Tracker The stations T1, T2, and T3 were located between the magnet and the calorimeters. They were split into an Inner and Outer Tracker, as shown in Figure 2.6. The main objective of the IT and OT were to reconstruct the trajectory of the particles between the VeLo and the calorimeters, and to measure the momentum with the aid of the magnet.

The IT had the same format, resolution, and composition as the TT, and together they formed the so-called Silicon Tracker. The OT had a spatial resolution of approximately 200 µm in the horizontal coordinate and was composed of cylindrical gas drift tubes with a diameter of 5 mm and 0.075 mm in thickness. The total coverage of the OT reached a region of 600 cm × 490 cm while the IT covered the inner part around the beam pipe of approximately 120 cm × 40 cm.

The VErtex LOcator (VeLo)

The Vertex Locator (VeLo) is the first detector to interact with the collision products and is part of the Tracking System. It is mounted around the collision point and measures the particle of trajectories in the region close to the collision point. The measurements made

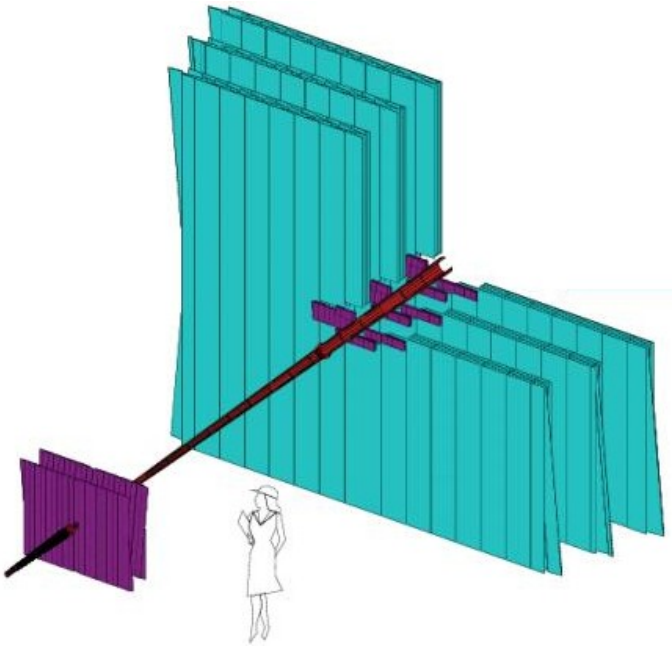


Figure 2.6: Position of the tracking stations. The TT can be seen in purple in the bottom left. On the right, in green, the three stations T1, T2, and T3 can be seen. The Inner Tracker is shown in purple at the center of the T stations.

by VeLo are used to reconstruct the trajectories and the production vertices (primary) and decay vertices (secondary) of c and b hadrons.

The VeLo was composed of 34 semi-disks installed around the collision region, as shown in Figure 2.7. The semi-disks were installed in a vacuum tank. The semi-disks were composed of silicon micro-strip detectors that provide a reconstruction of primary vertices with a resolution of about 40 µm in the z direction and 10 µm in the radial direction. For secondary vertices, the resolution was slightly worse and could vary between 150 µm and 300 µm depending on the number of hits used in the reconstruction.

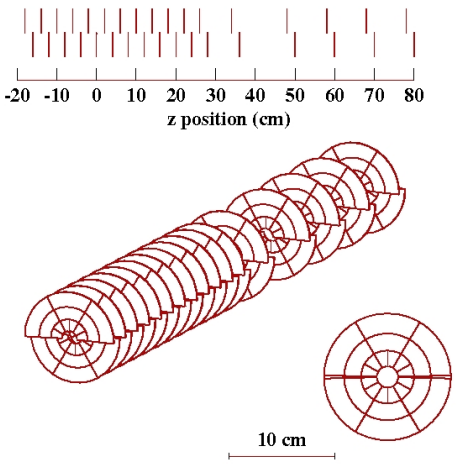


Figure 2.7: VeLo geometry. In total, there are 34 semi-disks installed around the region of the collision.

The RICH Detectors

The Ring Imaging Cherenkov (RICH) [9] is the detector responsible for identifying charged particles with momentum between 1 GeV/c and 150 GeV/c. Its operation principle is based on Cherenkov radiation, the radiation emitted by particles that have a speed higher than the phase velocity of light in the medium in which they propagate. This emitted radiation is captured after being reflected by mirrors inside the RICH, and its angles of emission and reflection are measured. With this information, one measures the velocity that the particle had when entering the detector and determines the particle mass, thus identifying the particle (pions, kaons, etc.).

The LHCb has 2 RICH detectors: RICH1 is located right after the VeLo and operates in the range of 1-65 GeV/c, while RICH2 is located just after the tracking stations and operates in the range of 15-150 GeV.

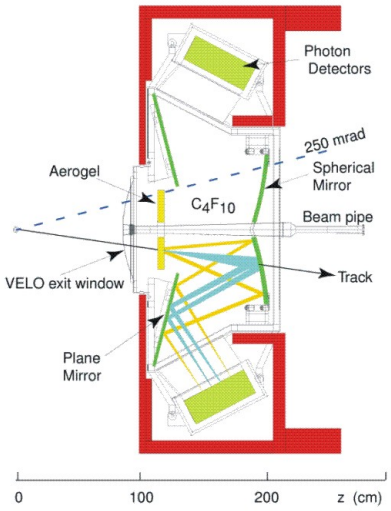


Figure 2.8: Side view of the detector RICH1.

Calorimeters

The calorimeters measure the energy deposited by particles that slow down and eventually stop traversing the calorimeter, with the main goal of identifying neutral hadrons, electrons, and photons. The LHCb has two types of calorimeters [10]: Hadronic (HCAL) and Electromagnetic (ECAL). The HCAL identifies particles whose main interaction is given by the strong force (hadronic showers), while ECAL identifies particles (electrons, positrons, and photons) that interact electromagnetically.

The calorimeter system previously included the Scintillating Pad Detector (SPD) and PreShower Detector (PS). The SPD (Scintillator Pad Detector) identified charged particles, that way allowing electrons to be distinguished from photons. The PreShower detector identified electromagnetic particles.

The final stage of particle identification occurs in the ECAL and HCAL. The first, in addition to electrons and photons, also helps to identify neutral pions while HCAL identifies

protons, neutrons, and charged pions. Figure 2.9 shows the distribution of cells in the LHCb calorimeters.

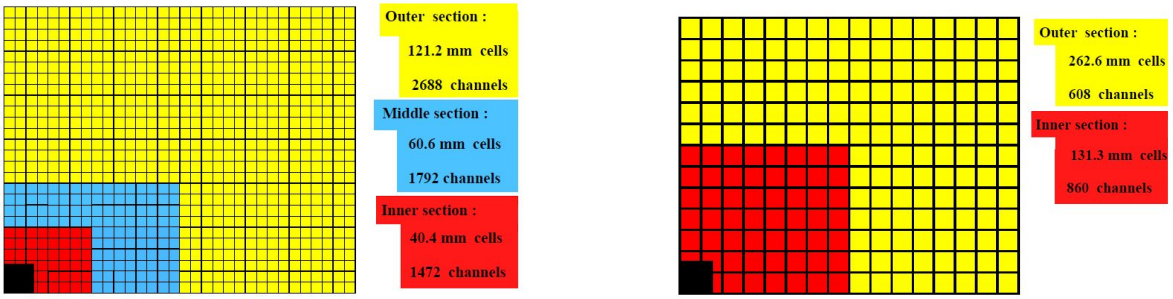


Figure 2.9: Illustration of the granularity in the SPD, PS, and ECAL (left) and in the HCAL (right).

The structure of the calorimeter varies according to its components. The SPD, PS, and ECAL consist of cells organized in the same structure. The size of each cell is 4×4 cm on the inside of the detector, 6×6 cm in the middle, and 12×12 cm on the outside. The SPD and PS use 15 mm thick scintillator plates connected to photomultipliers by optical fibers. ECAL alternates scintillation plates 4 mm thick and lead plates 2 mm thick. HCAL alternates 4 mm scintillating plates with iron plates 16 mm thick.

Muon System

The Muon System [11] is a set of detectors forming five muon stations (M1 to M5), measuring the trajectory of the muons that pass through the experiment. It is located at the end of the experiment as muons are very unlikely to be contained by the calorimeter. The muon system covers the entire acceptance of the experiment and has an area of approximately 435 m^2. Figure 2.10 shows the position of the five muon stations in the experiment. The M1 station is located in front of the PS, while the other four stations are located right after HCAL, separated by iron filters with the same dimensions. Altogether there are 1380 multi-wire proportional chambers, of 20 different sizes, installed according to the occupancy of particles in the different regions of the station.

Each muon chamber is composed of cells filled with a combination of 3 gases: carbon dioxide, argon, and tetrafluoromethane. When crossing a cell, the muon ionizes this combination of gases, and an electrode located in the center of the cell captures the electrons resulting from ionization, inducing a current signal that is processed by the detector's front-end electronics.

Trigger System

The LHCb's Trigger System is the system responsible for filtering out the events that will be stored in the LHCb's storage cluster.

The LHCb's detectors capture events at the same rate as the collisions: 40 MHz. The ultimate goal of the Trigger System is to reduce the event storage rate from 40 MHz to

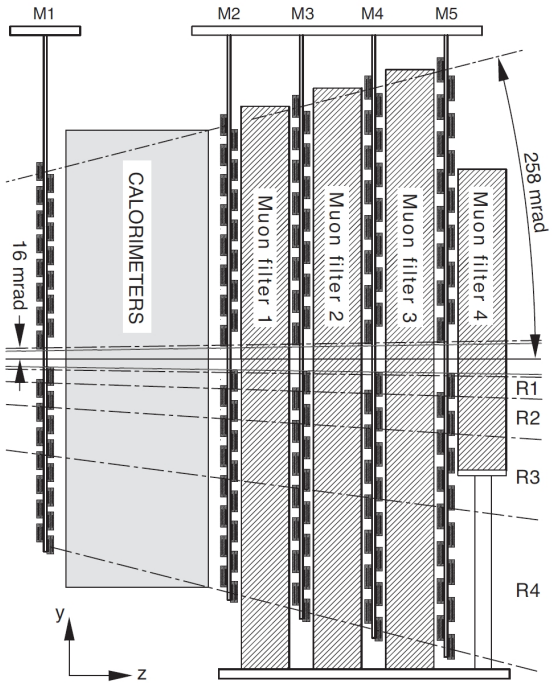


Figure 2.10: Side view of the Muon System.

the order of kHz. The system was divided into two parts: the Level-0 Trigger (L0) and the High-Level Trigger (HLT).

L0 The Level 0 Trigger was a trigger implemented at the hardware level. It received data directly from the detectors to decide whether or not to keep the event. The volume of data generated at each event is very large, and the electronics used to send and acquire data in the experiment did not have enough bandwidth to recover data from all the collisions. The objective of the L0 was to reduce the event rate to 1 MHz. The L0 is implemented in custom electronic modules located in the vicinity of the experiment.

The L0 Trigger received information from the calorimeters and the muon system. The criteria for the decision were based on the following parameters:

- Total energy deposited in hadronic and electromagnetic calorimeters.
- Transverse momentum of trajectories reconstructed in the muon stations.
- Particle identification by the calorimeters.
- Multiplicity in the calorimeters.

The cutoff values for each of these parameters were adjusted in order to guarantee the reduction of events to the maximum desired value of 1 MHz while keeping optimal efficiency in retaining the desired c- and b-decays.

HLT The High-Level Trigger decision is implemented in a CPU farm. Several algorithms run in the computers, which receive the data selected by the L0 trigger and perform a complete event reconstruction in order to identify the decays of interest. The HLT reduces the number of events sent to storage to a rate of a few kHz.

Control System

The LHCb Online System provides the backbone for taking data, as well as for controlling, configuring, and monitoring the entire experiment. It comprises three components: DAQ (Data Acquisition), TFC (Trigger and Fast Control) and ECS (Experiment Control System).

DAQ The Data Acquisition System ensures the data flow from the detectors to the permanent storage location. The acquisition process was triggered by the Level 0 Trigger, and the DAQ is responsible for ensuring the transmission of all data to the HLT.

The front-end electronics data selected by the L0 were sent to custom electronic modules called TELL1s, and by these further routed through a gigabit ethernet network to the HLT farm, after some internal processing. The data received by the TELL1 were processed in four FPGAs, by applying algorithms for data compression, zero suppression, and others; the data were then collected by a fifth FPGA (SyncLink) and formatted in a raw IP packet, and routed to an event builder network through a four-channel Gigabit-Ethernet module. The simplified block diagram is shown in the Figure 2.11.

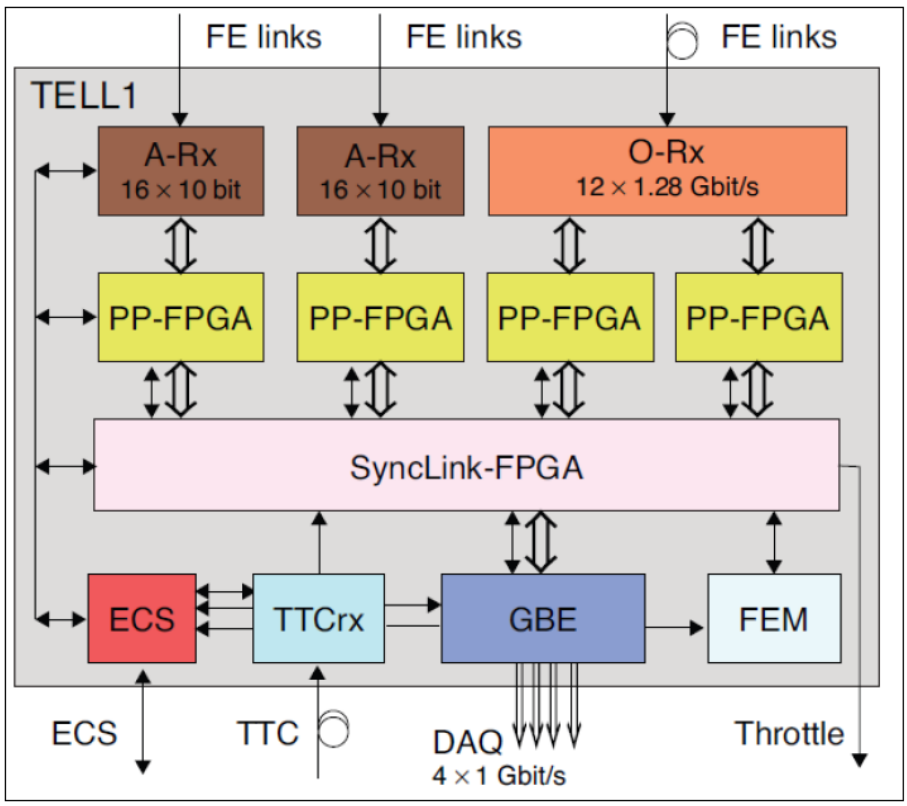


Figure 2.11: Simplified block diagram of the data acquisition module TELL1.

TFC The Trigger and Fast Control System (TFC) is the system responsible for distributing the LHC beam clock, the decisions of the L0 Trigger, synchronous reset signals, and other fast control commands. The TFC supervises all stages of data transmission from the front-end electronics to the back-end electronics.

ECS The Experiment Control System (ECS) is the system responsible for the configuration and control of the detectors, as well as for the integration of these with the Online System. The ECS uses several control protocols, including the Serial Protocol for ECS (SPECS), a proprietary protocol developed at CERN, the Controller Area Network (CAN) and Ethernet, and other custom protocols. The ECS software is based on the Siemens WinCC OA software.

The main systems controlled by the ECS are:

- Detector Operations
  - Gas Distribution System
  - High Voltage Modules
  - Power Modules
  - Temperature Control
- Trigger and DAQ Systems
  - Front-end electronics
  - Events Reconstruction
  - HLT trigger parameters
- Experiment's Infrastructure
  - Cooling and Ventilation System
  - Power Distribution
- Magnet Control, Accelerator Systems, Safety Systems, etc.

In Figure 2.12, one can find a simplified illustration of the control system and its control chain.

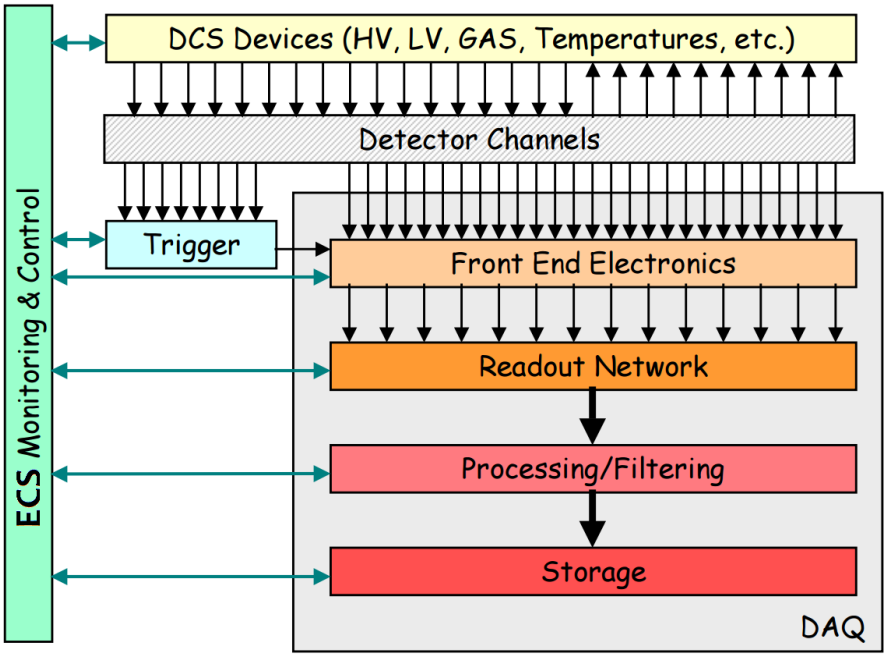


Figure 2.12: Diagram showing the different operations being controlled in parallel by the ECS system.

Chapter 3

The LHCb Upgrade

Figure 3.1 shows the LHC / HL-LHC operation schedule for the coming years. Although this schedule is experiencing delays due to the COVID-19 pandemic, the main features remain the same: data-taking periods alternate with Long Shutdowns (LS) for the collider and the experiments' upgrade. The LHC underwent a major upgrade during the LS2, aimed at increasing its luminosity, as shown in Figure 3.1. In a complementary way, the LHCb experiment will also be upgraded in order to adapt to the new higher luminosity provided by the LHC accelerator. LHCb has been upgraded during LS2 and is currently being commissioned.

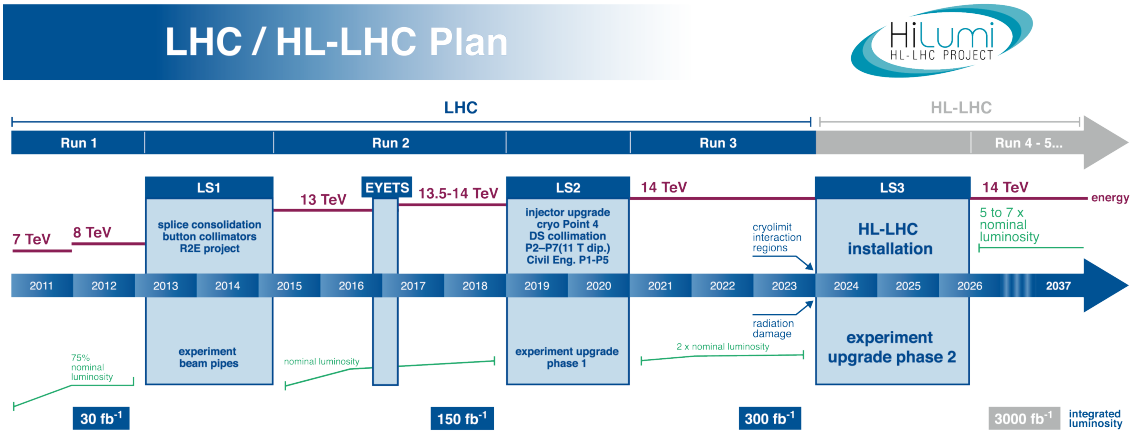


Figure 3.1: Operation schedule of the LHC for the following years.

With the upgrade, LHCb is expected to increase its physics yield for muon decays by a factor of 10 and for hadronic channels by a factor of 20. The data will be taken at an instantaneous luminosity of $2 \times 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. This corresponds to ten times the luminosity of previous data-taking periods and will allow LHCb to increase the volume of data annually collected from $1\,\mathrm{fb}^{-1}$ to up to $10\,\mathrm{fb}^{-1}$ [12, 13, 14].

The LHCb upgrade foresees the elimination of the Level 0 trigger and the data acquisition at a rate of 40 MHz. The increase in the luminosity will result in a significant increase in the level of incident radiation, in the detector occupancy level, and in the volume of data generated, among others. The radiation levels require the validation of all system

components and the use of appropriate electronics. The volume of data requires the replacement of all electronics for the data acquisition of the experiment and also of the front-end electronics (FEE) of the sub-detectors. Because the FEE is embedded in the detection modules, the VELO, IT, TT, and RICH detectors will also need to be replaced. Finally, the higher level of occupancy in the Outer Tracker requires a complete overhaul of its architecture.

3.1 Trigger and Data Acquisition

One of the biggest limitations of the current experiment is the need to reduce the L0 Trigger rate to 1.1 MHz. This reduction is achieved through the filtering parameters of the L0 Trigger. This is currently the main factor that prevents LHCb from taking advantage of the higher luminosity regime that the LHC upgrade will offer. For this reason, it was decided that the L0 Trigger would be completely eliminated, and the data acquisition of the experiment would be performed at the full rate of 40 MHz. All events will be transmitted to the back-end electronics, and the filtering of the events will be done entirely through software algorithms running on the HLT farm. A trigger implemented completely in software adds an enormous degree of flexibility, allowing the experiment to use a complex set of conditions to filter events previously limited by the physical connections of the L0 Trigger.

The elimination of the L0 Trigger and the 40 MHz readout requires the replacement of all data acquisition electronics in the experiment. Initially, the format designed for the data acquisition system was similar to the one currently used by the experiment, the main difference being the bandwidth required to support the much larger data volume. This format would consist of a motherboard in the ATCA standard (Advanced Telecommunications Computing Architecture), equipped with four mezzanine modules in the AMC standard (Advanced Mezzanine Card), named AMC40s. The AMC40 would be the interface between the front-end electronics and the computer network; it would be in the counting room and transmit the data to the network through Ethernet links, processing data at a rate of 108 Gbps. It could be used both as a data acquisition module as well as a control module.

Several AMC40 units were delivered assembled on a test platform called MiniDAQ, consisting of a module containing a CCPC (Credit Card PC) and an AMC connector. The MiniDAQ with the AMC40 was in the radiation tests of the SciFi Front-End Electronics described in Chapter 5.

Later on, a new format of the acquisition system was envisaged. An acquisition module called PCIe40 [15], followed by a second version called PCIe40v2; the second version of the board is often just called PCIe40 and will be referred to as such in this thesis. Such a board is being used in the experiment since LS2. It is similar to the AMC40, but it has a PCI express connector instead of an AMC connector and an FPGA with a larger capacity. It was developed to be embedded in the servers that will then receive the data via the PCI Express bus.

The PCIe40 is an electronic board that comprises an FPGA Arria 10, 48 bidirectional optical links, two SFP+ modules, and the PCI Express interface which allows it to be

connected to the server so that the data can be collected and slow control commands can be issued. Each PCIe40 can process a maximum data rate of 100 Gbps. Due to the fact that the PCIe40 will be installed in a region without incidence of radiation, it uses commercial optical transceiver modules called MiniPODs and instances of the GBT-FPGA firmware implemented in the Arria 10 FPGA. The image of the PCIe40 can be seen in Figure 3.2.

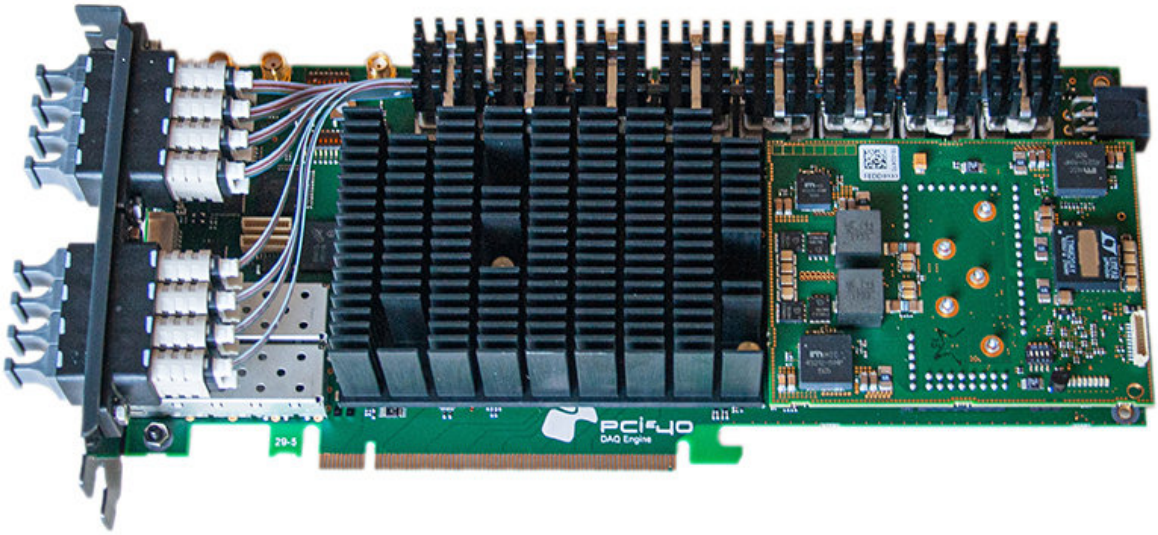


Figure 3.2: Final version of the control and data acquisition board PCIe40, used as DAQ modules for the whole LHCb experiment.

3.2 Tracking System Upgrade

The tracking stations T1, T2, and T3 of the Inner Tracker and Outer Tracker detectors were designed to provide their best performance at an instantaneous luminosity in the range of approximately $2 \times 10^{32}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. The LHCb after the upgrade will operate at a higher luminosity, up to $2 \times 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. The geometry of the previous tracking system was chosen so that the maximum occupancy in the hottest regions of the Outer Tracker was limited to 10 % at a luminosity of $2 \times 10^{32}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. This limitation is imposed by the efficiency of the pattern recognition in the track reconstruction algorithm. Improvements were made to this algorithm that allowed the Outer Tracker to operate with up to 25 % occupancy without significant loss in the efficiency of track reconstruction. However, the luminosity foreseen after LS2 required the change of the modules of the Outer Tracker in the regions of the highest luminosity. All front-end electronics would need to be replaced anyway to accommodate for the new 40 MHz readout.

Several possibilities were discussed, and it was finally decided to completely replace the Inner and Outer Tracker with a novel detector formed by scintillating fibers, read out by silicon photomultipliers (SiPM). The main requirements for this new detector are listed below:

- The detection efficiency should be higher than 99%.

- The spatial resolution in the magnetic field bending plane should be better than 100 µm.
- The readout electronics should operate at 40 MHz.
- The detector should be capable of maintaining the required performance for an integrated luminosity of at least 50 fb^{-1}.

The detector also has to fit the space occupied by the previous detector and must also tolerate the radiation levels expected for its operational life. According to the study carried out in Ref. [16], the peak luminosity regions are foreseen to receive ionizing doses of approximately 35 kGy and 25 kGy in stations T1 and T3, respectively. Figure 3.3 shows the total dose expected in the x-y plane, 783 cm away from the collision point (currently the 5° UV face of the OT T1 tracking station).

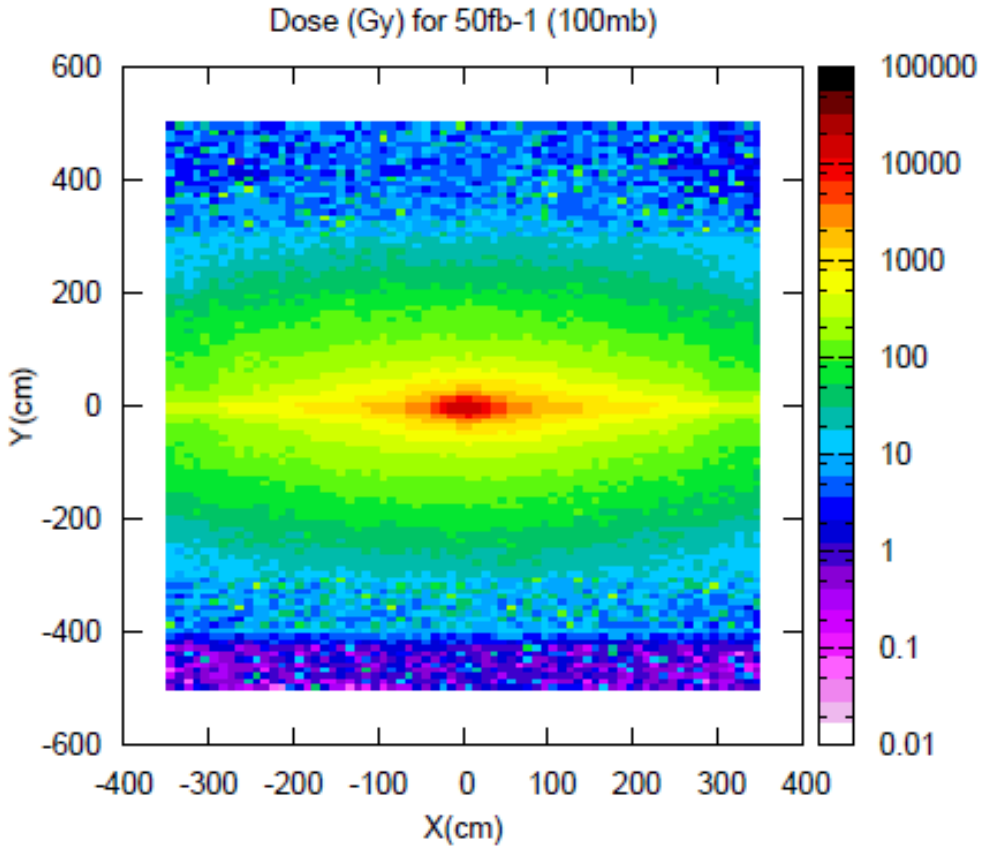


Figure 3.3: Expected integrated dose in the x-y plane, at a distance of z=783 cm from the interaction point, after an integrated luminosity of 50 fb^{-1}.

The Scintillating Fiber Tracker will be discussed in detail in Chapter 4 and is, at the of writing, installed and being commissioned.

Chapter 4

The Scintillating Fiber Detector

The Scintillating Fiber Detector (SciFi) is the main tracker in the LHCb spectrometer and is expected to measure charged-particle tracks with an efficiency greater than 99 % and a spatial resolution better than 70 µm (100 µm being sufficient to guarantee the desired momentum resolution). The task of the SciFi is to provide the hit information necessary to reconstruct the tracks and thus to determine the momentum of charged particles that pass through the detector; moreover, it will provide position information to the RICH particle identification system.

The SciFi detector will consist of three tracking stations between the magnet and the RICH2 detector. Each SciFi station will consist of 4 detection layers. The three stations are centered around the LHC beam pipe. The four detection layers of each station are located at a distance of about 20cm from each other and at angles of 0°, +5°, -5° and 0° with respect to the vertical axis, in the so-called "XUVX" arrangement. The circular hole in the center of the detector allows the active detection layers to come to a distance of about 20mm from the beam pipe.

The choice of the ±5° angles has been made to maximize the resolution in the direction in which the magnetic field deflects the particles, while sufficiently reducing the number of combinations in the hit pattern recognition, at the expense of the spatial resolution in the vertical direction.

Figure 4.1 shows a 3D model of the SciFi detector between the magnet and the RICH2.

Each detection layer is divided into 12 (for station T3) or 10 (for stations T1 and T2) scintillating fiber modules. Each module is 5m high and 0.52m wide. The complete detector covers an area 5 meters high and 6 meters wide in the X-Y plane. In total, the SciFi detector comprises 128 modules. Figure 4.2 shows how the modules are arranged.

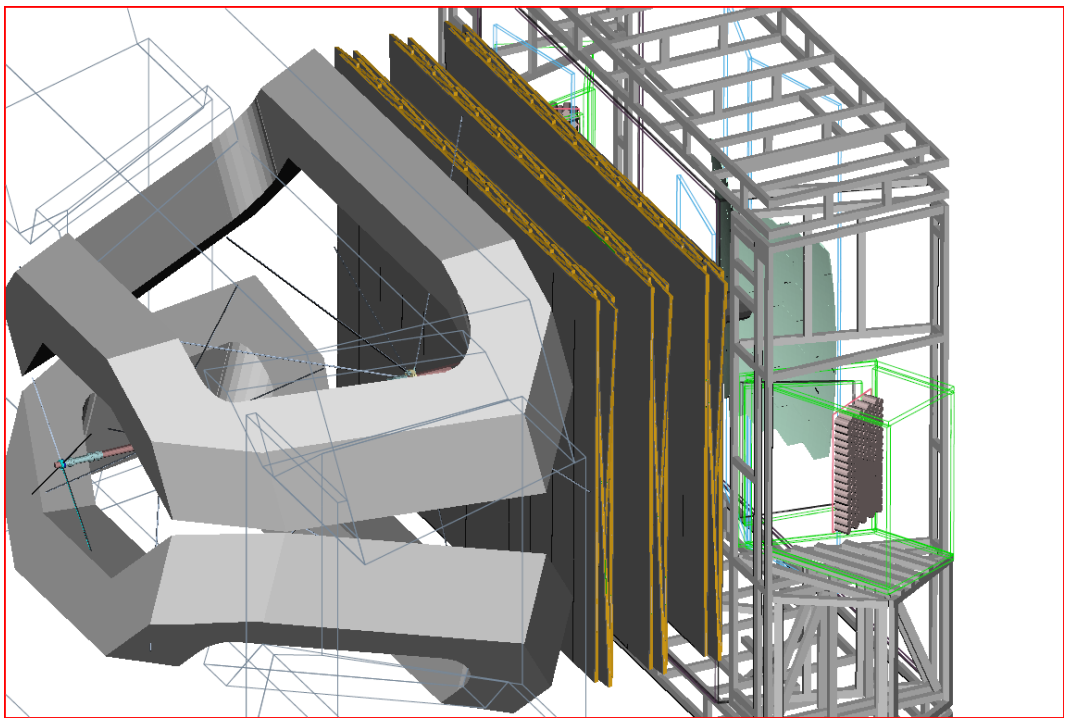


Figure 4.1: 3D model of the three SciFi tracking stations, between the Magnet (to the left) and the RICH2 detector (to the right).

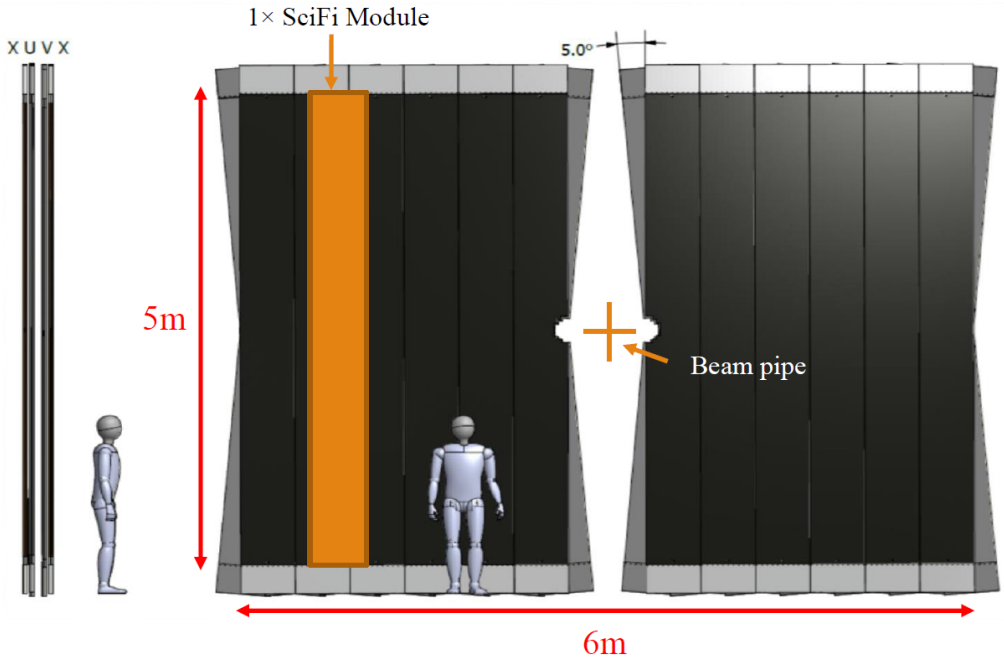


Figure 4.2: 3D model of the arrangement of the SciFi modules in station T3.

4.1 Scintillating Fibers

The scintillating fibers are the active detection element of the SciFi tracker. These are round fibers with a diameter of 250 µm. They have a polystyrene core coated with two layers whose refractive index is decreasing from the inside out. The fiber core is "scintil-

lating": charged particles when crossing this material, deposit energy in it, exciting the atoms contained therein and causing a reaction that leads to the emission of photons. The inner coating layer is made of polymethyl methacrylate (PMMA), and the outer layer is made of a fluorinated polymer. The design of the coating layers of the fibers is made to optimize the internal reflection of light, which is then guided toward the fiber end, where it is detected by a photon sensor.

Figure 4.3 shows an illustration of a SciFi fiber with its layers and the refractive index of each of them.

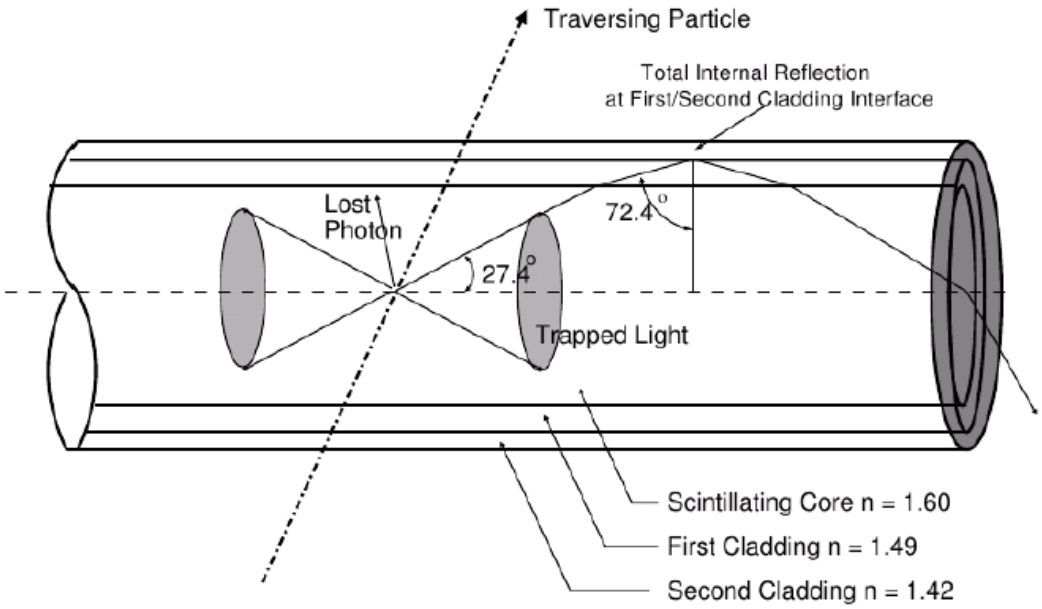


Figure 4.3: Photons are produced at the fiber's core by scintillation: part of them are then trapped inside the fiber, and travel along the fiber until they reach one of the ends.

4.2 Design

Each $5 \times 0.5\,\mathrm{m}^2$ detector module is subdivided into eight mats of 2.5 m long scintillating fibers. Four 13cm wide mats are placed side by side at the top of each module and four at the bottom. The modules closest to the beam pipe are modified to accommodate the LHC beam pipe at the center of the detector. The mats are 13 cm wide "packs" of fibers glued in a stacked way, forming six layers of fibers as seen in Figure 4.4.

At the outer ends of the fiber mats, a readout unit called Read-Out Box (ROB) is attached, which houses all the electronics that perform the readout of the detector. At the inner ends of the fiber mats, facing the center of the detector, there are mirrors so that the scintillating light reaching this end is reflected back toward the opposite end where the ROB is located.

A ROB consists of a "Cold Box", housing 16 Silicon Photomultipliers (SiPM), and a Front-End Box described in detail in Section 4.4. Figure 4.5 shows an exploded view of a ROB, indicating its different components.

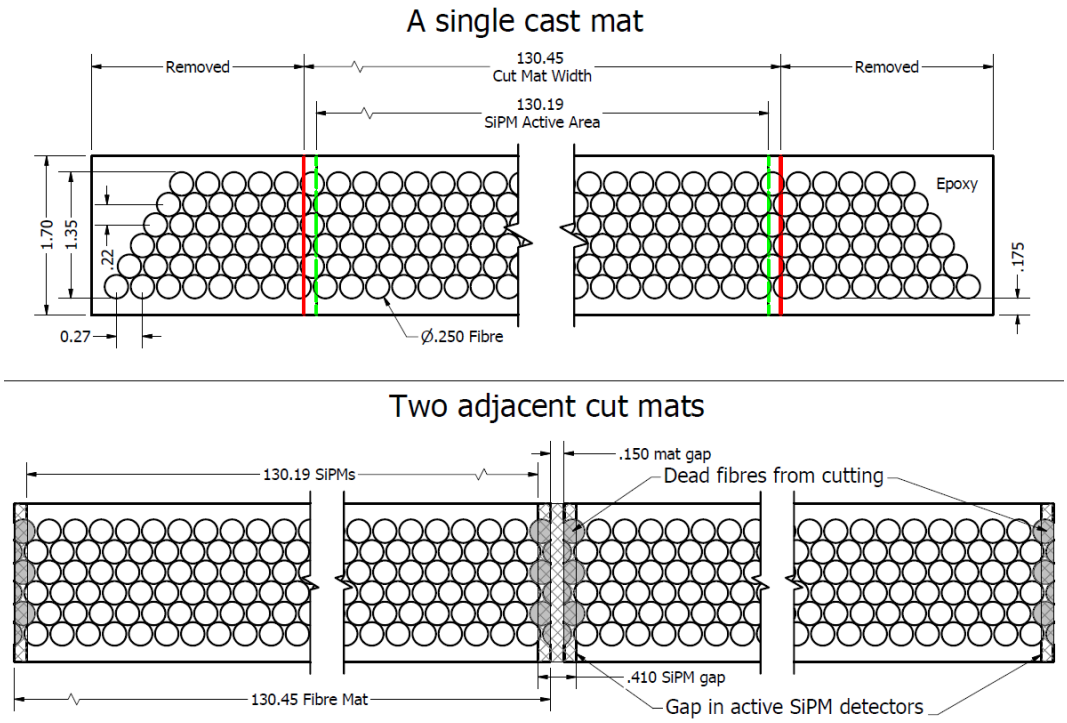


Figure 4.4: Transversal cross-section of the fibers of a single mat (top) and two adjacent mats (bottom).

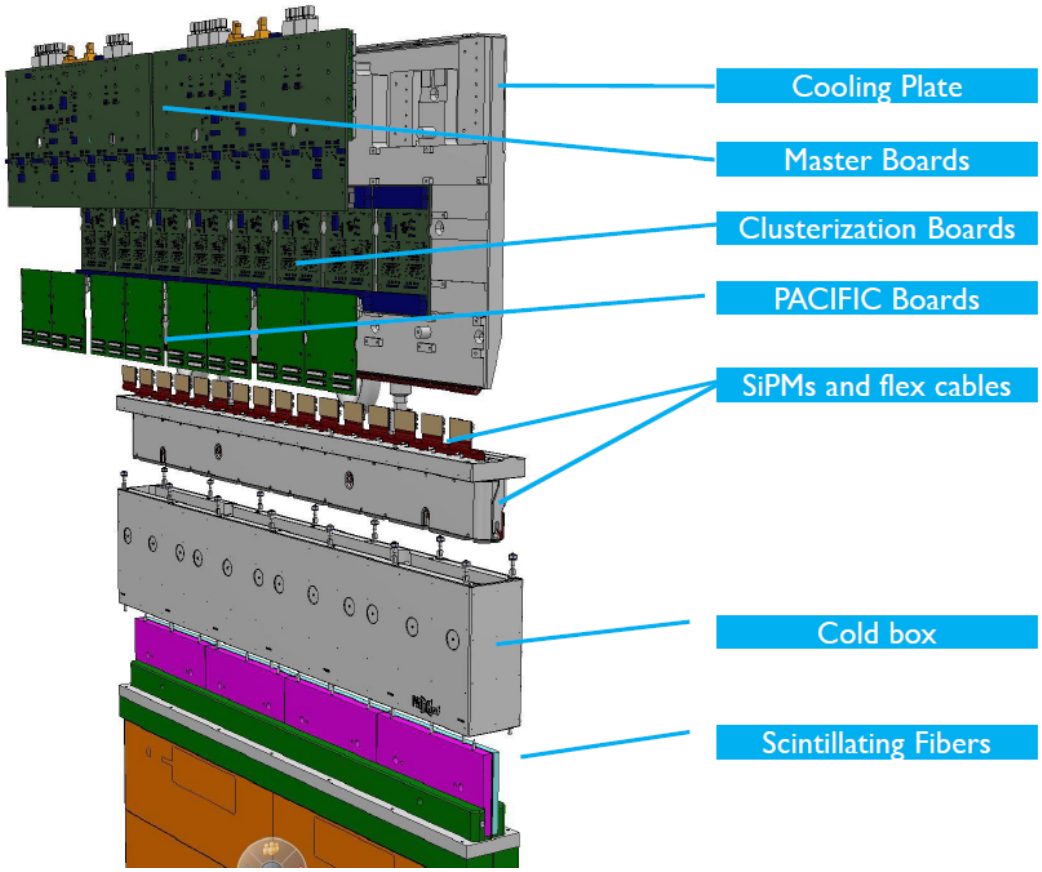


Figure 4.5: Exploded view of a read-out box.

The main functions of the Cold-Box are:

- the alignment of the SiPMs with respect to the fiber mats.
- the cooling of the SIPMs down to -40°C.
- the optical, electrical and thermal insulation of the SiPMs.

A tight optical coupling between the fibers and the SiPMs is of paramount importance for the detector performance: the SciFi design matches the SiPM and fiber mats geometry as to avoid fiber re-routing and "clear" fibers, but demands the distance between fibers and SiPMs to be no larger than 200 µm and their relative alignment in the transversal direction to be better than ±50 µm. Figure 4.6 shows conceptually how the coupling between the ROB and the fiber module is obtained.

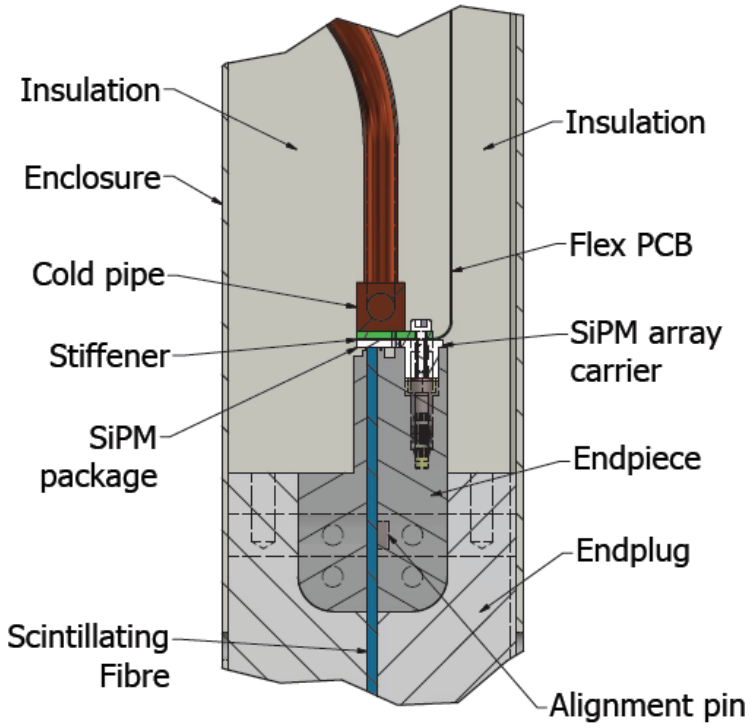


Figure 4.6: Side view cut of the junction of the fiber module with the read-out box.

The Cold Box is connected to a cooling system that keeps the SiPMs at -40°C. A 3D-printed titanium bar with three parallel channels runs through the ColdBox, split into four sections to accommodate for thermal expansion effects. The SiPM arrays have been glued to this cooling bar by a custom-made pick-and-place machine capable of positioning the SiPMs with a precision of ±20 µm. The coolant has been chosen to be NOVEC649, a drop-in replacement of the perfluorocarbons traditionally used in LHC detectors, but with very low global warming potential. Figure 4.7 shows a set of 16 SiPMs glued to the cooling bar.

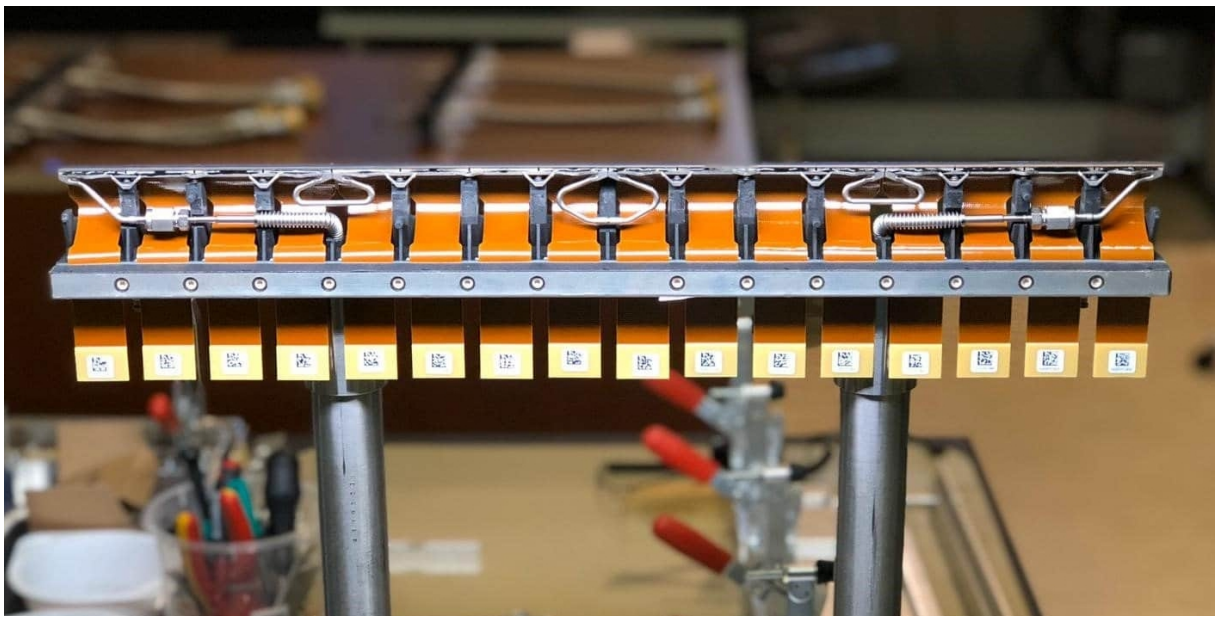


Figure 4.7: A set of 16 SiPMs glued to the cooling bar.

4.3 Silicon Photo-multipliers

The Silicon Photomultipliers (SiPM) are solid-state devices used to detect photons. SiPMs are used to detect light from the fibers in the SciFi detector.

The SiPMs are formed by an array of avalanche photodiodes on a silicon substrate. Current technologies allow the creation of avalanche photodiodes with sizes up to 20 µm, allowing a matrix to have up to 1000 photodiodes in a square millimeter. The diodes operate in Geiger mode and are connected in parallel to form the SiPM output. Each diode corresponds to a pixel, and the avalanche effect can be triggered by a single photon. When triggered, the diode saturates and generates a fixed and measurable current level. This level does not depend on the number of photons reaching the same pixel, since a single photon is enough to saturate the diode. The current measured at the SiPM output will be proportional to the number of diodes that have been saturated, that is, the number of illuminated pixels, that is, the number of photons (as long as the probability of multiple photons hitting the same pixel is sufficiently low).

The SiPM model chosen to be used in the SciFi presents a photodetection efficiency of approximately 40 % and a gain of approximately 10^6 e/γ. It has 128 channels, composed of 2 silicon wafers side by side, with 64 channels each. Each of the 128 channels is formed by a matrix of 4 by 26 pixels, totaling 104 pixels per channel. The pixels have dimensions of 57.5 µm × 62.5 µm.

As we can see in Figure 4.9, when a particle passes through the fibers, several SiPM pixels are expected to be illuminated. Although each channel is 250 µm wide (Figure 4.8), the number of pixels illuminated in the adjacent channels can be used as input to a clustering algorithm that can determine the hit position with a resolution better than the "binary" one of 250 µm/$\sqrt{12}$.

The efficiency of the SiPMs decreases proportionally to the total radiation dose received.

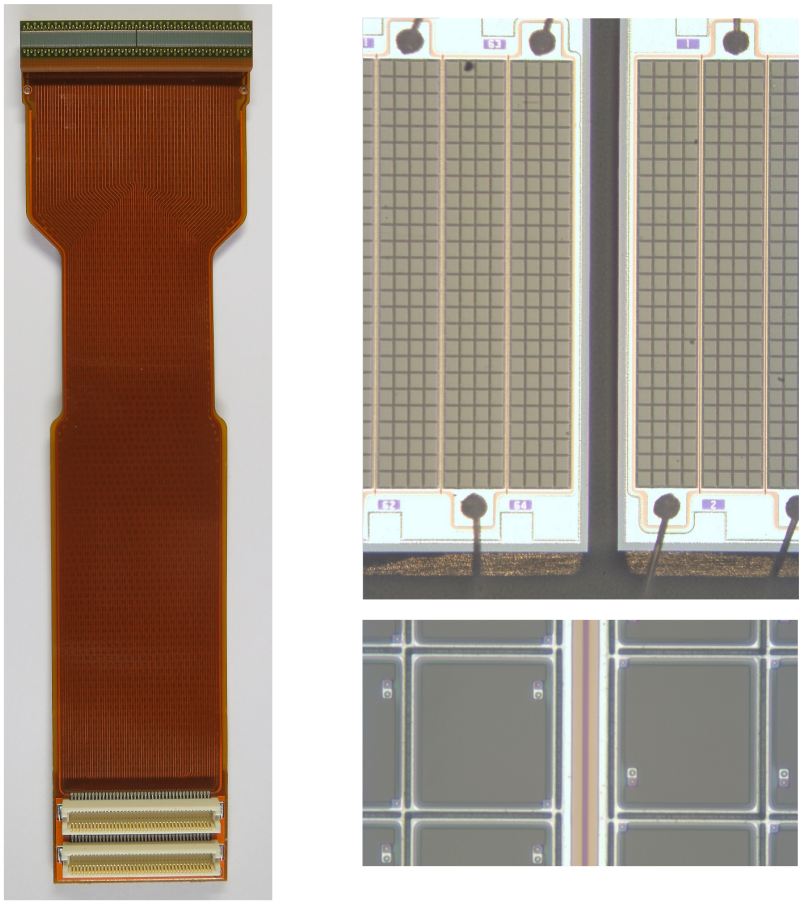


Figure 4.8: Left: flat cable with the SiPM PCB soldered on the top, containing two SiPM dies. Top right: enlarged image of the gap between the two SiPM dies. Bottom right: enlarged image of the gap between channels showing the individual pixels of the SiPM die.

This happens because the radiation damage caused to the silicon structure of the SiPM diodes increases the rate of pixels being fired when no photons trigger the diodes. Such a rate is called Dark Count Rate (DCR). When the DCR is high enough, the hits measured due to noise will cause a decrease in efficiency. The DCR, however, is reduced by a factor of 2 for every 10°C decrease in temperature. In the SciFi, cooling down the SiPMs to a temperature of -40°C will ensure that the efficiency requirements of the detector are met for the total radiation dose expected during its operation.

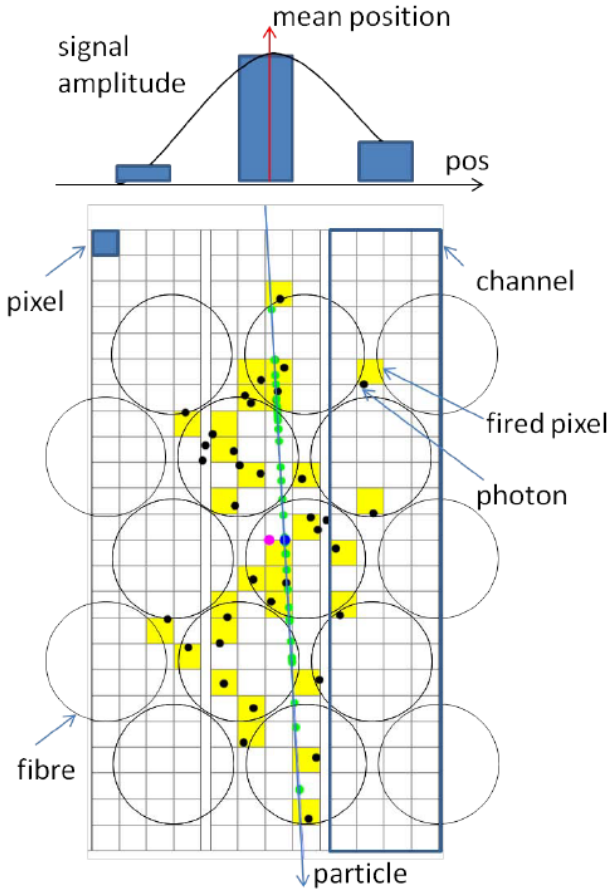


Figure 4.9: Photons produced by the scintillation of the fibers reach the pixels highlighted in yellow.

4.4 The Front-end Electronics

The front-end electronics (FEE) is the interface between the SiPM signals and the acquisition modules (the so-called Back-End Electronics). It is attached to the detector and is responsible for reading the detector signals and sending the data to the BE electronics. In addition, the FEE is also responsible for controlling and monitoring the detector, receiving commands, and sending the corresponding reports on the status of the components. As previously seen, the FEE will be contained in the ROBs at both ends of the fiber modules.

The SciFi FEE consists of three types of modules that perform readout, processing, and sending of the data:

- **Pacific Board:** reads, amplifies, and digitizes the data from the SiPMs.
- **Clusterization Board:** groups the data in clusters to be sent to the back end.
- **Master Board:** it is the module that controls the others, provides power, clock, and is responsible for serializing and sending the data.

The SciFi FEE makes use of the components of the Giga-Bit Transceiver Project (GBT) [17]. The front-end contains the GBTx and GBT SCA chips. The GBT SCA is used to communicate with all the devices on the front end through I2C, SPI, and JTAG protocols. The GBTx is responsible for serializing and sending the data. More details are given in Section 4.4.4

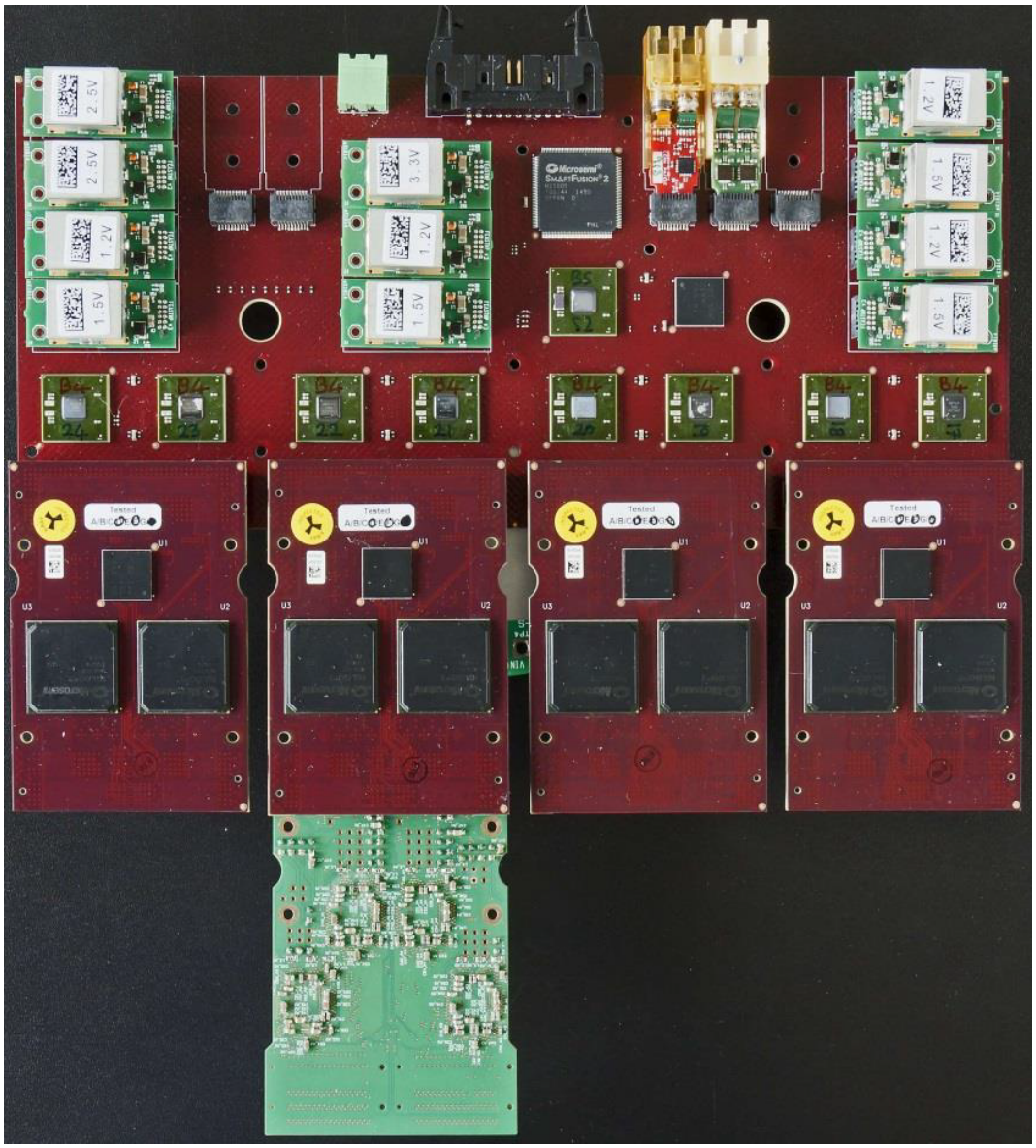


Figure 4.10: Prototype of the front-end boards that comprise the FEE. The large one at the top is the Master Board with 11 DC-DC converters and two optical modules plugged in. In the center, connected to the Master Board, there are four Clusterization Boards. At the bottom, there is a single PACIFIC board connected to the Clusterization Boards.

4.4.1 The Data Path

The data path in the Front-End electronics can be summarized as follows: when light reaches the pixels of the SiPM, electrical pulses are generated. The PACIFIC chip reads such pulses, amplifies, digitizes, and outputs a digital signal of two bits representing the

threshold level of the pulse (see section 4.4.2). The clusterization FPGAs receive the 2-bit signals of all 128 SiPM channels, create the clusters, and format them in data frames that fit the GBT protocol. The data is then passed to the GBTx chips on the Masterboard, which serializes the data and sends it to the BE electronics through the VTTx modules. This data path is shown in the block diagram of Fig. 4.11 for a half FE Box.

The data coming from each of the eight clusterization FPGAs follows the path to the corresponding GBTx chip on the Masterboard, indicated as "DATA GBT" in Fig. 4.11. Each of the four VTTx modules receives the data from two of the eight data GBTs. Numbering them from left to right, they are arranged in the following way: GBTs 1 and 3 transmit through VTTx 1, GBTs 2 and 4 transmit through VTTx 2, GBTs 5 and 7 transmit through VTTx 3, and GBTs 6 and 8 transmit through VTTx 4.

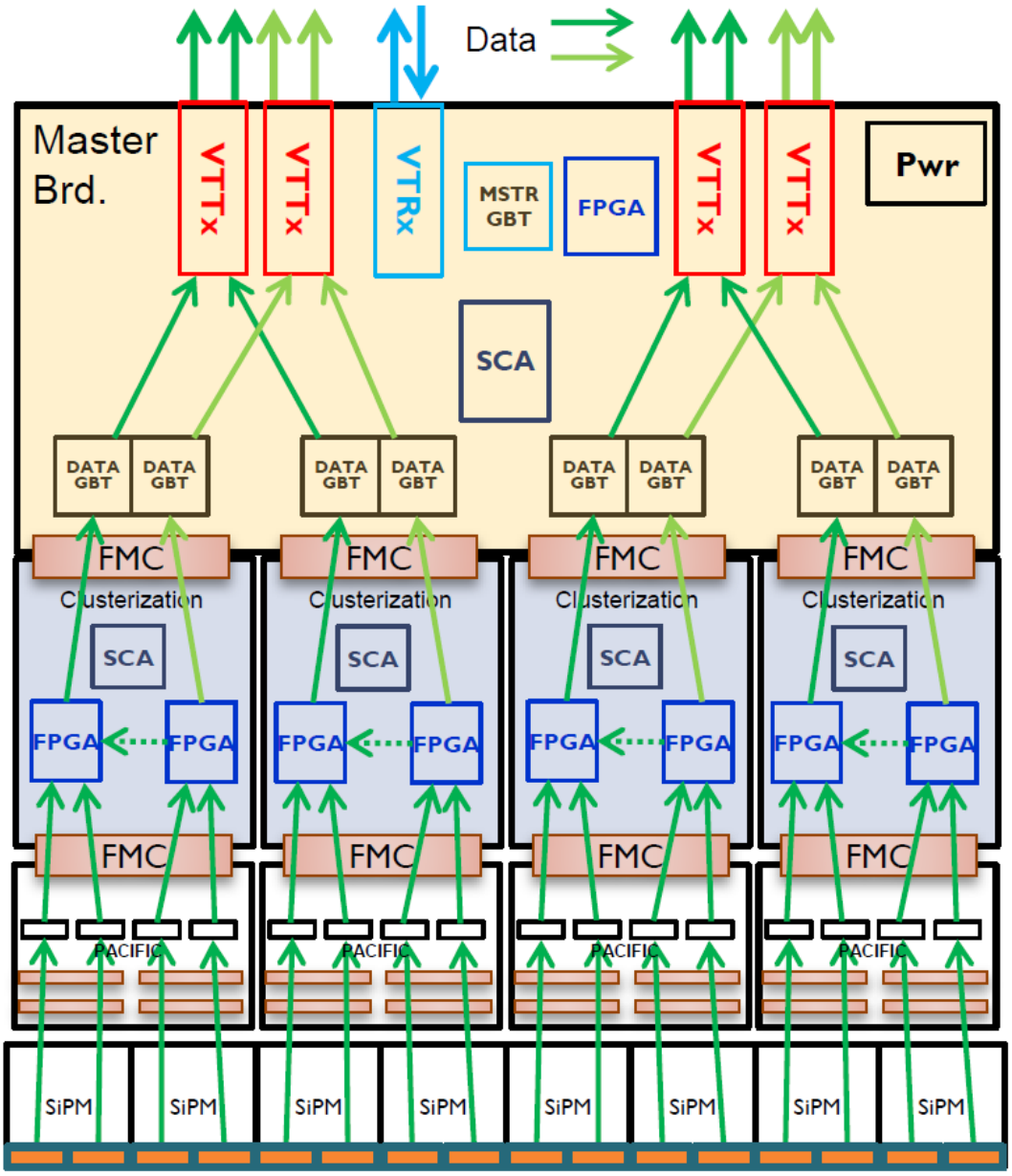


Figure 4.11: Block diagram of the SciFi FEE, displaying the data path from each SiPM to the corresponding VTTx module for a half FE Box.

4.4.2 The PACIFIC Chip

The Pacific Board is the module that reads the SiPM signals. It contains 4 "*low-**P**ower **A**SIC for the s**CI**ntillating **FI**bres tra**C**ker*" (PACIFIC) [18] chips.

PACIFIC is an application-specific integrated circuit (ASIC) with mixed signals, developed using 130 nm CMOS technology. It has 64 input channels, consuming less than 8 mW per channel and with a sampling rate of 40 MHz. The PACIFIC is controlled via I2C. It is responsible for amplifying and digitizing each SiPM signal, converting it into a 2-bit digital value.

In Figure 4.12, we can see the signal processing steps of the PACIFIC chip.

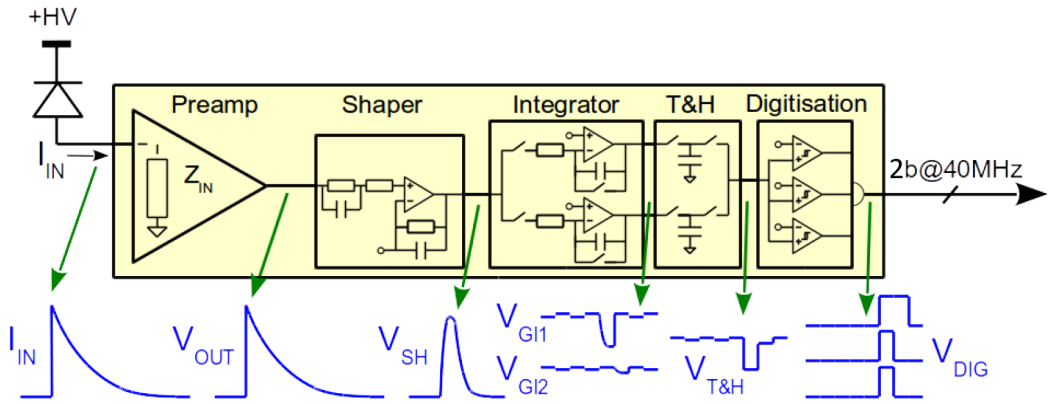


Figure 4.12: Signal processing chain of each read-out channel of the PACIFIC ship.

The stages in Figure 4.12 are explained below:

- The Preamp component is a transimpedance amplifier that converts the current signal from the SiPM into an identical voltage signal with a configurable gain. The gain can be chosen out of 4 options (0.5x, 1x, 1.5x, and 2x) through the I2C interface.
- The Shaper is a signal processing block that performs the cancellation of poles and zeros. The purpose is to remove the tail of the SiPM signal and cancel overshoots and undershoots. The values for the poles and zeros are also configurable through I2C.
- The Integrator is a block that uses the classic integrator architecture comprised of an operational amplifier with a resistor at the signal input for the negative input and a capacitor for the output. This block, however, uses two integrators interleaved according to the clock function, in order to avoid dead time. While one is active, the other is discharging and vice versa.
- The T&H block is a "Track and Hold", a circuit that samples the signal value and keeps it for a while, so that the signal is stable during the conversion time of the analog-to-digital converter.
- The Digitisation block is responsible for digitizing the signal. The digitizer output is a 2-bit binary signal formed from the outputs of 3 comparators whose threshold signals can be configured via the I2C interface.

Each Pacific Board contains 4 PACIFIC chips, thus allowing the reading of 2 SiPMs of 128 channels per PACIFIC card. Each Master Board supports up to 4 Pacific Boards, so a Master Board receives the signal from 8 SiPMs, and consequently, each ROB reads 16 SiPMs. After the signal is digitalized, it is sent to the Clusterization Board.

4.4.3 Clusterization

The Clusterization Board is the module responsible for zero-suppressing the data coming from the PACIFIC chip and sending them to the back-end. The Clusterization Board has a GBT SCA and two Microsemi Igloo2 FPGAs that are referred to as " Clusterization FPGAs ". The function of the GBT SCA is to configure the Pacific Board FPGAs and the PACIFIC chips. The FPGAs are responsible for data processing, zero suppressing the data through a clustering algorithm. A Pacific Board will be connected to each Clusterization Board, and each Master Board will be connected to 4 Clusterization Boards.

The raw data read from the SiPM cannot be sent directly to the back-end, because the volume of data, $128 \times 2 \times 40$ Mbps, exceeds the 4.48 Gbps throughput of the GBTx ASIC. Most channels record no hits in most events since the number of particles produced in collisions is relatively small compared to the number of detector channels. This fact allows the use of a technique to deal with the excess of data, which is called Zero Suppression. It suppresses the channels without hits before sending the data. This saves bandwidth that would otherwise be wasted by sending zeros after zeros. The zero suppression needs to add to the data the addresses of the channels that were not suppressed and is practically only worth it if the occupancy is sufficiently low. In addition to removing the channels without hits, the FPGA also "clusters" hits, combining the information of neighboring channels, which leads to a further data reduction.

The clustering algorithm processes the detector's channel data in order to judge what will be considered a particle or what will be considered noise. The clusterization FPGA receives the data in a 2-bit format per channel. These 2 bits mean the value read by the PACIFIC as a function of the thresholds. There are three thresholds and, therefore, four possibilities (2 bits). The algorithm analyzes whether the total charge deposited in a given channel and in the adjacent ones (see Fig. 4.9) is enough to consider that the reading is a particle or even if the value corresponds to two or more particles that hit adjacent channels. Figure 4.13 shows an example of how this is achieved.

The leftmost peak in Figure 4.13 was considered a cluster. In the center, there is a reading that was zero suppressed because the sum of the channel and adjacent values did not add up enough to represent a particle. The rightmost structure was interpreted by the algorithm as two clusters because the number of channels reached was more than four and also because the total energy would be enough to consider it two. Obviously, the working of such an algorithm will depend on the values chosen from the three thresholds, called neighboring, seeding, and high thresholds, respectively. The FPGA packs each cluster into a 9-bit value: 7 bits correspond to the SiPM channel number and two to the position of the cluster center of mass within the channel length.

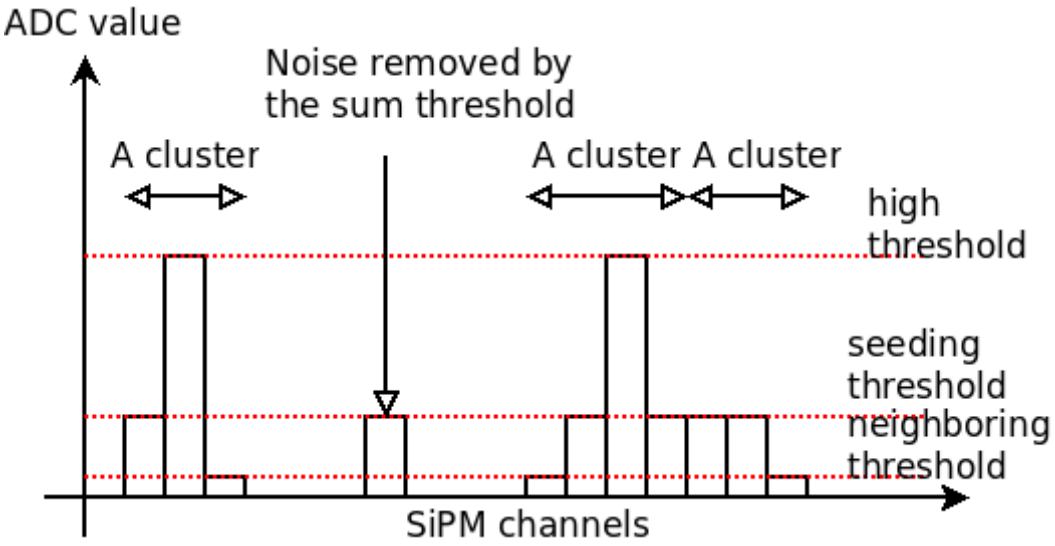


Figure 4.13: An example read of the SiPM channels showing which group of signals would be considered a cluster by the algorithm.

4.4.4 The GigaBit Transceiver

The GBT (GigaBit Transceiver) is a project with the objective of providing a set of radiation-resistant integrated circuits (a chipset) to allow the implementation of a 4.8Gbps optical bi-directional link for the LHC experiments upgrade. The link implemented by the GBT aims at transmitting data between the front-end and back-end electronics, simultaneously serving applications such as data acquisition, trigger, timing, and control. The GBT has its own communication protocol. In the GBT project, there is also a branch, the GBT-FPGA project, developing an HDL code that allows the direct interface between FPGAs and links based on the GBT project.

The GBT chipset comprises the following integrated circuits:

- The GBTx is the chip that serializes and deserializes the link data and implements the protocol layer of the link, for encoding and decoding the data.
- The GBT-SCA (GBT-Slow Control Adapter) provides the control interfaces. The GBT-SCA translates commands received through the GBT link into commands in common protocols used for control, such as I2C, SPI, and JTAG. Additionally, it has analog-to-digital and digital-to-analog converters that can be used for different purposes (voltage control and monitoring, temperature readings, etc.).
- The GBTIA is a transimpedance amplifier whose purpose is to amplify the current coming from a PIN diode. It operates at a rate of up to 5 Gbps and was developed to deal with the degradation of sensitivity of the PIN diode due to the incidence of radiation.
- The GBLD is the integrated circuit responsible for feeding the transmitter with data and modulating the current supplied to the laser to achieve the best electro-optical conversion.

In the upgrade of the LHCb experiment, GBT chips are used in the front-end electronics in order to meet the needs of high-speed, radiation-resistant optical links. In the back-end electronics, which is located in an area without incidence of LHC radiation, it is not necessary to use the GBT chips. In this case, a commercial optical module is used, and the GBT protocol layer is implemented in firmware running on an FPGA, to reduce the cost of producing application-specific integrated circuits. In Figure 4.14, we can see the functional diagram of a GBT link.

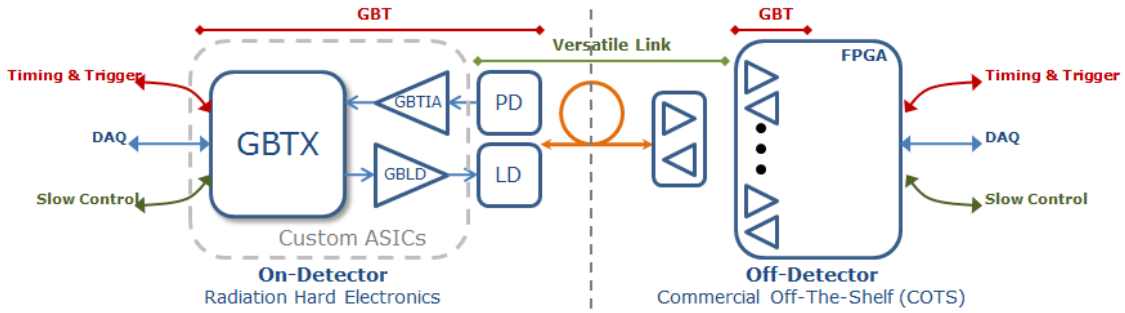


Figure 4.14: Diagram showing the functionality of the GBT project chipset.

The GBT was developed in order to minimize data loss through "Single Event Upsets" (SEU), which is a change in the logical state of a memory cell (a bit of a register, for example) due to the passage of an ionizing particle. The GBT chips are produced using 130 nm CMOS technology and adopt a triple redundancy architecture. This consists of creating two additional registers for each original register: whenever a difference among any of them arises, they will be rewritten with the predominant value among the three of them. Thus, in order to have data loss by SEUs, it would be necessary for a SEU to occur in the same bit of 2 different registers during the same interval of time of a clock cycle.

The GBTx chip retrieves the clock signal from the GBT link itself ("clock recovery"). This means that a shared clock signal is not required. They just need to be operating at the same frequency. An internal PLL of the receiving GBTx is calibrated according to the edges of the received data signal, and a clock for the deserializer is generated.

The GBT protocol consists of 120-bit words and provides three different configurations: GBT mode, WideBus mode, and 8B/10B mode. In 8B/10B mode, the 120 bits of the GBT word are simply divided into 12 words in 8B/10B decoding. In GBT mode, four bits are dedicated to a header that indicates whether the word refers to valid data or not, 4 bits are dedicated to control, 80 bits are dedicated to data, and 32 bits are reserved for a forward error correction algorithm. Figure 4.15 shows the word format in GBT mode. The WideBus Mode has the four-bit header and control bits as the GBT mode and it uses all the remaining 112 bits for data, giving up the forward error correction system.

The GBT encoder consists of 3 steps: scrambler, RS encoder, and interleaver. The block diagram of the encoder and decoder is shown in Figure 4.16. The RS Encoder (Reed-Solomon Encoder) is only used in GBT mode, and it is responsible for the forward error correction system. The scrambler is an algorithm that changes the bit values based on the previous words, and a seed word, whose objective is to prevent that, in a data sequence composed predominantly of '0' or '1', the average DC level of the transmission medium deviates too much from the middle level between Gnd and Vcc. This is a procedure used

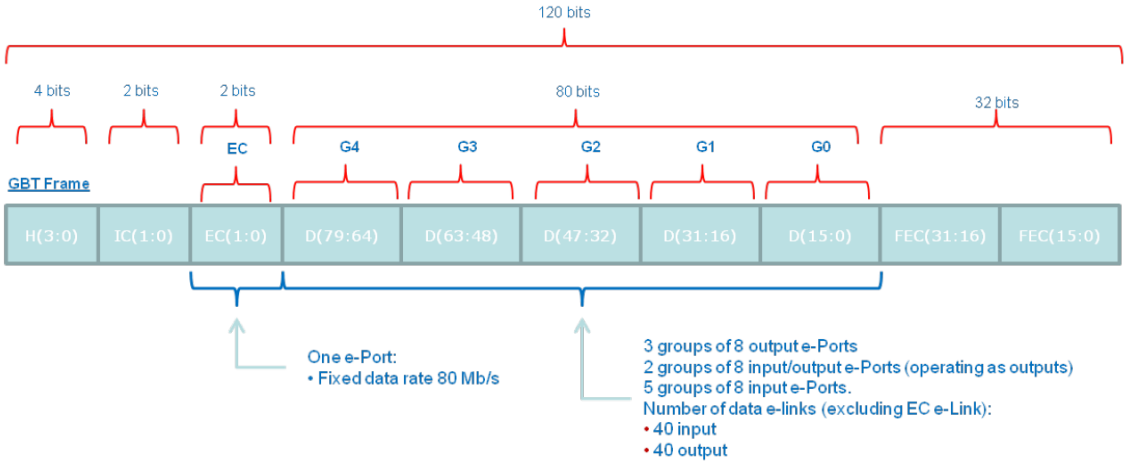


Figure 4.15: Structure of the 120-bit frame used in the GBT protocol.

in communication standards to prevent the full charge or discharge of a coupling capacitor that is in the data line, which in turn may cause a continuous high or low signal at the receiver. The interleaver is an algorithm that changes the position of specific bits in order to facilitate the reconstruction of the word when a sequential loss of bits occurs.

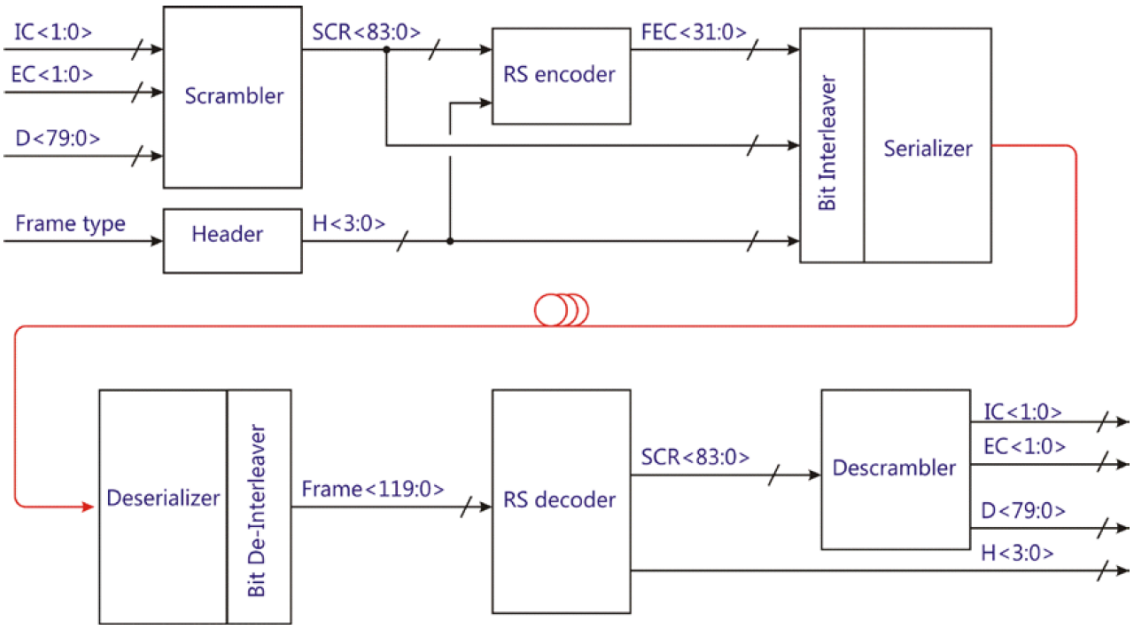


Figure 4.16: Block diagram of encoder and decoder of the GBT-FPGA.

4.4.5 Front-end Control and Monitoring

The FEE is controlled through a bidirectional GBT link, represented by the blue arrows in Fig. 4.17. Apart from the 8 GBTx chips used for data transmission, the master board has an additional GBTx chip named Master GBT (labeled as “MSTR GBT” in Fig. 4.17) connected to a VTRx module, which implements a GBT link dedicated to the control of the FEE. In the LHCb context, the control signals are divided into two categories, usually referred to as “slow control” and “fast control”.

The fast control refers to real-time control signals with strict timing requirements. Such signals need to have fixed latency and be distributed to all the FEE in every bunch-crossing. The GBTx can be configured to output certain bits of the received GBT word in various pins. In the SciFi FEE, such pins from the Master GBT are routed to the FPGAs in the master and clusterization boards. Those bits are interpreted by the FPGAs and can represent a variety of commands such as "FE reset," which sends a reset pulse to the FE devices, as "Non Zero Suppressed," which causes the clusterization FPGAs to send raw data instead of clusters, as "BX reset," which resets the bunch-crossing identifier counters on the FPGAs, as "Calib," that triggers the light injection system to flash LEDs embedded in the fiber modules, among others options. The full list of commands and their description can be found in [19]. These bits are generated centrally by LHCb's Timing and Fast Control (TFC) system and control the behavior of all the FEE of the detector at every bunch-crossing.

The slow control refers to commands with no strict timing constraints, mostly implemented via read and write access to registers on the FEE devices. These commands also arrive through the GBT word received by the Master GBT, however, they are assembled from a small number of bits from many GBT words received in sequence. These commands are transmitted through e-links to the GBT-SCAs, which can trigger read and/or write access to the different protocols implemented by the GBT-SCA. Those commands are originated from LHCb's Experiment Control System (ECS) and used to configure and monitor many parameters in the FEE.

Every configurable device in the FEE is connected via I2C to one of the 5 GBT SCA chips. The FPGAs also have an SPI and JTAG connection to their respective GBT SCA, which can be used to program new firmware into the FPGAs.

The paths for slow control in the FEE can be seen in Fig. 4.17. The yellow dotted lines represent the e-links, the purple dashed lines represent SPI and JTAG connections, whereas the green continuous lines represent I2C connections.

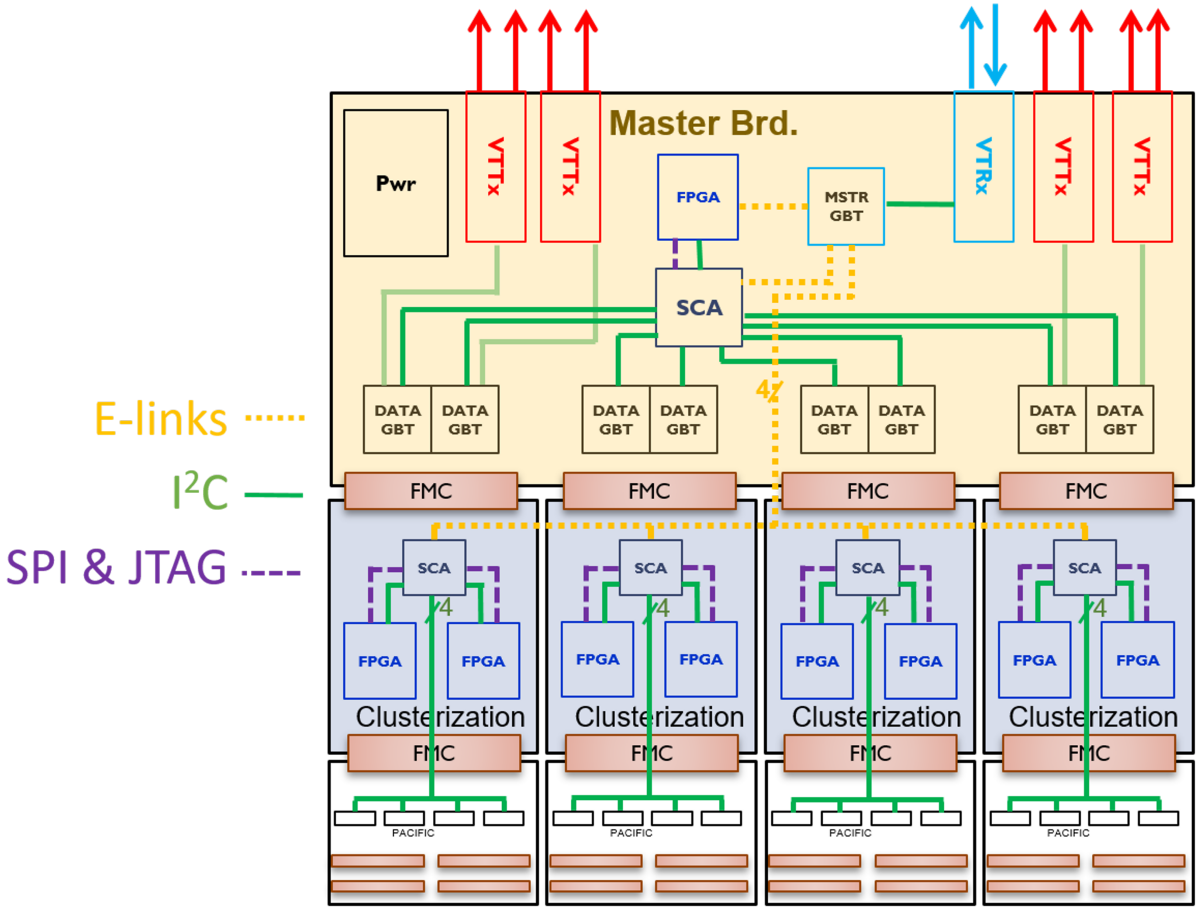


Figure 4.17: Block diagram of the SciFi FEE, displaying the control paths for I2C, SPI, JTAG, and GBT E-links.

4.5 Monte Carlo Simulation and Expected Performance

The expected performance of the SciFi detector was extensively studied with Monte Carlo simulations during the detector design phase [20]. Detector parameters were tuned on the base of a number of figures of merit, most importantly, the detector efficiency and resolution. The results of the simulations shown in this section are the obvious term of comparison for the test-beam results presented in this dissertation.

The expected performance in terms of efficiency and resolution has been obtained with the BOOLE software [21] containing a Monte Carlo simulation of the SciFi detector in the expected running conditions [20]. BOOLE simulates the generation and transport of photons inside the fibers as well as the digitization of the signal in the SiPMs and the FE Electronics from hits caused by particles with a variety of momentum.

Figure 4.18 shows (left) the cluster efficiency as a function of the momentum of the particle and (right) the distribution of the cluster efficiency per event for particles with $p > 5\,\mathrm{GeV/c}$. As one can see, a cluster efficiency above 98% is expected over a wide momentum range.

Figure 4.19 shows the cluster resolution for (left) lite clusters and (right) full clusters for

particles with $p > 2\,\mathrm{GeV/c}$. Lite clusters are proxies of the clusters actually encoded by the FE Electronics, while Full Clusters contain the full information from the clustering algorithm, part of which had to be sacrificed due to bandwidth limitations. The red curves denote the results of a double-Gaussian fit to the data. As can be seen from Figure 4.19, the expected lite cluster resolution (slightly worse than the full-cluster resolution due to the 12-bit fractional position furthermore reduced to one bit only) is 77 µm, thus well within the performance design limit of 100 µm.

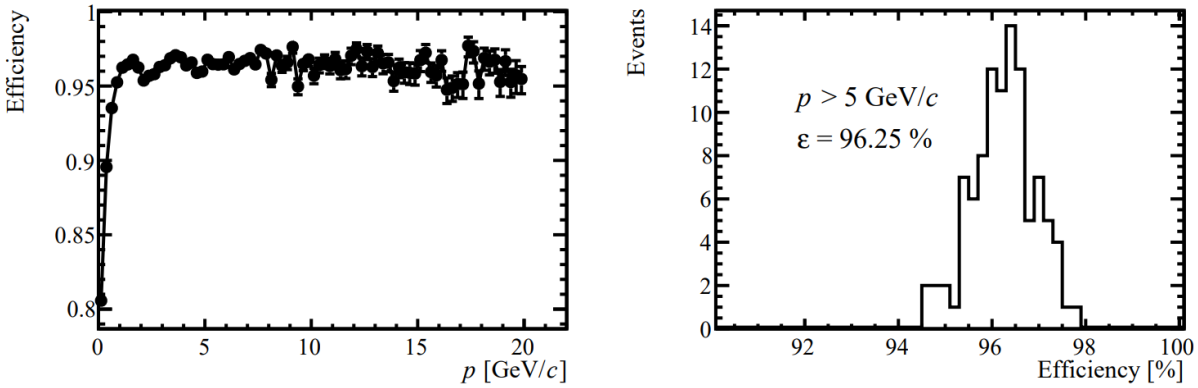


Figure 4.18: (left) Cluster efficiency as a function of the momentum of the particle and (right) distribution of the cluster efficiency per event for particles with $p > 5\,\mathrm{GeV/c}$. These plots have been obtained using only clusters of at most four channels.

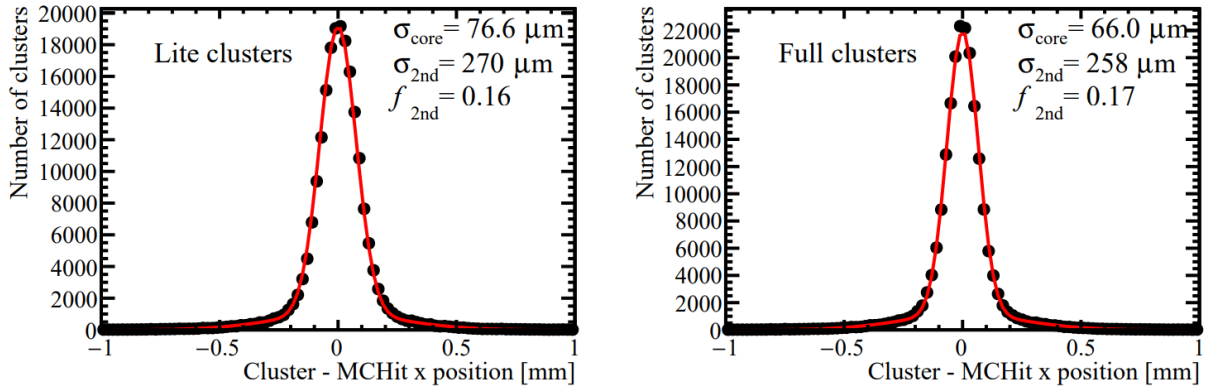


Figure 4.19: Cluster resolution for (left) lite clusters and (right) full clusters for particles with $p > 2\,\mathrm{GeV/c}$. The results from the double-Gaussian fits are superimposed, where σ_{core} and σ_{2nd} define the widths of the Gaussian, and f_{2nd} defines the fraction of the second Gauss. For these plots, no limit to the maximum cluster size was set.

Chapter 5

Radiation Tests of the SciFi Front-End Electronics

This chapter describes in detail the tests that were performed with the SciFi front-end electronics in order to validate it for operation up to the total radiation dose expected and beyond. The tests were performed at the **C**ERN **H**igh-energy **A**ccele**R**ator test facility (CHARM) [22] in two campaigns, both consisting of one week of preparation and one week of irradiation. During the first campaign, the Device Under Test (DUT) was a single half-ROB; in the second campaign, two half-ROBs were tested.

The CHARM facility at CERN makes use of a 24 GeV/c proton beam extracted from the Proton Synchrotron (PS) that interacts with a metallic target in order to generate a mixed-field radiation environment resembling that present in the vicinity of the LHCb Experiment. For the tests described here, a target made of copper was used, and the DUT was placed at position R10 in the CHARM irradiation room; Figure 5.1 shows the particle lethargy spectra (obtained with FLUKA [23] simulation) expected in these conditions.

The hottest spot in the SciFi electronics is expected to receive a radiation dose of up to 100 Gy while accumulating the $50\,\mathrm{fb}^{-1}$ of integrated luminosity aimed at by LHCb. A total dose of 200 Gy was requested, and CHARM provided a total of 175 Gy and 133.2 Gy for the first and second campaigns, respectively. The dose was delivered to the DUT almost uniformly over time, having one and four interruptions during the first and second campaigns, respectively. Figure 5.2 shows the total accumulated radiation dose over time, highlighting the periods during which irradiation was on. Table 5.1 shows the detailed dates and doses of the tests.

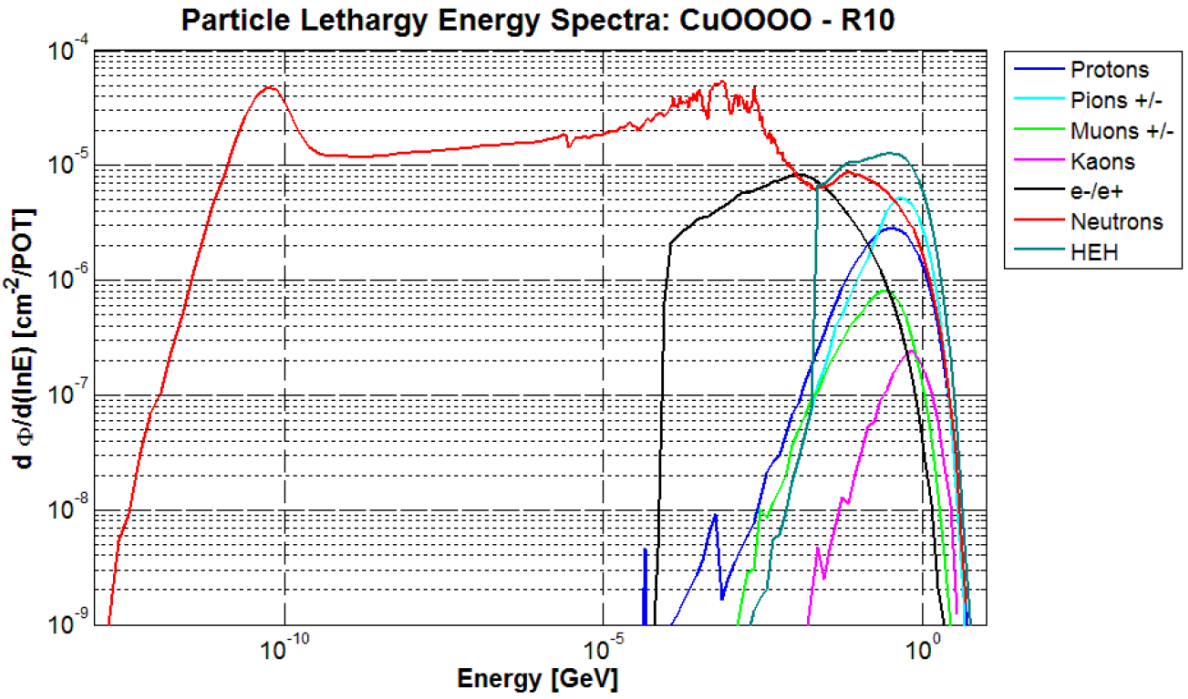


Figure 5.1: Particles lethargy spectra for the target and position used in the SciFi FEE tests.

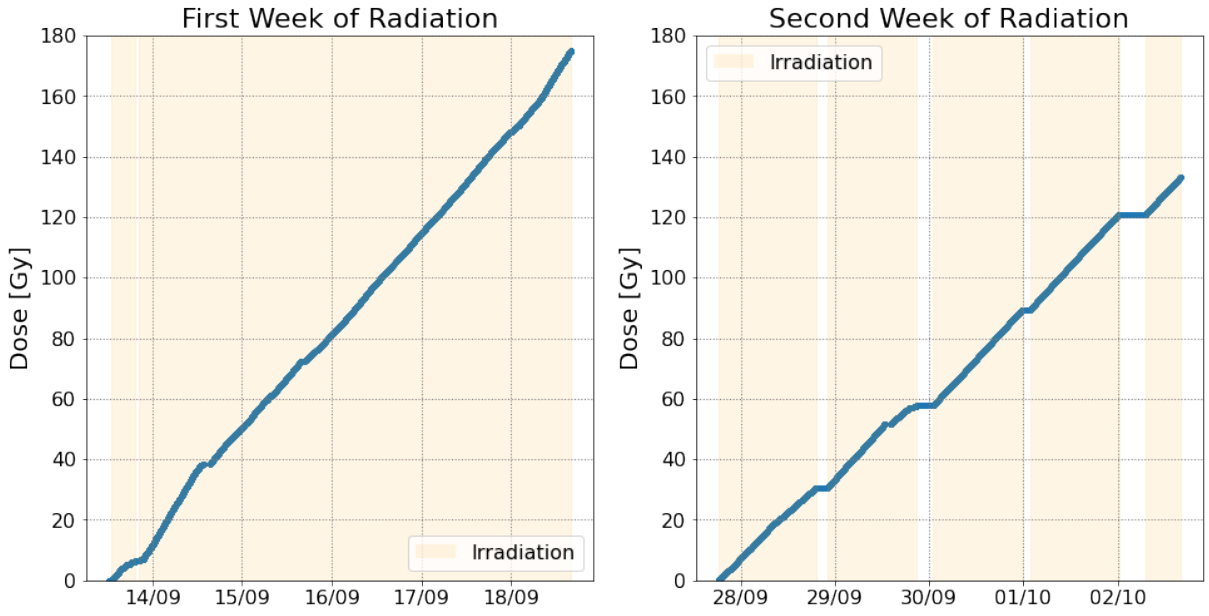


Figure 5.2: Total dose over time for both irradiation campaigns.

	Week I	**Week II**
Beginning	13-09-2017 13:02	27-09-2017 18:23
End	18-09-2017 16:00	02-10-2017 16:01
Dose	175 Gy	133.2 Gy
1 MeV eq. fluence	$1.7 \times 10^{11}\,\mathrm{cm}^{-2}$	$1.3 \times 10^{11}\,\mathrm{cm}^{-2}$
High Energy Hadrons fluence	$5.4 \times 10^{11}\,\mathrm{cm}^{-2}$	$4.1 \times 10^{11}\,\mathrm{cm}^{-2}$

Table 5.1: Radiation levels accumulated during the tests.

5.1 Objectives

Radiation can affect electronics in many different ways. When designing electronics to operate in radiation environments, it is important to choose components that have been qualified for the specific application. In the case of the SciFi FE electronics, most of the individual components had already been qualified. However, the behavior of the Microsemi IGLOO2 M2GL090 FPGAs (see section 4.4) posed a number of uncertainties. These FPGAs are 65nm, non-volatile, flash-based, low-power FPGAs designed to retain reliability in the presence of background radiation. Typical applications include aviation and space. Although they had already been the subject of some studies [24], the specific application in LHCb requires a dedicated investigation. The objective of the tests described in this chapter was to measure and understand the effects of radiation on the FPGAs and qualify the SciFi front-end as a whole for the radiation dose expected in the experiment, including a safety margin.

The effects on the FPGAs that have been studied in this chapter are:

- **Speed Degradation:** an effect that is expected to depend on the Total Ionizing Dose (TID). The propagation speed of electrical signals within the FPGA degrades with the accumulated dose. This reduces the maximum frequency at which the FPGA can operate and can also induce timing errors that did not occur before irradiation.

- **Leakage Current:** also expected to depend on the TID. The leakage current of certain devices can vary with the radiation dose. This can raise the power requirements of the experiment; therefore, it is important to know in advance how much the power consumption may vary with radiation.

- **FPGAs Reprogrammability:** after a certain accumulated dose, the FPGAs lose their ability to be reprogrammed. Any further attempt at reprogramming leads to the loss of their previous configuration and renders the device unusable. With more than four thousand FPGAs present on the SciFi FE electronics, it is crucial to know this point of no return, after which reprogramming will no longer be possible, and therefore, debugging and changing the firmware would not be possible anymore.

- **Single Event Upsets (SEU):** when a passing particle deposits charge at the base of a transistor, it may charge or discharge volatile memory cells as a consequence, causing the so-called bit-flips. Such bit-flips can cause the corruption of data and, in the worst case, the change of important configuration registers, possibly causing the hardware to malfunction. The test aimed to evaluate the rate of SEUs expected in the experiment and to measure the efficiency of Triple Modular Redundancy in preventing SEUs.

- **Single Event Latch-ups (SEL):** depending on the instantaneous dose, an SEU can cause a short circuit in the chip. Structures like inverter gates and buffers have two transistors between the ground and the power vias, which are always in opposite states, to prevent short circuits. An SEU can cause both transistors to conduct, causing a short circuit, which can lead to overheating and damage to the device.

5.2 Measurements Methodology

In this section, it will be presented the procedure to measure each of the radiation effects studied in this thesis.

Speed Degradation

In order to measure the FPGA speed degradation, a ring oscillator was used. A ring oscillator is a device with an odd number of inverters connected in a ring topology. Such devices cannot be stable because an odd number of inverters always behave logically as a single inverter, and the inverted output is connected back to the input. Once the inverted signal of the output propagates back to the input, it will cause the signal to be inverted again; after the new output signal propagates back to the input, it will be inverted to the original value, and the cycle will repeat. Figure 5.3 shows a ring oscillator of 3 stages, highlighting the logical value in each of the nodes. The dashed arrow indicates the propagation direction of the signal from the output to the input.

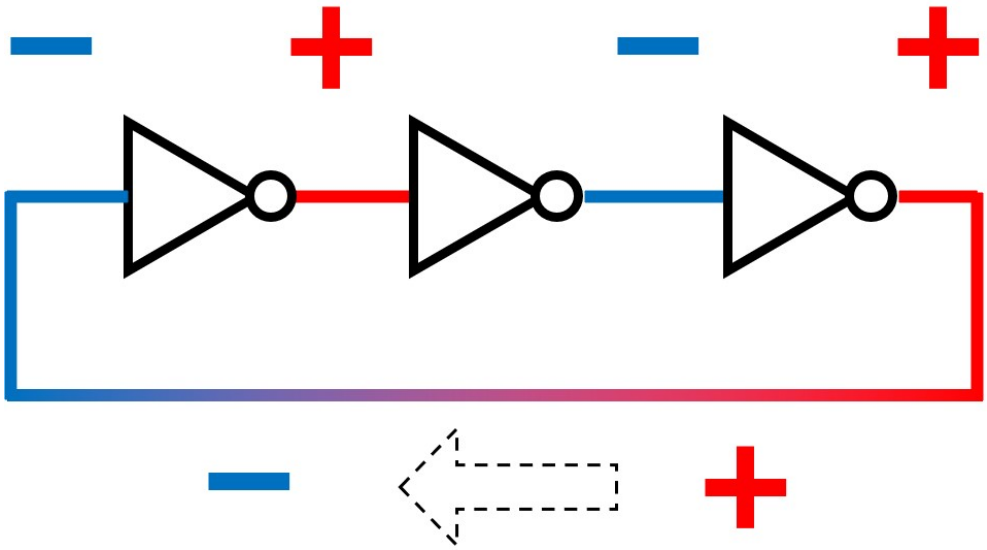


Figure 5.3: Diagram of a ring oscillator with three inverters.

The described structure presents an oscillatory behavior where the period of oscillation is the total propagation delay of the ring circuit. Since the propagation delay is dominated by the logic gates, the frequency is given by

$$f = \frac{1}{2tn} \tag{5.1}$$

where t is the time delay for a single inverter and n is the number of inverters. Implementing a ring oscillator in the FPGA, it is possible to measure its speed degradation by measuring the frequency of the ring oscillator. For the tests at CHARM, a ring oscillator of 101 inverters was used per FPGA.

In order to read the frequency of the ring oscillator, its output is connected to a counter, the value of which is periodically stored in a register (and reset), clocked by a stable

external reference clock source that does not vary with radiation. In this test, the clock was provided by a GBT chip. The data is then acquired by reading this register value through the control system. The value read divided by the frequency of the reference clock gives the oscillation frequency of the ring. From this, the time delay t in 5.1 can be monitored.

Leakage Current

To measure the leakage current, a Power Monitor Tool was built by the Nikhef Electronic Department. The front-end boards were powered through this tool, equipped with current and voltage sensors to monitor the power consumption of the SciFi FEE. The hardware was LabView compatible, and the data was acquired through a LabView project running on a dedicated computer. The increase in leakage current from the components of the board would cause a measurable increase in the total power consumption.

FPGAs Programmability

In order to study when the FPGAs lose their programmability, we selected nine of the 24 FPGAs and reprogrammed them periodically until all of them failed. The programming was done through the slow control system, and the results of each programming iteration were recorded for later analysis.

Single Event Upsets

The Microsemi IGLOO2 M@GL090 FPGAs have two kinds of volatile memory: the flip-flops from the logic units, which can be combined by means of firmware to form the registers, and the hard RAM memory contained in the chips. The occurrence of SEUs in both of them has been studied.

In order to measure the number of SEUs in the FPGA's registers, a 16 kbits long shift register was implemented. The input of the shift-register was an alternating signal, and the output is connected to a counter which counts whenever two consecutive bits have the same value. In a binary alternating pattern, any bit-flip would cause three consecutive bits to be the same. This would cause the counter to count twice. Therefore, the value read from the counter represents twice the amount of bit-flips that occurred through the whole shift register.

The measurement of the number of SEUs in the RAM memory was performed by reading 8 kbits of memory per FPGA through the ECS system every 30 seconds. The amount of SEUs is then determined by an analysis software that compares the values of the memory in each read cycle and computes the number of bits that changed.

Single Event Latch-ups

The SELs are also measured thanks to the Power Monitor Tool: once a SEL happens, a current spike is expected to happen due to the short circuit created by the SEL. The Power Monitor Tool monitors and logs such spikes.

5.3 Experimental Setup

The CHARM facility offers a control room with a patch panel that connects directly to the irradiation hall. The user prepares in advance all the necessary cabling in order to connect the DUT's through the patch panel. CHARM also provides a "preparation room," which is a room with a pair of patch panels exactly like the ones found in the control room and in the irradiation area (including the same length of cables) such that the user can prepare all DUT's as well as the control and DAQ systems in advance. In this way, the user can perform a complete test of their setup, minimizing the time to be spent in the irradiation area and the chance of failure leading to unwanted downtime. Once the preparation is done, the user assembles the DUT in a rack provided by the CHARM team and waits for the access window, which usually lasts for 3 to 4 hours and happens once a week. Figure 5.4 shows the DUT comprised of two partially assembled ROBs mounted in a rack.

During the access window, the rack where the DUT is mounted is carried to the radiation facility by a remotely controlled robot, and the CHARM team moves the necessary cables provided by the user to the experimental area. The user then must access the experimental area and connect the cables from the patch panel to the DUT. As the area is next to the irradiation room, part of the surroundings is activated. The access is, therefore, limited and strictly regulated. Once the connections are made, the robot carries the rack with the DUT installed to the desired position inside the irradiation room. Figure 5.5 shows in the top left the robot carrying the rack with the DUTs to its position, in the top right diagram with a red arrow indicating the path of the robot, and in the bottom right the irradiation room. Once the user verifies that the control and data acquisition system is able to operate the DUT, the access can be terminated, and the user must wait for the CHARM team to finish the service of the facility, start the beam, and introduce the target for the irradiation period to start. During the installation periods of the two campaigns, the present writer spent a total of 10 to 15 minutes in the area, and his operation dosimeter registered a total of 30 µSv of received dose. In terms of exposure, this is the equivalent of four hours at the cruise altitude of a commercial airliner. Once the hall is closed, the DUT will be unreachable until the next access a week later. The irradiation occurs in this period, which the user should use to collect data.

During the first irradiation campaign, only one of the ROBs (ROB 2_3 in Fig. 5.4) was tested; both were tested during the second campaign. Therefore ROB 2_3 was exposed to a higher dose than ROB 0_1.

Figure 5.4: Two ROBs being prepared for the radiation tests.

Figure 5.5: Top left: the robot carrying the rack with the DUTs to its position. Top right: the diagram with a red arrow indicating the path of the robot. Bottom left: the logo of the CHARM facility. Bottom right: the irradiation room.

5.4 Control and Data Acquisition

For control and data acquisition, two MiniDAQs version 1 were used. The MiniDAQ v1 is a module based on the AMC40 board, which was the first prototype of LHCb's DAQ board for the upgrade. It was designed to provide control and data acquisition to the subdetector's front-ends through optical links. The LHCb online group provided firmware and a set of libraries for the SCADA software (WinCC) that allow to read and write registers on the SciFi FEE.

Based on these tools, a WinCC-based control system was developed to configure and monitor the front-end boards during the tests. It included scripts to automatically configure all the components of the ROB and put it into the expected state to perform a data run. All configuration is done by writing registers. Once a data run is started, a script is periodically called every 30 seconds in order to perform a read cycle, i.e., to read all the registers from all the devices in the ROB. This implies that the control links of the ROB (see Section 4.4.5) were used to acquire the data. The scripts save the data from the registers in text files, adding timestamps. The scripts would also perform a certain level of automatic error checking and alert the shifter through messages in a smartphone app. A General Purpose Interface Bus (GPIB) controller was also created using an Arduino board in order to allow the shifter to remotely change parameters in the power supplies in order to be able to control the voltage provided, the current limit, and also power cycle a ROB.

5.5 Analysis and Results

The analysis of the data was performed using Python. A set of Jupyter Notebooks was created to load the data, compute results, and produce plots. The next sub-sections present the data analysis and the corresponding results.

5.5.1 Uncertainties

All the results from this chapter are presented as a function of the radiation dose. Therefore, the uncertainty in each result will come from two different sources: the measurement performed with the SciFi electronics and the measurement of the radiation dose performed by the CHARM team.

For the radiation dose measurement, a global uncertainty of 35% is estimated, taking into account the error derived by both the Secondary Emission Chamber and RadMon dosimetry uncertainties. This value was estimated by the CHARM team as documented in [22].

When combining the uncertainties in the SciFi measurements and in the radiation dose measurement, the total uncertainty is dominated by the latter (with the exception of the programmability test due to the limited data). In order to ease the comparison of the results, the uncertainties in the results presented in this chapter include only those due to the SciFi measurements unless explicitly stated otherwise.

When the total uncertainty is presented, it is the result of the propagation of the uncertainties in the measurement of the quantity X normalized on the radiation dose

$$X_{\text{total}} = \frac{X_{\text{SciFi}}}{D_{\text{rad}}} \tag{5.2}$$

, i.e.

$$\sigma = \sqrt{\left(\frac{\sigma_{SciFi}}{X_{SciFi}}\right)^2 + \left(\frac{\sigma_{D_{rad}}}{D_{rad}}\right)^2 - 2\frac{\sigma_{SciFi,rad}}{X_{SciFi}D_{rad}}} \tag{5.3}$$

5.5.2 Speed Degradation

The procedure to compute the speed degradation of each FPGA is the following: record the data of the ring oscillator counter and the radiation dose over time, interpolate the dose values corresponding to the timestamps of the counters and plot the counter values versus the accumulated radiation dose. The degradation of the ring counter frequency is a direct measurement of the speed degradation, and from this, one can calculate the change in the FPGA's propagation delay as a function of the dose according to Eq. 5.1.

A technical issue, however, complicated the analysis: the ring oscillators would erratically start with different frequencies. The blue line in Figure 5.6 represents the frequency over time of one of the FPGAs before irradiation was even started: the jumps in frequency indicate that the ring oscillators were not properly initialized. The initial condition ideally should be that all the inverters have an opposite value in the output with respect to the input except for one, namely the one through which the signal is propagating at that given moment; when not properly initialized, the ring oscillator can start in a random state in which multiple inverters have the same value in their inputs and outputs, that would cause them to oscillate at random frequencies.

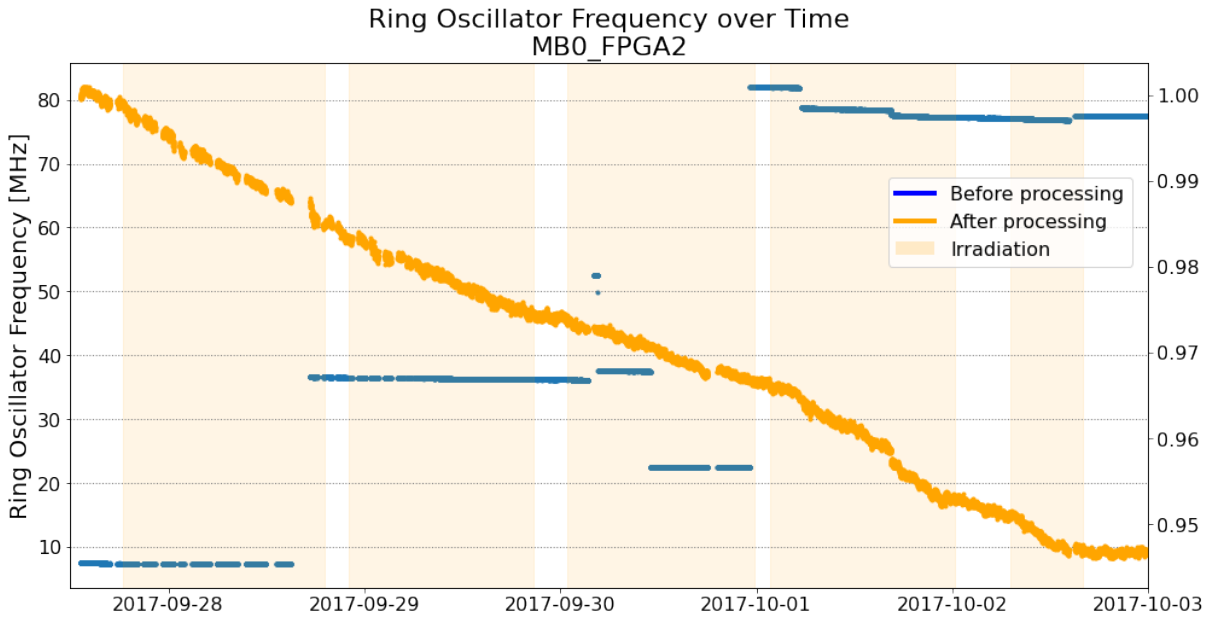


Figure 5.6: Ring oscillator frequency over time. In blue: raw values read from FPGA. In orange: normalized values after processing.

This effect should have been prevented by a mechanism implemented to reset the ring to the proper initial state, which should have been set automatically by software before every run, but apparently, due to a firmware or software problem, this reset was not applied properly.

In the analysis, this unwanted effect could be circumvented because we are interested in the frequency variation due to the accumulated radiation dose and not in the absolute oscillating frequency. Therefore, to obtain the variation, each continuous segment of data can be corrected to its previous value, starting from the value of the last segment (that could be identified by the beginning of a new run, by a missing block of data or by a sudden variation larger than 0.3%). After applying such correction, we obtain the orange line from Figure 5.6.

The frequency data after this correction can then be plotted versus the accumulated dose, as shown in Figure 5.7, which illustrates the speed degradation of 15 FPGAs along 100 Gy of accumulated dose, starting from 30 Gy. [1]

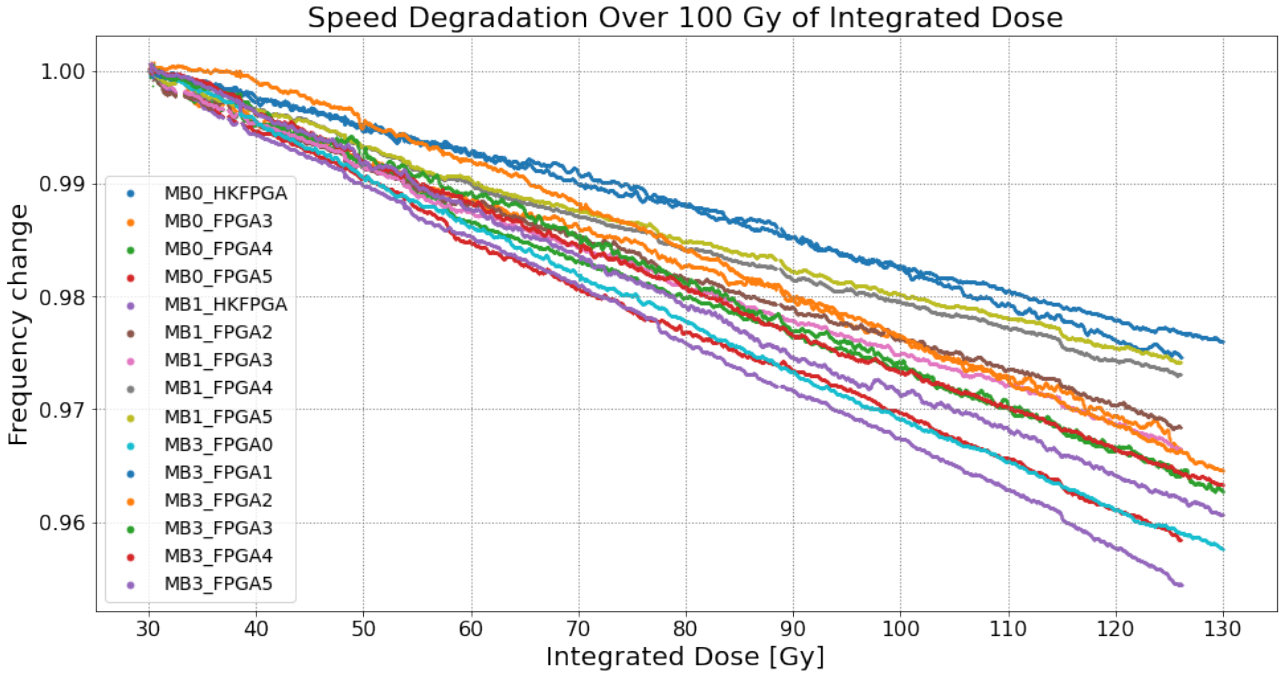


Figure 5.7: Speed degradation vs accumulated ionizing dose (30-130 Gy) for the FPGAs tested.

In Figure 5.7, one can easily note the linear relation between speed degradation and dose in this range of integrated dose. A line was fitted to the data to determine the speed degradation ratio of each FPGA, and the Pearson's chi-squared test was used to evaluate the goodness of the fit. Figure 5.8 shows the slopes from the linear fit for each FPGA.

The speed degradation was on average 3.47 % with a standard deviation of 0.64 % for 100 Gy of integrated dose. Taking into account the uncertainty from the CHARM radiation measurement, the speed degradation rate is given in Table 5.2.

[1] Some FPGAs from ROB 0_1 exhibited a smaller slope over the first 30Gy, possibly because, during these periods, those FPGAs were undergoing the reprogramming test. Moreover, the FPGAs from ROB 2_3 didn't record data up to around 30 Gy of irradiation. For all these reasons, the comparison in Fig

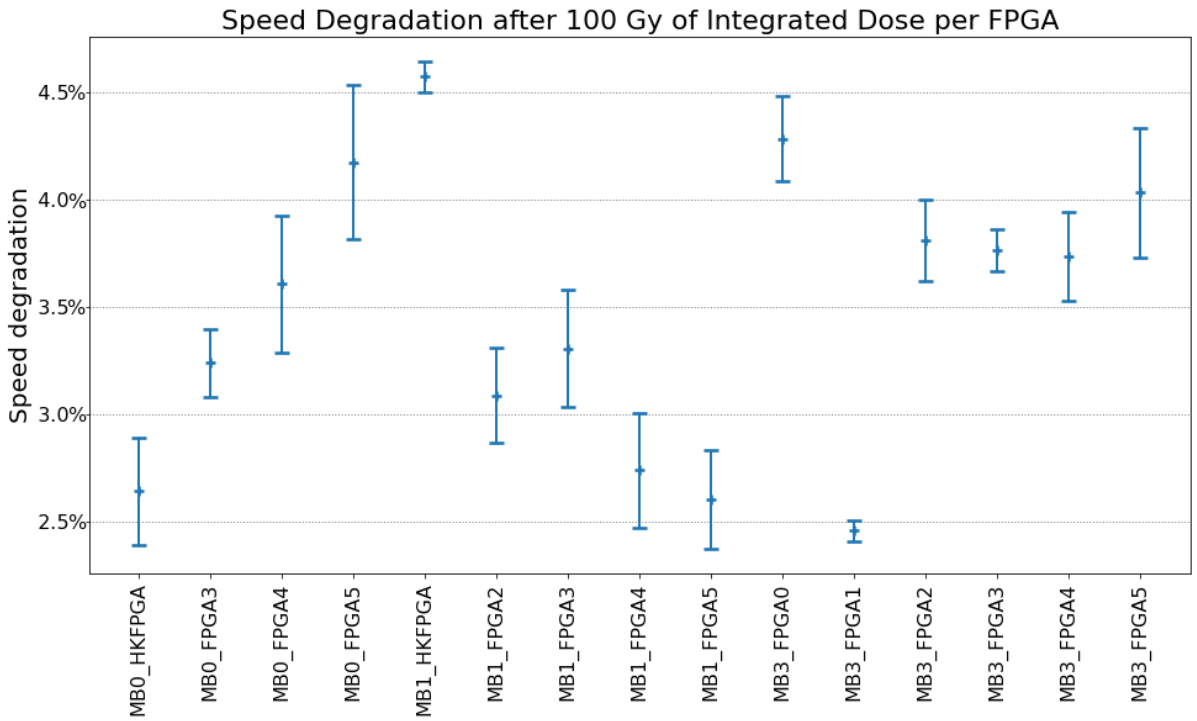


Figure 5.8: Results of speed degradation per FPGA.

Speed Degradation per Total Ionizing Dose
$(0.35 \pm 0.14)\,\%\mathrm{Gy}^{-1}$

Table 5.2: Final result of Speed Degradation.

After approximately 150 Gy of accumulated dose, the relation is no longer linear, as shown in Figure 5.9.

Figures 5.10 and 5.11 present the speed degradation over time for the FPGAs from ROB 0_1 and ROB 2_3, respectively. The speed degradation is fully dominated by the radiation effect during the tests: as soon as the irradiation stops, the speed degradation gets close to zero (in some cases, it even anneals slightly). The second point is the difference due to the total accumulated dose: the FPGAs in Figure 5.10 were not irradiated during the first campaign (up to 104 Gy at the beginning of the plot), while those in Figure 5.11 were (279 Gy before the beginning of the plot).

Another factor that seemed to impact the speed degradation is the act of reprogramming the FPGA. Unfortunately, it is difficult to draw conclusions from the data because data taking had to be stopped while reprogramming the FPGAs, and the issue of initialization of ring oscillators renders it impossible to know frequency changes while data taking is suspended. However, even with this limitation, it was possible to observe that, after reprogramming, the frequency of some FPGAs would rise for a while. Figure 5.12 shows the plot of speed degradation versus accumulated dose for the eight FPGAs that have been reprogrammed during the second campaign. Six out of seven could not be reprogrammed beyond 30 Gy. The one that remained operational beyond 30 Gy (orange line on Figure

5.7 runs from 30 Gy.

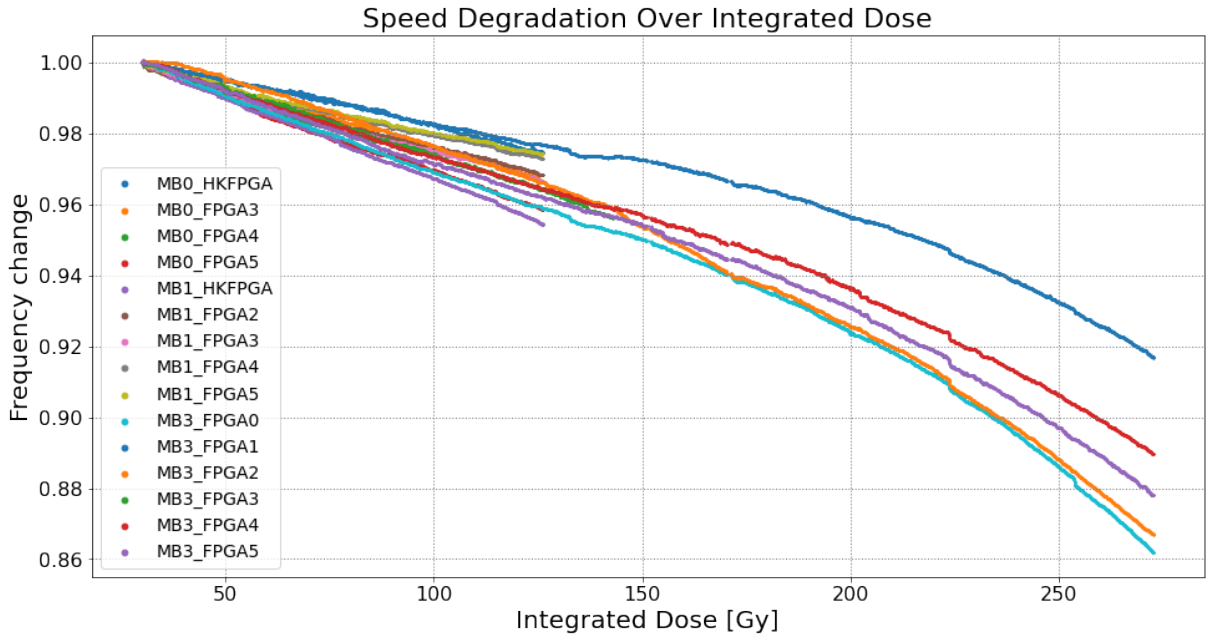


Figure 5.9: Speed degradation vs accumulated ionizing dose (30-280 Gy). Note that only some FPGAs have been tested up to 280 Gy.

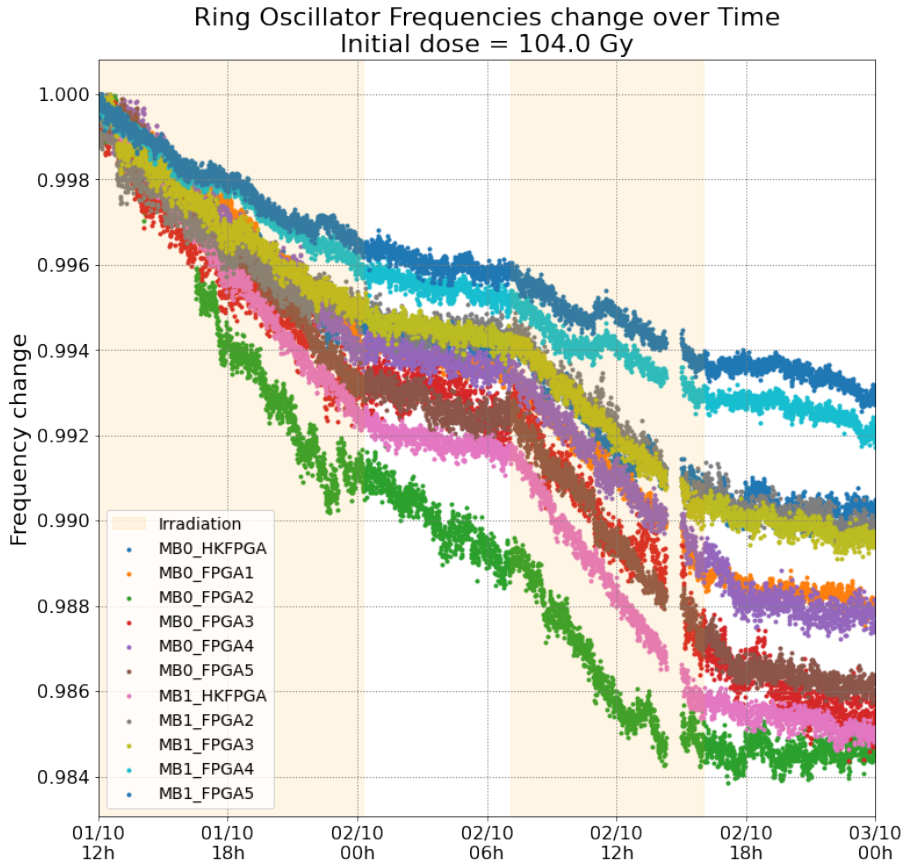


Figure 5.10: Speed degradation of the tested FPGAs from ROB 0_1, over a period of 36 hours during the second week of irradiation. Irradiated periods are highlighted in yellow.

5.12) exhibited a speed increase of about 1.25% and then a degradation of about 3% for

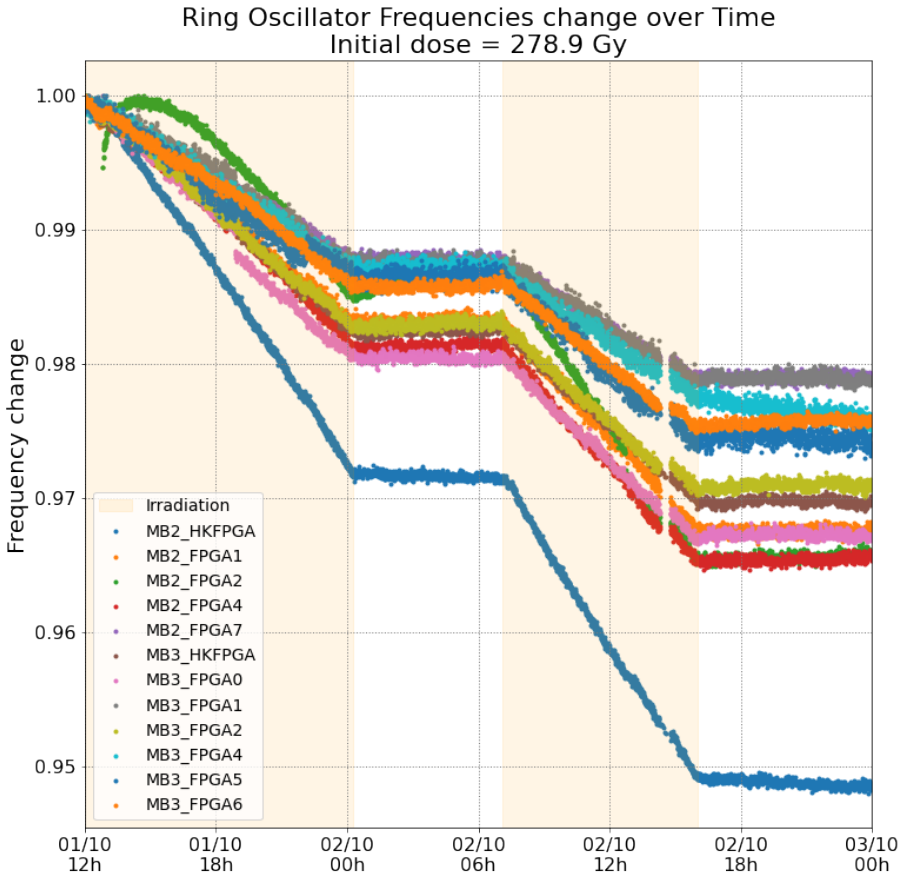


Figure 5.11: Speed degradation of the tested FPGAs from ROB 2_3, over a period of 36 hours during the second week of irradiation. Irradiated periods are highlighted in yellow.

the next 100 Gy, similar to those in Figure 5.7.

5.5.3 Leakage Current

The objective of this analysis is to determine whether the power consumption of the SciFi electronics will change over time due to radiation. Some electronic components are susceptible to a leakage current increase when exposed to a certain accumulated ionizing dose, which, over time, can significantly impact the power consumption of the front-end boards.

The idea is to plot the power consumption data as a function of the accumulated radiation dose to verify if there is any correlation. However, the data requires some prior processing as the power consumption can change for a variety of reasons. Figure 5.13 shows as an example the power consumption of Master board 2 during the second week of irradiation. Note that there are many spikes and sudden shifts in consumption. These happen when applying a new configuration or resetting a component in the board, as well as when programming FPGAs and, obviously, when power cycling the board. On the other hand, the power consumption variation caused by radiation is expected to be smooth. This is confirmed by the observation that sudden changes and shifts take place even when no irradiation is applied.

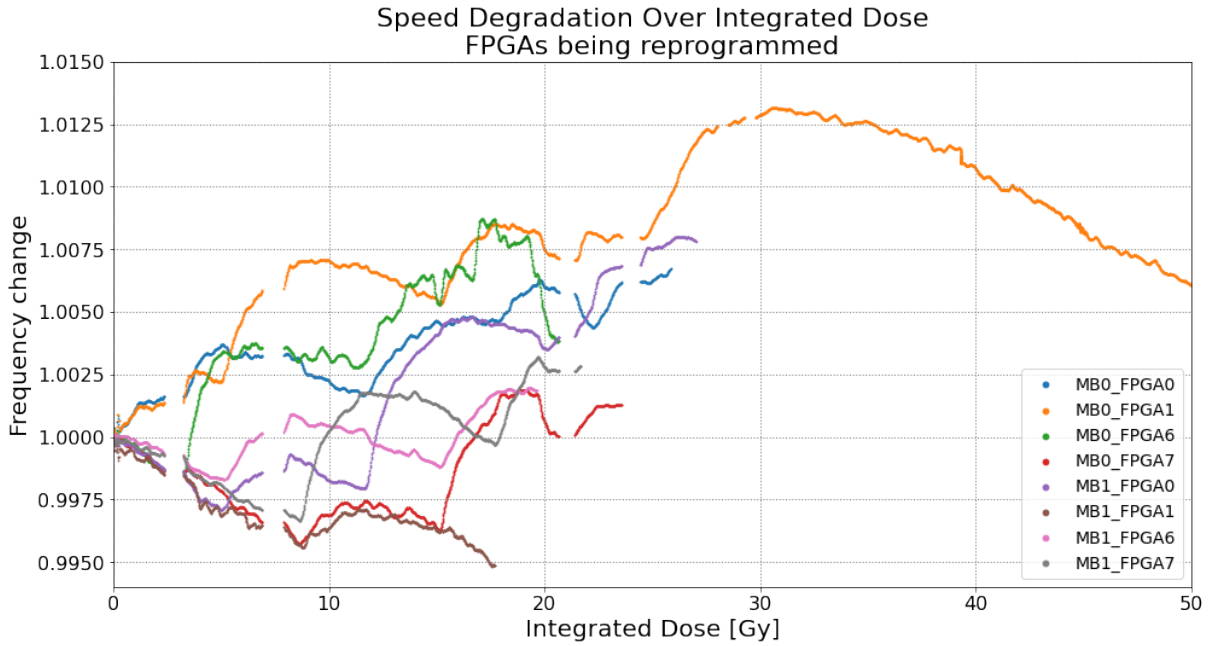


Figure 5.12: Speed degradation versus accumulated dose for the FPGAs reprogrammed during the second campaign.

Having said that, extracting the smooth effect due to irradiation requires the correction for the variations caused by other factors. The method chosen to accomplish this can be summarized as follows:

- Take the moving average of the data (a range of 10 elements was used).
- Take the moving variance of the data (a range of 20 elements was used).
- Remove an interval of data around any region in which the variance is higher than a certain threshold (an interval of 140 data points around the regions with variance higher than 0.005 was used)
- Normalize on the first data point of each segment of data and connect all the segments together.

Figure 5.14 shows the same data as Figure 5.13 after having the regions of high variance removed, and Figure 5.15 presents it after having normalized each segment.

Figures 5.16 and 5.17 were obtained using the procedure explained above; they show the variation in power consumption for all the Master boards during the first and second irradiation campaigns, respectively. During the first campaign, Master boards 0 and 1 were not irradiated. One can observe small variations in both directions through the irradiation period (but also outside of the irradiation periods, as one can see with MBs 0 and 1 during the first week.) No board at the end of an irradiation period ended up consuming more than 2% additional power, and, in fact, only 2 out of the 6 measurements showed a positive variation of power.

Therefore, one can conclude that no significant correlation between power consumption and radiation dose was found for the range of radiation studied.

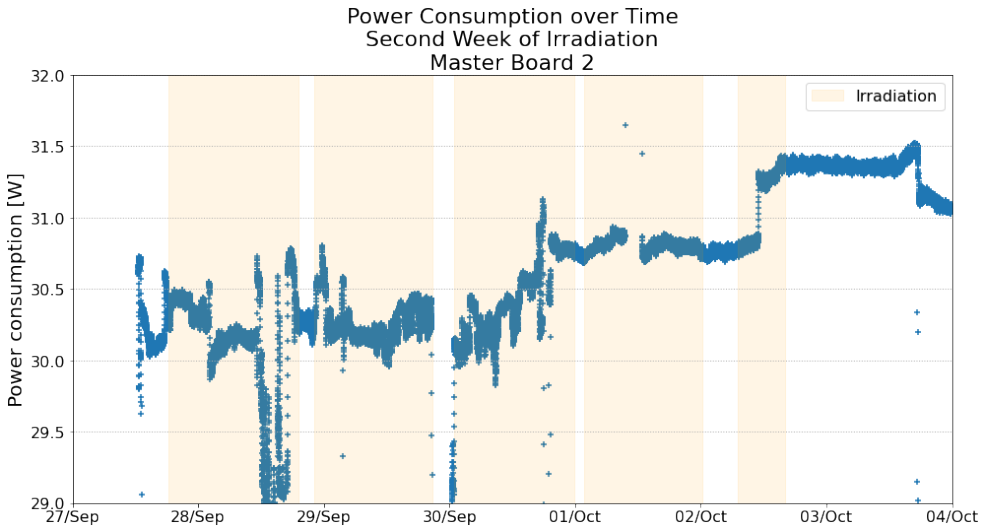


Figure 5.13: Power consumption of Master Board 2 during the second irradiation campaign after applying a moving average of 10 elements.

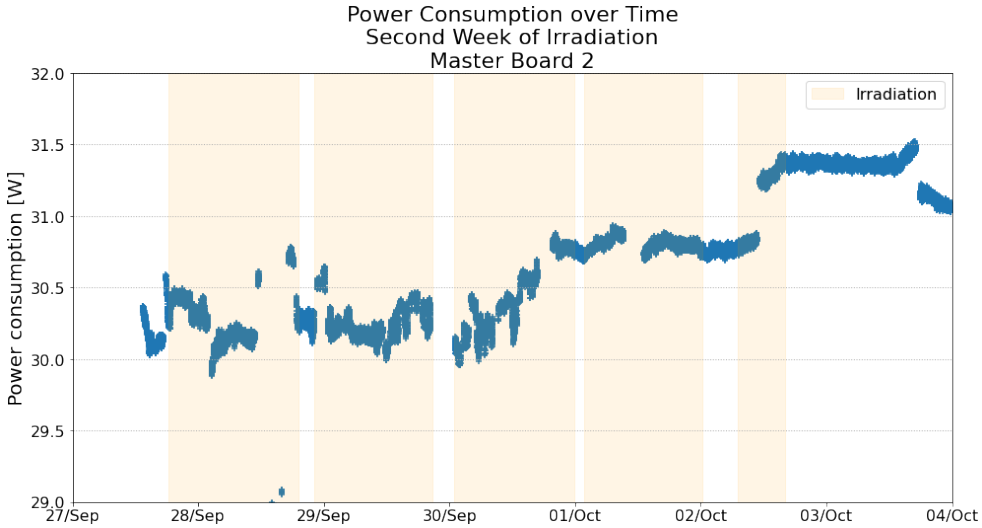


Figure 5.14: Power consumption of Master Board 2 during the second irradiation campaign, after removing the regions of high variance.

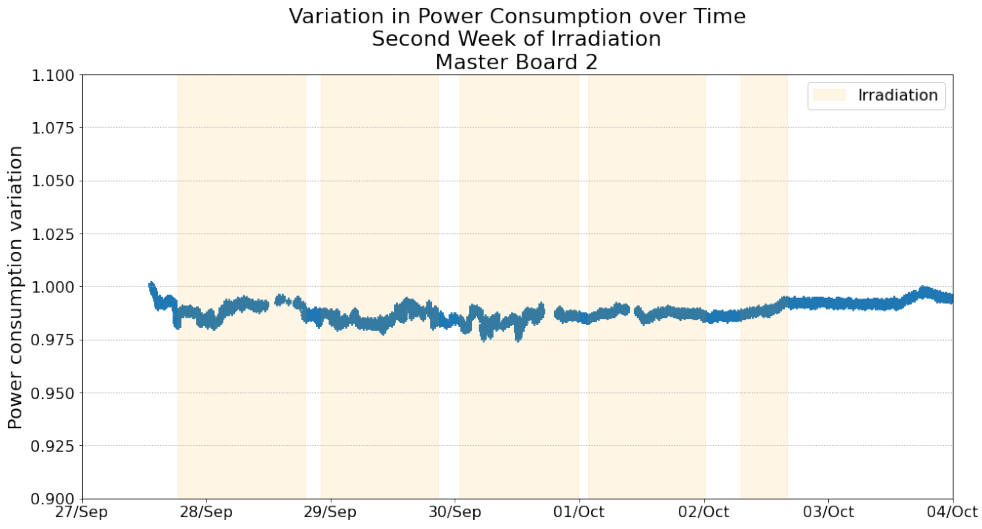


Figure 5.15: Power consumption of Master Board 2 during the second irradiation campaign, after normalizing each segment of data.

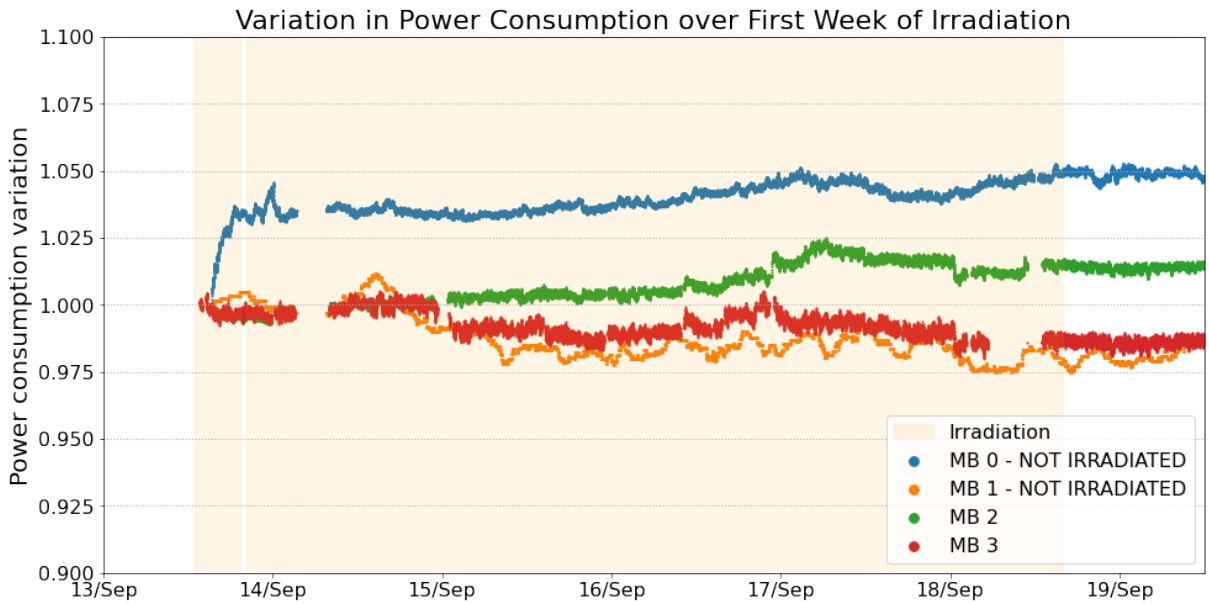


Figure 5.16: Power consumption variation during the first irradiation campaign.

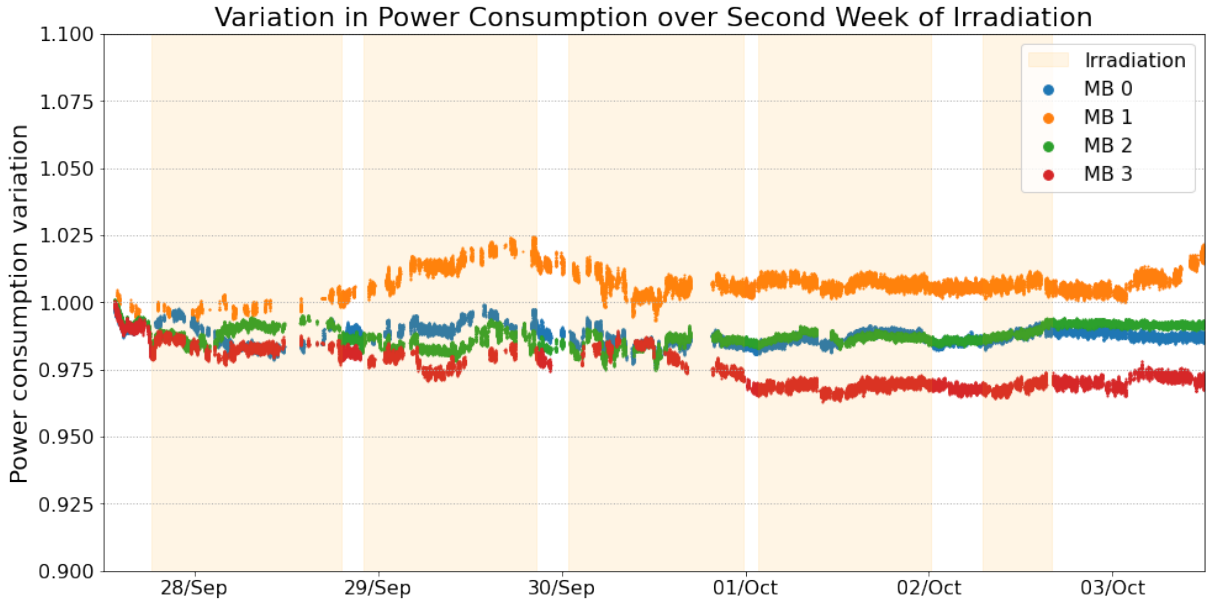


Figure 5.17: Power consumption variation during the second irradiation campaign.

5.5.4 FPGA programmability

During the radiation tests, a total of ten FPGAs were tested for programmability, nine of which until the programming cycle failed and rendered the device useless. The last one was kept operational to provide a later comparison with the ones that had not been reprogrammed during irradiation.

By comparing the dose at which each FPGA failed to be reprogrammed with the one from the last successful programming, one can determine the dose range at which each of them became non-programmable. Figure 5.18 shows these dose ranges.

Considering the probability distribution of failure to be uniform among such intervals,

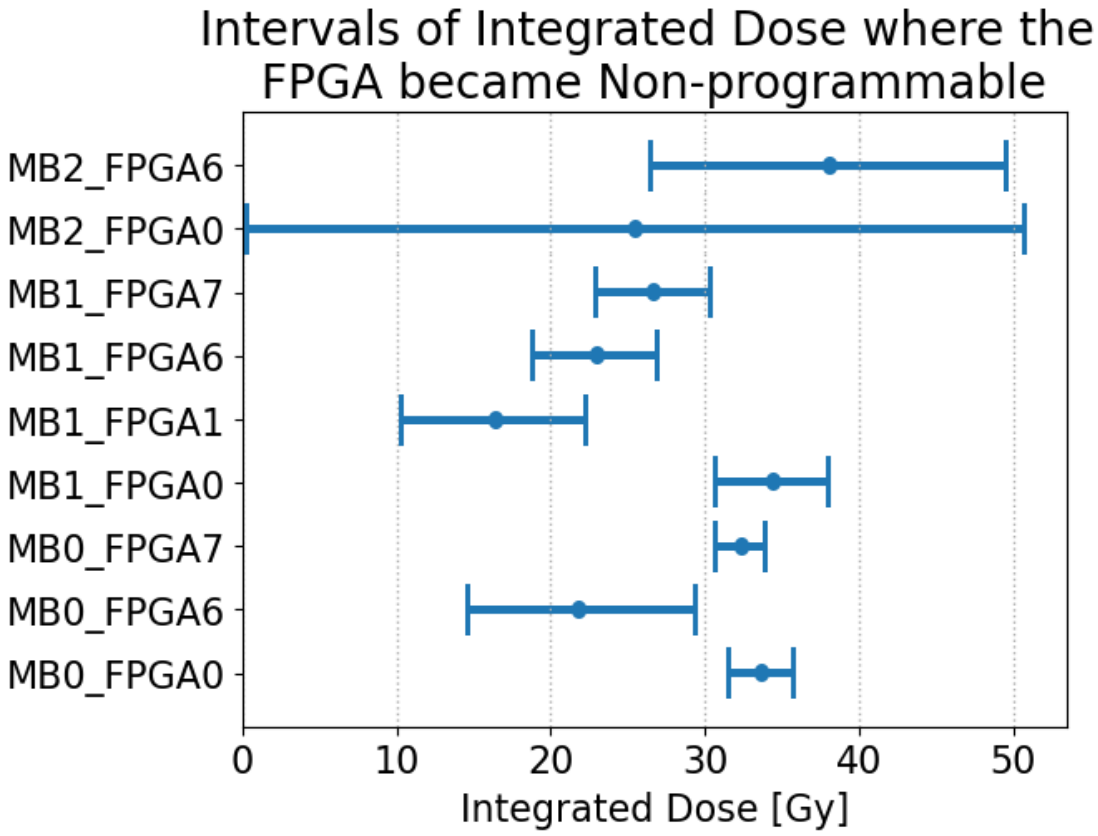


Figure 5.18: Intervals between the last successful programming attempt and first programming failure for each tested FPGA.

the sum of all the intervals would converge to the probability mass function of the FPGA failing over the integrated dose. Applying such sum to our data gives us the histogram found in Figure 5.19.

5.5.5 Single Event Upsets

The main objectives of this analysis were to estimate the rate of SEUs taking place in the FPGAs' RAM memory and registers during the experiment operation and to measure how efficient the Triple Modular Redundancy is in preventing them.

SEUs in the RAM Memory

The amount of SEUs was evaluated by comparing the values of the memory at each reading cycle with the value at the next one and counting the number of bits that changed in between read cycles.

For reasons explained further in this section, the analysis considered only the reading cycles in which a maximum of three FPGAs had a SEU, and the total amount of SEUs was smaller than 20 in all FPGAs during that read cycle.

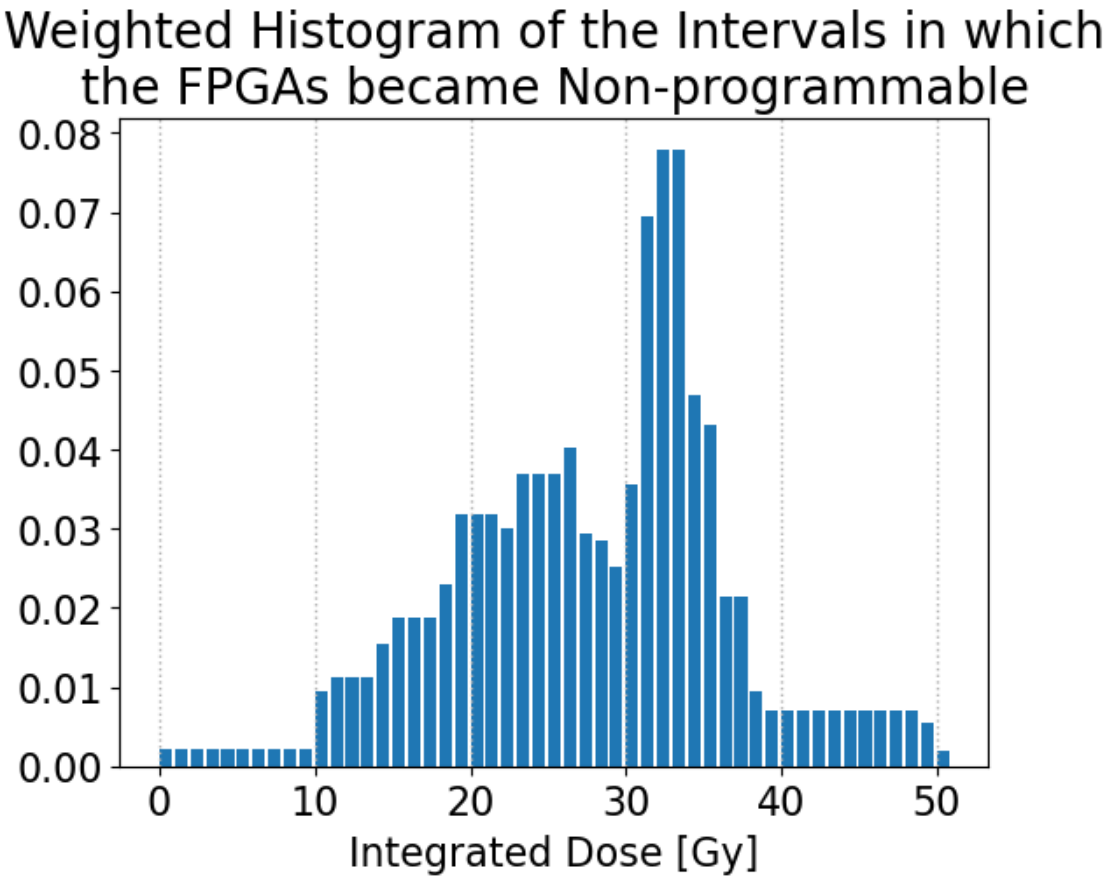


Figure 5.19: Weighted histogram of the intervals in which the FPGAs became non-programmable.

Figure 5.20 and 5.21 presents the number of SEUs per FPGA as a function of time in the second irradiation campaign for ROBs 0_1 and 2_3 respectively. As expected, the amount of SEUs increases linearly with the integrated dose, and when not being irradiated, the FPGAs do not experience any significant amount of SEUs. The FPGAs on ROB 0_1 exhibit a rate of SEUs of $(0.58 \pm 0.11)\,\mathrm{Gy}^{-1}$ while the ones in ROB 2_3 $(0.79 \pm 0.10)\,\mathrm{Gy}^{-1}$.

The SEU rate was higher for the FPGAs of ROB 2_3 than for those of ROB 0_1. Figure 5.22 hints at the fact that this could be due to the position of the ROBs in the radiation area, which could lead to different doses being accumulated by different areas of the ROBs. There seems to be a trend: FPGAs on the right position of the board measured a lower number of bit-flips, consistent with the variation we expected along the horizontal axis, as the beam was closer to the left side of the ROBs.

The multiplicity of SEUs per readout cycle is shown in Figure 5.23. Only events whose readout cycle lasted for $(30 \pm 3)\,\mathrm{s}$ have been selected. The events were also filtered by the amount of FPGAs that had a bit-flip during the reading cycle. Three different cuts have been used: a maximum of one FPGA, a maximum of three FPGAs, and no limit to the number of FPGAs having bit-flips during the read cycle. We can observe that in the left part, the incidence of multiple SEUs in the same read cycle decreases exponentially. In the right part, there is one case in which 22 bit-flips happened in the same FPGA and 11 cases in which more than 25 bit-flips happened, once in a single FPGA and 10

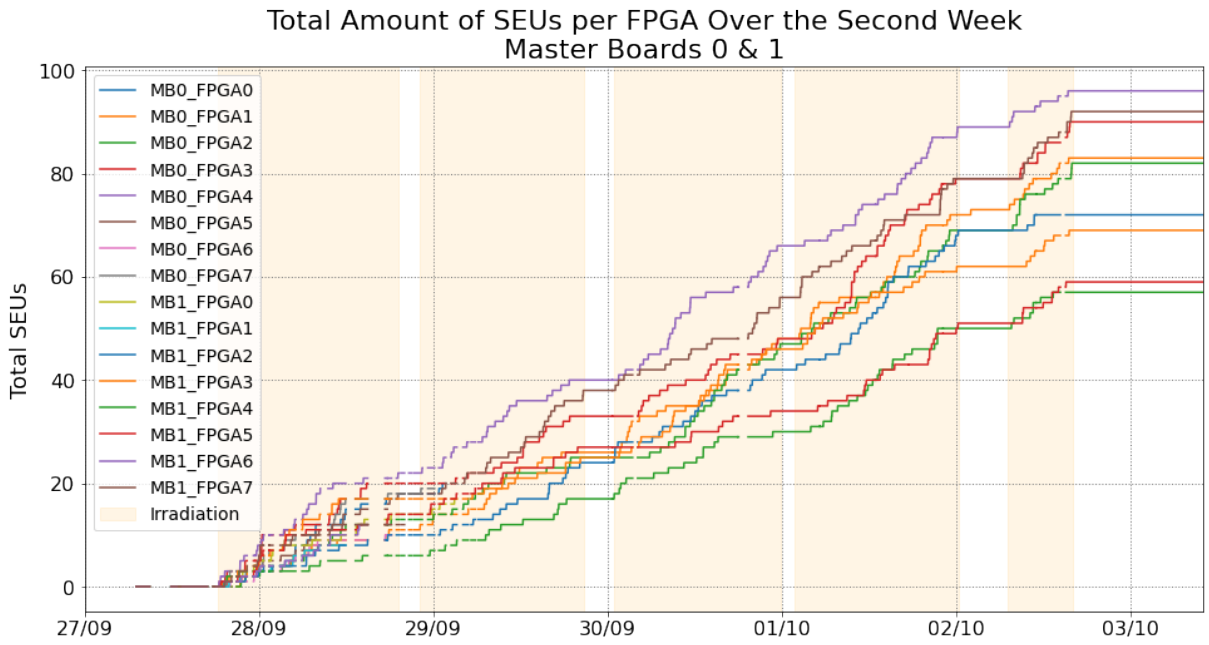


Figure 5.20: Total amount of SEUs over time, per FPGAs of the ROB 0_1, during the second week of irradiation.

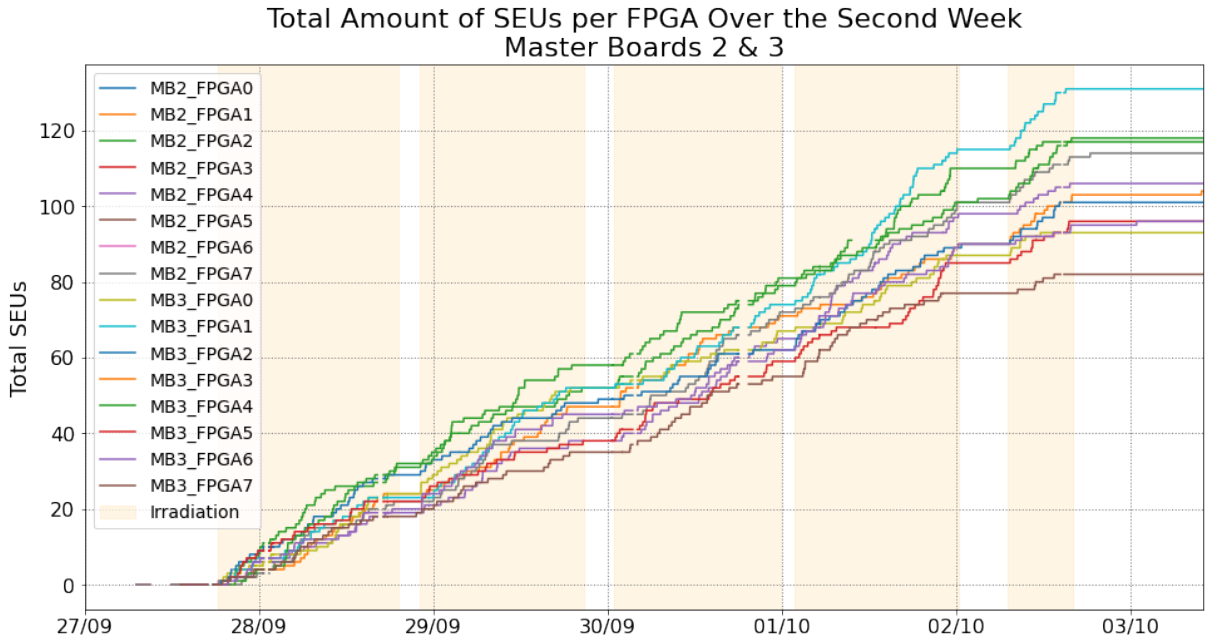


Figure 5.21: Total amount of SEUs over time, per FPGAs of the ROB 2_3, during the second week of irradiation.

times in multiple FPGAs. This is extremely unlikely to have happened due to SEUs, and therefore, such events can be attributed to a readout error from the data acquisition system or to a change in the RAM caused by a device reset or a power cycle.

In Figure 5.24, the SEUs increase as a function of the integrated dose. The dose values are interpolated, selecting only the read cycles that happened during the irradiation periods; then, for each readout cycle interval, the amount of SEUs is divided by the amount of

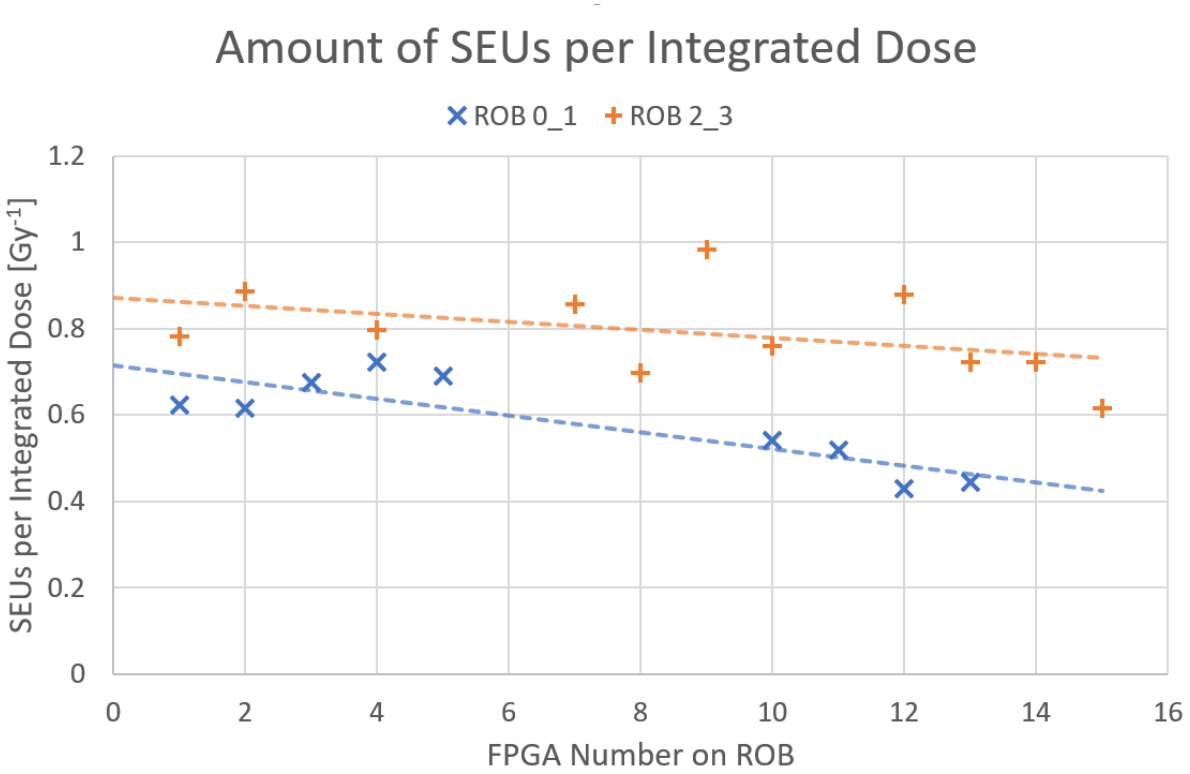


Figure 5.22: Amount of SEUs per integrated dose. The horizontal axis represents the FPGA position in the ROB, 0 being the leftmost and 15 being the rightmost.

FPGAs that were correctly read out through such period. Additionally, in order to avoid pathological events like those on the right of Figure 5.23, a cut is applied, requiring the number of bit-flips to be lower than 20. By fitting a straight line to the data in Figure 5.24 the average amount of SEUs per integrated dose can be found. Dividing by the amount of memory bits that were used during the test (8192 bits), the average amount of SEUs per memory cell (taking into account also the uncertainty from the CHARM radiation measurement) can be found, as shown in Table 5.3.

ROB	**SEU rate per RAM memory cell**
ROB 0_1	$(1.0 \pm 0.4) \times 10^{-8}\,\mathrm{Gy}^{-1}$
ROB 2_3	$(1.2 \pm 0.4) \times 10^{-8}\,\mathrm{Gy}^{-1}$

Table 5.3: Rate of SEUs per RAM memory cell.

SEUs on the Logic Units

To evaluate the number of SEUs happening in the logic units of the FPGA, a similar procedure was followed. Only ROB 0_1 underwent the test as ROB 2_3 had a bug in its SEU counting mechanism that could not be fixed in time for the test.

Unlike the case of the SEUs in the RAM, the distribution of bit-flips per readout cycle does not decrease exponentially but exhibits a long tail, as shown in Figure 5.25. The

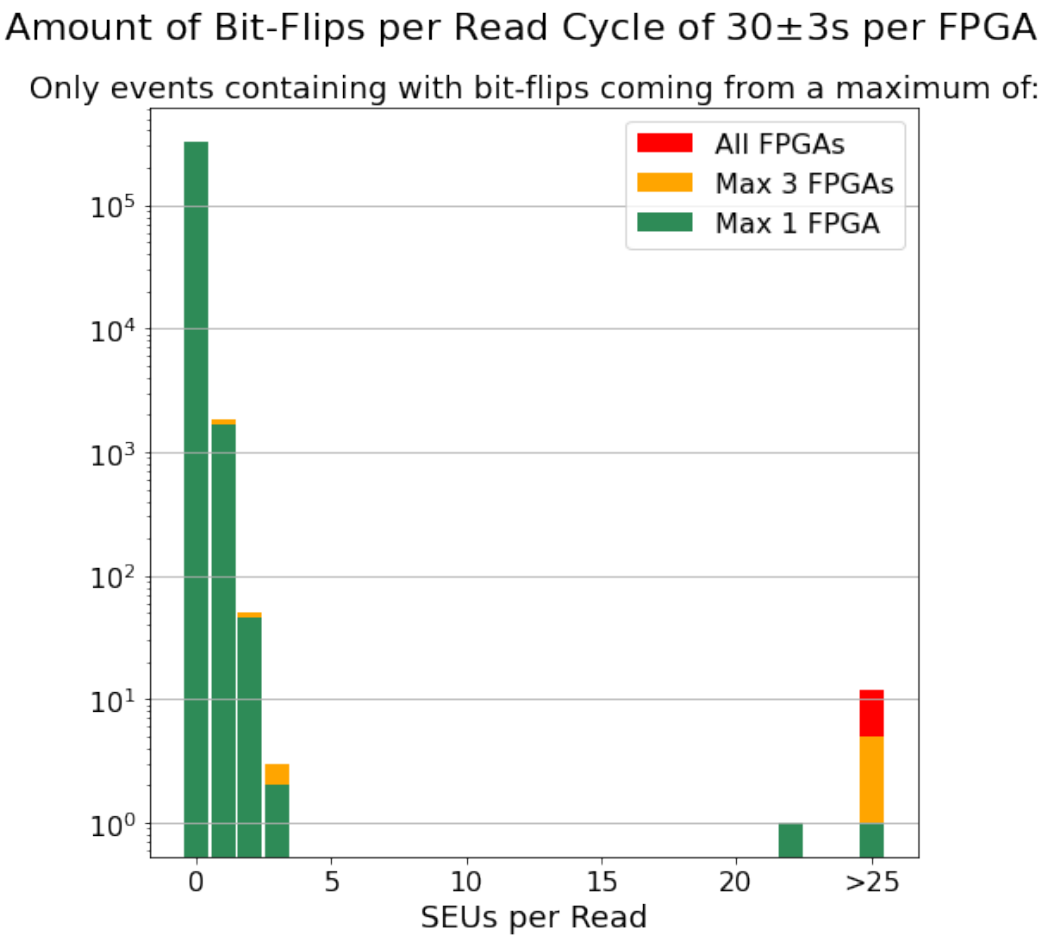


Figure 5.23: Histogram of the distribution of the number of bit-flips happening in the same reading cycle of (30 ± 3) s.

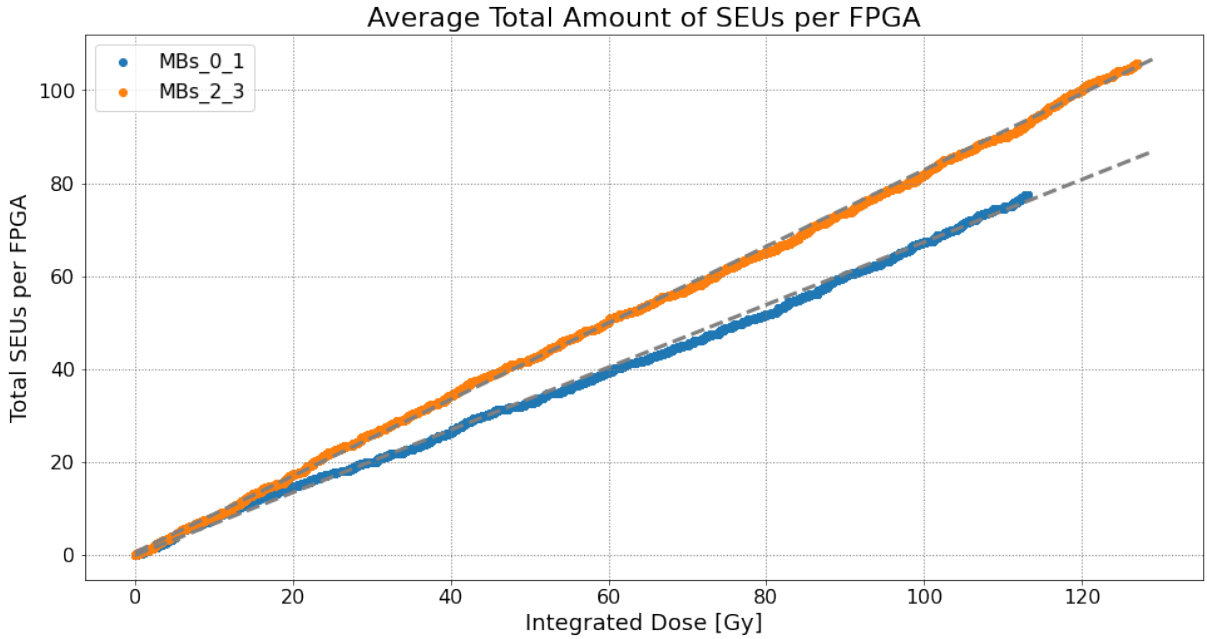


Figure 5.24: Average amount of RAM memory SEUs as a function of the integrated dose.

vast majority of the events with more than five bit-flips actually occurred in more than 3 FPGAs at once. Given the fact that a particle cannot generate SEUs in multiple FPGAs and that having SEUs happening in more than 3 FPGAs in the same readout cycle is extremely unlikely, one can safely assume that all the red bars in Figure 5.25 are not due to irradiation. Apart from resetting, reprogramming, and power cycling, another reason that might have caused such bit-flips in multiple FPGAs at once is the fact that the FPGAs' shift registers to measure SEUs (see Section 5.2) were driven by an external signal from the MiniDAQ in the control room. Any transmission error could trigger the bit-flip counter in multiple FPGAs. For that reason, only events in which the number of FPGAs with bit-flips in a read cycle is smaller or equal to three were considered. Such selection limits the distribution to the yellow bars in Figure 5.25.

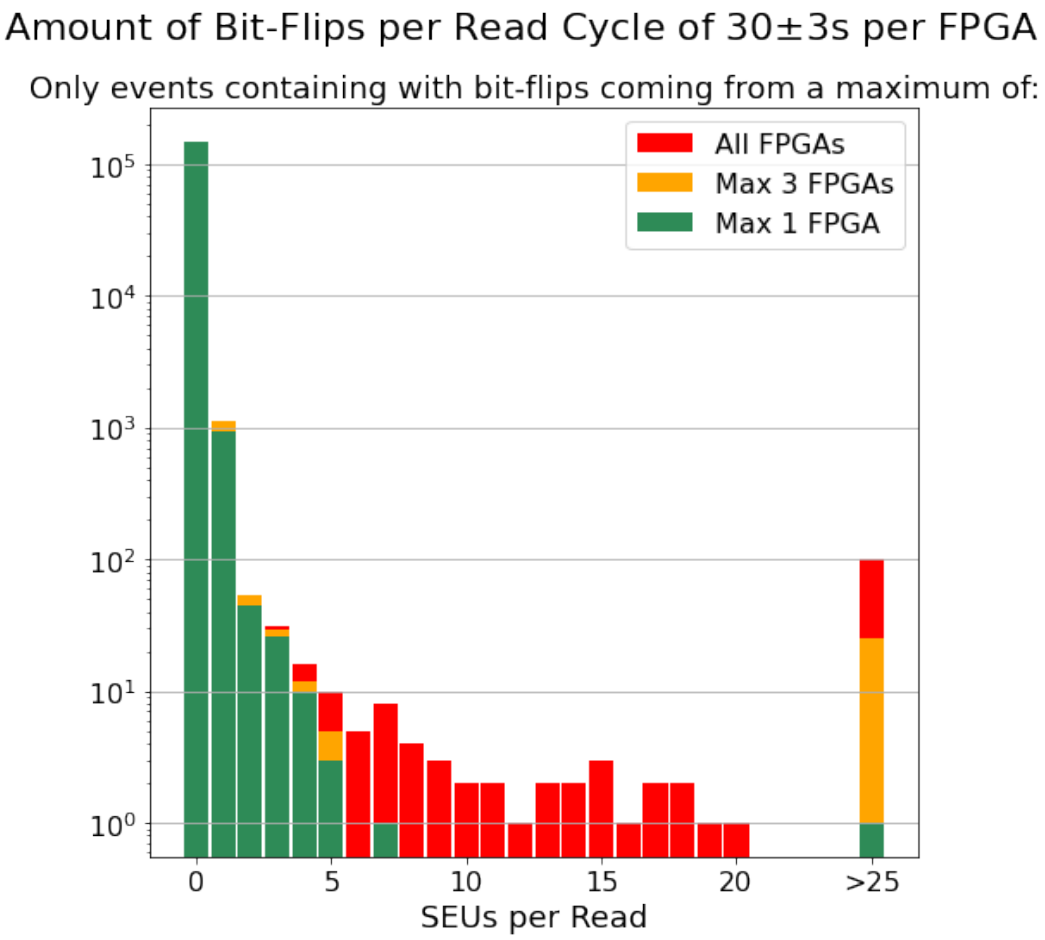


Figure 5.25: Histogram of the distribution of the number of bit-flips happening in the same reading cycle of (30 ± 3) s, with respect to the maximum amount of FPGAs that suffered a bit-flip in the said cycle.

In Figure 5.26, the average number of SEUs is shown as a function of radiation dose for the shift register using normal flip-flops and for those implementing Triple Modular Redundancy. In both cases, the relation is linear within the interval measured. The normal flip-flops presented a number of SEUs about five times larger than those implemented with triple redundancy.

The average amount of SEUs measured for each method (taking into account the uncertainty in the CHARM radiation measurement) can be found in Table 5.4.

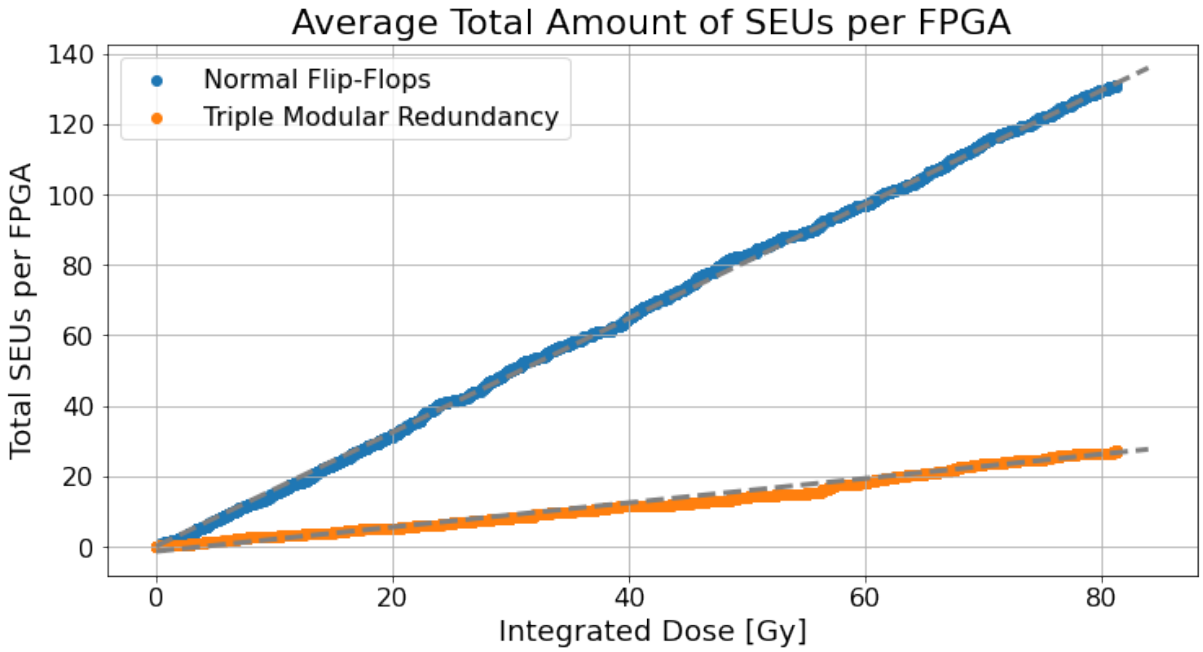


Figure 5.26: Average amount of flip-flop SEUs per FPGA over integrated dose. In blue: normal flip-flops. In orange: flip-flops implemented with triple modular redundancy.

Method	**SEU rate per flip-flop**
Normal Flip-Flops	$(6.1 \pm 2.1) \times 10^{-10}\,\mathrm{Gy}^{-1}$
Triple Modular Redundancy	$(1.3 \pm 0.4) \times 10^{-10}\,\mathrm{Gy}^{-1}$

Table 5.4: Rate of SEUs per flip-flop for both flip-flop implementations.

5.5.6 Single Event Latch-ups

During the two weeks of irradiation, 11 FPGAs received a total dose of 133 Gy, while 13 FPGAs received a total dose of 308 Gy.

A single event latch-up can partially or completely damage the device. However, during the whole extent of the tests, no FPGAs were damaged, and all of them completed the tests fully operational (with the exception of the nine FPGAs that failed reprogramming).

During the course of the tests, three times, an FPGA became unresponsive; in all three cases, after a power cycle, the FPGA was fully operational again. This could be due to a single event latch-up; however, even if that would have been the case, the DUT re-emerged unscathed.

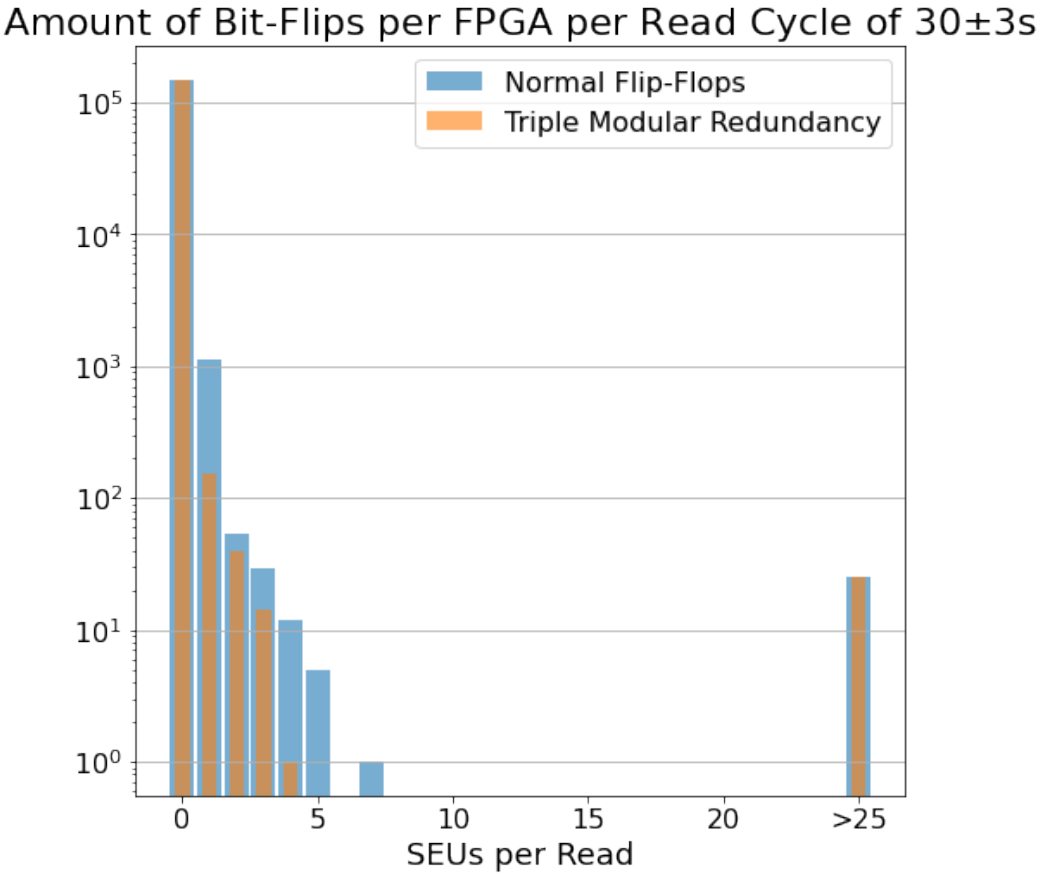


Figure 5.27: Histogram of the distribution of the number of bit-flips happening in the same reading cycle of (30 ± 3) s. In blue: normal flip-flops. In orange: flip-flops implemented with triple modular redundancy.

5.6 Conclusions

The radiation tests performed at CHARM were a successful campaign that lasted a month, including two weeks of irradiation that produced enough data for the desired results.

Regarding speed degradation, the tested FPGAs presented a degradation of (3.47 ± 1.37) % for 100 Gy of integrated dose. Degradations of such order present no risk to the experiment.

Regarding leakage current, it was not observed any significant increase in power consumption in any of the Master boards during the whole extent of the tests.

The programmability of the FPGAs will cease on average around a dose of 20 to 30 Gy. From 9 FPGAs tested, the earliest to fail happened after a dose of 22 Gy. Due to the lack of statistics, it is difficult to infer the variance of such a value, which means that from the 4608 FPGAs in the experiment, many could fail before 20 Gy. Therefore, if no FPGA loss at all is desired, reprogramming them after the electronics is exposed to radiation is not recommended.

Regarding single-event upsets, the rate at which bit-flips will occur in the FPGAs has been successfully measured, and this rate, when applied to the number of control registers per

FPGA, does not pose a risk to the experiment. Triple Modular Redundancy proved to be an efficient method to mitigate SEUs, reducing them by a factor of 4.7.

Regarding single-event latch-ups, no FPGAs were damaged due to SELs during the whole extent of the tests.

Therefore, concerning speed degradation, leakage current, single event upsets, and latch-ups, the SciFi front-end electronics has been fully qualified for operation under the radiation dose expected for the LHCb experiment during the run 3, within a factor two of safety.

Chapter 6

Beam Test for Characterizing the SciFi Module and Electronics

The first tests performed on mass-production SciFi modules and readout electronics using a particle beam are commonly referred to as "test beam". The test beam discussed in this dissertation consisted of the first "slice test" of two full-width scintillating fiber modules coupled to the SciFi front-end electronics. The data acquisition was performed with the so-called "MiniDAQ2", a preliminary standalone version of the LHCb PCIe40 DAQ System. The test was performed in the North Area of the CERN Prevessin site, using the beamline H8A from the SPS accelerator. This chapter illustrates the test beam objectives and the experimental setup, whereas Chapter 7 focuses on the software framework developed for the analysis of the test-beam data, and Chapter 8 presents the results of the analysis.

The main objective of the test beam was to understand the single-hit efficiency and resolution in various configurations. The experimental setup and its components are described in Section 6.1, the data format in Section 6.2, and the various settings tested during the test beam in Section 6.3.

6.1 Experimental Setup

The test beam setup could be broken down into four main components: the SciFi detector module, the TimePix 3 telescope [25], the MiniDAQ2 data acquisition module, and the trigger system. The SciFi module is the device under test. The telescope is a well-characterized detector used as a reference for the measurements performed by the SciFi module. The MiniDAQ2 provides the clock and the synchronization signals to the previous two components, as well as reads out and stores the data from the SciFi front-end electronics. The trigger system is composed of fast scintillators read out by NIM modules to detect when an event occurred and generate a trigger signal for the other components. Figure 6.1 shows a photo of the test beam setup, with an arrow indicating the position and direction of the particle beam and the x, y, and z arrows indicating the coordinate system.

Figure 6.2 provides a functional diagram of the test beam setup shown in Figure 6.1.

Figure 6.1: The experimental setup used in the SciFi test beam.

6.1.1 The SciFi Setup

Two separate 2.5m long halves obtained by cutting one complete SciFi production module were assembled and equipped for the test beam. Each half-module was equipped with a complete ROB (see Section 4.2): a cold-box with 16 SiPMs, two light injection mezzanines, and a complete front-end electronics box. The two half-modules were mounted in an aluminum profile frame at a distance of 140mm from each other in the beam direction. Figure 6.3 shows a drawing of the mechanical assembly used in the test.

In order to maintain stable the temperature of the electronics and of the SiPMs, a cooling system was used to provide chilled water through the aluminium water-cooling block of the front-end box and the Novec feed-through of the cold-box, respectively. A monitoring system was set up to monitor the temperatures during the test.

The position of the SciFi detector with respect to the beam was controlled by an electronically controllable table movable along the X and Y axis with sub-millimetric precision.

6.1.2 The Timepix3 Telescope

The Timepix3 Telescope [25] is a high-performance particle detector based on the Timepix3 ASIC [26], providing very precise measurements of the particle's Time-of-Arrival and

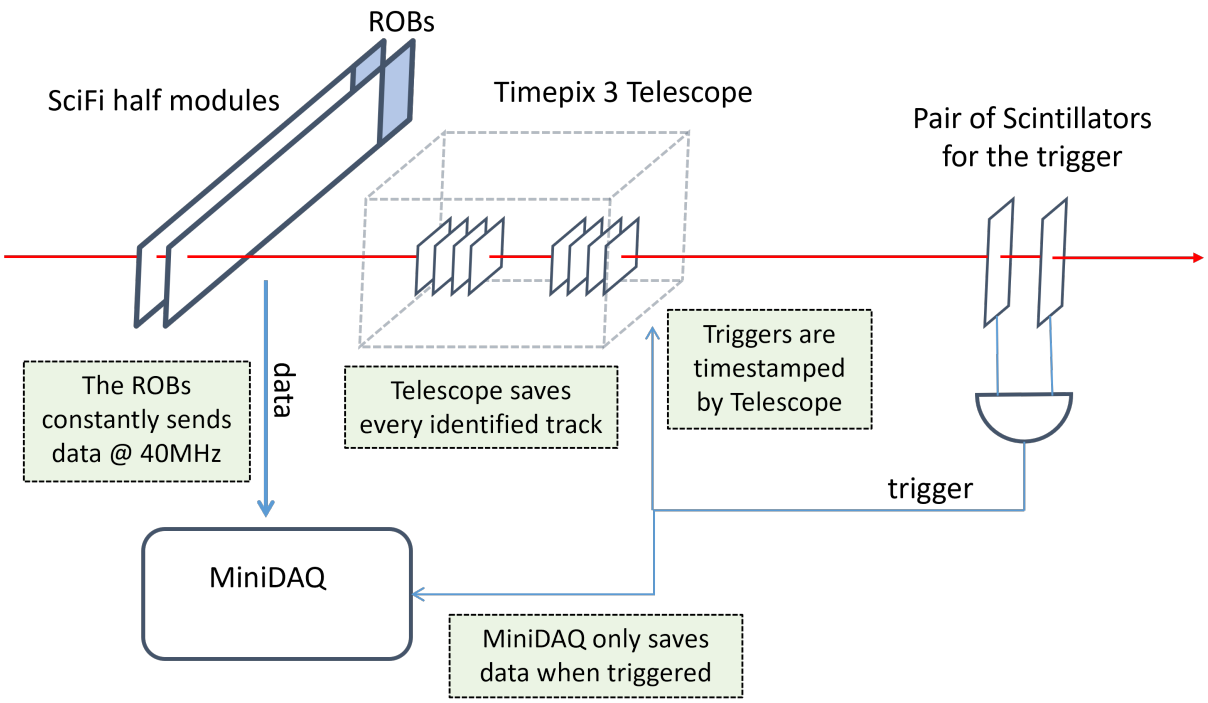


Figure 6.2: Functional diagram of the test beam experimental setup showing its main components and their interconnections.

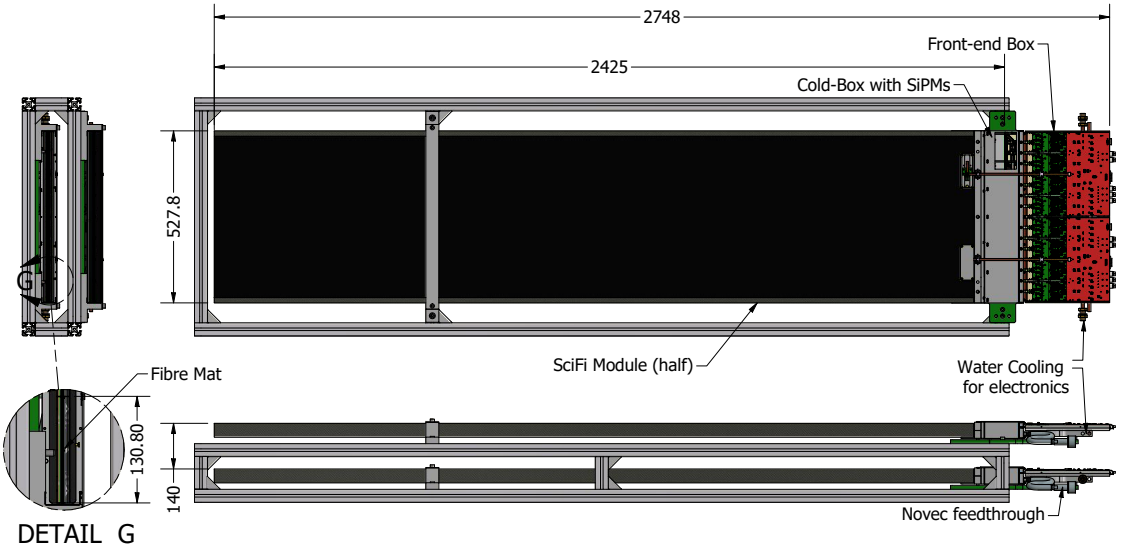


Figure 6.3: Mechanical setup supporting the two SciFi half-modules used during the test beam.

pulse height (Time-over-Threshold) over the eight detection planes contained in the telescope. It includes an analysis package combining the data from all the detection planes to reconstruct tracks with a resolution of less than 2 µm for particles passing close to the center of the telescope. Figure 6.4 shows the spatial resolution of the telescope for the X and Y coordinates as a function of Z. Each data point seen on these plots corresponds to one of the eight detection planes.

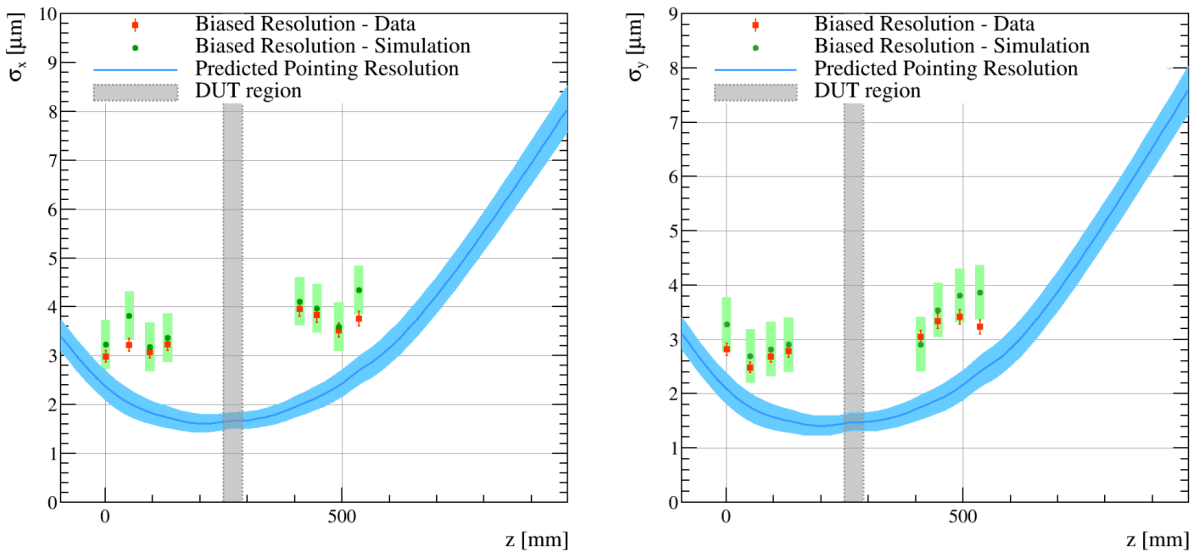


Figure 6.4: Spatial resolution of the Timepix3 Telescope as a function of the plane position. The plots on the left and right display the resolution in the X and Y coordinates.

The tracks provided by the telescope are used as a reference to search for hits in the SciFi modules. Figure 6.5 shows the mechanical design of the telescope, with its eight detection planes, each containing a Timepix3 chip denoted in the picture by a small square traversed by a straight line representing the particle beam. The whole structure is enclosed in the metallic box shown in Figure 6.1. The telescope analysis software extends the tracks from the telescope to the location of the SciFi modules. The two modules are 754 mm and 614 mm away from the zero position in the telescope, corresponding to Z = -754 mm and -614 mm in Figure 6.4. This track extrapolation slightly worsens the resolution, which at the position of the SciFi modules is approximately 13 µm.

For each track, the telescope also provides a timestamp associated with it, with a time resolution of about 280 ps. To be able to identify which telescope tracks match the events recorded by the MiniDAQ, the telescope generates an additional timestamp for the external trigger signal coming from the scintillators.

6.1.3 Trigger and Synchronization

The synchronization between MiniDAQ and the telescope is done by using the same signal as a clock for both systems. The telescope provides a TTL clock output which is converted to LVDS and fed into the MiniDAQ by one of the pins in the Molex connector. This is crucial to keep both systems in synchrony in order to allow the analysis software to match the events from the MiniDAQ to the timestamps provided by the telescope.

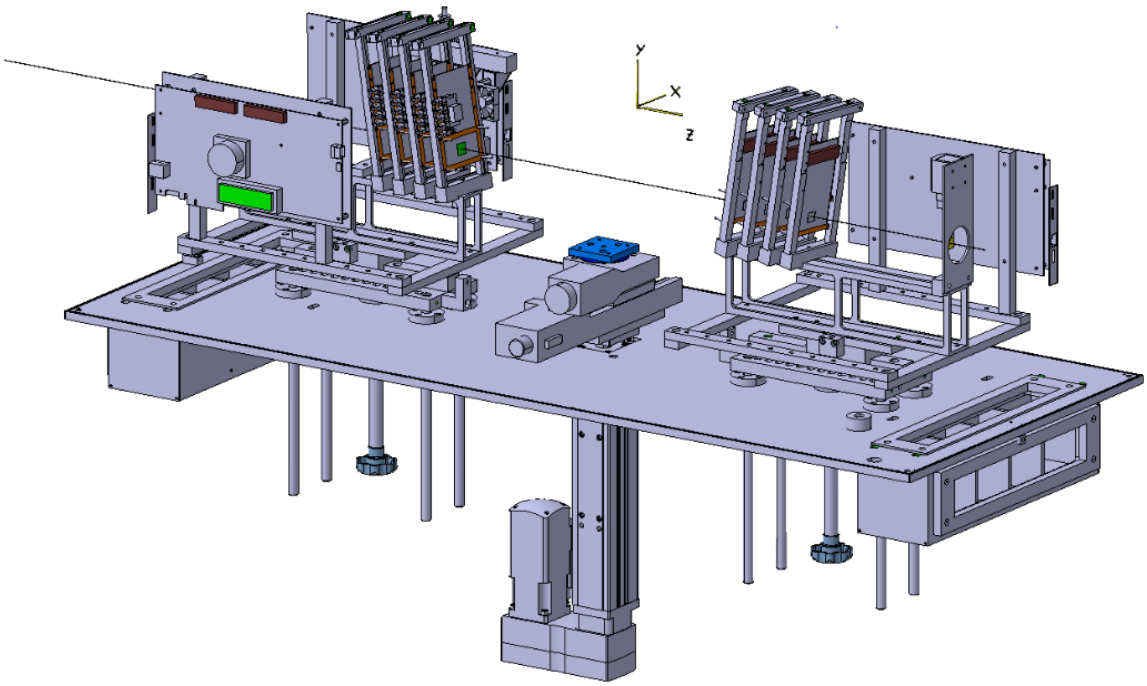


Figure 6.5: Mechanical design of the Timepix3 telescope.

The front-end electronics for the LHCb Upgrade were designed to send to the back-ends all recorded events at a rate of 40 MHz. In the test beam, however, we record spills of around five seconds, containing, on average, a million particles randomly distributed over time: during a single spill, the front-end will send, on average, around 200 million data events, of which most are actually empty events. Therefore a filter is needed to discard the empty and only keep the events of interest. The objective of the trigger system is to select only the events corresponding to the crossing of real particles.

An external input signal can be provided to the MiniDAQ2 module instructing the firmware whether to save an event to disk or not. In the test beam, this was chosen to be an external LVDS signal (fed into the MiniDAQ through a Molex connector with 17 pins) generated by the scintillators trigger logic implemented with NIM modules. This same signal (in TTL logic) was also provided to the telescope, which reads it out and adds a timestamp to it. A block diagram describing the trigger logic can be seen in Figure 6.6.

In order to allow both systems to start and stop recording events synchronously, a veto signal was implemented that prevented any trigger from reaching the MiniDAQ and telescope before both devices were configured and started acquisition. The veto signal is electrically generated by an Arduino board, driven by a Python script that, at the start of a run, sends the command to release the veto to the Arduino through a USB interface.

The signal is a coincidence between the signals of the two scintillators. A BUSY signal of 200ns is also implemented and fed with the trigger into an AND gate. It also allows us to send fake triggers from the control room as well as monitor the trigger output from the fan-out outputs.

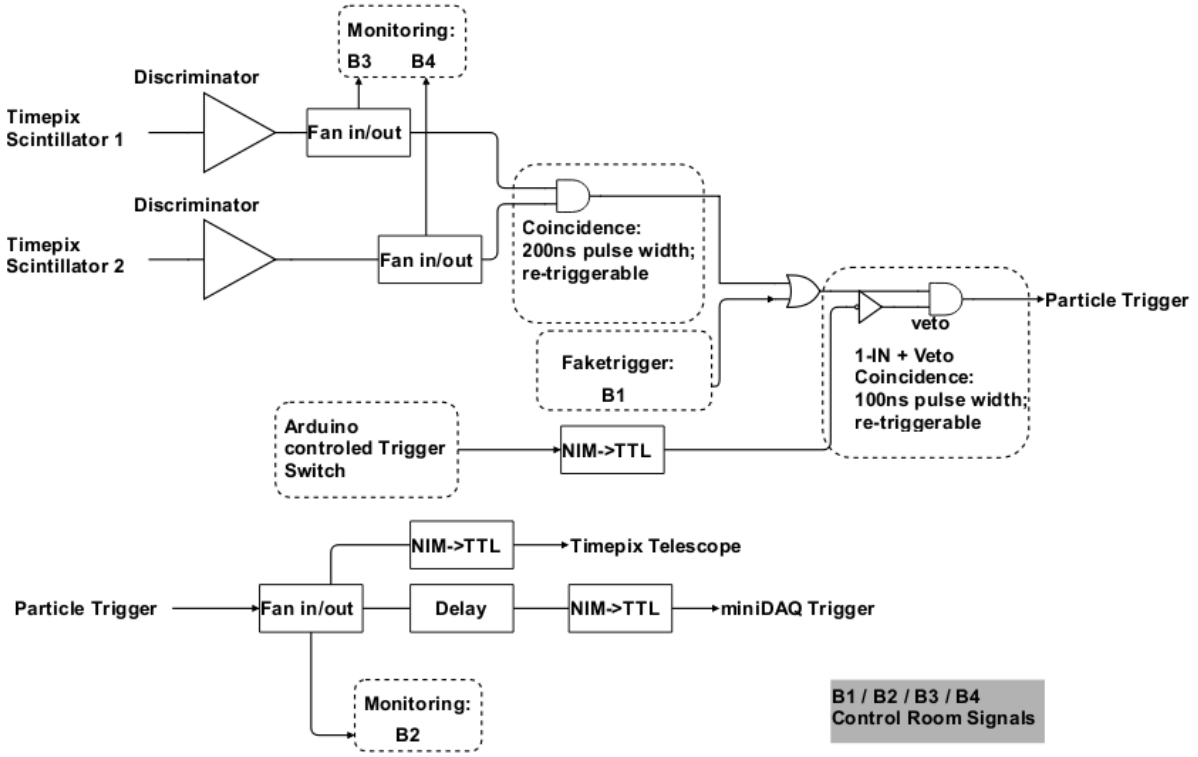


Figure 6.6: Block diagram of the trigger logic used during the test beam.

6.1.4 Data Acquisition

For data acquisition from the SciFi front-end, the MiniDAQ2 module was used. This is a stand-alone system containing a PCIe40 board (described in Section 3.1) that provides both the ECS and TFC commands required to control the LHCb front-ends as well as several GBT input links for data acquisition.

Data from the front-end electronics are transmitted over the GBT links directly to the event-builder server PC via the PCIe40 card. Each GBT link carries a frame of 120 bits at 40 MHz, resulting in a line rate of 4.8Gb/s per link. Physically, the optical links are packed in 12-fiber MPO ribbons at the input of the board; the optical-to-electrical conversion is handled by MiniPOD optical modules.

The GBT offers a "Widebus" mode, in which 112 bits of useful data are transmitted without any error correction mechanism. This mode offers a usable data rate of 4.48 Gb/s (instead of 3.2Gb/s with forward error correction). To optimize the data flow capabilities, the Scifi Tracker electronics community has decided to use this Widebus format to send data from the front-end boards to the PCIe40 boards. Such a mode was used during the full length of the test beam.

During each data run, the MiniDAQ2 writes the current event whenever a trigger is received from the trigger system. A data file is then generated containing the data from all such events. Figure 6.7 shows a summary of what data is saved every time a trigger is received. The data format is explained in more detail in Section 6.2.

The Timepix3 telescope only records the timestamp of the trigger received. The track data

has its own independent trigger system, which, for its part, generates its own timestamp. As both systems are synchronized with the common clock, it is possible to match the timestamp of the track with the timestamp of the trigger, as they will always have a fixed time delay with respect to each other. However, it is possible that, whereas the scintillators detect a particle and generate a trigger that translates into a MiniDAQ event and a trigger timestamp, the telescope does not detect such particle and therefore does not write the track data. The opposite is also possible. These cases do not happen often. Many data runs recorded millions of events without a single event being missed by any of the systems. When this happened, it was corrected by the event matching software described in Section 7.1.

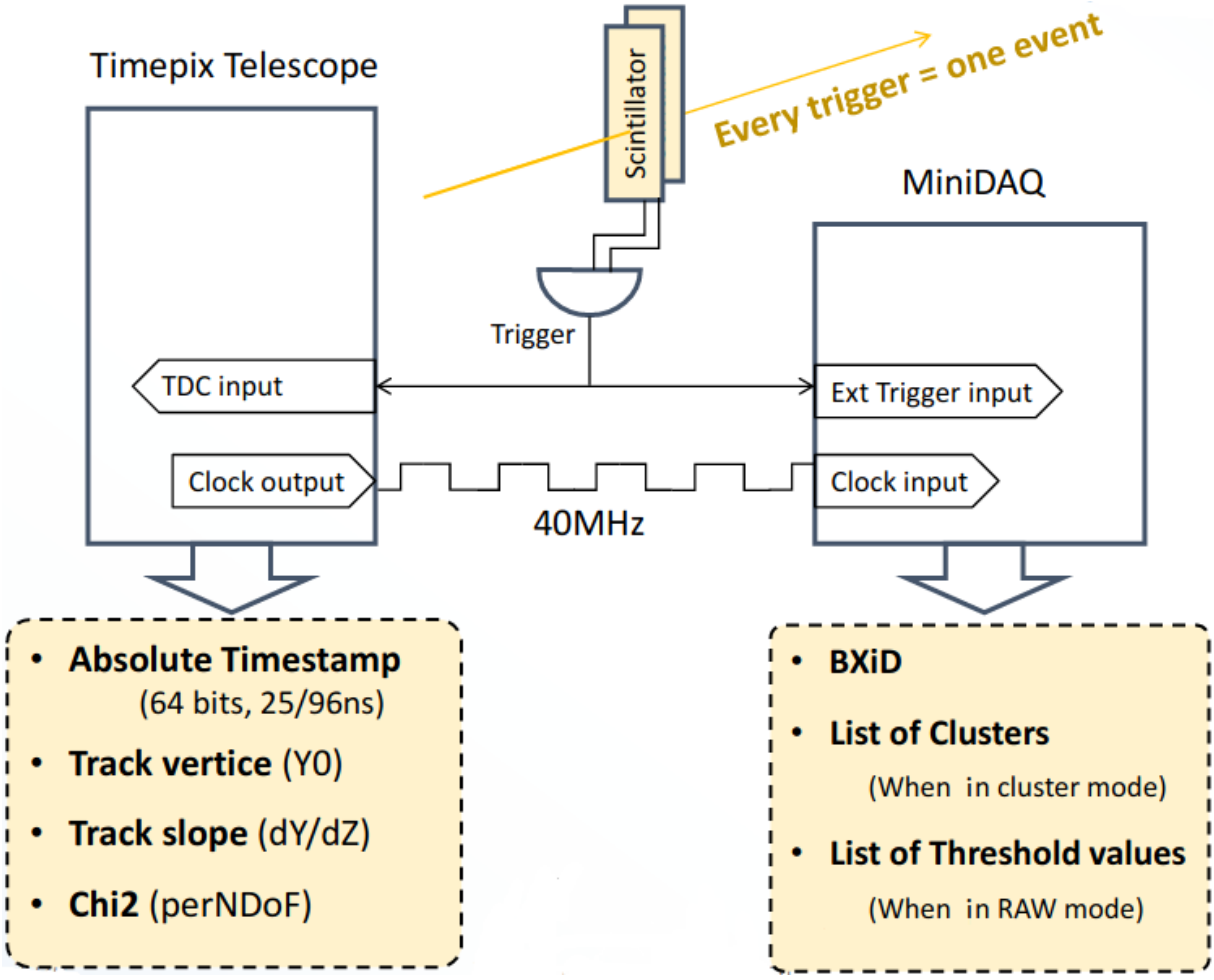


Figure 6.7: Diagram representing a trigger generated from a particle passing through the scintillators and the data that is saved as a consequence of such trigger.

6.2 Data Format

As previously described, for each passing particle, the trigger system generates a trigger, and an event is saved to disk by the telescope and by the MiniDAQ, as shown in Figure 6.7. In this section, it will be described the format of those data.

6.2.1 MiniDAQ Data

Two different SciFi data formats have been used during the test beam when reading out the data: the cluster and the raw data format. Both formats share a common 20-bit header, as shown in table 6.1.

Table 6.1: Bit allocation in common header used in both raw and cluster data format.

Bit(s)	Name	Description
111:100	Bunch crossing ID	12 bit bunch crossing ID within orbit. Ranges from 0 to 3653.
99	Raw data format flag	0 for cluster and 1 for raw data format.
98	Parity bit	Parity of 19 other header bits.
97	TFC flag	Set to 1 when receiving a TFC command.
96:92	Cluster count	Number of formed clusters.

The raw data format contains the highest possible amount of information about the hit signals: for each channel, 2 bits are read out that encode the number of PACIFIC thresholds that have been exceeded. However, since only 112-20 bits = 92 bits are available, for each SiPM at 40 MHz, only the raw data from 46 channels can be transmitted. Therefore each SiPM is split into four "windows" of 32 channels: only the data of one such window can be read out in a given run. The detailed bit allocation of the 92-bit payload of the raw data format is shown in table 6.2.

The cluster data format is the default format; up to 10 clusters per SiPM per bunch crossing can be transmitted. The cluster position was encoded with 8 bits; resulting in a precision of half SiPM channel. The position is computed by taking the mean of the hit channels, weighted by the number of passed PACIFIC thresholds - the so-called "barycenter" of the cluster. An ordinary cluster can consist of up to 4 neighboring channels. In the case of more than four involved channels (as seen in Figure 4.13), two positions are returned that encode the beginning and the end of the cluster. To distinguish this *big* cluster from two ordinary clusters, a *big* flag is set together with the given end position of the cluster. This results in a total of 9 bits per cluster (8 bits to encode the position itself in units of half SiPM channels and 1 bit as a flag to indicate the end of a *big* cluster). The detailed bit allocation of the 92-bit payload of the cluster data format is shown in table 6.3.

By default, each trigger signal causes one event to be written to disk. In addition, there is the possibility to also record the events of the previous and following bunch crossings. This mode is called TAE (**T**ime **A**lignment **E**vents); the number of extra events recorded before and after is adjustable by means of the *TAE half window* parameter. Data taken in TAE mode allows for performing spillover studies, as well as aligning the recorded data with the trigger and telescope system.

A software parser to read the recorded data, available as `scifi40_frgreader` in the git repository *https://gitlab.cern.ch/lhcb-scifi/lhcb-scifi40-software*, was used to turn the acquired data into the ROOT TTrees that are used as input for the Analysis Framework Software described in Chapter 7.

Table 6.2: Bit allocation in the payload of raw data format.

Bit(s)	Name	Description
91	TFC Sync flag	Raised when received TFC Sync command.
90	TFC HdrOnly flag	Raised when received TFC HdrOnly command.
89	TFC NZS flag	Raised when received TFC NZS command.
88	TFC SnapShot flag	Raised when received TFC SnapShot command.
87	TFC FERst flag	Raised when received TFC Front-End reset.
86	TFC CalibB flag	Raised when received TFC Calibration B trigger.
85	TFC CalibA flag	Raised when received TFC Calibration A trigger.
84	TFC BcntRst	Raised when received TFC Bunch crossing ID reset.
83:76	0xCD	Reserved: Fixed to "0xCD".
75:68	ChBlock	Quarter of SiPM channels to which raw data belongs. 0x00: SiPM Channels 0-31 0x01: SiPM Channels 16-47 0x10: SiPM Channels 32-63 0x12: SiPM Channels 48-79 0x20: SiPM Channels 64-95 0x23: SiPM Channels 80-111 0x30: SiPM Channels 96-127
67:64	0x0	Reserved: Fixed to "0x0"
63:62	Raw data Ch[31]	Exceeded thresholds of the 32nd channel within quarter.
61:60	Raw data Ch[30]	Exceeded thresholds of the 31st channel within quarter.
59:58	Raw data Ch[29]	Exceeded thresholds of the 30th channel within quarter.
...	...	...
3:2	Raw data Ch[1]	Exceeded thresholds of the 2nd channel within quarter.
1:0	Raw data Ch[0]	Exceeded thresholds of the 1st channel within quarter.

Table 6.3: Bit allocation in the payload cluster data. The least significant bit in the 9-bit cluster data holds the *big* flag, while the remaining 8 bits encode the cluster position with a precision of half the SiPM channel.

Bit(s)	Name	Description
91:90	Reserved	Not in use.
89:81	Cluster 1	If applicable: Position and big flag of the 1st built cluster.
80:72	Cluster 2	If applicable: Position and big flag of the 2nd built cluster.
71:63	Cluster 3	If applicable: Position and big flag of the 3rd built cluster.
62:54	Cluster 4	If applicable: Position and big flag of the 4th built cluster.
53:45	Cluster 5	If applicable: Position and big flag of the 5th built cluster.
44:36	Cluster 6	If applicable: Position and big flag of the 6th built cluster.
35:27	Cluster 7	If applicable: Position and big flag of the 7th built cluster.
26:18	Cluster 8	If applicable: Position and big flag of the 8th built cluster.
17:9	Cluster 9	If applicable: Position and big flag of the 9th built cluster.
8:0	Cluster 10	If applicable: Position and big flag of the 10th built cluster.

6.2.2 Telescope Data

The data from the telescope are provided to the test beam user in the format of ROOT TTrees already containing the processed events, i.e., the recorded trigger, timestamp, and track parameters but not the raw data. Such processing is done by an offline reconstruction software called "Kepler", based on the GAUDI event-processing framework [27], which uses the raw data to provide track fitting, alignment, and charge calibration, among other tasks. Reference [25] provides a detailed description of the tasks performed by the Kepler software.

The telescope outputs a ROOT file containing two TTrees named Trigger and Tracks. The Trigger TTree stores the trigger information recorded by the TDC input, which was used to timestamp the signal from the scintillators. The Tracks TTree records a variety of parameters about the reconstructed tracks. The parameters used by the SciFi Test Beam Analysis Framework are described in Table 6.4.

Table 6.4: Data contained in the TTrees provided by the telescope software.

TTree	Parameter	Description
Trigger	TgTime	Timestamp of the trigger signal received.
Tracks	TkX	X axis coordinate of the track origin.
Tracks	TkY	Y axis coordinate of the track origin.
Tracks	TkTx	Slope of the track over the X axis.
Tracks	TkTy	Slope of the track over the Y axis.
Tracks	TkChi2PerNdof	Chi^2 value per degrees of freedom of the track fitting.

6.3 Data Runs and Settings Tested

A total of 436 data runs have been recorded during the test beam. Many of these were done in order to set up and calibrate the hardware and therefore were not used in the analysis. For all the others, a number of one million events was set as a target of accumulated data per setting. Figure 6.8 shows the run plan used for the test beam, which summarizes all the settings used for the data runs. The main variables tuned during the beam tests were the impact position of the particles along the SciFi module (X-position in Figure 6.8) and the front-end chip settings (PACIFIC shaper settings in Figure 6.8).

For the analysis described in this thesis, only the runs in raw data mode were used, as they provide better precision when compared to cluster mode. The data run selection is described in Chapter 8.

Y-position	X-position	PACIFIC shaper setting : threhold set	Data mode	Cluster algorithm
mirror	33 × positions full module scan @ Module 0 (A1-A4 mats)	Feb_g3 : 15/25/45	Cluster/Raw	old version
	9 × positions @ Module 0	Aug_g3 : 15/25/45	Cluster/Raw	
	1 × position @ A3 mat	Feb_g3: 15/25/45, 05/15/25, 35/55/95 Aug_g3: 15/25/45, 05/15/25, 35/55/95	Cluster/Raw Cluster-TAE Raw-TAE	
130cm-from-mirror	1 × position @ A3 mat	Feb_g3: 15/25/45, 05/15/25, 35/55/95 Aug_g3: 15/25/45, 05/15/25, 35/55/95	Cluster/Raw Cluster-TAE Raw-TAE	
	9 × positions @ A3 mat	Feb_g3: 15/25/45 Aug_g3: 15/25/45	Cluster/Raw	
mirror	1 × position @ A3 mat / B2 mat	Feb_g3: 15/25/45	Cluster/Raw Cluster-TAE Raw-TAE	new version
	33 × positions full module scan @ Module 0 / Module 1	Feb_g3: 15/25/45	Cluster/Raw	
		1M raw events for each setting, SiPM @ 3.5V overVoltage		

Figure 6.8: Table summarizing all setting combinations that were used during the test beam.

Chapter 7

Test Beam Analysis Framework

In this chapter, the Analysis Framework developed to analyze the test beam data will be described in detail. The goal of this framework is to provide all the necessary tools to perform the analysis of individual runs and to generate the desired global results using all the runs of interest.

The framework developed has been adopted by the SciFi Test-Beam Group to perform all analyses and obtain the results demonstrating the performance of the SciFi detector.

The Framework can be divided into three main parts:

- **The Event Matching Software**

  It is the software that matches the events in the MiniDAQ TTree (SciFi data) with those in the Trigger TTree and the Tracks TTree (telescope data) and generates, as a result, a lookup-table which will be used by the Run Data-Analysis software. The Event Matching Software will be described in detail in Section 7.1.

- **The Run Data Analysis Software**

  It is the software that reads the data from a specific run, using the lookup-table from the Matching Software as a reference, and computes results for this specific run. It produces the plots of interest and writes the run-specific results in a text file. The Run Data-Analysis software will be discussed in Section 7.2.

- **The Results Plotter**

  The Results Plotter is a collection of Python libraries and Jupyter Notebooks that use as input all the run-specific results of interest generated by the Run Data-Analysis and generates the results that are the final goal of the test beam. The Results Plotter will be illustrated in Section 7.3.

7.1 Event Matching Software

The Event Matching Software is a c++11 program responsible for generating a Root TTree [28] to be used by the Run Data Analysis Software as a lookup table indicating

which track and trigger events from the telescope TTrees correspond to which events in the MiniDAQ TTree. The software was able to match 99.99% of the events from all the runs of interest, i.e., all runs that had no issues reported in the logbook. The only exception was run 30216, which failed the matching for unknown reasons; however, as such run was a redundant run taken after two consecutive runs with the same settings, no time was invested into understanding why the events from run 30216 could not be matched.

The need for event matching exists because the data from both recording devices (telescope and MiniDAQ) are not necessarily aligned in time. Despite the effort taken to release the veto electronically in a way that both systems start and stop counting triggers simultaneously in order to count the same amount of events, it was difficult to prevent one of them from missing an event. As the MiniDAQ events don't have a timestamp, it is challenging to know which events from one device correspond to the ones in the other. Additionally, the telescope acts as two independent devices when recording triggers and tracks, making it possible for the telescope to miss a trigger but not a track and vice-versa.

7.1.1 Time Alignment Between Trigger and Track events

Matching the events in the Triggers TTree with those in the Tracks TTree is straightforward since both systems share the same clock and are, therefore, synchronous, and both systems record an absolute timestamp with a resolution of 260 picoseconds. The timestamp due to a particle recorded by the track system is always shifted by the same amount of time with respect to the timestamp due to the same particle recorded by the trigger system, with the exception of a few rare events where such time shift is recorded to be approximately 3 ns larger. This points to a small systematic error that happens in the trigger generation or in the timestamp readout, however, this will not affect the analysis results as this measurement is used only by the event matching. Figure 7.1 shows a histogram of the time difference between the trigger and track events caused by the same particle, the mean of which was measured to be 261.5 ns. For that reason, a certain trigger event is considered to have a corresponding track event if its timestamp is within an interval of 25 ns from the trigger event timestamp added by 261.5 ns.

7.1.2 Time Alignment Between Telescope and MiniDAQ

Matching the telescope events with the MiniDAQ ones is more complicated due to the fact that the MiniDAQ events have no absolute timestamp, instead, they just register the bunch ID (BXId) of the crossing in which the particle arrived. The BXId works as a 25 ns clocked counter. However, it is reset at 3564 (number of bunch crossings in an LHC orbit) or approximately 89.1 µs; consequently, two consecutive MiniDAQ events with, e.g., BXIds 100 and 200, may not necessarily have happened 100 BXIds (or 2.5 µs) apart from each other, as an unknown amount of BXId resets might have happened in between them. Moreover, the MiniDAQ might have missed some events. The telescope data is then used to infer the amount of time passed between MiniDAQ events and then finally match them to the telescope ones. The method for matching the events is described below.

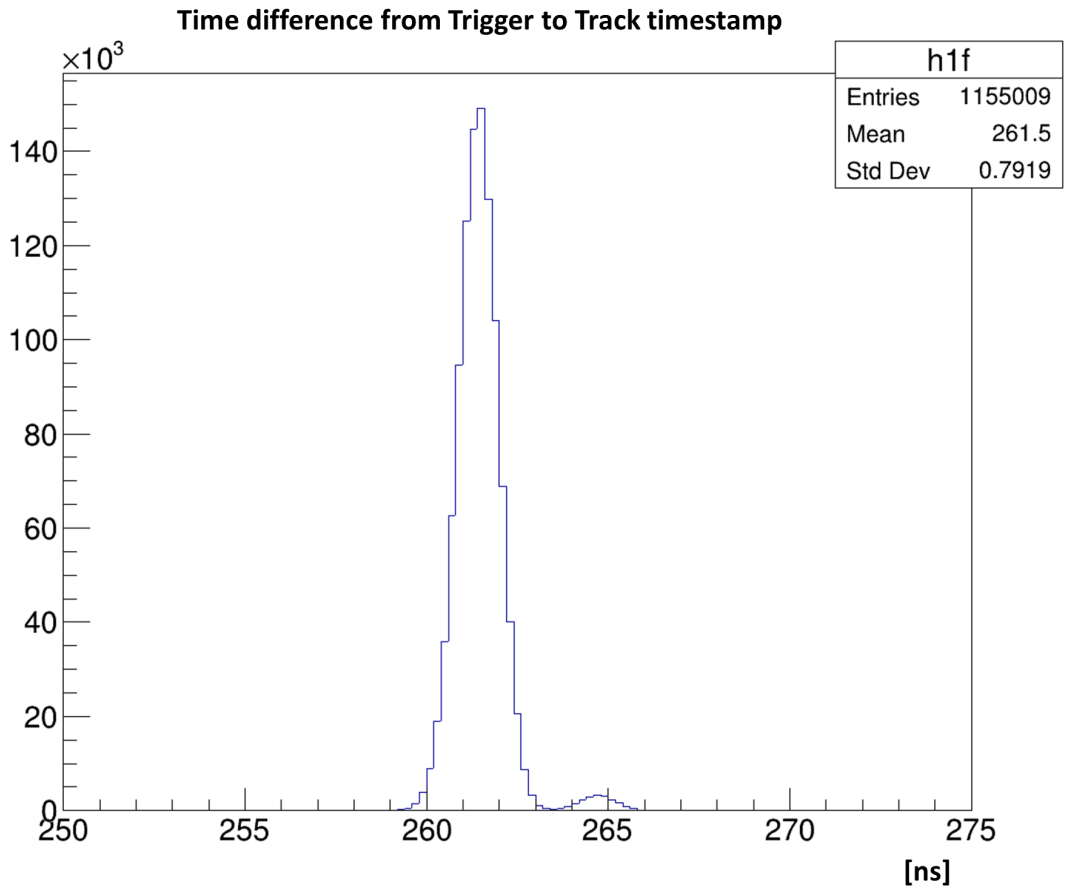


Figure 7.1: Time difference between the recorded timestamp of a trigger event to the timestamp of its corresponding track.

Let's consider the ideal case in which neither the telescope nor the MiniDAQ have missed any event, and let's use the telescope timestamp (t_0) of the first event as our time reference. The first MiniDAQ event will have a BXId corresponding to t_0, which we will refer to as "initial_offset". As long as no events are missing, the quotient of the telescope's timestamp by the BXId reset time ($3564 \times 25\,\text{ns}$) will always return as the remainder the BXId of the corresponding MiniDAQ event minus the initial_offset. Therefore, once the initial_offset is found, one can verify that pairs of telescope-MiniDAQ events are correctly matched by verifying that:

$$BX_{\text{SciFi}} = ts_{\text{tele}} - 3564 \times 25 \cdot \left(\frac{ts_{\text{tele}}}{3564 \times 25}\right) + BX(t_0) \tag{7.1}$$

The initial_offset $BX_{\text{SciFi}}(t_0)$ is found by getting the remainder of the quotient of the telescope's first event's timestamp by the BXId reset time ($3564 \times 25\,\text{ns}$):

$$BX_{\text{SciFi}}(t_0) = t_0 - 3564 \times 25 \cdot \left(\frac{t_0}{3564 \times 25}\right) \tag{7.2}$$

Then, it is confirmed by checking that Equation 7.1 is satisfied in the next 5 events. As the veto-releasing system was implemented electronically (see Section 6.1.3) and happens

simultaneously (in the same clock cycle) on the MiniDAQ and the telescope, it is very unlikely that the first event will be misaligned. In fact, all the runs of interest had no misaligned "beginning" (i.e., the first five events).

Once the Initial_offset has been found, the software progressively checks if the offset is still consistent for each pair of events. As long as the events from both sources are aligned, 7.1 should always hold. If it finds an inconsistency, it can only mean that either the telescope or the MiniDAQ have missed an event or, more rarely, that there was a data transmission error.

If the inconsistency is due to transmission errors or any phenomenon that might change the header of the data, only this specific pair of events will be inconsistent, but the following ones will still be aligned. To verify this possibility, the software evaluates the consistency of the following 5 events without skipping any events on both sides. If the next events are still inconsistent, it assumes that one of the devices missed an event and proceeds with the realignment routine.

From the moment in which one of the devices misses an event, all following events will be misaligned by one event. The software can correct for this misalignment by skipping one event on the side that did not miss the event. To find out which side missed the event, the software skips one event on one side and checks if the offsets of the next 5 events are consistent with 7.1 again. If it's not, it repeats the same procedure for the other side. To address the possibility of one of the devices having skipped multiple events in a row, the software will proceed to repeat the said procedure but now skipping two events instead of one. It will iterate this procedure, incrementing the number of events to be skipped, until it either recovers consistency with 7.1 or reaches the limit of events to be skipped.

The Event Matching Software then performs the matching by adding all the consistent pairs of events to a vector, each element of which contains three integers corresponding to the entry numbers of the MiniDAQ, the trigger, and the track events at their respective data TTrees. Figure 7.2 illustrates how the entries of the data TTrees are matched in this lookup table. Once it finishes looping through the whole data, it then saves the vector to disk, as described in the next section.

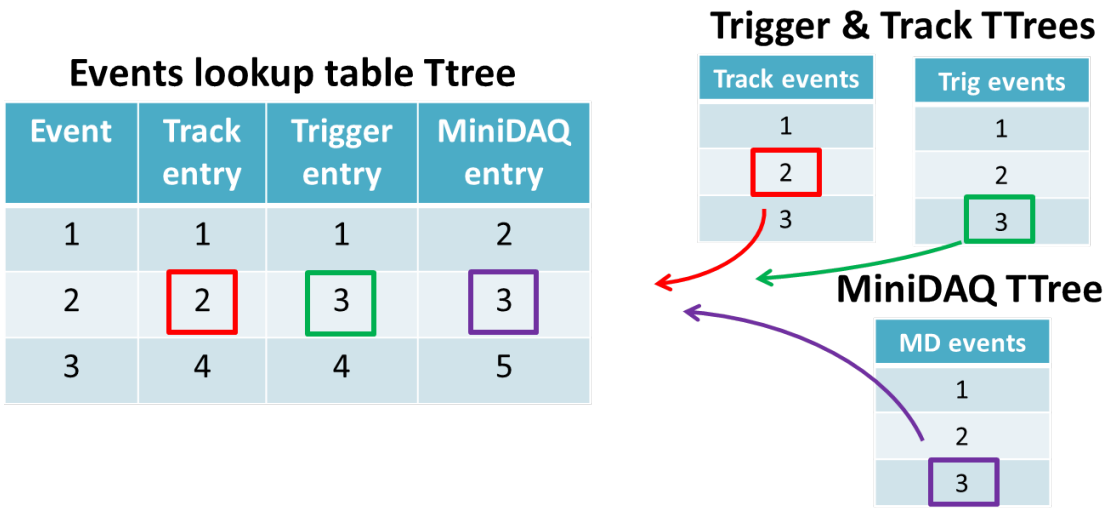


Figure 7.2: Illustration of the event matching mechanism.

7.1.3 The Matching Output File

The Event Matching Software outputs a ROOT file that contains one directory for each run for which events have been matched. All the run settings stored in a .ini file by the acquisition software during the data taking are copied over into the ROOT file in the sub-directory called "ini_params". The MiniDAQ, Trigger, and Track events that have been matched have their respective entry numbers stored in branches under the sub-directory named "lookup_table".

The structure of the output ROOT file can be seen in Figure 7.3 ("run_params" and "alignment" are obsolete).

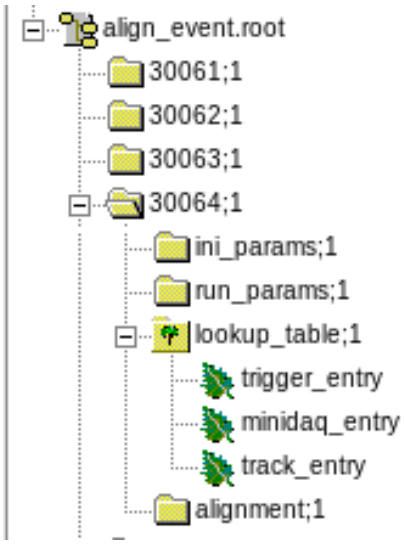


Figure 7.3: Structure of the event matching TTree.

7.2 Run Data Analysis Software

The Run Data-Analysis Software is a c++17 program performing the analysis of individual runs of the SciFi 2018 test beam. It takes as input the run number and a configuration file containing all the parameters needed for the software to run. As output, it generates a result file comprising the run settings, the analysis parameters, and the analysis results. Furthermore, several plots are generated according to the list of analyses being performed.

The objective of this software is to provide the results of interest, for each individual run, in a way that the Results Plotter software can use them as input to perform higher-level analysis and produce global plots and global results (efficiency, resolution, etc.) of the test beam analysis.

The analysis parameters can be passed on the command line and/or through a configuration file. Those include the paths to the data files, the lookup ROOT file, the run number, the hardware serial numbers, the number of events to be analyzed, the list of analysis classes to be called, and the parameters specific to the analysis performed by those classes.

The Run Data-Analysis Software adopts Object Oriented Programming. The main constituents of the software are:

- **The run data structure:** contains all run data and is described in detail in Section 7.2.1.
- **The EventHandler class:** reads the data files and stores the data in the run data structure. The EventHandler is explained in Section 7.2.2.
- **The analysis classes:** a list of classes derived from the Analysis Base Class, which receives the data event by event and performs the analyses. The Analysis Base Class is explained in Section 7.2.3. Once the run data structure is fully loaded by the EventHandler, the software calls the analysis classes one by one in the order they appear in the configuration file. Those classes are responsible for tagging events, applying cuts and filters, and computing results by looping over all the events in the run data structure and finally saving the results and plots to disk. An overview of all the analysis classes is given in Section 7.2.4.

7.2.1 The run data structure

It is the structure that contains the data types to store all the test beam settings, the analysis parameters, and the events data, as shown in Figure 7.4. A single object is instantiated at the beginning, containing the data structure and all the methods necessary to the Event Handler and subsequently to all the analysis classes.

7.2.2 Class EventHandler

The EventHandler class is the one actually loading all event data into the Data Events List in Figure 7.4. The EventHandler uses the lookup file created by the Event Matching Software to know which are the entries of the matched events, both on the Telescope and MiniDAQ data files. It then loads the actual data from such events into the Data Events List of the data structure. It also reads the test beam settings from the .ini file generated by the test-beam DAQ software during the data taking and stores them into the Run Info of the data structure.

7.2.3 Analysis Base Class

The Analysis Base Class is the class from which all analysis classes are derived. The objective is to provide a standard set of virtual functions of the same type, which allows the software to have a list of classes (as a std::vector) through which it can loop. Then, for each analysis class, the virtual methods derived from the Base Class will be invoked.

The virtual methods contained in the Analysis Base Class are listed below:

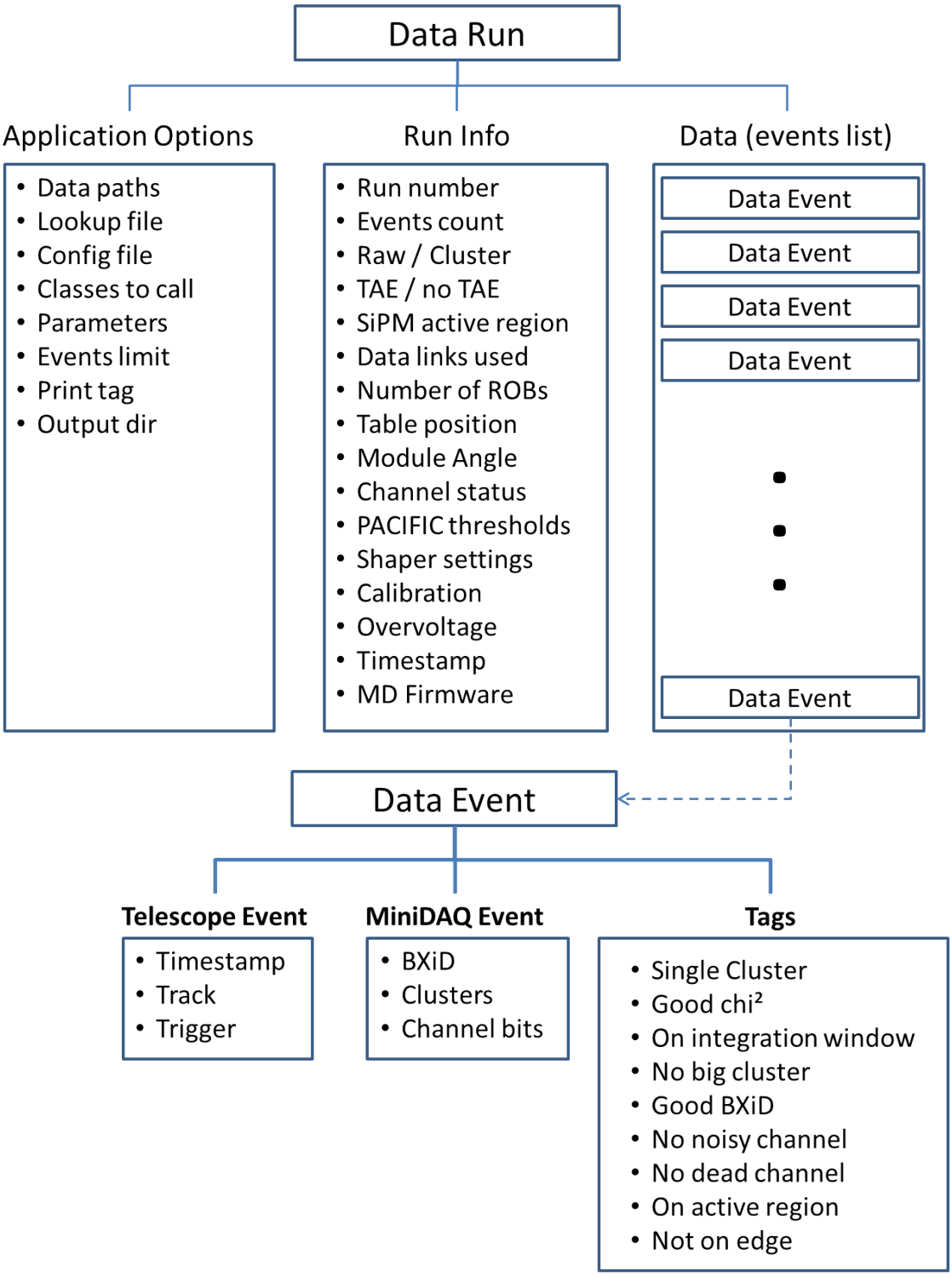


Figure 7.4: Simplified overview of the run data structure.

- **nametag:** returns a string that identifies which analysis class it is. This enables the user to choose which classes are going to be called when executing the software. The software only calls the methods specified by the user, either in the configuration file or through the command line.

- **addOptions:** notifies to the caller parameter values, allowing the developer to easily assign values to the parameters of a newly created class, both from the command

line or from the configuration file.

- **process:** executes whatever the class is supposed to do with the data. For classes that generate results, all computations should be done in this method.
- **plot:** if the class generates plots, this method is responsible for generating such plots from the results.
- **save:** if the class generates results, this method saves such results to the results text file.

7.2.4 List of Analysis Classes

The analysis classes used for the work described in this thesis are listed below, followed by a brief explanation of their purpose:

- **ClusterFilter:** filters off the clusters from a noisy region caused by a bad SiPM (whenever the datatype is not "raw").
- **Clusterization:** performs the offline clusterization from the raw data. More details are given in Section 7.2.5.
- **PreTagger:** tags the events matching the following criteria: events with a single cluster, events with track fitting chi^2 within a certain range, and events within the good range of BXIds. The tags are later used by other analysis classes to apply selections to the data. The ranges of such tags can be defined through the program options.
- **TWB:** Stands for "Time Within Bunch-crossing". It is the class responsible for finding the time width of the telescope TDC bins and generating weights for them, to be used when searching the PACIFIC integration windows. It is described in more detail in Section 7.2.6.
- **IntWin:** finds the limits of the plateau of the PACIFIC integration window and tags the events that are within such limits. This procedure is described in Section 7.2.6.
- **Alignment:** finds the alignment of the SciFi module with respect to the telescope. It finds values for the rotation around the beam axis, the translation along the beam axis, and the translation along the vertical axis (perpendicular to the SiPM channels). The alignment procedure is illustrated in Section 7.4.
- **Residuals:** computes the residual for every event containing a cluster and a track. Later used to compute the spatial resolution.
- **PosTagger:** It is the class responsible for tagging the events in which a track has passed close to the SiPM middle gap, to the SiPM edges, into a noisy channel, or into a dead channel. The maximum distance from each of these is a parameter of the class, which can be set from the configuration file or through the command line interface when invoking the analysis software. Some later analysis classes filter such

events in order to avoid any bias on the results caused by clusters that are touching the edges of the SiPM die, or that would include a noisy or dead channel. The method and the criteria for selecting which channels are treated as noisy or dead are discussed in Section 7.5.

- **Resolution:** computes the spatial resolution for a specific run. How this is done is described in detail in Section 7.2.5.

- **Efficiency:** computes the overall single hit efficiency for a specific run. (See Section 7.6.2).

- **ChannelStudies:** computes the statistics of channel hits and outputs it to the results text file. This is used for inferring noisy and dead channels. This process is explained in Section 7.5.

- **Weights:** generates the results needed to study how different sets of weighting coefficients affect the clusterization algorithm. (See Section 7.7).

7.2.5 Offline Clusterization

As explained in Section 4.4.3, clusterization is the process of grouping signals from adjacent channels into a single value that represents the best estimate of the position at which a particle hits the detector. In the experiment, the clusterization algorithm will run online, that is, as part of the process of collecting and processing data by the front-end boards. "Offline clusterization" refers to the case in which this process happens after data taking, using data saved to storage.

The motivation for implementing an offline clusterization is to study the effects of different clusterization methods applied to the same data set. For that to be possible, one needs to collect data in raw mode (no zero suppression) in order to apply the clusterization algorithm by software at a later stage. For that reason, in all configurations tested during the test beam, data were collected in raw mode.

The code used for the analysis presented herein has been derived from the class *FTFP-GAClusterCreator* belonging to the project Boole [21] from the LHCb software repository, with only a few differences. The main differences are listed below:

- The code uses modified data types with respect to the original, a reduced version from which many parameters used only for simulation have been removed.

- The code takes as an input a list of the noisy channels and ignores any data coming from such channels. This is done to prevent creating clusters from channels that are constantly firing. The list of noisy channels comes from the study described in Section 7.5.

The Clusterization Class

This is the analysis class created to perform the offline clusterization. It encapsulates the offline clusterization code into a method, called for every event of the run data as part of the main program loop.

The class can receive many parameters, either from the command line interface or from a configuration file:

- Clusterization weighting coefficients: these are the weights to assign to each PACIFIC threshold level when the clusterization algorithm is calculating the barycenter of the cluster.
- Automatically set coefficients: this option allows the software to select the best set of weights for a given PACIFIC threshold setting instead of using a manually chosen set of weights. This value comes from a configuration value, containing the set of weights to be used for each of the three threshold settings used during the test beam.
- Use fractional bit: when enabled, this option will cause the algorithm to use a single bit of precision for the fractional part of the cluster position, emulating what is done in the FPGA implementation of the algorithm. When disabled, the algorithm will use a floating point to represent the cluster position, therefore having more precision for the fractional part.
- Keep edge clusters: option to consider also the clusters created on the SiPM edge and central dead areas.
- Cluster maximum width: option to set the maximal cluster width.

Moreover, after clusters are created for an event, the class also tags the event with at least one of the following tags: *notEmpty*, *singleCluster* and *noBigCluster*, which respectively means an event with at least one cluster, an event with a single cluster and an event with no big clusters.

7.2.6 Determining the PACIFIC Integration Window

As the PACIFIC chip digitizes the detector signal by integrating the hit charge in a given time window, it is of great importance that we filter the events keeping only those in which particles have hit the detector at a moment in which the full charge could be integrated by the PACIFIC. This is not necessarily the case because the detector readout has been designed to operate synchronously with the LHC collisions, such that particle hits will always be digitized within the PACIFIC integration window. However, in the test beam, particles arrive randomly with respect to the internal readout clock.

Arrival Time Within the Clock Cycle

To measure the time at which a hit signal has arrived with respect to the SciFi clock, we used the fine timestamp provided by the telescope. This timestamp has a resolution of 260 ps (a 96th of a clock cycle). The distribution of the particle arrival time with respect to the SciFi clock exhibits the structure shown in Figure 7.5; the picture is for run 30268, but the structure is the same for all the runs of the test beam.

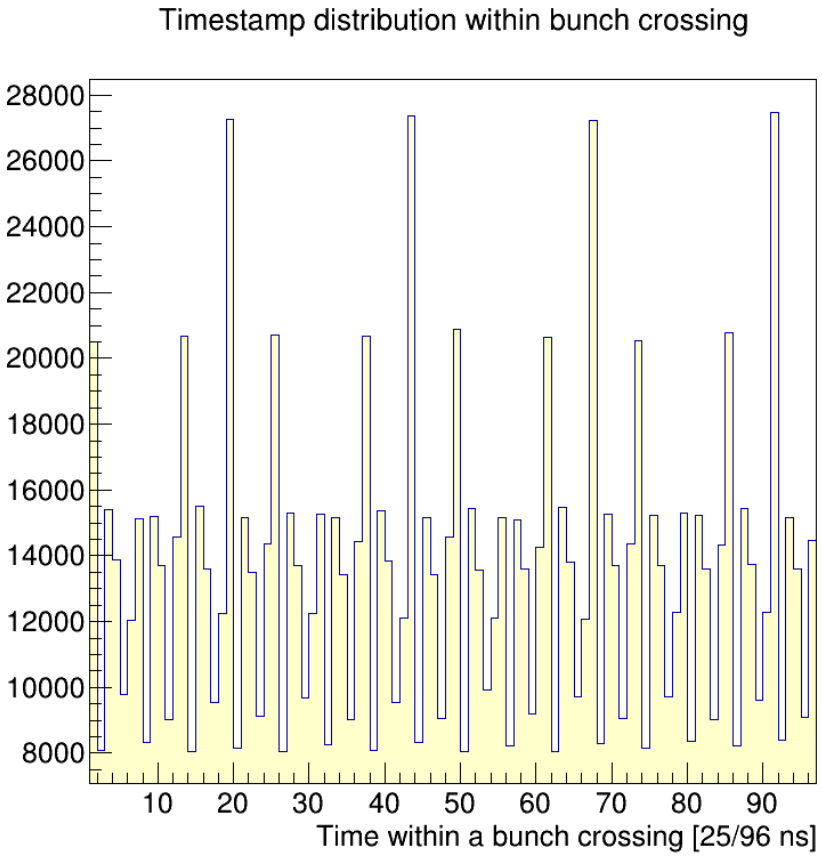


Figure 7.5: Histogram of the hit arrival time with respect to the Scifi clock for run 30268.

The beam time structure is such that one expects particles to arrive randomly distributed over time during the spill and, therefore, one would expect a uniform distribution instead of the structure in Fig 7.5. The structure of Figure 7.5 is due to the fact that the width of the time bins of the Telescope's TDC is not uniform. In order to correct for that, a list of weights was generated for each of the 96 time bins.

The class *TimeWithinBx* creates the time distributions like that in Fig 7.5, generates the list of weights per time bin, and assigns to each **SEILA** the time bin corresponding to the time of arrival of the particle. This information is then used by the next class, *IntegrationWindow*, to decide whether the hit time was within the PACIFIC integration window.

Setting the Integration Window Limits

In order to decide which events are within the PACIFIC integration window, we use the class *IntegrationWindow*, which defines a "plateau" in which events are efficiently integrated and tags all events fulfilling such conditions. This way, successive analysis classes can filter the events using this tag.

The *IntegrationWindow* class determines the time distribution of those hits that generated a cluster. It also corrects for the unevenness of the telescope TDC bins using the weights previously determined. An example of this distribution is shown in Figure 7.6. In the ideal case in which the PACIFIC would be able to integrate all charges efficiently regardless of their arrival time, this distribution would be flat. The effect of the limited PACIFIC integration window is clearly visible in Fig. 7.6. The actual integration window can vary significantly depending on certain settings in the front-end electronics, especially the PACIFIC shaper settings, which can extend or shorten the signal length. However, the pattern in Fig. 7.6 is quite general: there is a plateau in which the events are efficiently integrated.

Note that the fact that the shape of the integration window measured in the test beam (Figure 7.6) does not resemble a plateau is solely due to the position of the plateau within the clock cycle: in the test beam, the plateau happened to be on the edges of the SciFi clock period.

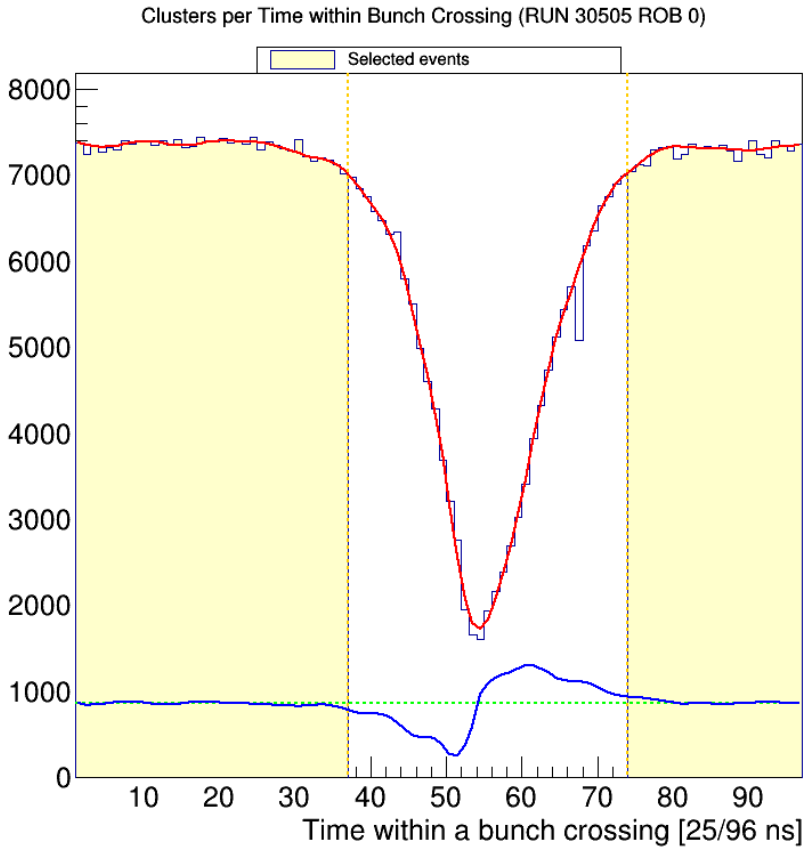


Figure 7.6: Histogram of cluster time distribution for run 30505 ROB 0.

Using the histogram in Fig. 7.6, the class proceeds to find the limits of the plateau. It

first smooths the histogram by a factor of 10, using the *TH1D→Smooth()* method. The result is denoted by the red line. Then, starting from somewhere in the center of the plateau, the algorithm scans the bins in both directions until it finds the 95% bin: either with content less than 95% of the difference between the maximum and the minimum of the histogram or with a derivative of the smoothed line (denoted by the blue line) higher than 1% of the difference between the maximum and the minimum of the histogram. The plateau limits are conventionally defined between the two 95% bins.

The class then runs through all events, tagging those that are within the plateau. This allows the next analysis classes to select only those events that have been efficiently integrated, as would happen in the LHCb experiment.

7.2.7 The Residuals Class

The *Residuals* class is the one responsible for finding the residual distribution of the clusters. A residual is defined as the difference between the position of which the particle should actually hit and the position measured by the SciFi.

The former is predicted by extrapolating the track from the detection plane of the Telescope to the plane of the SciFi detector. The residual is then defined as

$$\text{Residual} = y_{\text{track}} - y_{\text{SciFi}} \tag{7.3}$$

where y_{track} is the hit prediction and y_{SciFi} the cluster position measured by SciFi, Figure 7.7 illustrates the process. In order to find the residual distribution, the class loops over all the events of interest, computing the individual residual and filling a histogram. The histogram should follow a Gaussian distribution centered around the beam position.

The class stores the residual for each run (necessary to compute resolution) and the standard deviation as well as the mean of the residual distribution (used by the alignment procedure, see Section 7.4).

As the SciFi Module measures only one coordinate (Y in this case), only the track projection on this axis, at the point at which the track intersects the SciFi Module plane, is relevant for our analysis. This calculus has to take into account the angle θ_{SciFi} and the distance z_{SciFi} of the SciFi Module with respect to the Telescope, which is given by the alignment parameters, explained in Section 7.4. The track projection

$$y_{\text{SciFi}} = \cos\theta_{\text{SciFi}} \cdot \left(\frac{dy}{dz} \cdot z_{\text{SciFi}} + y_0\right) + \sin\theta_{\text{SciFi}} \cdot \left(\frac{dx}{dz} \cdot z_{\text{SciFi}} + x_0\right) \tag{7.4}$$

where $(\frac{dx}{dz}, \frac{dy}{dz})$ are the x, y slopes of the Telescope track and (x_0, y_0) is the track projection on the Telescope plane.

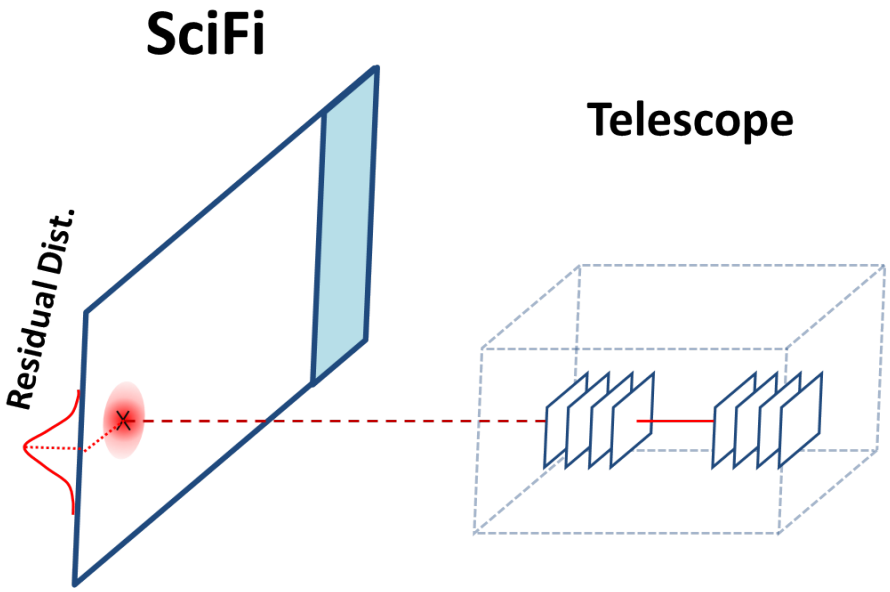


Figure 7.7: Illustration of how the residual distribution is computed for a run.

7.3 Results Plotter

The Results Plotter is a collection of Python libraries and Python-based Jupyter Notebooks to be used as a tool for plot creation and analysis at a higher level than the Run Data Analysis Software, described in Section 7.2. In particular, data from multiple runs are used to perform the analysis. It uses Python 3, the Pandas framework for data handling, and the library Matplotlib for data visualization.

It provides a variety of libraries with common code to be used by the notebooks:

- **resultTypes:** a python file containing a dictionary with the declaration of all the fields and respective types contained in the results file produced by the Run Data Analysis Software. It is not a class storing the data, but rather it works as a recipe for reading the results and creating the data structure to store its values.
- **data_loader:** a library with functions to read the data from the results file into a Pandas dataframe. It uses the *resultTypes* dictionary to identify which values should be read into the dataframe and for each of these, it creates a column with the appropriate type. It receives as a parameter the path to the results folder from the Run Data Analysis Software and automatically reads all the result files present in the result folder into the dataframe, creating a new row for each run.
- **plots_generic:** is a library containing functions to create graphical content (plots, color maps, etc.)
- **tb2018_plots:** is a library containing functions to generate many of the plots used by the Jupyter Notebooks.

The Jupyter Notebooks are the ones responsible for creating the final results of the analysis:

- **Alignment_Check:** it is the notebook that groups the runs with the same settings together to verify the quality of the software alignment. It is the one used for the procedure explained in Section 7.4.

- **Channel_Studies:** it is the notebook responsible for determining noisy and dead channels. It is the one used for the procedure described in Section 7.5.

- **Settings_listing:** it is the notebook created to identify and list the different settings for the test beam in order to provide a better overview of the analysis.

- **Resolution_Studies:** it is the notebook responsible for determining the position resolution for different settings and studying the dependency of the resolution on specific settings.

- **Weights_coarseScan:** it is the notebook that studies the resolution impact of the weights used in the clustering algorithm, discussed in detail in Section 7.7

- **Weights_fineScan:** it is the same as the previous one but uses a different set of data with a finer scan of the weights.

7.4 Software Alignment

Alignment can be described as the process of finding the actual position of the detectors with respect to the beam. The Telescope has its own alignment procedure that is carried out after the data is processed for each individual run. In order for the Analysis Software to properly match hits from the Telescope to the SciFi module, it is necessary to know the position and angle of the SciFi module with respect to the Telescope detector coordinates more accurately than directly measured during the installation in the test area. In Figure 7.8 it is illustrated the coordinate system used in the test beam. This section describes how the Analysis Software determines the correct alignment parameters.

7.4.1 Generating the Alignment Parameters

The software aligns the SciFi module using as degrees of freedom the translation along the Y and Z axes and the rotation over the Z axis. Note that, as the SciFi measures only the Y coordinate, small angles of rotation around the X and Y axes, as well as translation along the X axis, do not significantly affect the results.

It is important to note that even though the Telescope has not moved at all during the test beam period, and even though the SciFi modules only shifted along the Y and X axes by means of a precision table onto which the modules had been mounted, the Telescope processing software still generates (possibly slightly different) alignment parameters for each run. Because of that, it is necessary to generate the SciFi alignment values for every run that is analyzed, even if it is two consecutive runs with the exact same settings, taken minutes apart from each other.

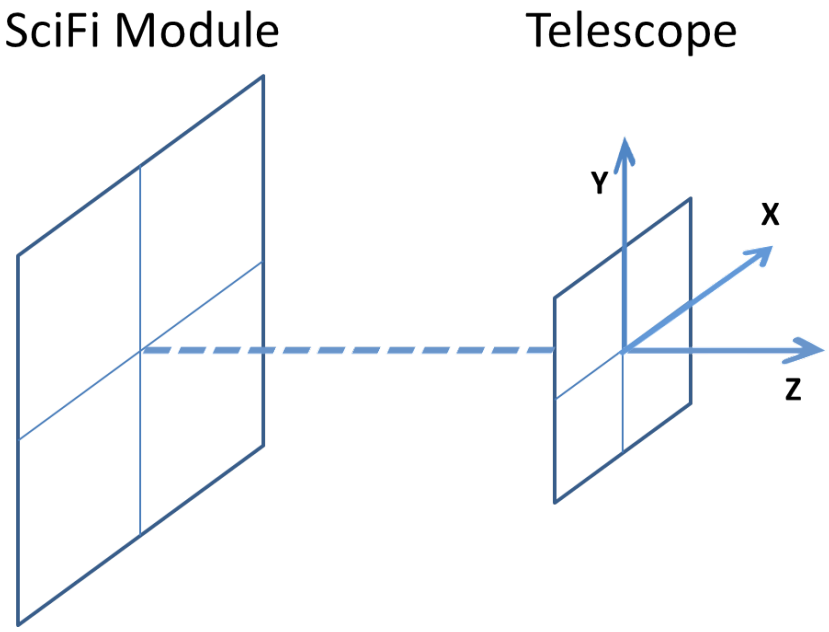


Figure 7.8: Coordinate system used by the Telescope and SciFi Module.

Linear Alignment over the Y axis

The correct position along the Y axis is obtained from the mean of the residual distribution by using the class *Residuals* as described in Section 7.2.7.

The mean of the residual distribution might not correctly represent the beam position and be slightly biased if the beam position is close to one of the ends of the module. This is because many clusters are formed far from the beam center (due to noise, scattering, etc.), and if the beam is closer to one end of the module, there will be an artificially larger number of clusters on the opposite side. To avoid this effect, the software runs the *Residuals* once considering the whole extension of the detector in order to find the approximate location of the beam, then runs the *Residuals* again but only considering a region of 20 mm around the previously found beam position, then runs it yet another time around 1 mm of the inferred beam center. For each iteration, the mean of the residual distribution will be closer to the actual beam position because fewer clusters formed far from the beam center will be taken into account. The result from the third and last iteration is considered to be the alignment value for Y.

Angular Alignment around the Z axis

The correct value for the Z angle of the SciFi module (see Fig. 7.9) is determined by varying the Z angle used in the analysis over a range of six degrees around the origin in order to find the angle that gives the smallest standard deviation for the residual distribution in that region.

The minimum standard deviation of the given range is found by successive approximations. First, three initial values (two in the extremities and one in the center) are taken,

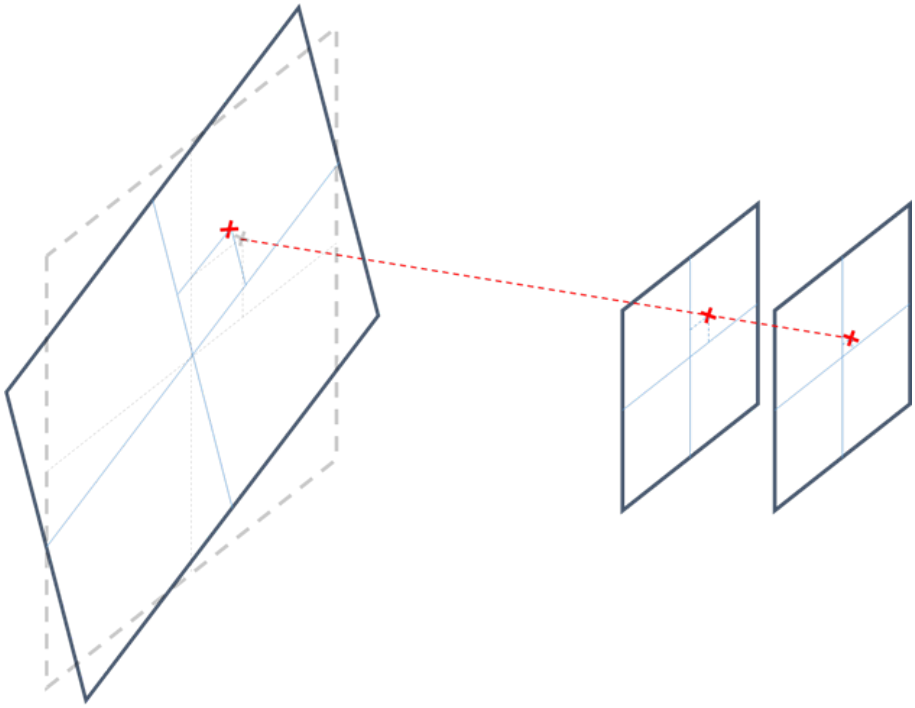


Figure 7.9: Illustration of the angular alignment around the Z axis by rotating the module.

then the middle between the two lowest points is taken and considered to be the new center value. The procedure is then repeated until the difference between the lowest points is less than 1 nm. Once this is achieved, the angle of the detector is considered to be the lowest point.

Figure 7.10 shows an example (run 30518 ROB0) of the standard deviation of the residuals vs the alignment.

Linear Alignment along the Z axis

The distance from the SciFi Modules to the Telescope detection plane along the Z axis has been fixed during the whole test beam. However, the Telescope's alignment software also tries to align the relative distances among its sensors (to counteract thermal expansion, among other effects). A wrong spacing of the sensors would alter the track slope.

Figure 7.11 illustrates how this would happen. The blue lines on the right side represent the position where the Telescope software thinks its detection planes are. The dashed orange line is the real position. As one can see, if the detection planes get closer to each other, the slope of the tracks increases. Such effects can be effectively corrected by shifting the SciFi module closer to the Telescope plane.

To find the optimal point for the translation over the Z axis, the same logic of the angular alignment applies, and the same algorithm of successive approximation is used, which generates plots like that in Figure 7.12.

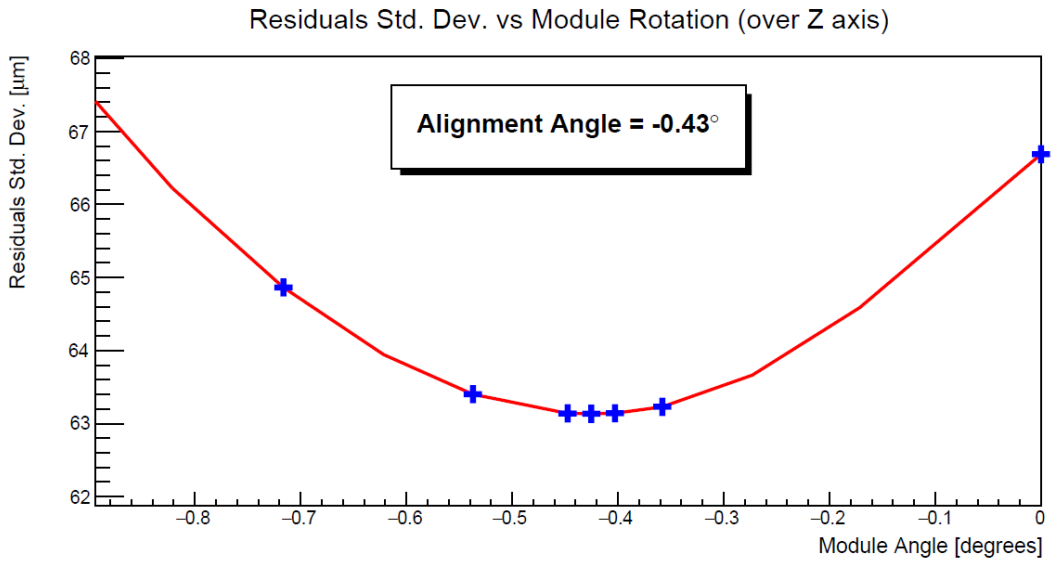


Figure 7.10: Standard deviation of the residual distribution as a function of the module angle.

Alignment order

The alignment is performed in the following order: translation along the Y axis, rotation around the Z axis, and translation along the Z axis. After that, translation in the Y axis is performed again to find precisely the beam position over the module.

7.4.2 Verifying the Alignment

To verify the results of the alignment procedure, we look at the standard deviation of the residuals distribution (see Section 7.6.1) and assume that the correct alignment will give the lowest standard deviation.

After applying the alignment parameters determined with this method, the resolution improves as expected, as shown in Figure 7.13 for all the raw runs of the test beam. The dots are the resolution results before alignment, while the crosses are the resolution after the alignment.

While the improvement is evident, the run-to-run variation of the results is still quite big. It is very important to understand whether these variations are due to bad alignment or to the actual changes in test beam settings.

In order to answer this question, all the runs with the exact same settings were grouped together. For each of these groups, the average resolution was computed. With a perfect alignment and no other source of errors, the variation among runs with the same settings should not be significant. A total of 52 runs were separated into 22 groups with common settings. Figure 7.14 shows the results of this analysis: the red dots denote the resolution before the alignment procedure, and the blue X that after the alignment.

Figure 7.15 illustrates how the resolution varies in groups of runs with the same settings.

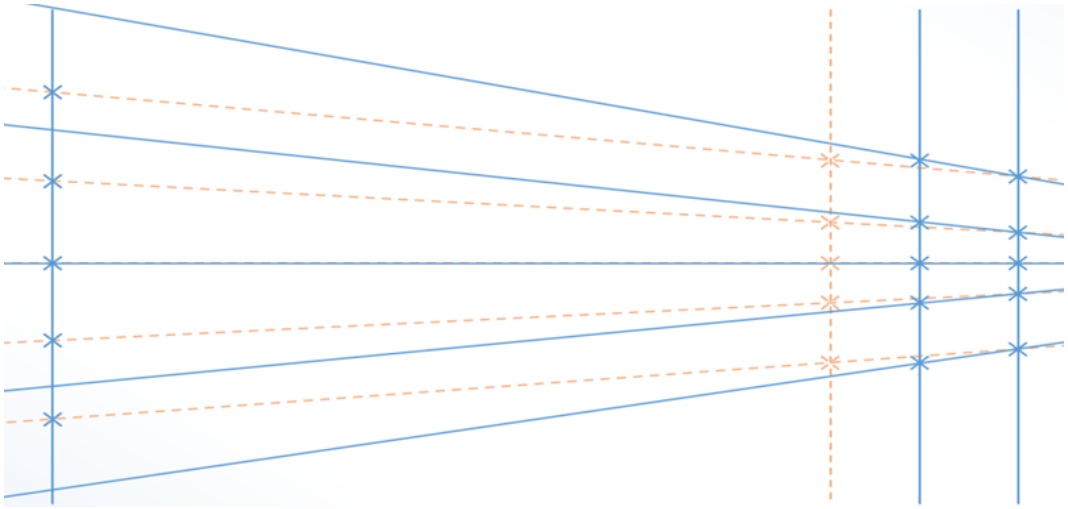


Figure 7.11: Illustration of how closer detection planes (blue lines on the right) create tracks with wider angles.

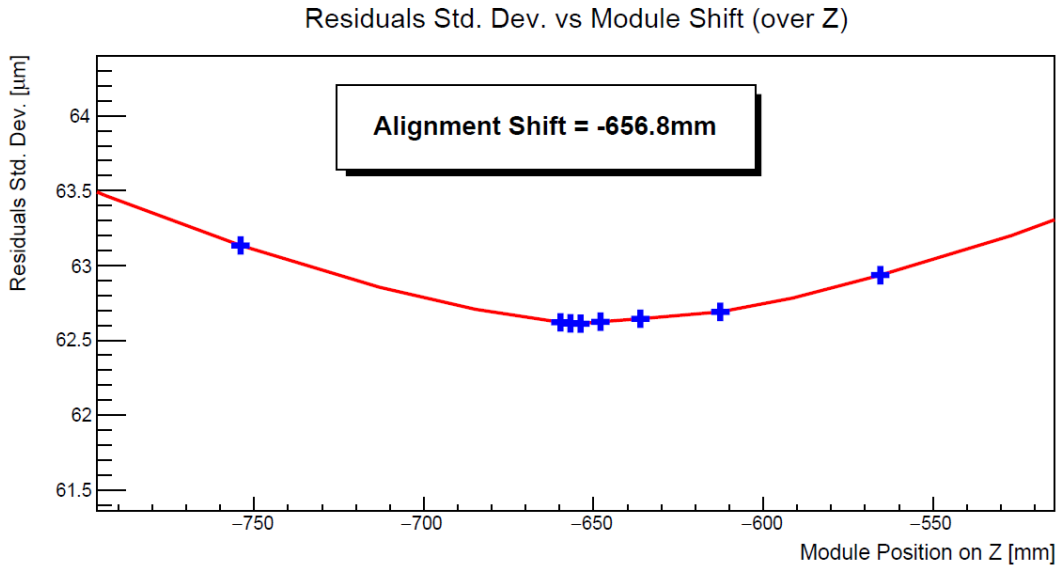


Figure 7.12: Standard deviation of the residual distribution as a function of the module translation along the Z axis.

Runs with the same settings are denoted by dots of the same color. Within each color group, each dot represents the resolution measured in that run divided by the average value of the resolutions for all runs in that group. Note that there are no runs with a resolution measured more than 1% from the group mean. Error bars represent the statistical errors of the resolution measurements.

Figure 7.16 shows the histogram with the data from the previous plot. The standard deviation of the resolution variation in runs with the same settings is 0.27% (corresponding to approximately 0.27 µm for the typical resolution). As this error is not significant, the alignment is considered satisfactory.

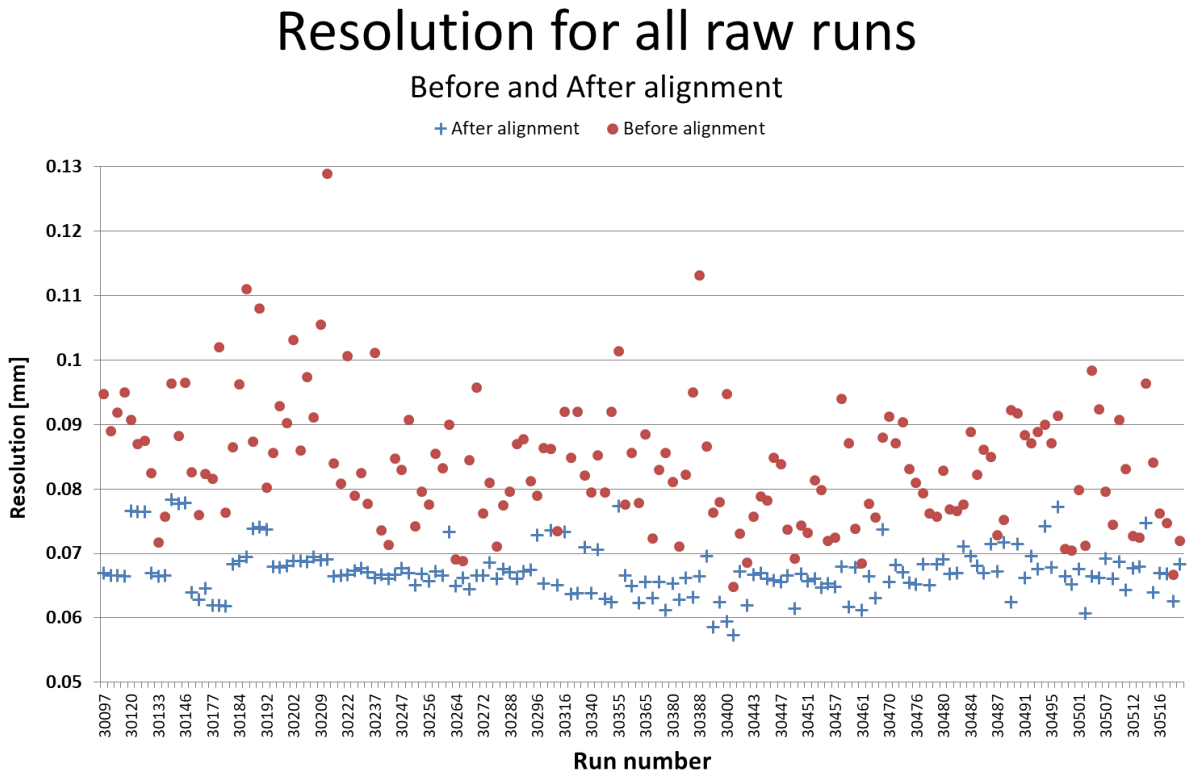


Figure 7.13: Resolution for all the raw runs, before and after alignment.

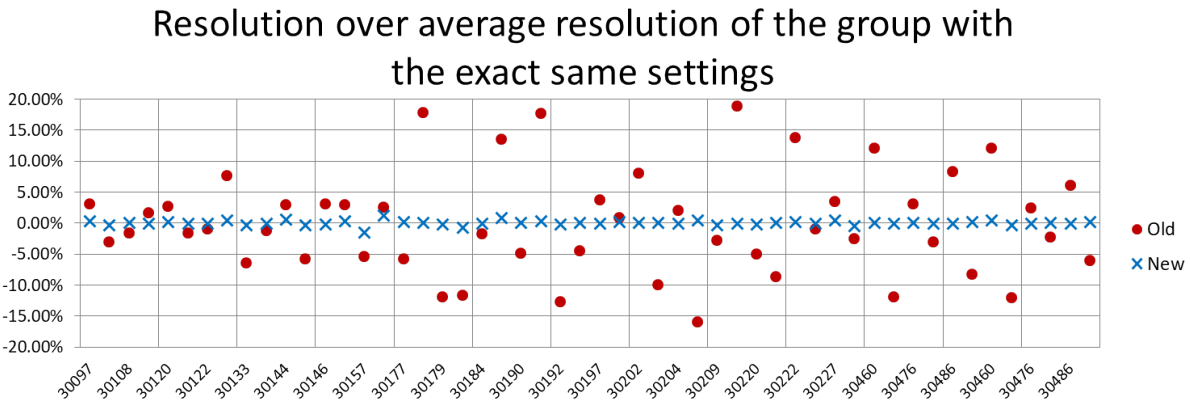


Figure 7.14: Resolution difference from raw runs to the average resolution of the group of runs with the same settings, before and after alignment.

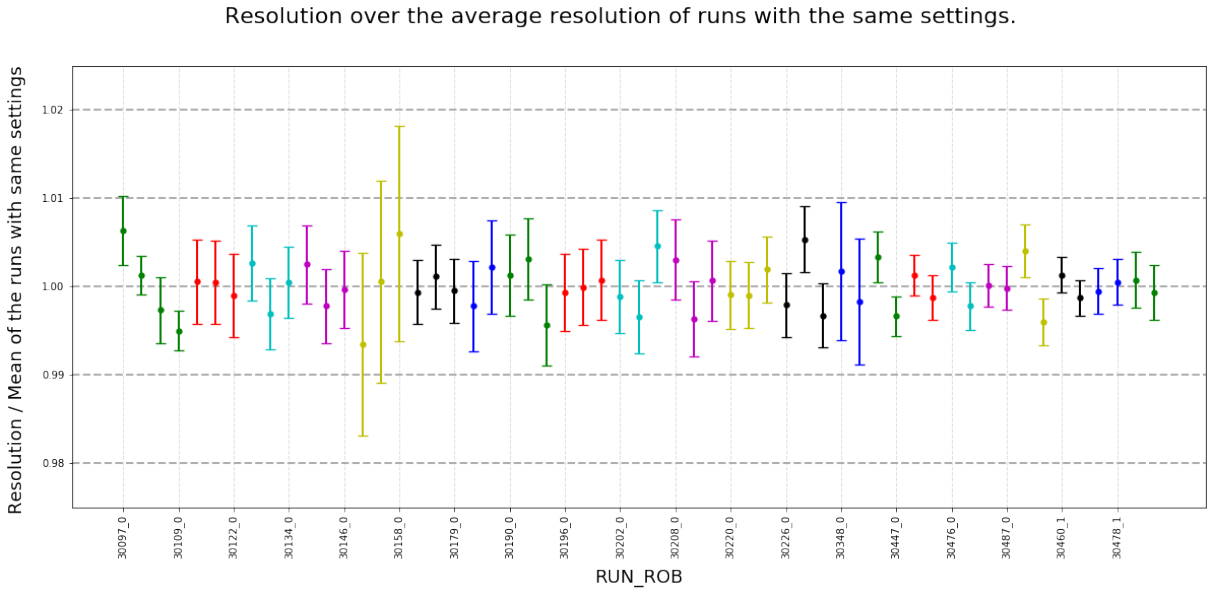


Figure 7.15: Runs with the same settings are denoted by dots of the same color. Within each color group, each dot represents the resolution measured in that run divided by the average value of the resolutions for all runs in that group.

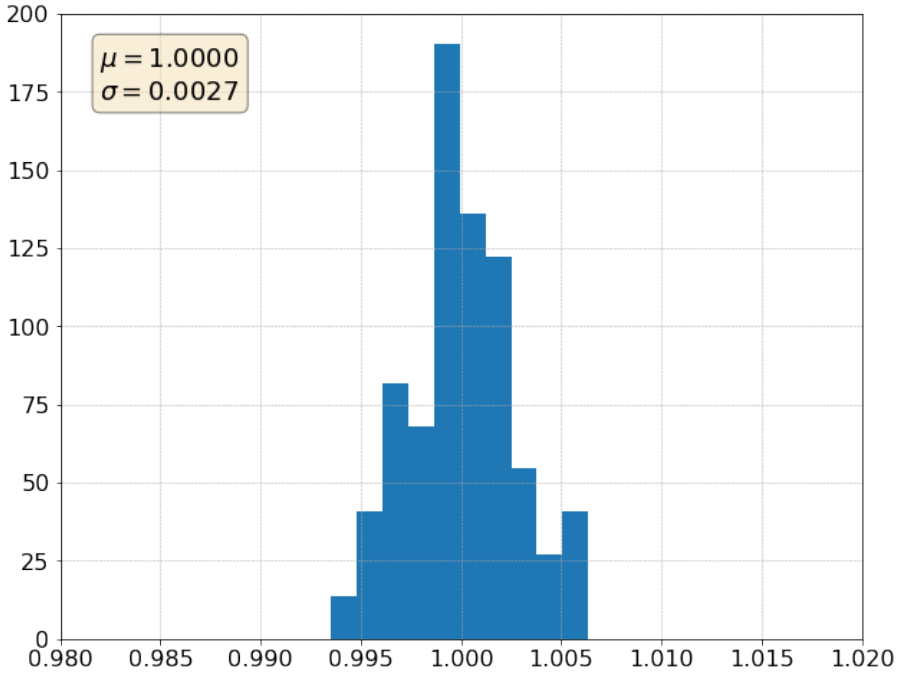


Figure 7.16: Histogram of the variation of resolution in runs with the same settings.

7.5 Handling Malfunctioning SiPM Channels

In the context of this analysis, by "noisy channels," we intend channels with a dark count rate (DCR), i.e., the rate of signal when there is no particle, high enough to affect the measurement. By "dead channel," we mean channels whose efficiency is low enough to affect the measurement. The threshold values have been chosen, as shown in the table below.

Noisy channel	DCR $> 0.5\%$
Dead channel	Efficiency $< 5\%$

Table 7.1: Criteria to define bad channels.

7.5.1 Filtering Channels

In order not to bias the measurement, the *Clusterization* class removes all channels that are marked as noisy before creating the clusters, and the *PosTagger* class tags all events containing track hits up to two channels away from a dead or noisy channel. This information is stored in a configuration file called *channels.cfg* that has 32 lines with 128 characters. Each line represents one SiPM, and each character represents the channel state, where 0 = good, 1 = dead, and 2 = noisy. A snippet of the lines for the two first SiPMs can be seen below:

```
# ROB 0 SiPM 0
00000000001100002000000000010101000000000000000000000000000000000000000000000000...
# ROB 0 SiPM 1
01100000000000000000020002000201000000000000000000000000020222000000000000000...
```

This configuration file is generated by a Python code contained in the Jupyter Notebook called *Channel_Studies.ipynb* (part of the *Results Plotter* described in Section 7.3) as explained in the following sections.

7.5.2 Gathering Channel Statistics

The class *ChannelStudies* loops over all the events, counting for each of the 2048 channels of the ROB, the quantities shown in the table below, which are then saved to a results text file.

Parameter	Description
trackHits_per_ch	The amount of Telescope tracks that predicted hit up to 3 channels away from the channel.
th0_per_ch_noTrackNearby th1_per_ch_noTrackNearby th2_per_ch_noTrackNearby	The number of times the channel reached respectively the first, second, or third PACIFIC threshold when **NO** Telescope tracks predicted hit up to 3 channels away from the channel.
th0_per_ch_hitByTrack th1_per_ch_hitByTrack th2_per_ch_hitByTrack	The number of times the channel reached respectively the first, second, or third PACIFIC threshold when a Telescope track predicts a hit up to 3 channels away from the channel.

Table 7.2: List of values saved to disk by the class *ChannelStudies*.

7.5.3 Evaluating Noisy Channels

Once the statistical parameters contained in Table 7.2 are available, an analysis is performed to determine which channels should be considered for the analysis and which should be ignored, following the criteria in Table 7.1.

In order to find out which channels are too noisy, a plot of the average amount of dark counts per event was created. This is obtained by dividing the amount of *th_per_ch_noTrackNearby* by the number of events with no nearby track hit:

$$\text{DCR} = \frac{\text{counts when no hit}}{\text{total events with no hit}} \tag{7.5}$$

The results are shown in Figure 7.17 for the 2048 channels of both ROBs.

All the channels exceeding 0.5% in Figure 7.17 were considered as noisy and excluded from the analysis.

7.5.4 Evaluating Dead Channels

For the evaluation of the dead channels, a similar analysis was made, using the number of misses (no count) when there was a track hit. This is a way of measuring the efficiency of a channel, but it differs from the single-hit efficiency measurement in Section 7.6.2 for which clusters are used. To evaluate dead channels, the channel efficiency was defined by the ratio of *th_per_ch_hitByTrack* to *trackHits_per_ch*:

$$\text{Channel Efficiency} = \frac{\text{counts when hit}}{\text{total events with hit}} \tag{7.6}$$

The results are shown in Figure 7.18 (Only ROB0 shown).

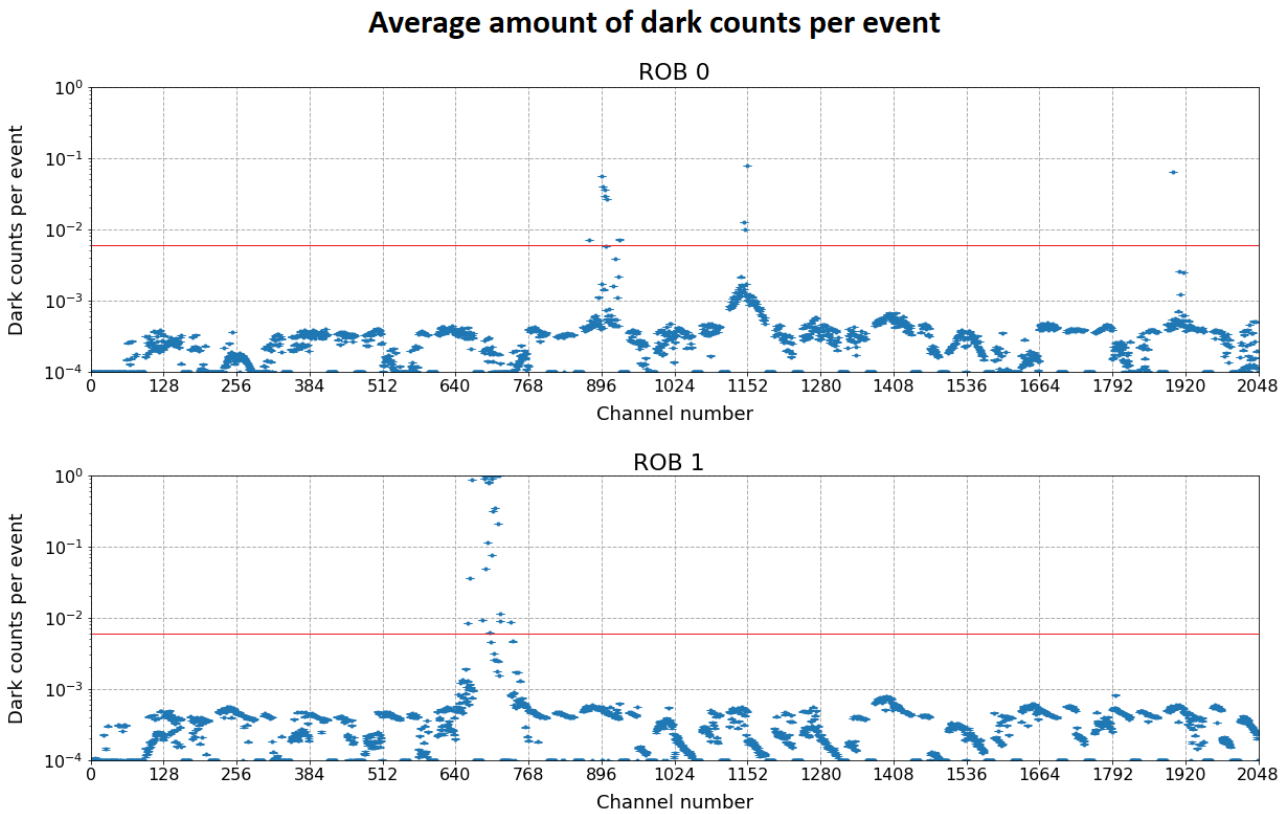


Figure 7.17: Average dark counts per event per channel. The red straight line denotes the 0.5% threshold defining noisy channels.

A complication is due to the fact that many channels have not been directly under the beam, therefore, there are not enough statistics to measure their efficiency with a small error. Furthermore, the channels next to an edge or the gap of the SiPM will present lower efficiencies.

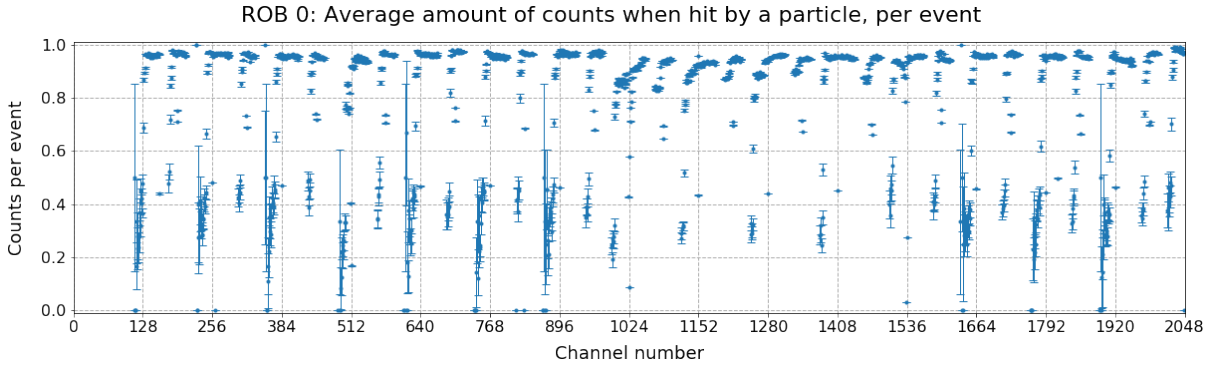


Figure 7.18: Average efficiency of a channel according to Equation 7.6.

To have a better understanding of the efficiency plot, the amount of hits each channel received was considered, shown in Figure 7.19 for all raw runs of the test beam for ROB0.

Using this information, the efficiency was re-calculated, including only the runs with a reasonable amount of statistics, and the results are shown in Figure 7.20. The background is on a scale from green (high amount of track hits) to red (low amount), and the white

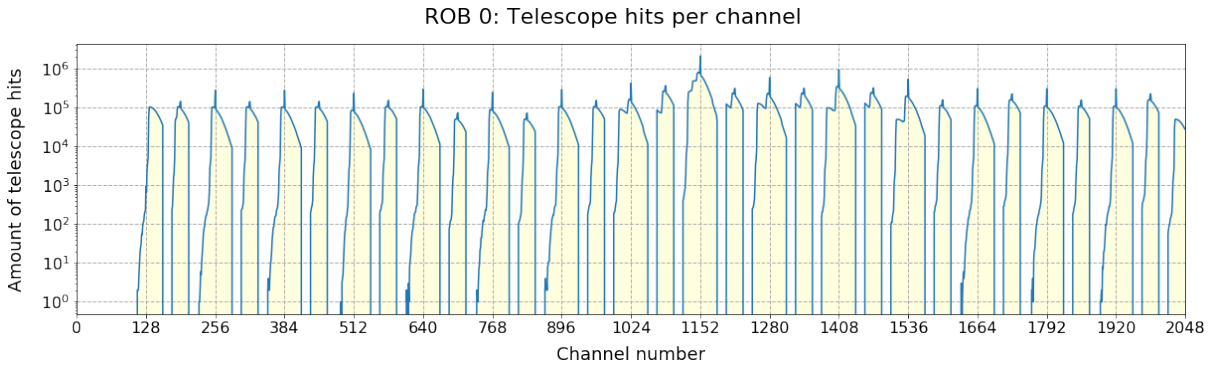


Figure 7.19: Total amount of track hits per channel, during all raw runs of the test beam, for ROB0.

background represents the regions that were not hit at all by the beam during the runs. The efficiency values of channels next to an edge or a gap are denoted by red dots in order to distinguish the channels with lower efficiency due to the gap effect from the actual dead channels.

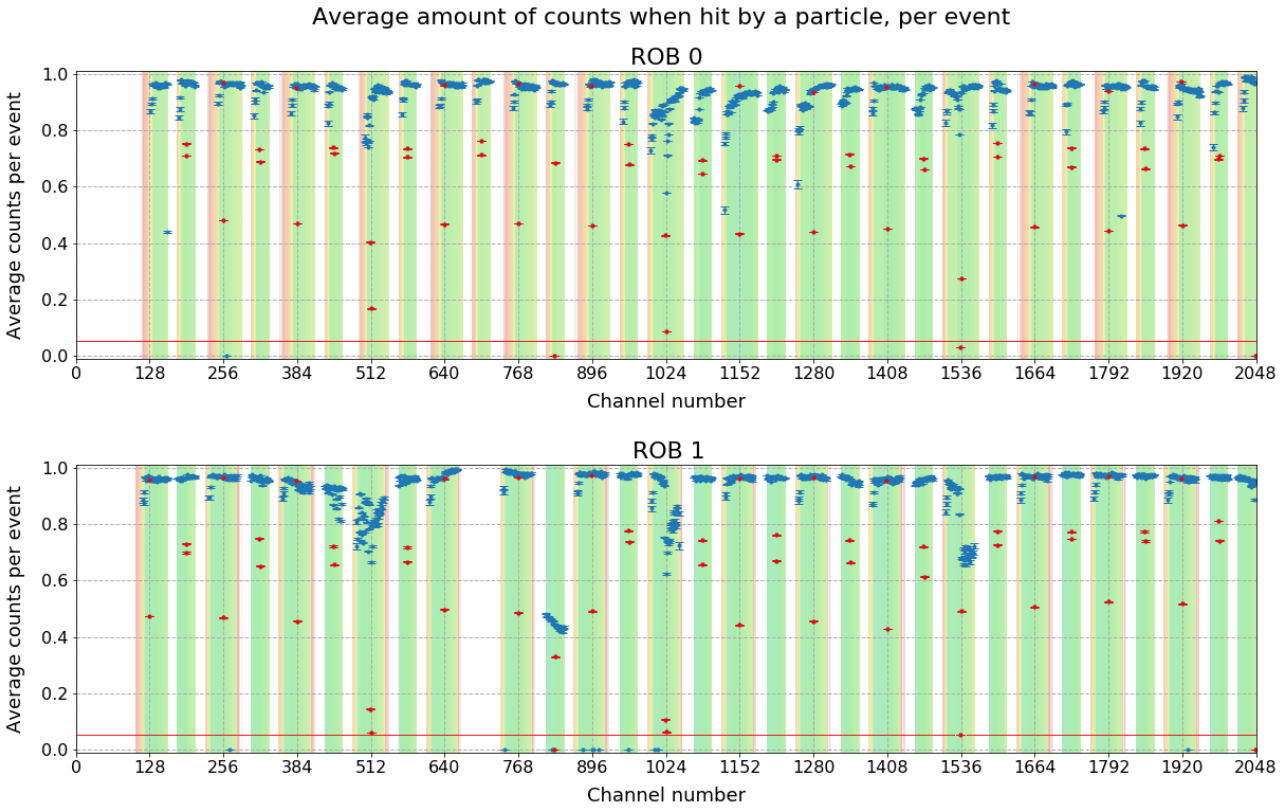


Figure 7.20: Average counts per event given a particle hit the channel, for all channels from both ROBs.

Following the criteria shown in Table 7.1, all channels with values in Figure 7.20 below 5% were excluded from the analysis. The red horizontal line denotes such a threshold.

7.6 Generating Run Based Results

7.6.1 Cluster Resolution

The class *Resolution* calculates the residual distribution and measures the resolution for a given run. It selects events, considering only the ones that fulfill the criteria below:

- Not empty;
- Within the PACIFIC integration window;
- Telescope track fitting: $0.5 < \text{chi}^2 < 2.0$;
- Not within the BXId limit: $50 < \text{BXId} < 60$;
- At least two channels away from edges and middle gap;
- Single cluster close to track hit;
- At least two channels away from dead and noisy channels.

Once the residual $y - y_{\text{track}}$ has been calculated for all the events as explained in Section 7.2.7, the cluster resolution can be determined from the standard deviation of the residual distribution. Such standard deviation is the square root of the sum of the squares of the SciFi intrinsic resolution and the Telescope resolution.

$$\sigma^2_{\text{residual}} = \sigma^2_{\text{SciFi}} + \sigma^2_{\text{telescope}} \tag{7.7}$$

Therefore the intrinsic resolution of the SciFi is:

$$\sigma_{\text{SciFi}} = \sqrt{\sigma^2_{\text{residual}} - \sigma^2_{\text{telescope}}} \tag{7.8}$$

As seen in Section 6.1.2, the track resolution of the Telescope at the distance of the SciFi modules is approximately $\sigma \approx 10\,\mu\text{m}$. Since it is much lower than the SciFi, the intrinsic resolution of the SciFi is very close to the standard deviation of the residual distribution itself. The Figure 7.21 shows a typical residual distribution.

The residual distribution (R) can be parameterized as a weighted sum of two Gaussian distributions:

$$R \propto f \cdot \exp\left(-\frac{(R-\mu_{\text{inner}})^2}{2\sigma^2_{\text{inner}}}\right) + (1-f) \cdot \exp\left(-\frac{(R-\mu_{\text{outer}})^2}{2\sigma^2_{\text{outer}}}\right) \tag{7.9}$$

where:
f is the relative weight of the inner and outer Gaussian;
μ_{inner} and μ_{outer} are the mean for the inner and outer Gaussian respectively;
σ_{inner} and σ_{outer} are the standard deviations of the inner and outer Gaussian, respectively.

The inner Gaussian represents particles that went straight to the detector, giving rise to a very sharp distribution. This is the dominant factor and is denoted by the dashed black

line in Figure 7.21. The outer Gaussian represents particles that suffered large scattering, and in Figure 7.21, it is denoted by the red dashed line.

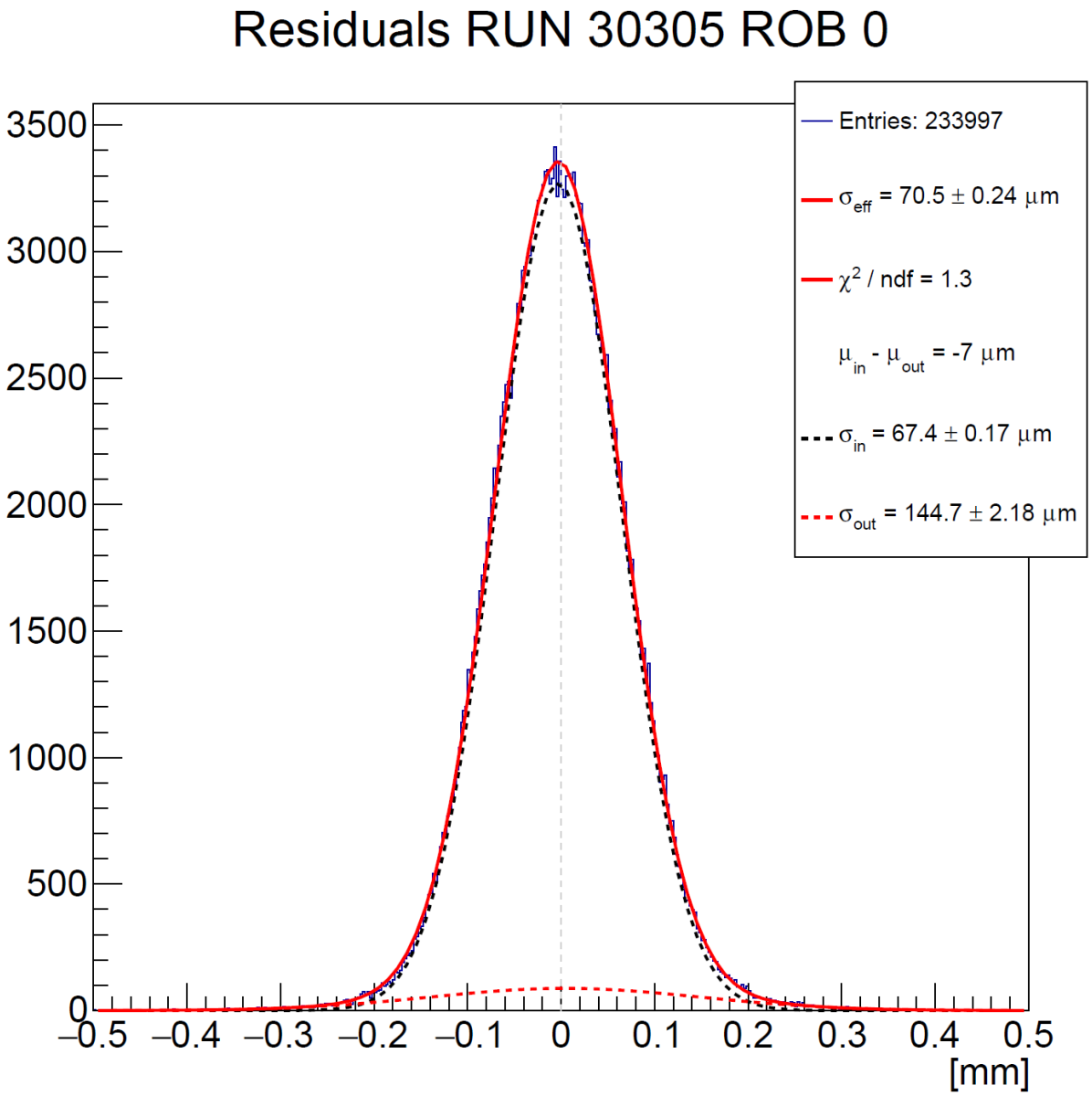


Figure 7.21: Residual distribution for run 30305 ROB 0. The solid red line denotes the result of the fit of Equation 7.9 to the data.

Once the parameters of the fitting have been estimated, the resolution σ_{eff} for such a run is given by the equation

$$\sigma_{\text{eff}} = \sqrt{f \cdot \sigma_{\text{inner}}^2 + (1-f) \cdot \sigma_{\text{outer}}^2} \approx \sigma_{\text{SciFi}} \tag{7.10}$$

and the corresponding statistical uncertainty can be calculated as

$$\sigma_{\text{effErr}} = \sqrt{\left(\frac{\sigma_{\text{in}} \cdot f \cdot \sigma_{\text{inErr}}}{\sigma_{\text{eff}}}\right)^2 + \left(\frac{\sigma_{\text{out}} \cdot (1-f) \cdot \sigma_{\text{outErr}}}{\sigma_{\text{eff}}}\right)^2 + \frac{2 \cdot f \cdot (1-f) \cdot \sigma_{\text{in}} \cdot \sigma_{\text{out}} \cdot \text{cov}(\sigma_{\text{in}}, \sigma_{\text{out}})}{\sigma_{\text{eff}}^2}} \tag{7.11}$$

where:
σ_{in} is the fit result of σ_{inner};
σ_{out} is the fit result of σ_{outer};
σ_{inErr} is the fit error on σ_{in};
σ_{outErr} is the fit error on σ_{out};
$\text{cov}(\sigma_{\text{in}}, \sigma_{\text{out}})$ is the fit covariance between σ_{in} and σ_{out}.

7.6.2 Single Hit Efficiency

The class *Efficiency* is the one responsible for evaluating the module's single hit efficiency for each run. It selects events, considering only those fulfilling the criteria below:

- Event with single or no tracks;
- Within the integration window;
- Telescope track fitting: $0.5 < \text{chi}^2 < 2.0$;
- Not within the BXId limit: $50 < \text{BXId} < 60$;
- At least two channels away from edges and middle gap;
- At least two channels away from dead and noisy channels;
- No big clusters present;
- Track on SiPM's region that is being read.

As the focus of this analysis was to measure the resolution, this class was created with the goal of verifying if the efficiency is affected by the parameters changed for the resolution studies. Therefore only the overall efficiency of the detector was determined (instead of that of individual channels), and the focus was on changes rather than absolute values.

The class counts the total amounts of events in the run that fulfill the criteria mentioned, henceforth referred to as *Total Count*. Then, from these events, it counts the subset of those that have a cluster within 1mm of the track's prediction (*SciFi Count*). The efficiency is then defined by:

$$\text{Efficiency} = \frac{\text{SciFi counts}}{\text{Total counts}} \tag{7.12}$$

7.7 Studying the Cluster Weighting Coefficients

One of the crucial ingredients of the event reconstruction is the weighting coefficients used by the clustering algorithm and how they affect the SciFi performance. The data for this study is generated by the class *ClusterWeightStudies*, which varies the weighting coefficients in the clustering algorithm in other to generate results for many combinations

of weights, to be later analyzed by a Jupyter Notebook from the Results Plotter mentioned in Section 7.3. The efficiency is also recorded to check that the weights do not significantly affect the probability of reconstructing a cluster or not, but merely affect the cluster barycenter calculation and, thus, the resolution.

7.7.1 Scanning different sets of weighting coefficients

As explained in Section 4.4.3, the clustering algorithm calculates the barycenter of a cluster by applying a specific weight to each channel of the cluster according to the threshold levels. There are three possible values for weights, as there are three threshold levels. Note that, as a weighted mean is used, the result is solely affected by the relative values of the weights. This means that using the values 1, 2, and 3 for the weights will generate the same result as using 2, 4, and 6. That said, being the three weights w_1, w_2 and w_3, we can treat the problem as effectively having only two parameters to vary:

$$\frac{w_2}{w_1} \quad \text{and} \quad \frac{w_3}{w_1}$$

In order to understand how resolution varies with the weights, the class performs a scan in two dimensions, where weight w_1 is kept fixed at the value 1.0 and weights w_2 and w_3 vary along a certain range. The range in which the weights are going to vary, and the amount of data points to be scanned can be set in the program options. The data points are chosen by the class such that they are equally spaced on a logarithmic scale.

Only data points with $w_3 > w_2 > w_1$ are scanned because the threshold values for the PACIFIC are always in increasing order, and therefore it would not make sense to attribute a higher weight to a threshold that needs less signal to be achieved.

Once the class is called, it creates the list of data points to be scanned, and then it loops through such list executing the following steps for each set of weights:

1. Update weight values on the *Clusterization* class.
2. Runs the Clusterization algorithm using the *Clusterization* class.
3. Regenerate the residual values using the *Residuals* class.
4. Compute resolution using the *Resolution* class.
5. Compute efficiency using the *Efficiency* class.
6. Save results to the results text file.

7.7.2 Plotting the results

Once the results are stored in the text files by the Run Data Analysis Software, the analysis proceeds with one of the Results Plotter's Jupyter Notebooks. The notebooks *Weights_coarseScan.ipynb* and *Weights_fineScan.ipynb* were created to respectively make

a scan along a wide range (1 to 100 in this analysis) and then zoom in to a smaller area, with more data points, in order to find more accurately where the values converge to. These notebooks use the Python libraries from the Results Plotter to load all the results from the weights scans into a Pandas dataframe in order to create the plots of interest for this analysis.

Plot of resolution per cluster weights combination

This is the direct visualization of the data generated by the weights scan. It shows a 2D heat map plot containing the resolution value for each pair $\left(\frac{W_2}{W_1}, \frac{W_3}{W_1}\right)$ for an individual run. Additionally, it highlights the combination of weights that gives the best resolution for such a run.

Figure 7.22 shows an example of such a plot for run 30441 ROB0, scanning the range from 1 to 100 in 10 bins. As $\frac{W_2}{W_1} < \frac{W_3}{W_1}$ only part of the heat map is filled.

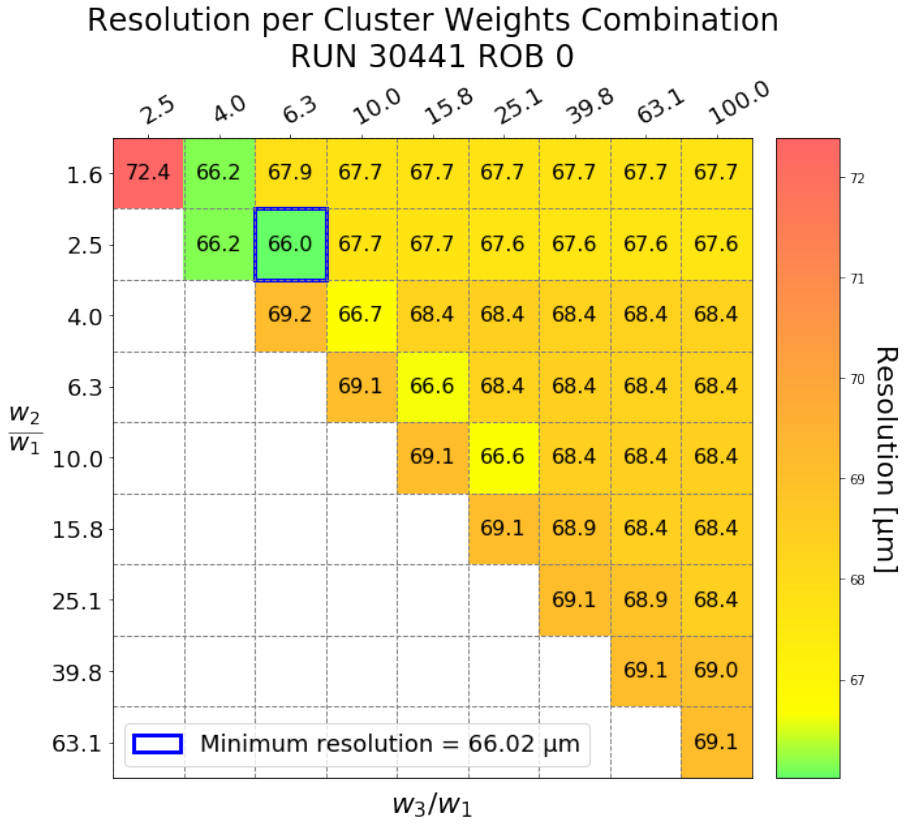


Figure 7.22: Plot of resolution per cluster weights pair for run 30441 ROB0.

Plot of the number of runs per best weights combination

Figure 7.23 is made to visualize which sets of weights are the best for most runs. Each cell in the heat map shows the number of runs for which a given set of weights provides the best resolution result. Figure 7.23 shows the mentioned plot, considering all runs taken with the raw data format.

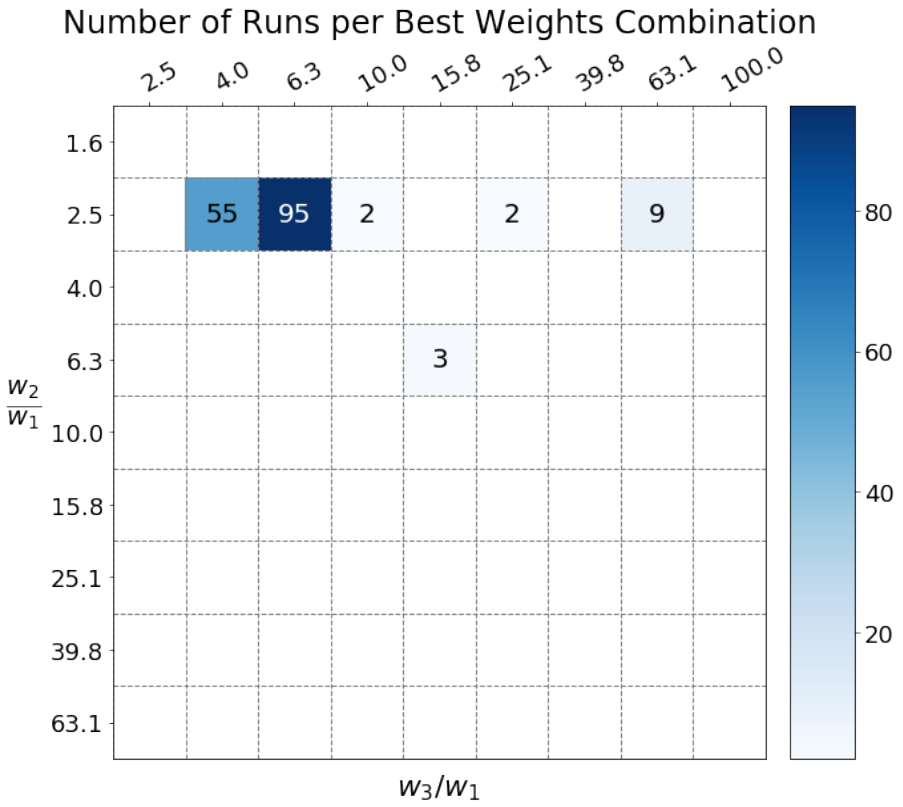


Figure 7.23: Number of runs for which a given pair of weights gives the best resolution.

Plot of score per best weights combination

In order to better evaluate the overall effect of each set of weights on all runs, it was created a score system for each weight pair. The score of a certain pair represents how much, on average, the resolution increases, in percentage, with respect to that obtained when using the best pair of weights for each of the runs used in the computation. Figure 7.24 shows the score plot for all runs taken with the raw data format.

Code Flexibility

The code making the plots described in this section was written in a very generic form. It is fairly easy to change which results are plotted and several other settings. The number of rows and columns of the plots are also not limited to the ones displayed but are automatically adjusted to the amount of data points tested during the weights scan. In this way, several different plots can be generated in addition to those shown in Figures 7.22, 7.23 and 7.24 for example. Figure 7.25 shows the plot of efficiency per best weights pair, and Figure 7.26 shows the resolution plot with 19×19 cells instead of 9×9.

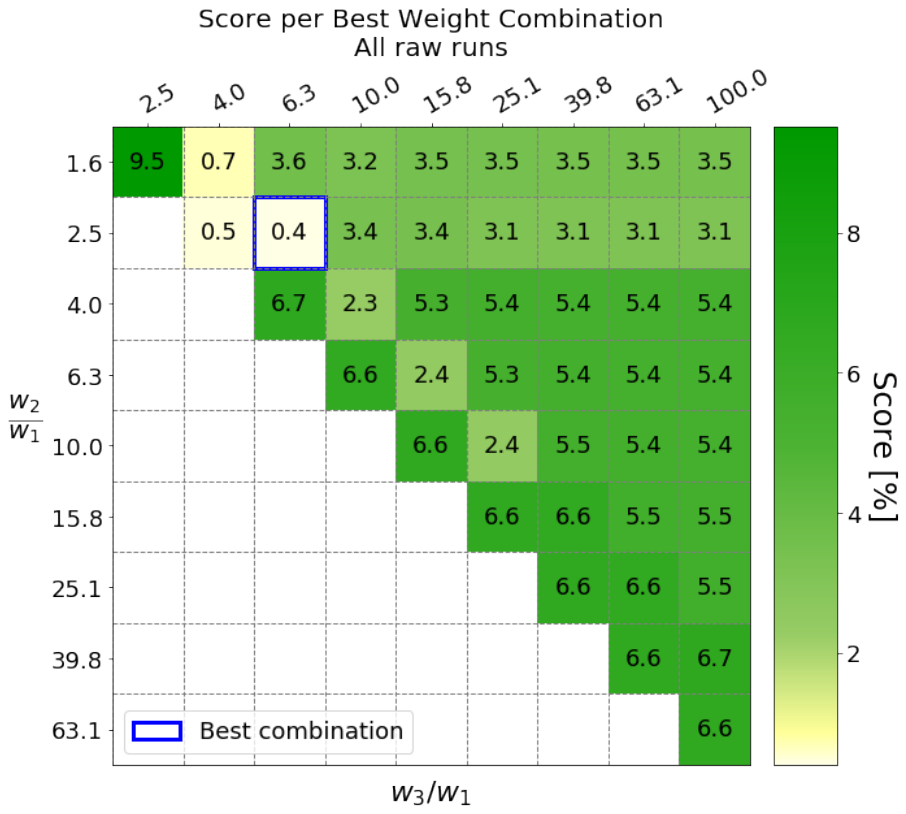


Figure 7.24: Plot of score per best weights combination.

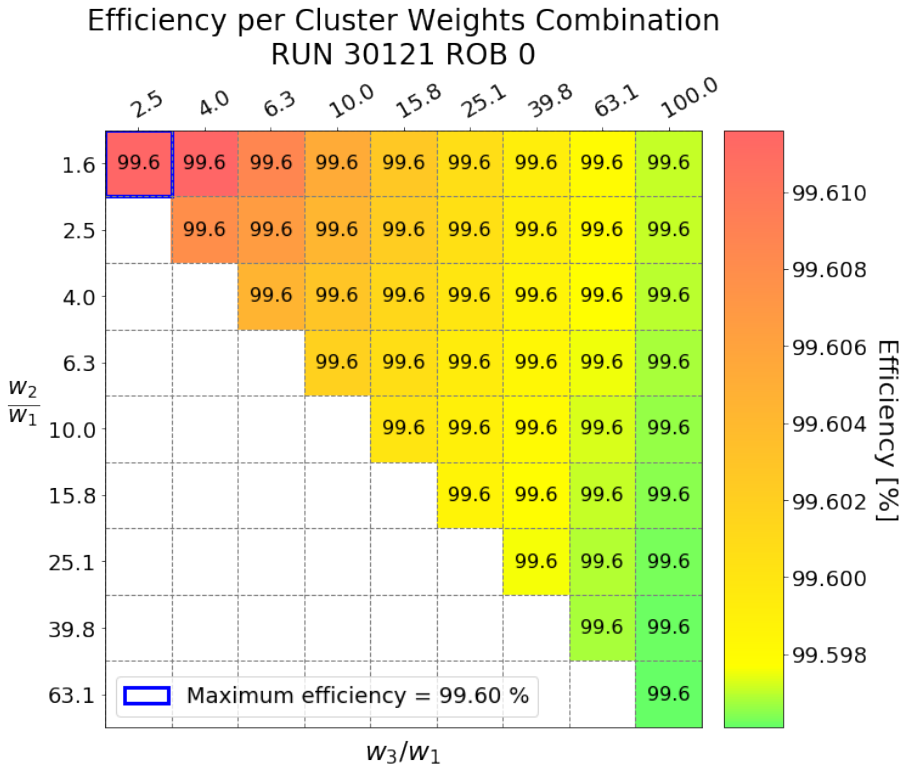


Figure 7.25: Plot of efficiency per cluster weights pair.

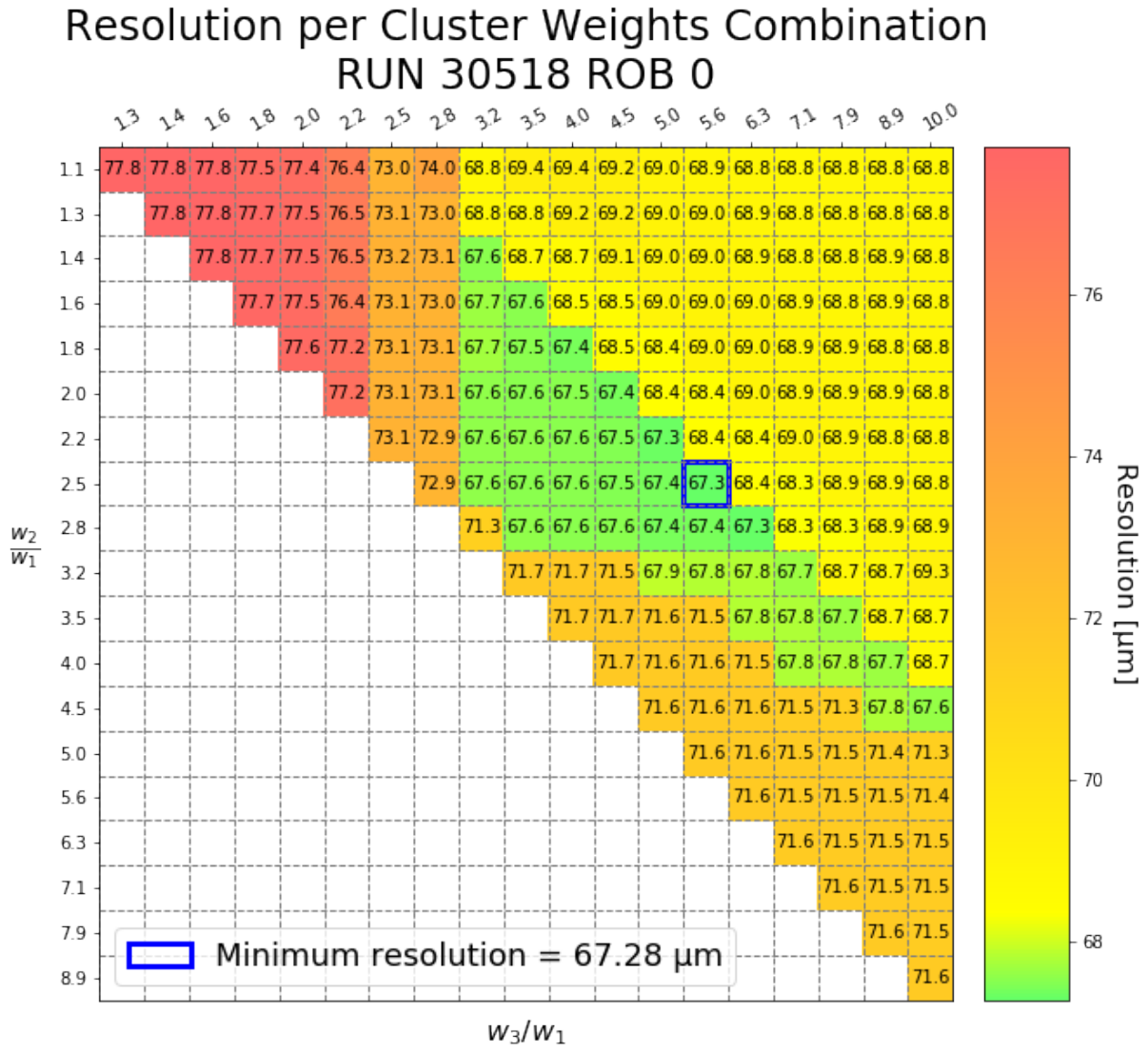


Figure 7.26: Plot of resolution per cluster weights pair for a scan taken with 20 steps on both axes.

Chapter 8

Test Beam Results

In this chapter, the results from the analysis will be presented and discussed. The analysis procedure used to reach such results has already been described in detail in Chapter 7 and the experimental setup in Chapter 6.

During the test beam, three hardware parameters have been varied, as shown in Table 8.1:

Parameter	**Values**
PACIFIC thresholds	05, 15, 25 15, 25, 45 35, 55, 95
Shaper settings	"February" "August"
Beam position	663 mm (Mirror) -40 mm (Center) -720 mm (SiPM)

Table 8.1: Main hardware parameters varied during the test beam.

During the analysis, three software parameters of the offline clustering algorithm have been varied, as shown in Table 8.2:

The following sections will discuss how these parameters affect the efficiency and the resolution of the SciFi detector.

Parameter	Values
Cluster precision	8 bits Floating point
Cluster max width	3 channels 4 channels 5 channels 6 channels
Cluster weights	10 steps from 1 to 100 20 steps from 1 to 10

Table 8.2: Main software parameters varied during the analysis

8.1 Data Selection

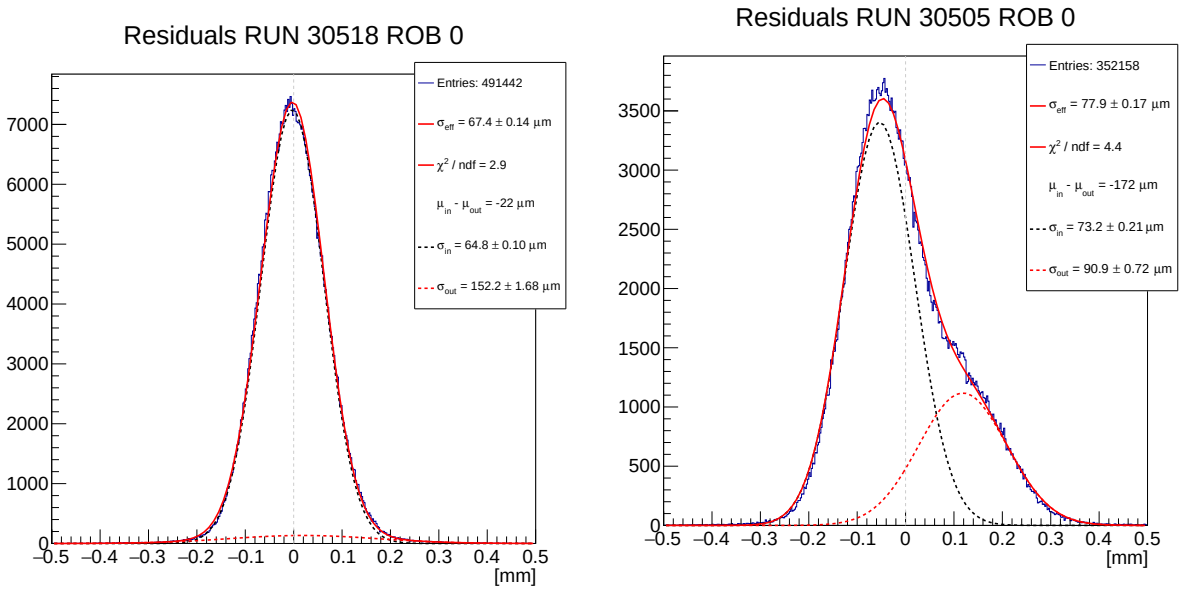
Sections 7.6.1 and 7.6.2 presented how events were selected when computing resolution and efficiency, respectively. From the individual run analysis results, some runs present an artifact in the residual distribution and were not used to obtain the results presented in this chapter. The excluded runs are listed in Table 8.3.

Run number	ROB
30214	0
30215	0
30262	0
30296	0
30355	0
30392	0
30443	1
30468	0
30468	1
30505	0
30505	1

Table 8.3: Runs excluded from the analysis.

The artifact present in the excluded runs is a second Gaussian peak in the residual distribution, the amplitude and center of which may vary. Figure 8.1 shows the difference between a typical run (to the left) and one of the runs containing the artifact (to the right). A discussion of the possible causes of this can be found in Section 8.1.1.

For the analysis of the detector's performance (Sections 8.3, 8.4, 8.5, 8.6 and 8.7) were chosen the runs listed in Table 8.4. In all these runs, the module position along the X axis was the closest possible to $-180\,\text{mm}$, the only position "common" to all runs and such as to minimize the influence of the beam position over the detector.



(a) Typical residual distribution.

(b) Residual distribution presenting the said artifact.

Figure 8.1: Difference between the residual distribution of a typical run and that of a run presenting the artifact discussed in Section 8.1.1.

Run number	Thresholds	Shaper Settings	Position Y	Position X
30344	51525	-720 mm (SiPM)	"August"	-177.0 mm
30336	51525	-720 mm (SiPM)	"February"	-177.0 mm
30316	51525	-40 mm (Center)	"August"	-180.0 mm
30305	51525	-40 mm (Center)	"February"	-180.0 mm
30146	51525	663 mm (Mirror)	"August"	-182.0 mm
30121	51525	663 mm (Mirror)	"February"	-182.0 mm
30340	152545	-720 mm (SiPM)	"August"	-177.0 mm
30332	152545	-720 mm (SiPM)	"February"	-177.0 mm
30311	152545	-40 mm (Center)	"August"	-180.0 mm
30300	152545	-40 mm (Center)	"February"	-180.0 mm
30133	152545	663 mm (Mirror)	"August"	-182.0 mm
30249	152545	663 mm (Mirror)	"February"	-182.0 mm
30352	355595	-720 mm (SiPM)	"August"	-177.0 mm
30396	355595	-720 mm (SiPM)	"February"	-177.0 mm
30324	355595	-40 mm (Center)	"August"	-180.0 mm
30320	355595	-40 mm (Center)	"February"	-180.0 mm
30158	355595	663 mm (Mirror)	"February"	-182.0 mm

Table 8.4: Runs used for the analysis of the performance of the detector given the test beam settings.

8.1.1 SiPM Misalignment

The Problem

When investigating the artifact shown in Figure 8.1, it was noted that it happened only in runs in which the beam was hitting the gap between the fiber mats. Each mat is associated with a section of the cold bar, containing 4 SiPMs glued to them (see Section 4.2). The cold bar is not a solid straight piece, and in between those sections, it is bent so that it allows the sections some flexibility to move slightly with respect to each other and thermally expand (see Figure 4.7). However, the Run Analysis Software performed the alignment considering all SiPMs perfectly aligned with respect to the one the beam is hitting. This is a valid assumption when the beam hits in-between pairs of SiPMs glued to the same cold-bar section. However, as the four sections of the cold bar have the same flexibility to move with respect to each other, during thermal cycles this assumption cannot be considered valid for SiPMs on different sections.

Figure 8.2 systematically displays the layout of the cold bar and the position of its 16 SiPMs. Note the four segments that constitute the cold bar (see Figure 8.2). Below the bar is a plot of the number of track hits per SiPM channel recorded during runs exhibiting the artifact (11 runs in total). Note that the beam is hitting two SiPMs across a cold-bar gap in all runs.

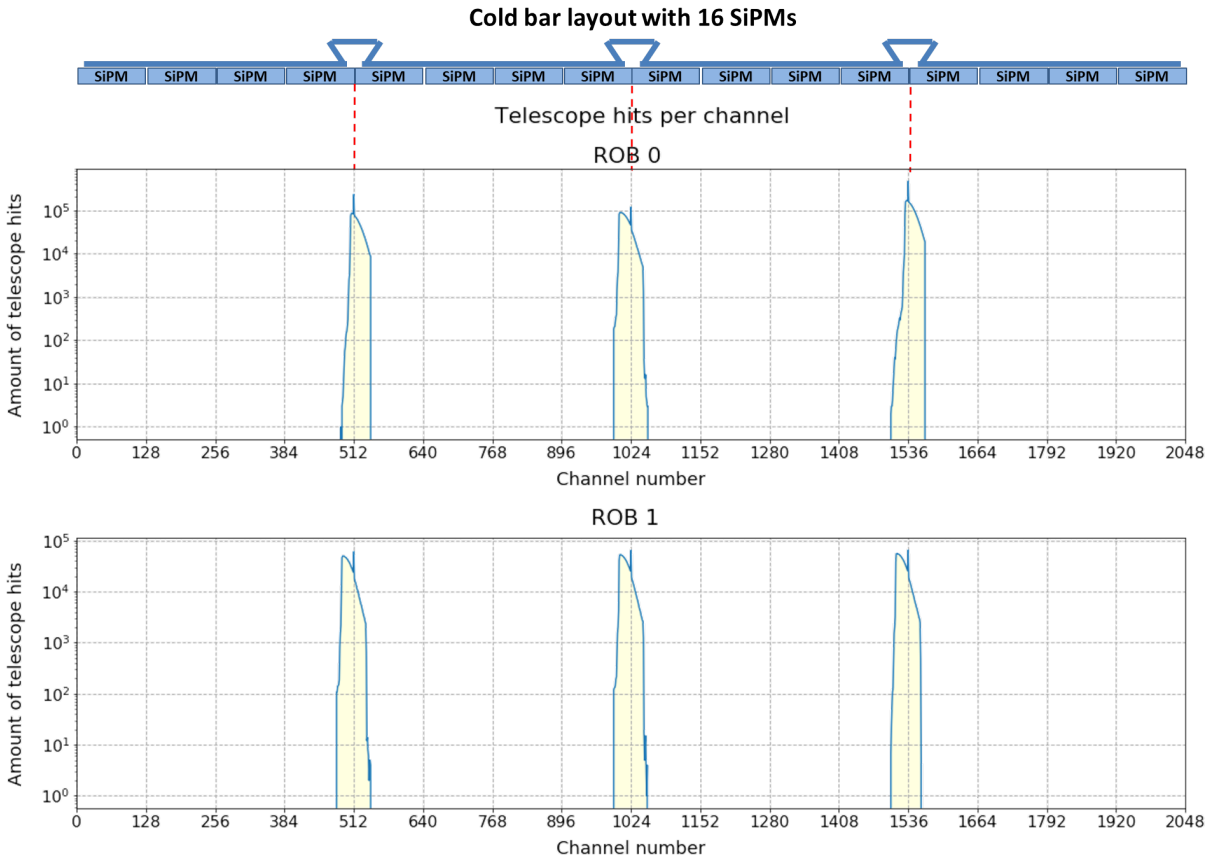


Figure 8.2: Total track hits per channel for the runs that show the artifact during the whole test beam.

Not correcting for possible deviations from the nominal inter-SiPM distance leads to the second Gaussian peak in the residual distribution, affecting the resolution computation by making the standard deviation bigger and, therefore, presenting a worse resolution.

The Solution

The problem could be solved by doing the alignment procedure while taking this effect into account and using separate degrees of freedom for each group of four SiPMs corresponding to a segment of the cold bar.

For the analysis presented in this thesis, however, we chose to exclude these 11 runs from the analysis as there is enough data from the other runs to achieve the desired statistical significance for the results.

As mentioned in 7.6.1, the residual distribution is formed by two Gaussian peaks with amplitudes, which are given by a factor f and $(1-f)$, respectively. This factor is typically around 98%. For runs in which the beam is hitting SiPM gaps, this factor is smaller than 97%.

Figure 8.3 displays the width of the residual distribution of all the runs of the test beam: the orange dots denote the runs with $f < 97\%$, corresponding to the 11 runs shown in Table 8.3.

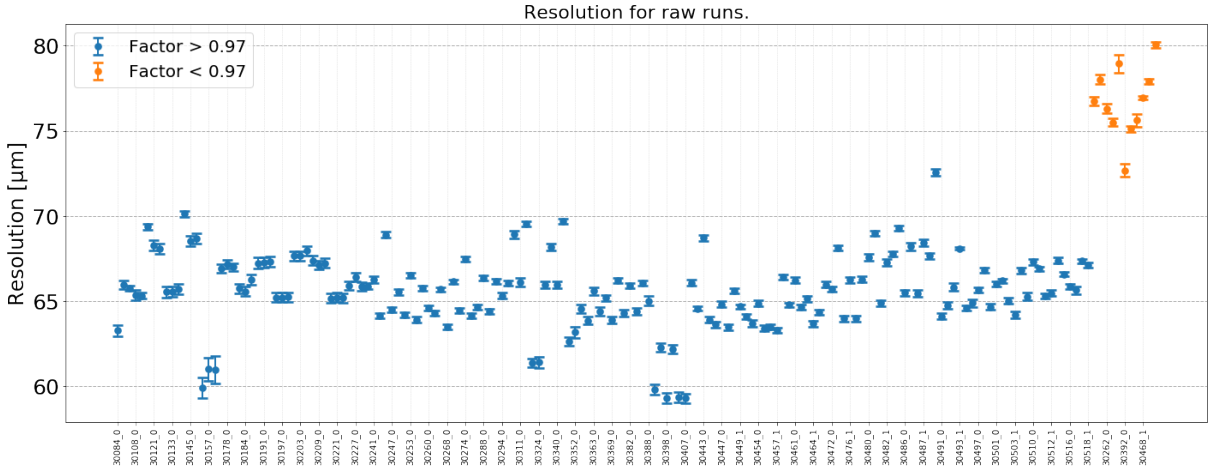


Figure 8.3: Plot of resolution for all runs, separated in runs with $f < 97\%$ and $f > 97\%$.

8.2 Optimal Clusterization Weighting Coefficients

This section will discuss the performance of the clusterization algorithm given the choice of weighting coefficients (see Section 7.7.) The weights might affect the position of the barycenter of a cluster. First, Section 8.2.1 shows that the efficiency is not significantly affected by the choice of the weights. Then, the impact on the resolution and the potential choice of the coefficients is discussed.

8.2.1 Efficiency Given the Weighting Coefficients

As the efficiency class only considers clusters within 0.5 mm of the track hit, clusters formed approximately two channels away from the track hit might cease to be considered if the barycenter changes by more than 0.5 mm. This is the case for a very low amount of events while the dominant part of the signal is within the range of 0.5 mm from the track, and therefore, the impact of weight changes on efficiency is minor.

Figure 8.4 shows the typical efficiency result for all the runs of the test beam. Note that the full range of the color scale represents less than 0.01% of the efficiency variation.

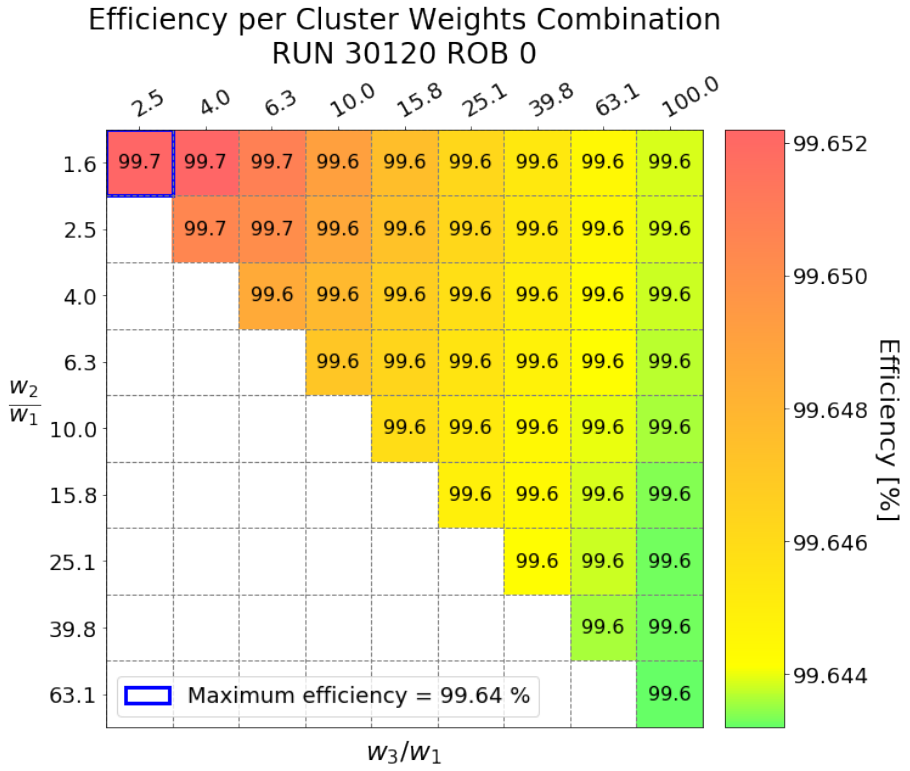


Figure 8.4: Efficiency for different combinations of weights.

8.2.2 Resolution Given Weighting Coefficients

When analyzing the resolution for different combinations of weights, it is clear that resolution can be drastically affected by the choice of weights. Figure 8.5 shows the resolution score per weights combination (as explained in Section 7.7.2) using 10 steps within the range 1-100.

It clearly appears that the values converge to the region around (2.5, 6.3). Zooming into that region and using 20 steps within the range 1-10 results in Figure 8.6. Figure 8.6 indicates that the set of weights 1.0, 2.2, and 5.0 is the optimal choice. This combination represents an average loss of 0.3% in resolution of each run when compared to the best combination for such runs.

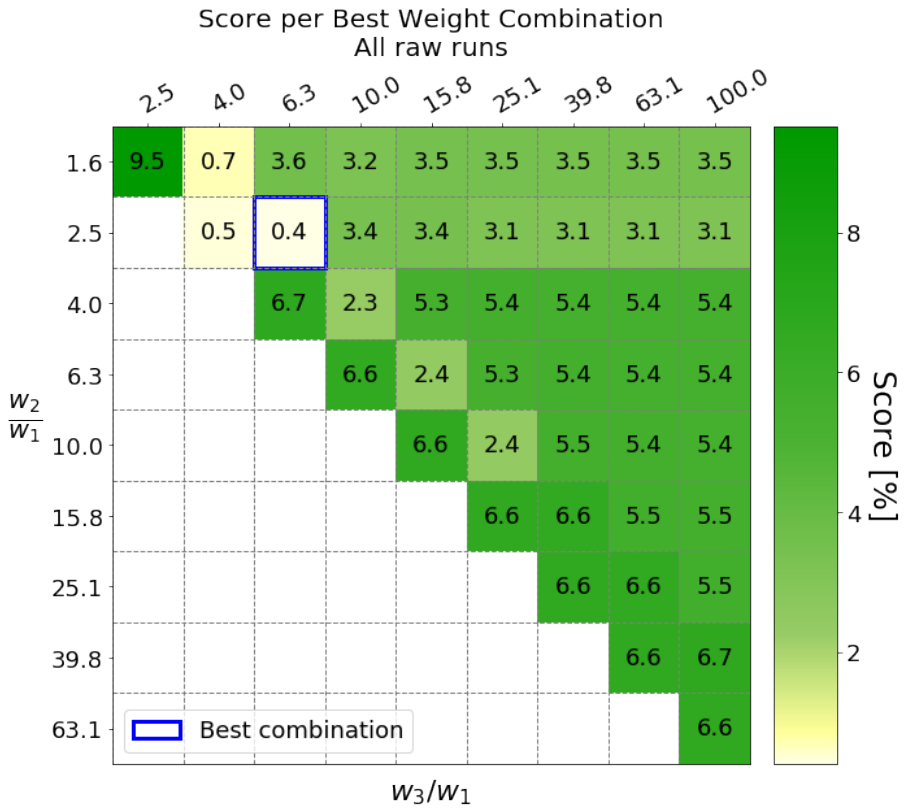


Figure 8.5: Resolution score for the given combinations of weights. Coarse scan.

8.2.3 Optimal Weighting Coefficients for the Test Beam Settings

Section 8.2.2 presented a set of weights that is optimal when considering all the different settings used during the test beam. To disentangle the contribution of each setting, we studied how the results are affected by the following settings:

- PACIFIC Shaper Settings;
- Module Position;
- PACIFIC Thresholds.

Optimal Weights Given PACIFIC Shaper Settings

To study how the PACIFIC Shaper settings impact the way the weighting coefficients affect the cluster resolution, we produce plots of score per weights combination for the two different Shaper settings, for each other combination of Position and Thresholds used during the test beam. The results are shown in Figure 8.7 and Table 8.5.

Figure 8.7 shows the overall result, taking into account the average score of each Shaper setting for all the different test beam settings. The plot on the left is the result for the Shaper setting "February," while the one on the right is for the setting "August." The PACIFIC Shaper doesn't significantly alter the trend of the plot. However, a different optimal set of weights was measured for each Shaper setting. The "February" settings

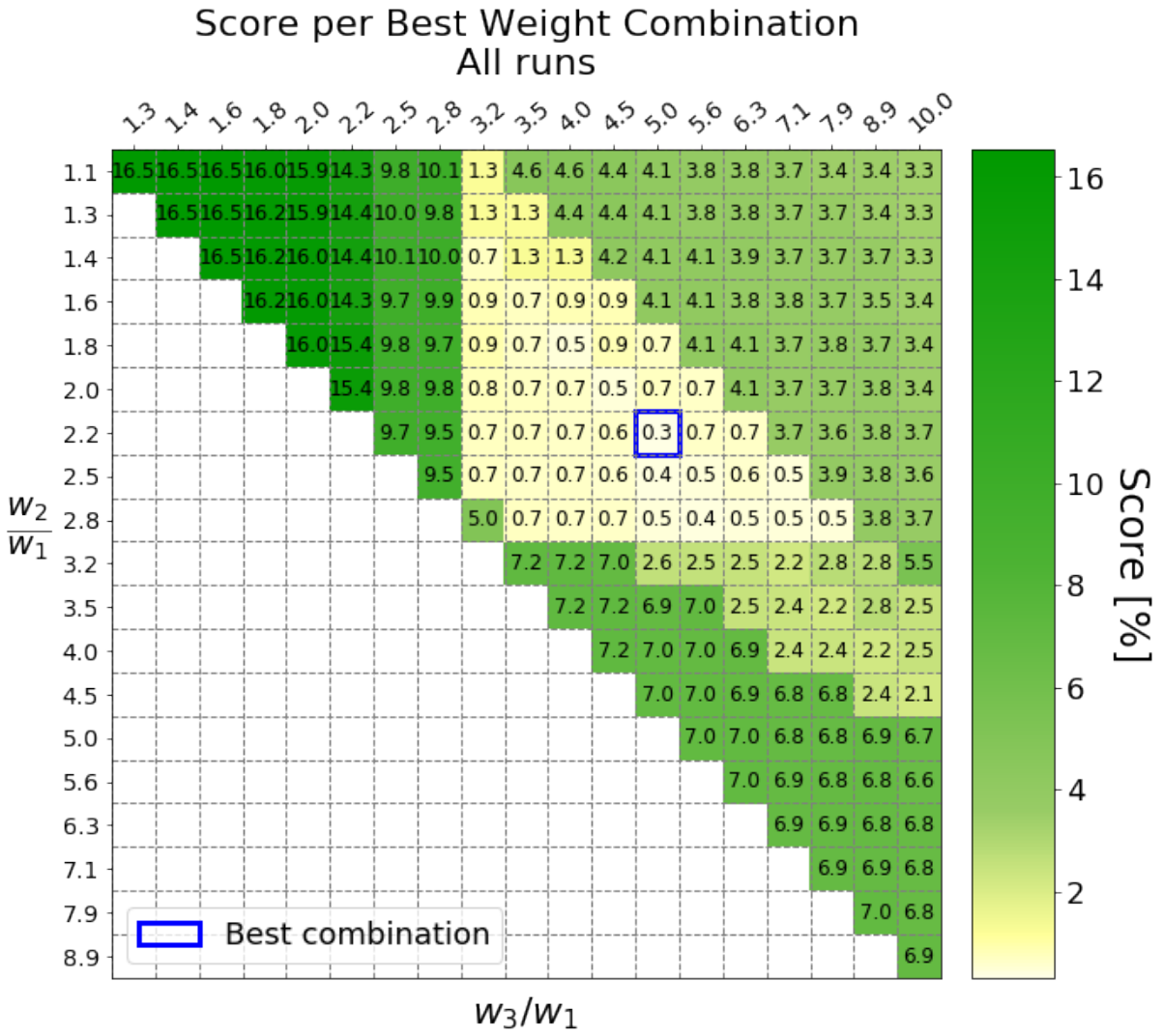


Figure 8.6: Resolution score for the given combinations of weights. Fine scan.

presented the better resolution when using weights 1.0, 2.2, and 5.0, being on average 0.2% worse than the best weights for each individual run. The "August" settings performed better with weights 1.0, 2.5, and 7.1, with a score of 0.4%.

Analyzing individually for each combination of test beam settings leads to the results presented in Table 8.5.

Optimal Weights Given Module Position

Similarly to how it was done with the PACIFIC Shaper settings, the impact of weights combination for the three different beam positions was studied. The left plot from Figure 8.8 displays the result for the runs where the beam was in the "Mirror" position ($Y_{table} = 663\,\text{mm}$) while in the right plot, the beam was in the "Center" position ($Y_{table} = -40\,\text{mm}$). The plot from Figure 8.9 shows the same for when the beam was in the "SiPM" position ($Y_{table} = -720\,\text{mm}$).

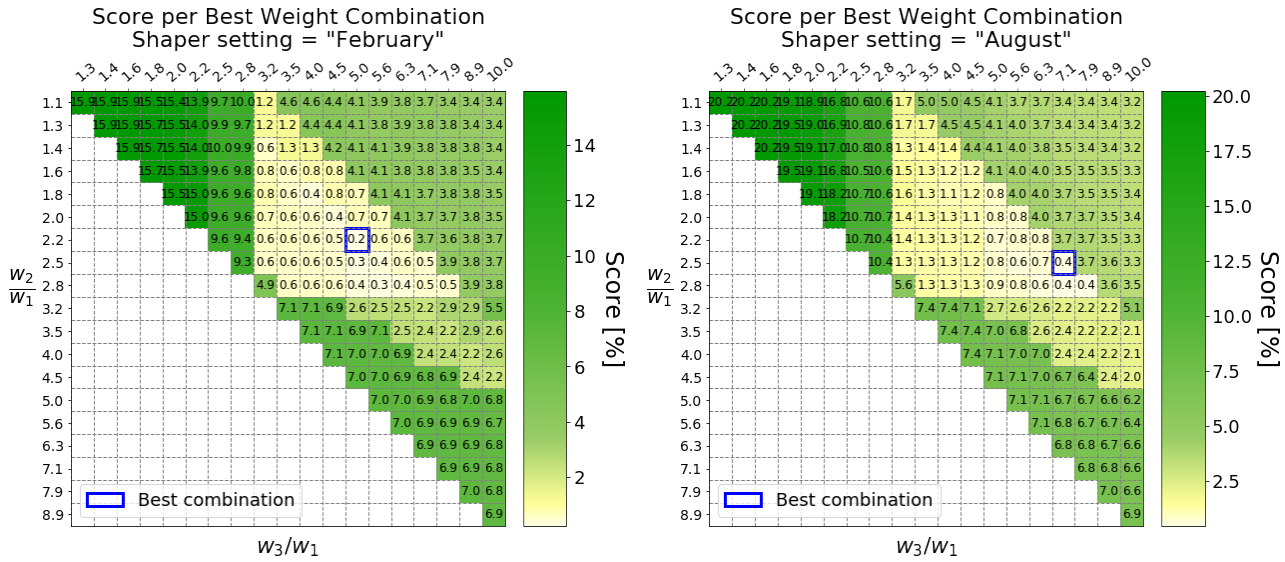


Figure 8.7: Resolution score given combinations of weights for shaper setting "February"(left plot) and "August" (right plot).

Test beam settings		**"February" settings**			**"August" settings**		
Thresholds	Position	Weights	Score	Runs	Weights	Score	Runs
0.5, 1.5, 2.5	Mirror	1.0, 2.5, 63	0.0%	3	1.0, 2.5, 63	0.0%	3
0.5, 1.5, 2.5	Center	1.0, 2.5, 63	0.0%	1	1.0, 2.5, 63	0.0%	1
0.5, 1.5, 2.5	SiPM	1.0, 2.5, 63	0.0%	1	1.0, 6.3, 16	0.0%	1
1.5, 2.5, 4.5	Mirror	1.0, 2.2, 5.0	0.1%	117	1.5, 2.5, 5.6	0.1%	9
1.5, 2.5, 4.5	Center	1.5, 2.5, 5.6	0.0%	1	1.5, 2.5, 5.6	0.0%	1
1.5, 2.5, 4.5	SiPM	1.5, 2.5, 5.6	0.0%	8	1.5, 2.5, 5.6	0.0%	6
3.5, 5.5, 9.5	Mirror	1.5, 2.5, 7.1	0.0%	3	-	-	0
3.5, 5.5, 9.5	Center	1.5, 2.5, 7.1	0.0%	1	1.5, 2.2, 5.6	0.0%	1
3.5, 5.5, 9.5	SiPM	1.5, 2.5, 7.1	0.0%	8	1.5, 2.2, 5.6	0.0%	1

Table 8.5: The best set of weighting coefficients for each test beam settings, displaying the runs with shaper settings "February" and "August" side by side for comparison.

Optimal Weights Given PACIFIC Thresholds

Figures 8.10 and 8.11 show the scores of each weight combination for three threshold settings (0.5, 1.5, 2.5), (1.5, 2.5, 4.5), (3.5, 5.5, 9.5). Note that the plot for thresholds 0.5, 1.5, and 2.5 (Figure 8.11) is in a different scale.

While the impact of the Shaper settings and the beam position is minor, the PACIFIC threshold settings drastically affect the trend displayed by the scores. The thresholds are, therefore, the dominant factor responsible for the score changes in Figure 8.6.

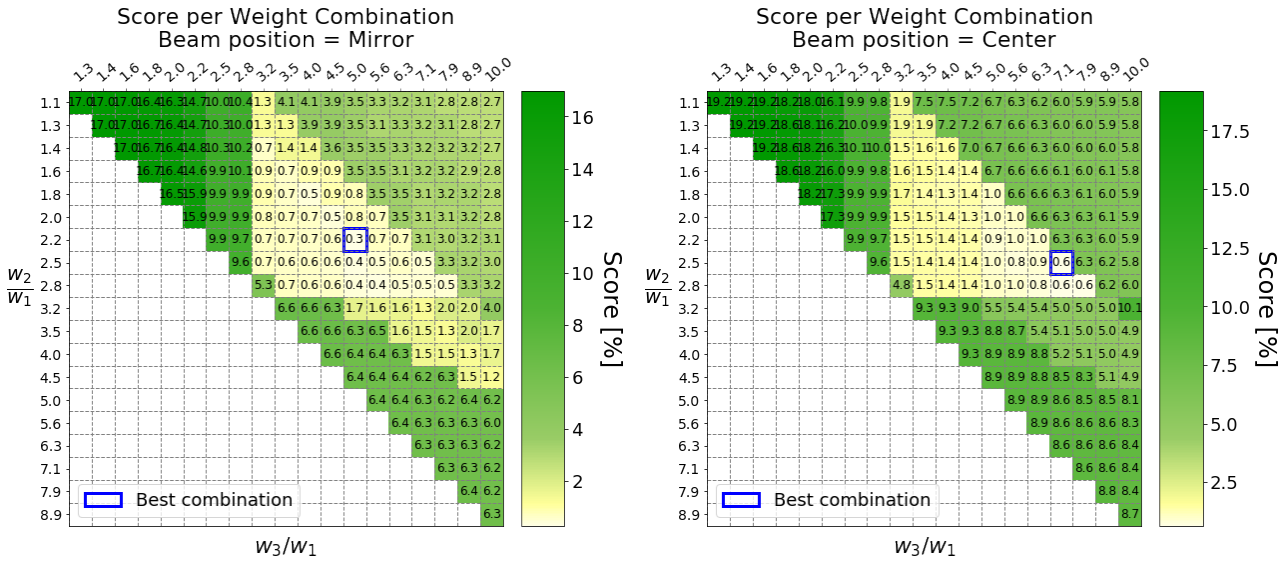


Figure 8.8: Resolution score for the given combinations of weights for the position of the beam along the fibers. Left plot: Mirror. Right plot: Center.

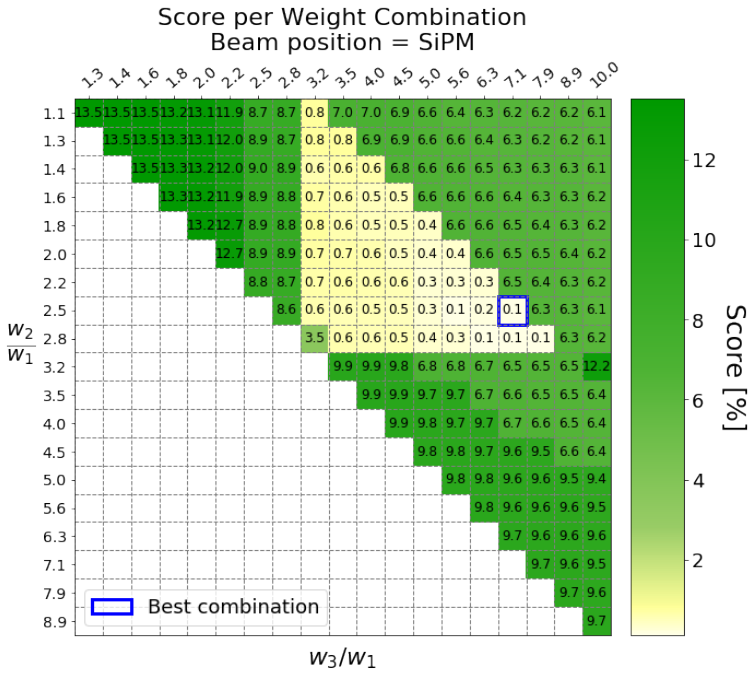


Figure 8.9: Resolution score for the given combinations of weights for the position of the beam along the fibers: SiPM.

8.2.4 The Special Case of Weights 1, 2, 6

Are the lowest scores well localized in a global minimum, such that we can firmly conclude that the weights combinations with the best score also provide the best overall resolution? The artifact denoted by the red line in Figure 8.12 seems to cast some doubts. The red line $W_3 = 3 \times W_2$ seems to correspond to a region of "discontinuity", in which scores are lower than what one would interpolate from neighboring bins.

Thus, along the red line, the score seems to have a local minimum. How deep this local

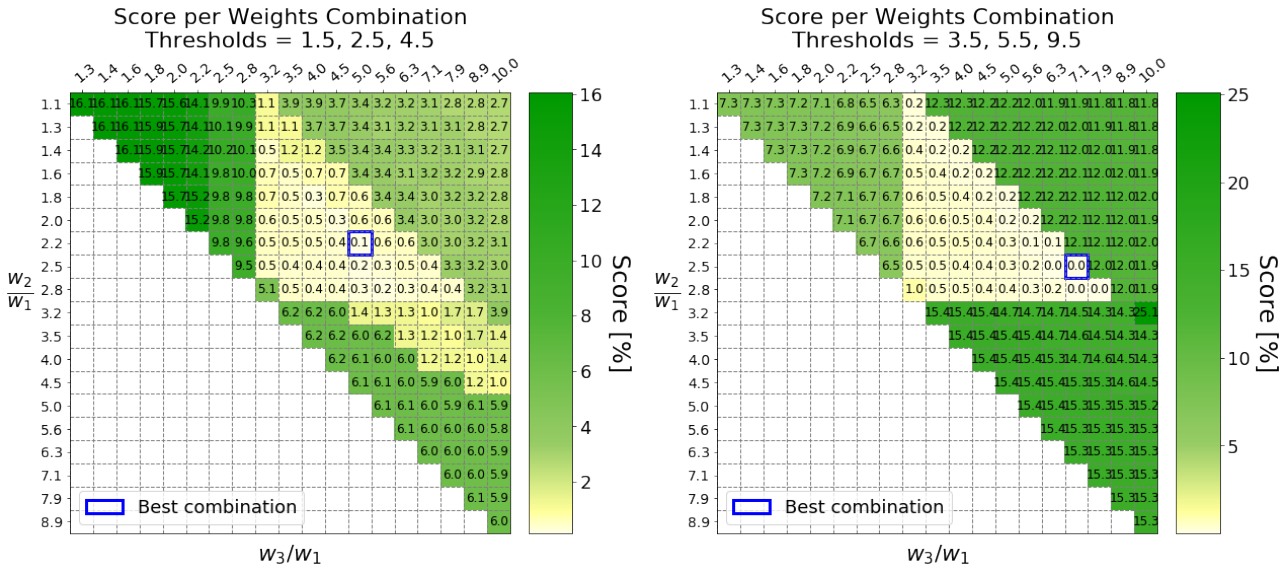


Figure 8.10: Resolution score given combinations of weights for PACIFIC thresholds. Left plot: thresholds 1.5, 2.5, 4.5. Right plot: thresholds 3.5, 5.5, 9.5.

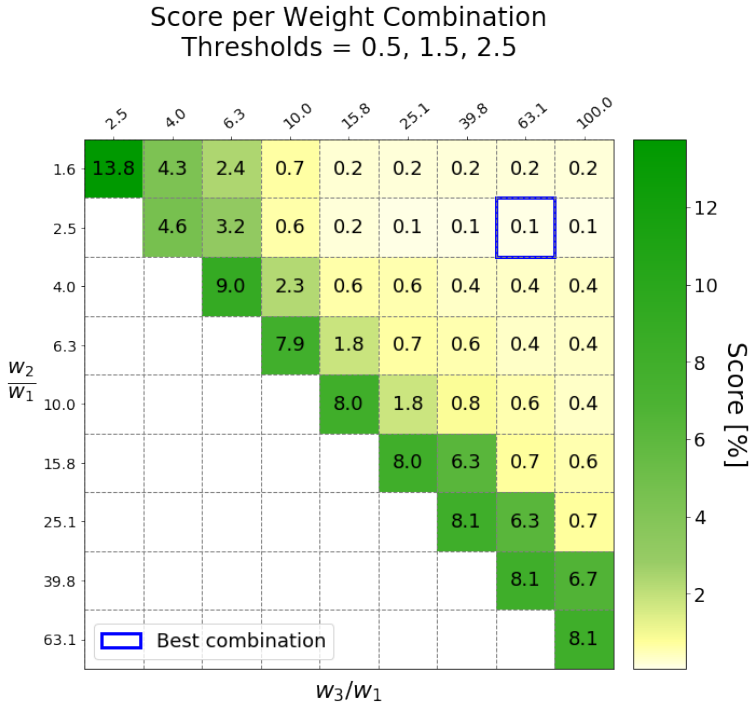


Figure 8.11: Resolution score given combinations of weights for PACIFIC thresholds 0.5, 1.5, 2.5.

minimum is, is affected by the choice of thresholds. The effect is minimal for thresholds [0.5 1.5 2.5] and [3.5 5.5 9.5], but for thresholds [1.5 2.5 4.5], the weights 1.0, 2.0, and 6.0 provide a local minimum 0.8% lower than the surroundings. Figure 8.12 shows the scores for such thresholds. One can observe that, while the bins (2.0, 5.6) and (2.0, 6.3) provide scores of 0.6% and 3.4%, respectively, the bin denoted by the red line (2.0, 6.0) in between them gives a score of -0.7%.

Figure 8.13 shows the resolution measured for all the runs with thresholds [1.5 2.5 4.5],

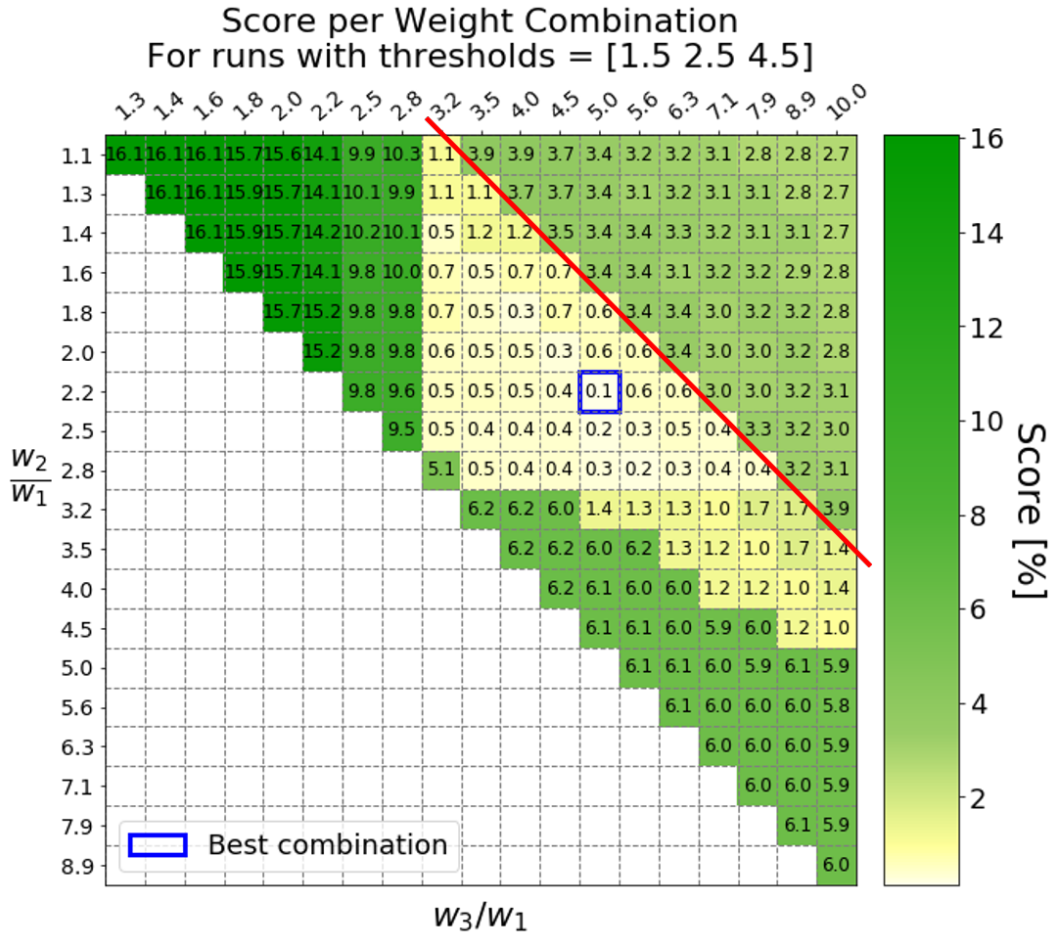


Figure 8.12: Plot of resolution score for PACIFIC thresholds 1.5, 2.5, 4.5, highlighting a local minimum region denoted by the red line.

for weights 1.0, 2.0, 6.0 (in blue) and 1.0, 2.3, 5.0 (in orange) and Figure 8.14 displays a histogram of the resolution of such runs, for both sets of weights. Both figures confirm the impression that for thresholds [1.5 2.5 4.5] the local minimum provides a better choice of weights. When the same histograms are produced for the other two threshold settings, though, shown in Figure 8.15, this is not the case.

The reason for this artifact is not fully understood, but there are strong indications that this is caused by numerical effects. The main concern is that such numerical effects will also happen in the FPGA implementation of the clustering algorithm.

If thresholds [1.5 2.5 4.5] are chosen for the experiment, it is recommended to investigate whether such artifact happens in the FPGA implementation as well and to study it systematically with a data injection interface allowing the FPGA to run test beam data or simulation data and check if the results are consistent with the software implementation. It would also be advisable to perform such a study at the beginning of the detector operation at the latest, when the FPGAs can still be reprogrammed to change the logic that executes the clusterization algorithm.

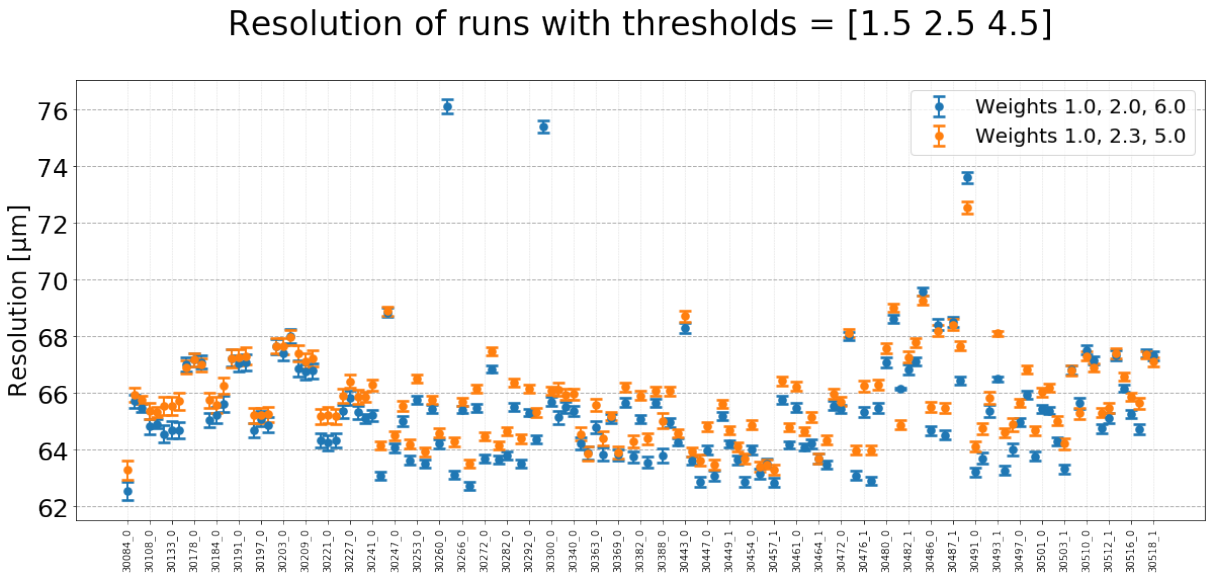


Figure 8.13: Resolution of all the runs with thresholds [1.5 2.5 4.5] for the sets of weights 1.0, 2.0, 6.0 (blue) and weights 1.0, 2.3, 5.0 (orange).

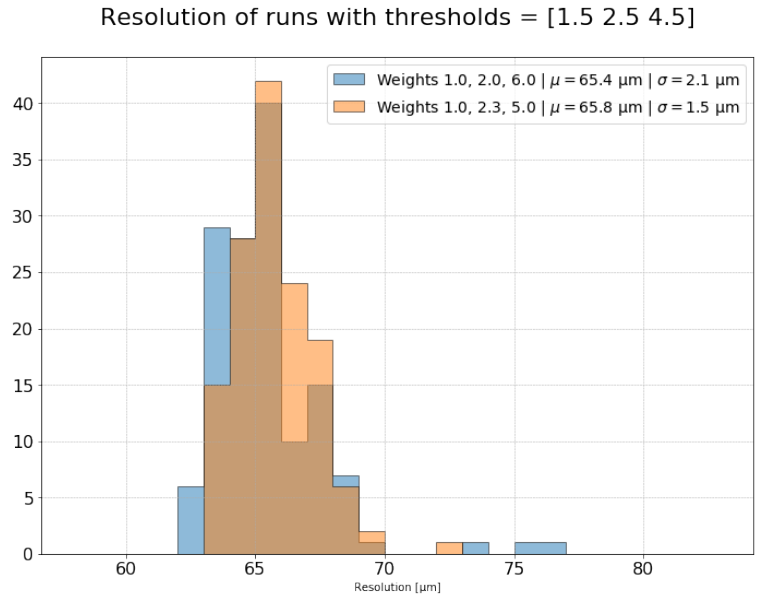
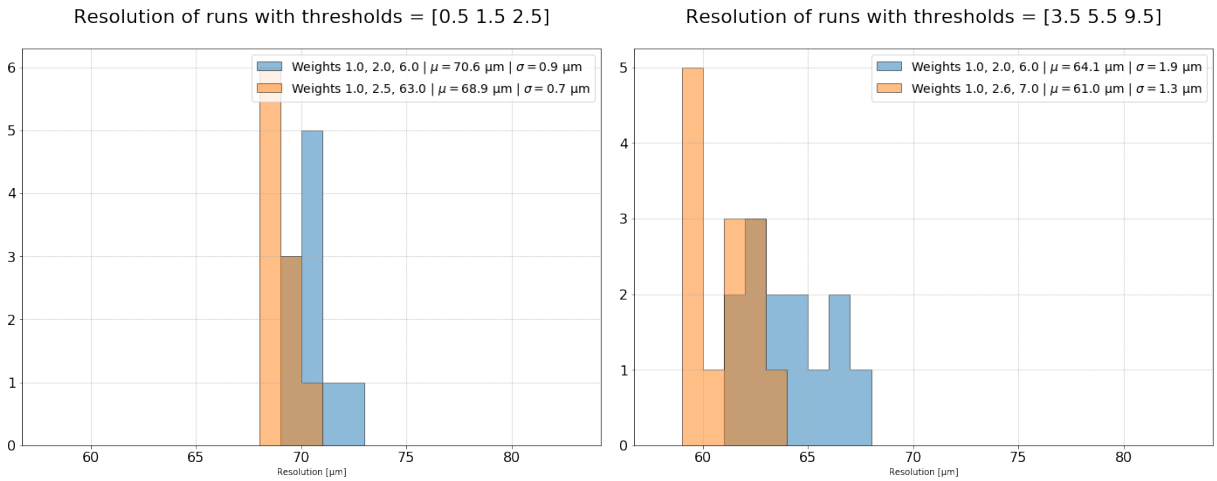


Figure 8.14: Histogram of resolution for all the runs with thresholds [1.5 2.5 4.5] for the sets of weights 1.0, 2.0, 6.0 (blue) and weights 1.0, 2.3, 5.0 (orange).

8.2.5 Conclusion

The analysis presented in the preview sections had the goal of understanding how the different test beam settings affect the optimal set of weights. The PACIFIC thresholds proved to be the dominant factor in defining which combination of weights should be optimal. Table 8.6 presents the optimal weights for all the combinations tested during the test beam.

For thresholds [1.5 2.5 4.5], the weights 1.0, 2.0, and 6.0 deliver better results than those in Table 8.6, but the reason is unclear and needs further studies in situ.



(a) Thresholds [0.5 1.5 2.5]. (b) Thresholds [3.5 5.5 9.5].

Figure 8.15: Histogram of resolution for all the runs with a specific threshold, comparing the optimal set of weights (orange) to the weights 1.0, 2.0, 6.0 (blue).

Thresholds	Shaper	Position	Weights	Score	Runs
1.5, 2.5, 4.5	February	Mirror	1.0, 2.2, 5.0	0.1%	117
1.5, 2.5, 4.5	February	Center	1.0, 2.5, 5.6	0.0%	1
1.5, 2.5, 4.5	February	SiPM	1.0, 2.5, 5.6	0.0%	8
1.5, 2.5, 4.5	August	Mirror	1.0, 2.5, 5.6	0.1%	9
1.5, 2.5, 4.5	August	Center	1.0, 2.5, 5.6	0.0%	1
1.5, 2.5, 4.5	August	SiPM	1.0, 2.5, 5.6	0.0%	6
3.5, 5.5, 9.5	February	Mirror	1.0, 2.5, 7.1	0.0%	3
3.5, 5.5, 9.5	February	Center	1.0, 2.5, 7.1	0.0%	1
3.5, 5.5, 9.5	February	SiPM	1.0, 2.5, 7.1	0.0%	8
3.5, 5.5, 9.5	August	Mirror	-	-	0
3.5, 5.5, 9.5	August	Center	1.0, 2.2, 5.6	0.0%	1
3.5, 5.5, 9.5	August	SiPM	1.0, 2.2, 5.6	0.0%	1
0.5, 1.5, 2.5	February	Mirror	1.0, 2.5, 63	0.0%	3
0.5, 1.5, 2.5	February	Center	1.0, 2.5, 63	0.0%	1
0.5, 1.5, 2.5	February	SiPM	1.0, 2.5, 63	0.0%	1
0.5, 1.5, 2.5	August	Mirror	1.0, 2.5, 63	0.0%	3
0.5, 1.5, 2.5	August	Center	1.0, 2.5, 63	0.0%	1
0.5, 1.5, 2.5	August	SiPM	1.0, 6.3, 16	0.0%	1

Table 8.6: The best set of weighting coefficients for each of the test beam settings.

8.3 Performance as a Function of PACIFIC Thresholds

To study the effects of the PACIFIC thresholds on the resolution and efficiency of the detector, we divide the data into groups of runs for which only the threshold settings differ.

Figure 8.16 shows the resolution results for each test beam settings combination, each threshold settings denoted by a different color. The resolutions of runs with thresholds [3.5, 5.5, 9.5] are around 62 µm, while runs with thresholds [1.5, 2.5, 4.5] and [0.5, 1.5, 2.5] are around 66 µm and 69 µm respectively. There is, therefore, a clear indication of how the PACIFIC thresholds affect the resolution. In fact, from Figure 8.16, one can also conclude that, although resolutions are affected by all the parameters varied during the test beam, the effect of the PACIFIC threshold is the dominating one.

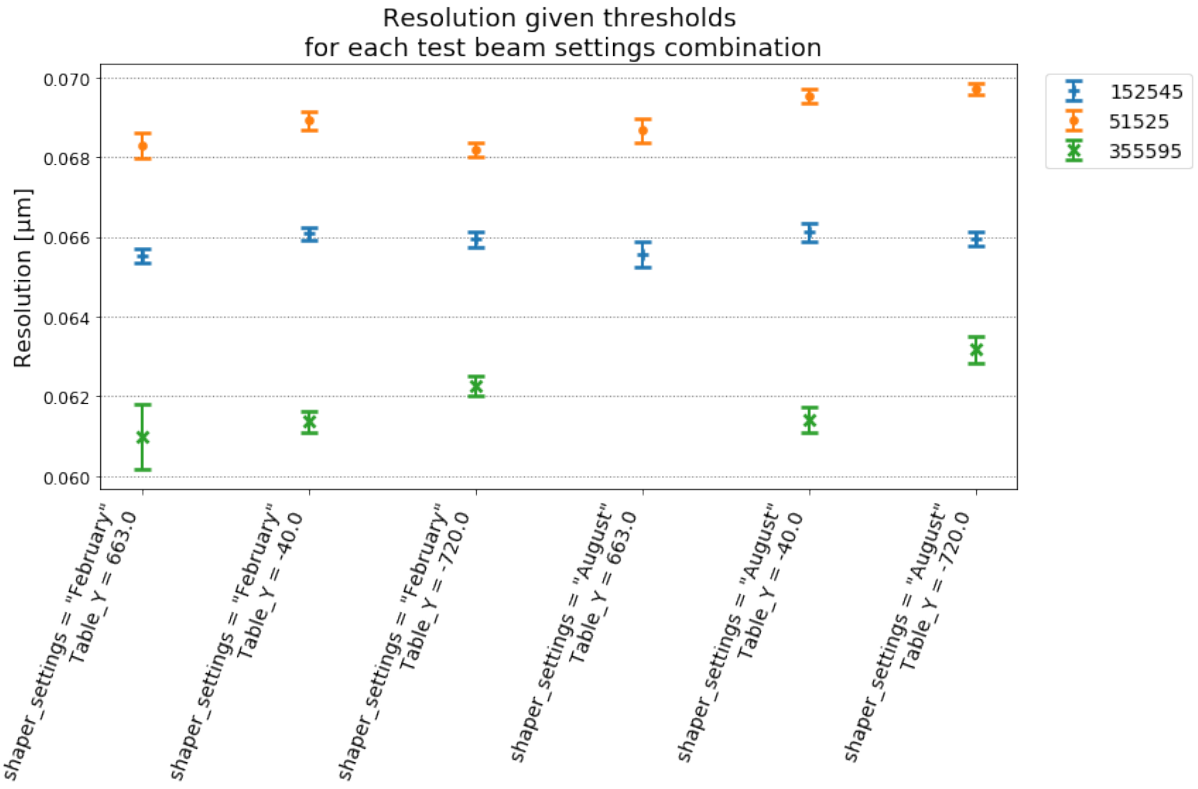


Figure 8.16: Resolution per test beam settings. The color codes denote the three PACIFIC threshold settings used during the test beam.

Figure 8.17 shows a plot similar to the previous one but for efficiency instead of resolution. As in the case of the resolution, the thresholds directly affect the efficiency as well. The runs with thresholds [3.5, 5.5, 9.5] exhibit low efficiency between 62% and 76%, while the other settings exhibit values above 96%. These two results are also shown in Figure 8.18 with a finer scale in order to ease the comparison. The runs taken with thresholds [0.5, 1.5, 2.5] show a rather flat efficiency of around 99.5%. The runs with thresholds [1.5, 2.5, 4.5] show slightly lower efficiencies between 97.2% and 98.6%, depending also on the other settings.

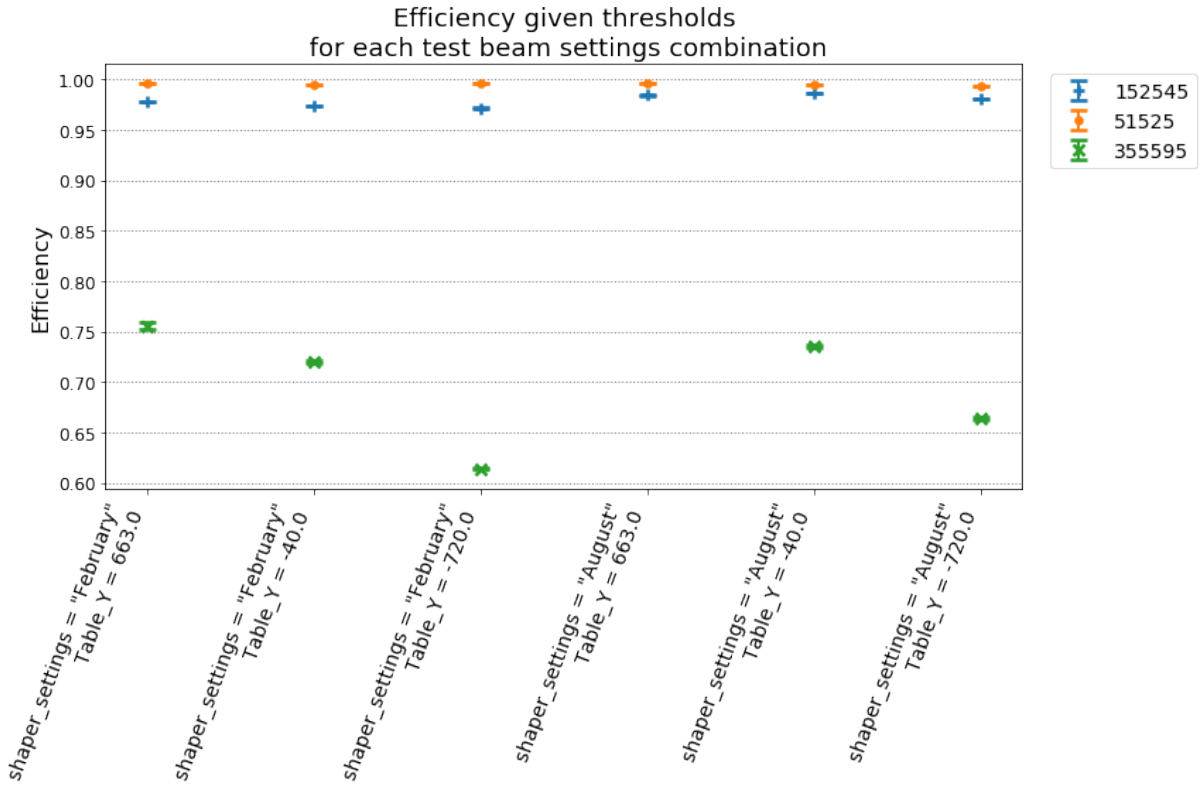


Figure 8.17: Efficiency per test beam settings. The color codes denote the threshold settings used during the test beam.

In view of these results, one can conclude that the PACIFIC thresholds are a dominant factor in the SciFi performance, their choice being a trade-off between resolution and efficiency: while the resolution increases for higher threshold values, the higher the thresholds, the lower the efficiency. Moreover, it is important to note that noise levels will put a practical limit to the lowest threshold values that can be used.

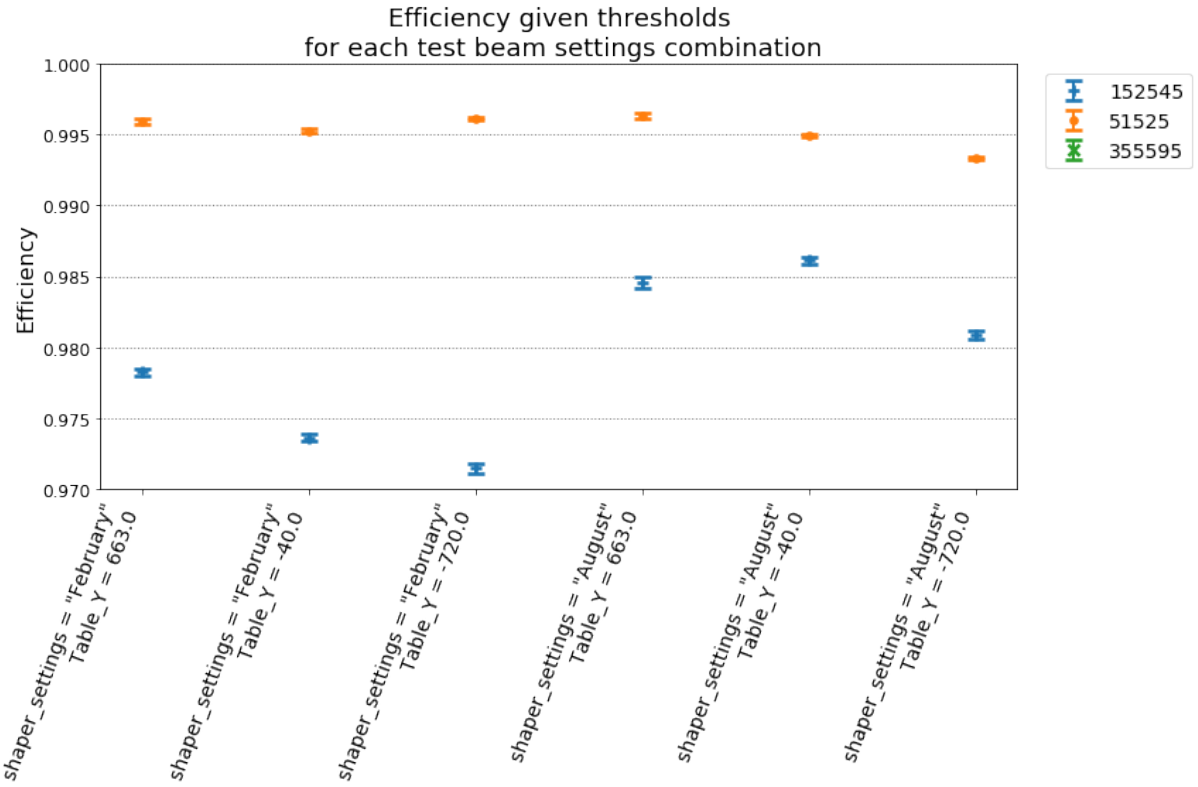


Figure 8.18: Same as Figure 8.17, zooming in the region with efficiency higher than 0.97.

8.4 Performance Dependence on PACIFIC Shaper Settings

To study how the PACIFIC shaper settings affect the resolution and efficiency of the detector, a procedure analogous to that described in Section 8.3 was used, grouping runs according to their shaper settings.

Figure 8.19 shows the resolution change due to the PACIFIC shaper settings for each combination of the other test beam settings. For the thresholds [1.5, 2.5, 4.5], the change in resolution due to the shaper settings is negligible. For the other combinations, the shaper setting August provides resolution up to 2% better than the setting February. In general, note that the shaper settings affect resolution much less significantly than the threshold settings.

Figure 8.20 is analogous to Figure 8.19 for efficiencies. One can see that the larger variations of efficiency are not due to the shaper settings: the drop in efficiency for the three settings on the right is due to the thresholds [3.5, 5.5, 9.5]. Figure 8.21 presents the same data but with a scale adjusted to zoom in the upper data points. The setting August performs 1% better than the February one for the thresholds [1.5, 2.5, 4.5]. For the thresholds [0.5, 1.5, 2.5], the shaper settings don't affect the result significantly, the biggest variation being around 0.25%.

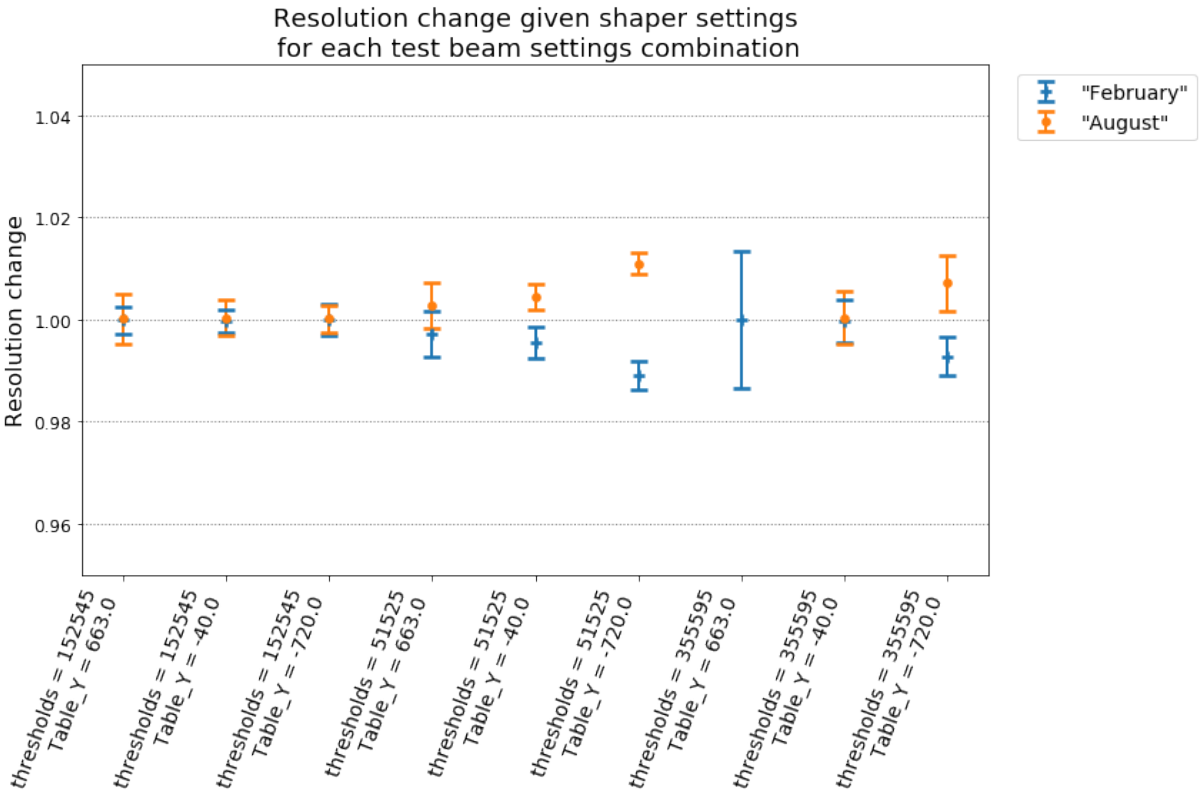


Figure 8.19: Plot of resolution per test beam settings for the two PACIFIC shaper settings denoted by the color code.

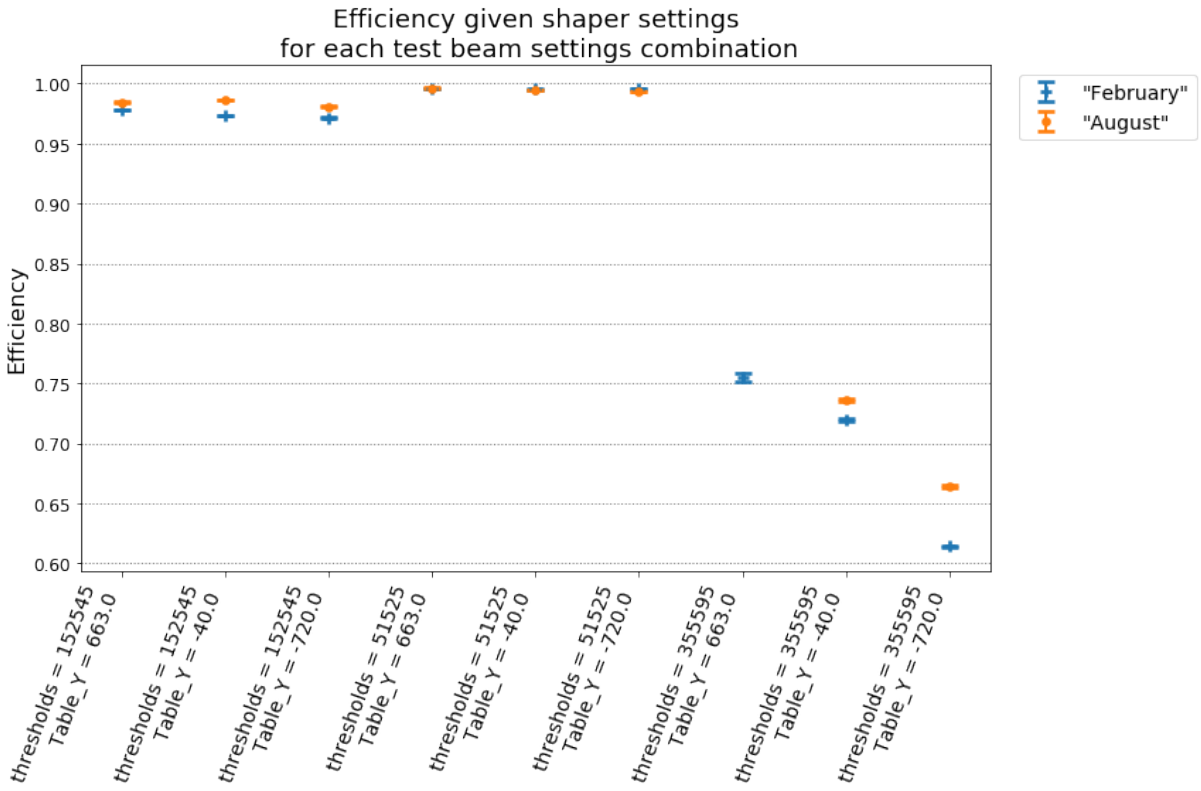


Figure 8.20: Plot of efficiency per test beam settings for the two PACIFIC shaper settings, denoted by the color code.

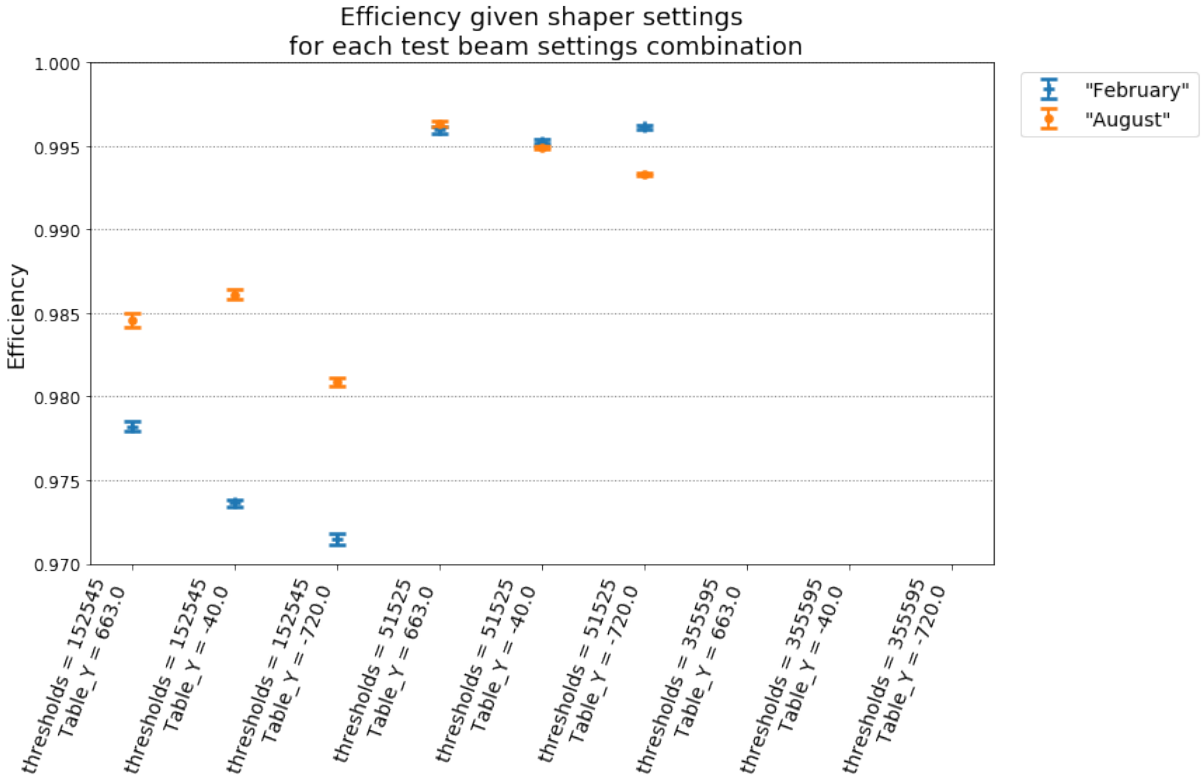


Figure 8.21: Same as Figure 8.20, but zooming in the region of efficiency of 0.97.

8.5 Performance Along the Fiber Module

The effect of the position of the particle hit along the fiber module was also studied, grouping runs according to the module position.

Figure 8.22 shows the resolution for the three different module positions for each of the other test beam settings combination. As for the shaper settings, the hit position along the fiber does not significantly affect resolution. For the thresholds [0.5, 1.5, 2.5] and [1.5, 2.5, 4.5] the resolution change is negligible (less than 2%). For the thresholds [3.5, 5.5, 9.5], hit positions closer to the Mirror (663 mm) result in a better resolution, up to 3%. The impact of the hit position on the resolution is thus significantly smaller than that of the thresholds.

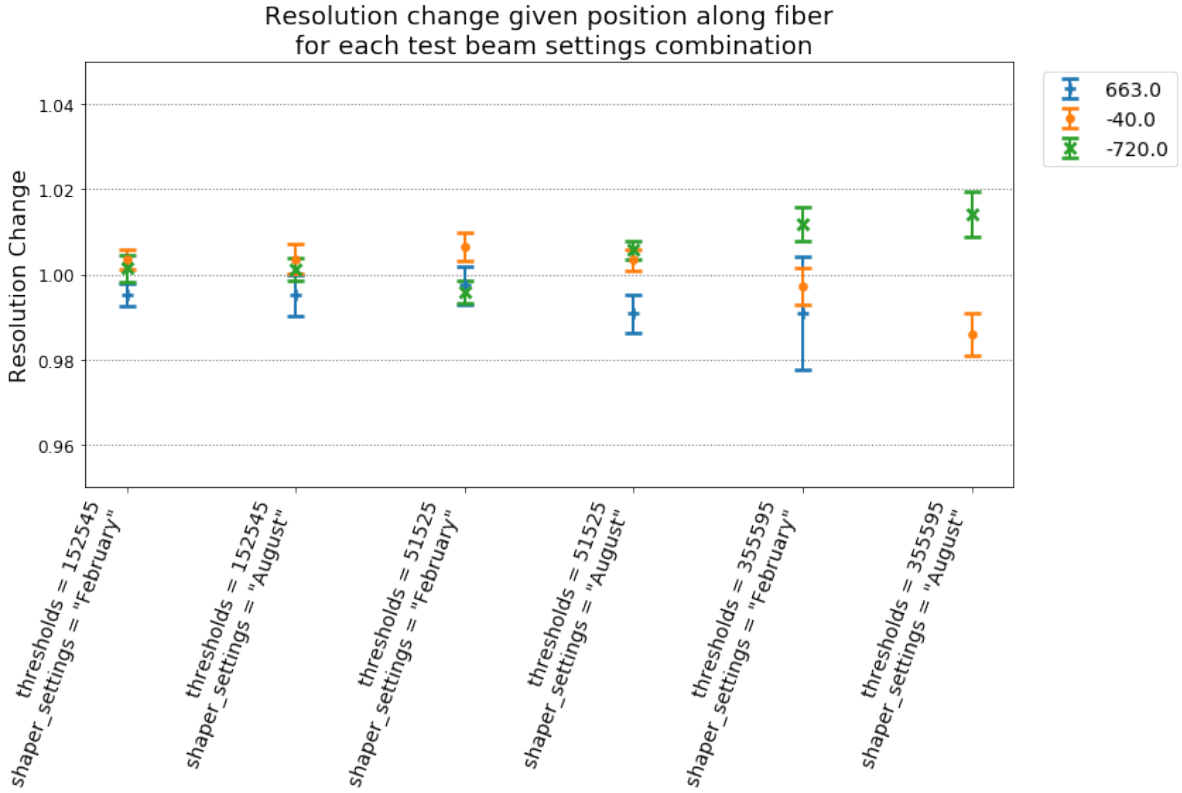


Figure 8.22: Plot of resolution per test beam settings for three module positions denoted by different color codes.

Figure 8.23 shows the results for efficiency. The data points in the lower right region of the plot show the runs with thresholds [3.5, 5.5, 9.5], having much lower efficiency (See Section 8.3). Zooming in the upper region of the scale as in Figure 8.24, one can more clearly observe that the runs in which the beam hit closer to the SiPMs consistently performed worse in terms of efficiency. The effect is smaller at lower thresholds [0.5, 1.5, 2.5]: with the shaper setting August, the region performed less than 0.5% worse than the Mirror, and for February settings, the difference is negligible. For the thresholds [1.5, 2.5, 4.5], the difference is bigger and reaches up to 0.7%.

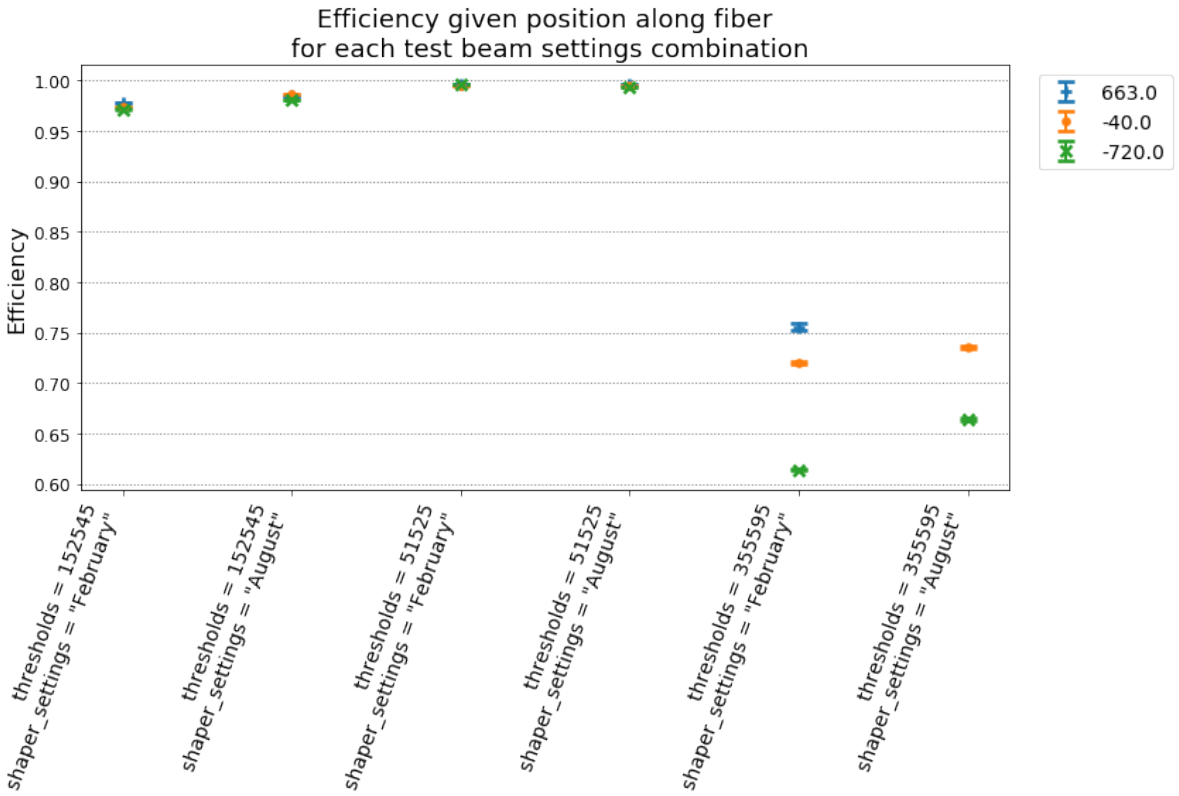


Figure 8.23: Plot of efficiency per test beam setting, for three different module positions denoted by different color codes.

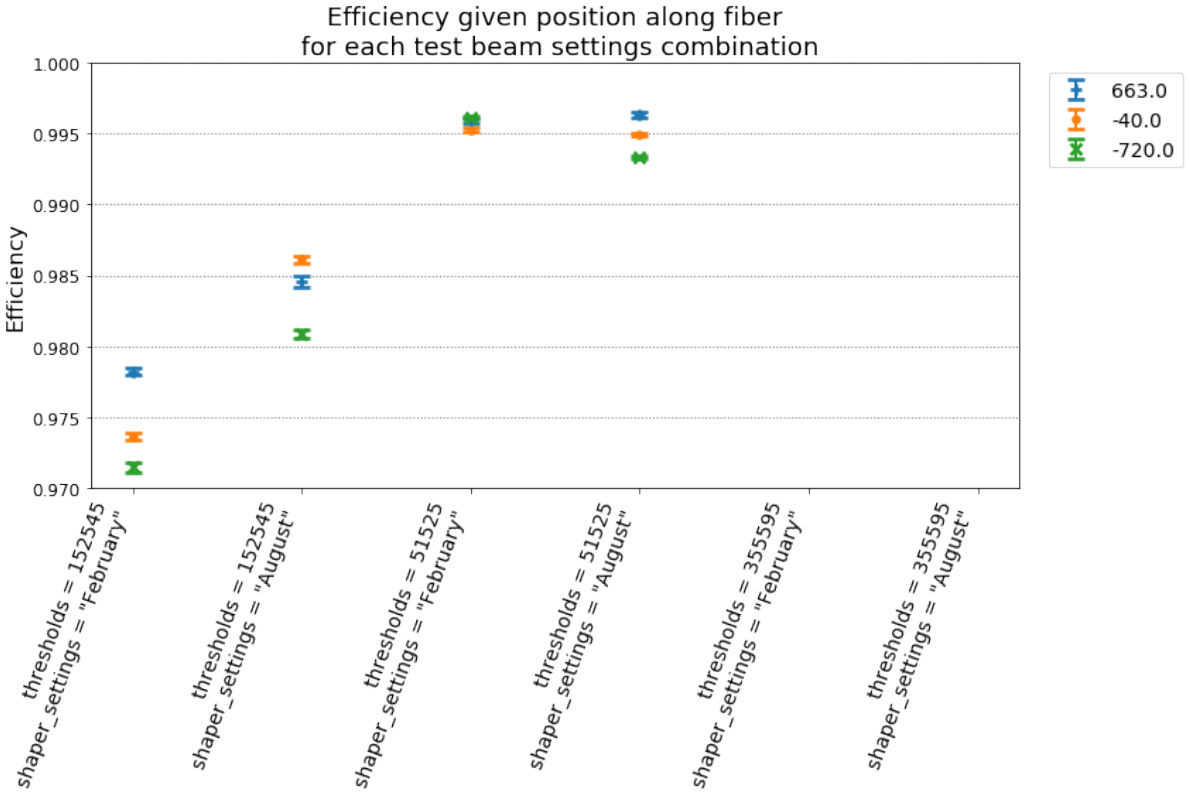


Figure 8.24: Same as Figure 8.23, zooming in the region with efficiency higher than 0.97.

8.6 Performance for Different Max Cluster Widths

The maximum cluster width allowed in the clustering algorithm is a parameter that has also been studied in terms of its effects on performance and resolution. As this parameter is set in the clustering algorithm, we could vary it offline by rerunning the algorithm for each data set. The results shown in this section have been obtained using all selected data.

Figure 8.25 shows the resolution for each of the four Max Widths tried (3, 4, 5, and 6) for all the valid test-beam runs. Smaller widths consistently provide better resolution. The vast majority of the runs exhibit the best resolution when Max Width = 3 and the worst when Max Width = 6.

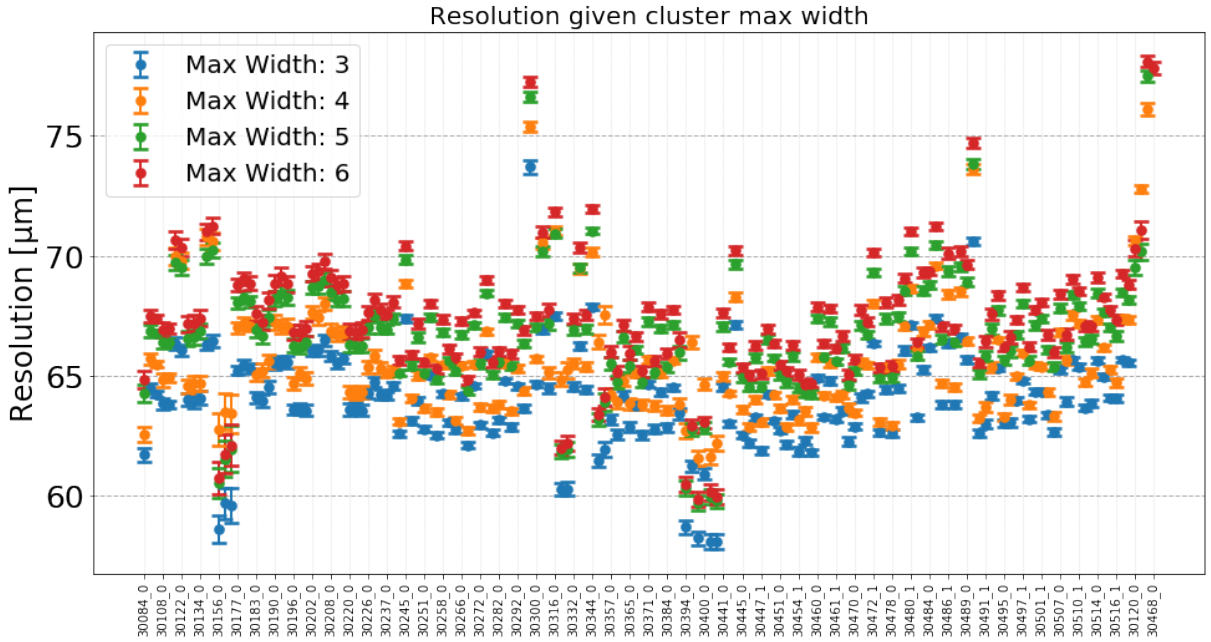


Figure 8.25: Resolution of all test-beam runs for four different settings of the Max Cluster Width in the clusterization algorithm, denoted by different color codes.

Binning the results from Figure 8.25 into a histogram results in the plot in Figure 8.26. The mean resolution varies from 64.0 µm for Max Width = 3 to 67.3 µm for Max Width = 6.

Remarks and discussion regarding the Max Width study It is important to note that the validity of the results of this study may be limited by various factors. The first one is the fact that during the whole test beam, the SciFi modules have always been perpendicular to the beam; therefore, this result may only be valid for the areas of the detector in which the particles cross the fibers perpendicularly, i.e., the regions close to the beam pipe. As the angle of the particles increases, it is expected that larger clusters will be created, and thus, consequently, the optimal max width for the clustering algorithm may vary. The factor is that we filtered the data by matching only the events with a single track recorded. If multiple clusters are created next to each other, depending on

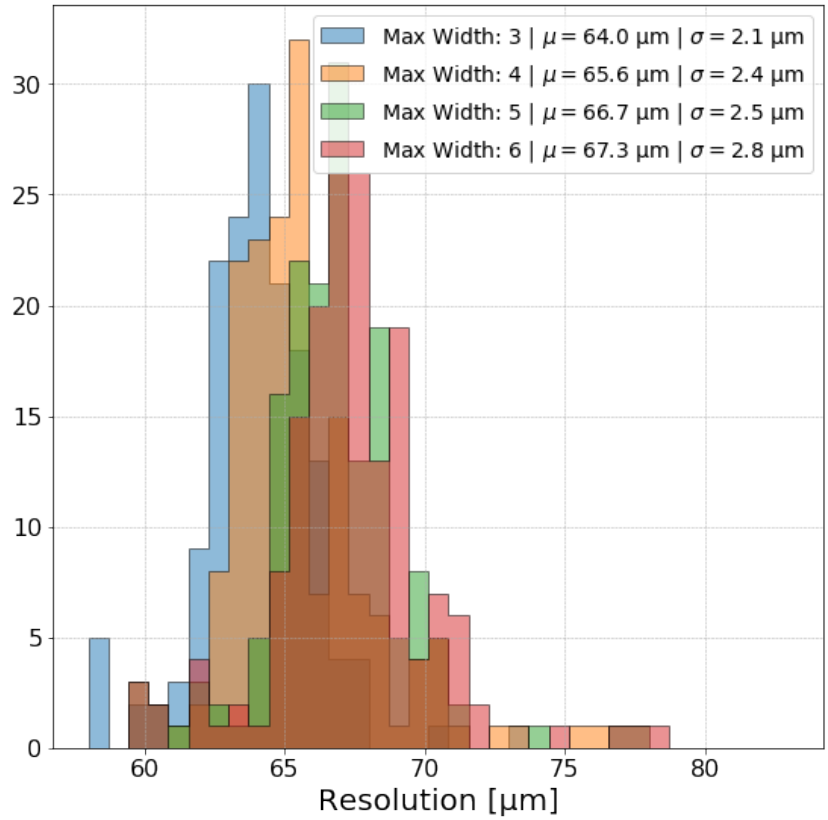


Figure 8.26: Histogram of resolution from all test-beam runs given different settings of the Max Cluster Width parameter in the clusterization algorithm, denoted by four different color codes.

the algorithm settings, it could be interpreted as multiple small clusters or a single big one, which will affect the resolution. This effect could not be studied in our analysis.

It is also worth noting that for the same reasons, the study as we performed is not sensitive to changes in efficiency unless one allows for multiple tracks in the detector, in which case the efficiency might be affected by the maximum cluster width.

8.7 Clustering Fractional bit limitation

The goal of the study presented in this section was to learn how the resolution worsens with respect to the case in which there were no limitations in data bandwidth. As the clustering was performed offline, we had the freedom to use any precision for the bits that record the cluster position. We then compared two different precisions (amount of bits representing the cluster position): eight bits as in the experiment and 64 bits.

Figure 8.27 shows the resolution results for both precisions for all valid test beam runs: as expected, the limited precision allowed by the DAQ system significantly affects the resolution. Figure 8.28 presents the data from Figure 8.27 in a histogram, allowing the estimate of the resolution loss in limiting the cluster precision to 8 bits, which is roughly 10 µm.

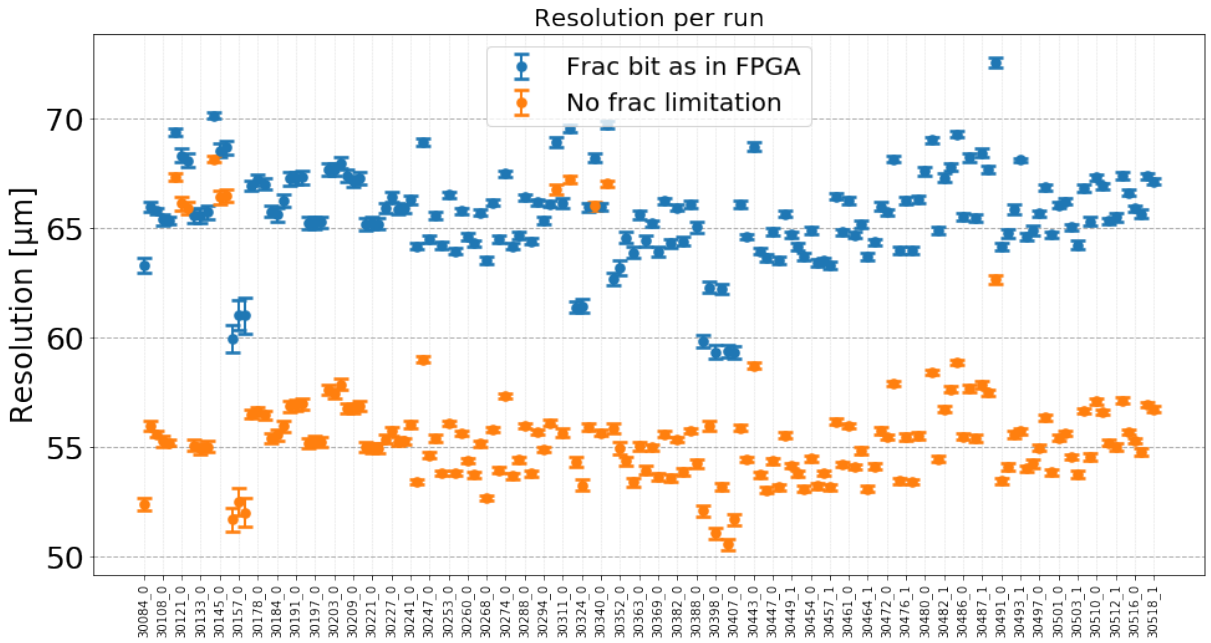


Figure 8.27: Resolution in all test beam runs, computed either limiting the fractional bit of the clusters to 1 bit (blue) or without any practical (64 bits) fractional bit limitation (orange).

Note that the requirement for the SciFi resolution to be better than 100 µm permits to absorb this loss of resolution and thus renders the combination of front-end electronics and DAQ sufficient for our goal.

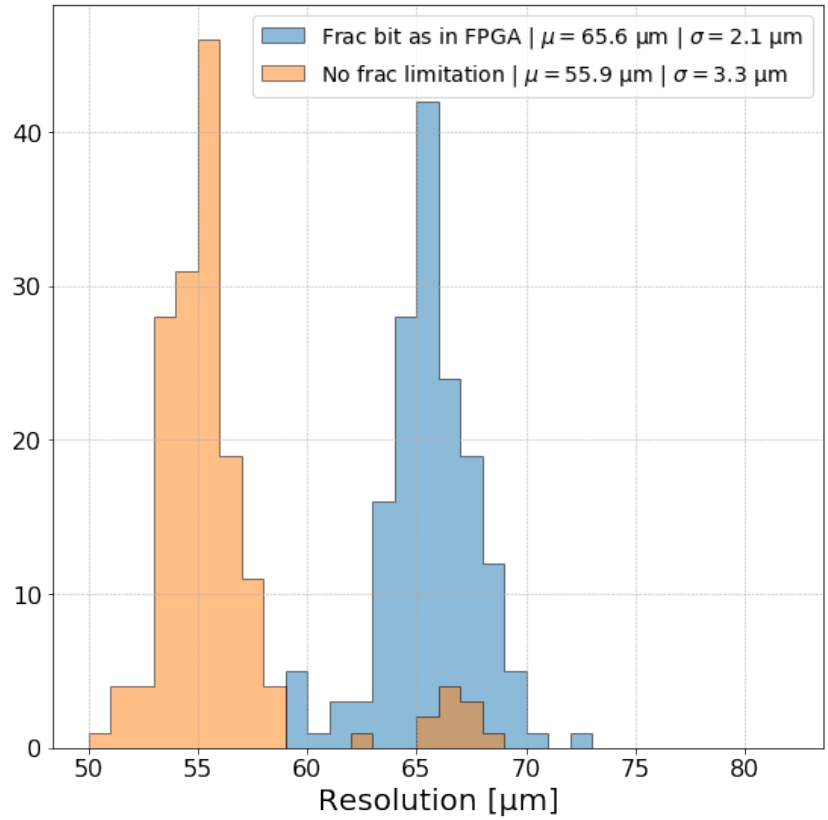


Figure 8.28: Histogram of resolution in all test beam runs, computed either limiting the fractional bit of the clusters to 1 bit (blue) or without any practical (64 bits) fractional bit limitation (orange).

8.8 Summary and Conclusions

In this chapter were presented the studies performed to understand how the parameters of the SciFi detector and its front-end electronics affect the detector's performance in terms of efficiency and resolution. The parameters studied were:

- Clusterization weighting coefficients;
- PACIFIC thresholds;
- PACIFIC shaper settings;
- Hit position along fibers;
- Maximum Cluster Width;
- Fractional bit precision.

A summary of all performance results presented in this chapter is given in Table 8.7 for all test beam settings combinations.

The weighting coefficients drastically affect the resolution of the detector and should be adjusted according to the settings of the electronics. The PACIFIC threshold settings are

Run	Thresholds	Position	Shaper	Resolution	Efficiency
30344	51525	SiPM	"August"	(69.70 ± 0.15) µm	(99.30 ± 0.01) %
30336	51525	SiPM	"February"	(68.20 ± 0.19) µm	(99.60 ± 0.01) %
30316	51525	Center	"August"	(69.50 ± 0.17) µm	(99.50 ± 0.01) %
30305	51525	Center	"February"	(68.90 ± 0.22) µm	(99.50 ± 0.01) %
30146	51525	Mirror	"August"	(68.70 ± 0.31) µm	(99.60 ± 0.02) %
30121	51525	Mirror	"February"	(68.30 ± 0.31) µm	(99.60 ± 0.02) %
30340	152545	SiPM	"August"	(66.00 ± 0.18) µm	(98.10 ± 0.03) %
30332	152545	SiPM	"February"	(65.90 ± 0.20) µm	(97.10 ± 0.03) %
30311	152545	Center	"August"	(66.10 ± 0.23) µm	(98.60 ± 0.03) %
30300	152545	Center	"February"	(66.10 ± 0.15) µm	(97.40 ± 0.02) %
30133	152545	Mirror	"August"	(65.60 ± 0.32) µm	(98.50 ± 0.04) %
30249	152545	Mirror	"February"	(65.50 ± 0.18) µm	(97.80 ± 0.03) %
30352	355595	SiPM	"August"	(63.20 ± 0.34) µm	(66.40 ± 0.14) %
30396	355595	SiPM	"February"	(62.30 ± 0.24) µm	(61.40 ± 0.10) %
30324	355595	Center	"August"	(61.40 ± 0.31) µm	(73.60 ± 0.13) %
30320	355595	Center	"February"	(61.40 ± 0.26) µm	(72.00 ± 0.12) %
30158	355595	Mirror	"February"	(61.00 ± 0.82) µm	(75.50 ± 0.36) %

Table 8.7: Overview of resolution and efficiency in all test beam runs for each settings combination.

the dominant factor affecting the resolution for given weights. The optimal set of weights can be found in Table 8.6.

The choice of PACIFIC thresholds directly affects both the resolution and the efficiency of the detector. It is, in fact, the dominant factor. The threshold settings are a trade-off: resolution gets better as one increases the thresholds while efficiency gets worse. When choosing a set of thresholds, it is important to take into account the noise levels.

The choice of the PACIFIC shaper settings does not significantly affect resolution. The effect is either negligible or limited to 2%. The efficiency is not affected for thresholds [0.5, 1.5, 2.5], and for thresholds [1.5, 2.5, 4.5], it can change the efficiency from 97.2% to 98.6% (depending on the other settings).

Concerning the hit position along fibers, the resolution is not significantly affected by it.

While the efficiency is lower closer to the SiPMs, the extent of this can be neglected when tuning the detector parameters, as the other settings affect the detector's performance considerably more.

Removing the hardware limitations on the fractional bit precision, it was possible to push the SciFi resolution down to around 56 µm. The resolution of 66 µm obtained with the affordable front-end and back-end electronics is nevertheless comfortably fulfilling the detector specifications.

Bibliography

[1] L. Evans and P. Bryant. Lhc machine. *JINST*, 3:S08001, 2008.

[2] ATLAS Collaboration, G. Asd, et al. The atlas experiment at the cern large hadron collider. *JINST*, 3:S08003, 2008.

[3] CMS Collaboration, S. Chatrchyan, et al. The cms experiment at the cern lhc. *JINST*, 3:S08004, 2008.

[4] LHCb Collaboration, A. Augusto Alves Jr, et al. The lhcb detector at the lhc. *JINST*, 3:S08005, 2008.

[5] ALICE Collaboration, K. Aamodt, et al. The alice experiment at the cern lhc. *JINST*, 3:S08002, 2008.

[6] TOTEM Collaboration, G. Anelli, et al. The totem experiment at the cern large hadron collider. *JINST*, 3:S08007, 2008.

[7] LHCf Collaboration, O. Adriani, et al. The lhcf detector at the cern large hadron collider. *JINST*, 3:S08006, 2008.

[8] James Pinfold, R Soluk, Y Yao, S Cecchini, G Giacomelli, M Giorgini, L Patrizii, G Sirri, D H Lacarrère, K Kinoshita, J Jakubek, M Platkevic, S Pospísil, Z Vykydal, T Hott, A Houdayer, Claude Leroy, J Swain, D Felea, D Hasegan, G E Pavalas, and V Popa. Technical Design Report of the MoEDAL Experiment. Technical Report CERN-LHCC-2009-006. MoEDAL-TDR-001, Jun 2009.

[9] LHCb Collaboration et al. Lhcb rich. Technical report, European Organization for Nuclear Research - CERN, 2000.

[10] LHCb Collaboration et al. Lhcb calorimeters. Technical report, European Organization for Nuclear Research - CERN, 2000.

[11] LHCb Collaboration et al. Lhcb muon system. Technical report, European Organization for Nuclear Research - CERN, 2001.

[12] Letter of Intent for the LHCb Upgrade. Technical Report CERN-LHCC-2011-001. LHCC-I-018, CERN, Geneva, Mar 2011.

[13] I Bediaga, J M De Miranda, F Ferreira Rodrigues, J Magnin, A Massafferri, I Nasteva, A C dos Reis, S Amato, K Carvalho Akiba, L De Paula, O Francisco, M Gandelman, A Gomes, J H Lopes, J M Otalora Goicochea, E Polycarpo, M S Rangel, B Souza

De Paula, D Vieira, C Gobel, J Molina Rodriguez, and et al. Framework TDR for the LHCb Upgrade: Technical Design Report. Technical Report CERN-LHCC-2012-007. LHCb-TDR-12, CERN, Geneva, Apr 2012.

[14] LHCb Trigger and Online Upgrade Technical Design Report. Technical Report CERN-LHCC-2014-016. LHCB-TDR-016, CERN, Geneva, May 2014.

[15] J. P. Cachemiche, P.-Y. Duval, R Le Gac, F Hachon, and F Réthoré. The PCIe-based readout system for the LHCb experiment. In *Topical Workshop on Electronics for Particle Physics*, volume 11, Lisbon, Portugal, September 2015.

[16] Neus Lopez March and Matthias Karacson. Radiation studies for the LHCb tracker upgrade. Technical Report LHCb-PUB-2014-022. CERN-LHCb-PUB-2014-022. LHCb-INT-2013-003, CERN, Geneva, Feb 2014.

[17] Gbt project home page. Available in: https://espace.cern.ch/GBT-Project/default.aspx. [Accessed in 13/May/2016].

[18] Jose Mazorra De Cos. PACIFIC : The readout ASIC for the SciFi Tracker planned for the upgrade of the LHCb detector. Sep 2015.

[19] Federico Alessio and Richard Jacobsson. System-level Specifications of the Timing and Fast Control system for the LHCb Upgrade. Technical report, CERN, Geneva, Feb 2014.

[20] Violaine Bellee, Martin Stefan Bieker, Jacco Andreas De Vries, Sevda Esen, Luca Pescatore, Jeroen Van Tilburg, and Julian Tarek Wishahi. Simulation of the SciFi detector. Technical report, CERN, Geneva, 2018.

[21] The boole project. Available in: http://lhcbdoc.web.cern.ch/lhcbdoc/boole/. [Accessed in 23/April/2023].

[22] Adam Thornton. CHARM Facility Test Area Radiation Field Description. Apr 2016.

[23] Giuseppe Battistoni, Till Boehlen, Francesco Cerutti, Pik Wai Chin, Luigi Salvatore Esposito, Alberto Fassò, Alfredo Ferrari, Anton Lechner, Anton Empl, Andrea Mairani, Alessio Mereghetti, Pablo Garcia Ortega, Johannes Ranft, Stefan Roesler, Paola R. Sala, Vasilis Vlachoudis, and George Smirnov. Overview of the fluka code. *Annals of Nuclear Energy*, 82:10–18, 2015. Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013, SNA + MC 2013. Pluri- and Trans-disciplinarity, Towards New Modeling and Numerical Simulation Paradigms.

[24] Microsemi. IGLOO2 and SmartFusion2 65 nm Commercial Flash FPGAs, Interim Summary of Radiation Test Results. Oct 2014. 51000013-2/10.14 Available at www.microsemi.com.

[25] K. Akiba, M. van Beuzekom, H. Boterenbrood, E. Buchanan, J. Buytaert, W. Byczynski, X. Cid Vidal, P. Collins, E. Dall'Occo, A. Dosil Suárez, and et al. Lhcb velo timepix3 telescope. *Journal of Instrumentation*, 14(05):P05026–P05026, May 2019.

[26] T Poikela, J Plosila, T Westerlund, M Campbell, M De Gaspari, X Llopart, V Gromov, R Kluit, M van Beuzekom, F Zappon, V Zivkovic, C Brezina, K Desch, Y Fu, and A Kruth. Timepix3: a 65k channel hybrid pixel readout chip with simultaneous ToA/ToT and sparse readout. *Journal of Instrumentation*, 9(05):C05013–C05013, may 2014.

[27] Marco Clemencic, Hubert Degaudenzi, Pere Mato, Sebastien Binet, Wim Lavrijsen, Charles Leggett, and Ivan Belyaev. Recent developments in the LHCb software framework gaudi. *Journal of Physics: Conference Series*, 219(4):042006, apr 2010.

[28] Root ttree class reference. Available in: https://root.cern/doc/master/classTTree.html. [Accessed in 23/April/2023].

Acknowledgements

This thesis only exists thanks to the support and kindness of many people to whom I am extremely grateful. My PhD journey was not the smoothest ride, but I succeeded, and I owe this to the most comprehensive and supportive supervisors I could have had and amazing friends who were there for me.

Antonio Pellegrino, thank you for helping me guide my career since before my PhD. All the advice, teachings, scolds, support, and fun moments will not be forgotten. FORZA LECCE!

Federico Alessio, thank you (firstly for ignoring the last sentence but mostly) for the incredible support and encouragement for me to finish the thesis and the PhD formalities, especially during the difficult times of the pandemic.

I'd like to thank all the professors and teachers who have supported me throughout my life, allowing me to obtain a doctoral degree. Thank you **Andre Massaferri** for the guidance during my master's and the opportunities that led me to CERN and later the PhD at Nikhef.

A huge thanks to all the people from the Nikhef LHCb team. To be more specific, thank you **Marcel** for managing the group so well and providing one of the best work environments I have experienced! Thank you **Niels** for the advice, support, and nice moments shared at the LHCb Control Room.

Many thanks to my paranymphs **Jacco** and **Maarten**, to meu irmán galego **Carlos**, and my bfys colleagues **Laurent**, **Katya**, **Mick**, **Elena**, **Laís**, **Sean**, **Roman**, **Daniel**, **Michele**, **Cristina**, **Silvia**, **Lera**, **Igor**, and **Hilbrand** for all the fun moments shared in Amsterdam.

Thank you **Wilco** for all the electronic expertise shared, and together with **Snow** and **Blake** for the many challenges we've faced together to put the SciFi to work.

Thank you **Jordy**, **Alice**, **Anamika**, and **Michiel** for such great adventures together. We were really good at holidaying.

Thanks to my Brazilian CERN Gang, in particular, **Ulisses**, **Diogo**, **Breno**, and **Lucas**, for a variety of fun moments, work adventures, and the support to write my thesis.

Thanks to Jacco, Laurent, and Jordy for translating the Dutch parts of my thesis and Lucas for helping me review it.

To conclude, a special thanks to my parents, **Osvaldo** and **Laís**, who have taught me to pursue knowledge as it is a possession no one can take away from you, and for setting my best examples of hard work and perseverance. I owe it all to you :)

Summary

Why, if matter and anti-matter particles are created and destroyed together, is everything we see made of matter? Finding out what tipped the balance is one of the challenges that the Large Hadron Collider Beauty (LHCb) Experiment [4] is set to meet. LHCb is located at the Large Hadron Collider (LHC) [1], the main particle accelerator at CERN, 100 meters underground on the border between France and Switzerland. The LHC, a 27 km long collider, was designed to induce collisions between protons (and heavy ions) at a center-of-mass energy of up to 14 TeV and with a luminosity of up to $10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$.

Following a major upgrade of the LHC during the so-called LS2, the LHCb experiment also underwent an upgrade in order to make it suitable to operate at higher luminosity. The increase in the luminosity will result in a significant increase in the level of incident radiation, in the detector occupancy level, and in the volume of data generated. To make efficient use of the large increase in data volume, the LHCb upgrade foresees the elimination of the Level 0 trigger and the acquisition of data at a rate of 40MHz. The radiation levels required the validation of all system components and the use of appropriate electronics. The volume of data required replacing all electronics for the data acquisition of the experiment and the front-end electronics (FEE). Because the FEE is embedded in the detection modules, the Vertex Locator, Inner Tracker, and Trigger Tracker also needed to be replaced. Moreover, the higher level of occupancy in the Outer Tracker required a complete overhaul of its architecture, which led to the conception of the "Scintillating Fiber Detector" (SciFi), which replaced the Outer Tracker and Inner Tracker.

The SciFi is now the main tracker in the LHCb spectrometer and is expected to measure charged-particle tracks with an efficiency greater than 99% and a spatial resolution better than 70 µm. The task of the SciFi is to provide the hit information necessary to reconstruct the tracks, enabling LHCb to determine the momentum of charged particles that pass through the detector; moreover, it will provide position information to the RICH particle identification system.

In order to validate the SciFi front-end electronics for operation up to the total radiation dose expected and beyond, tests were performed at the **C**ERN **H**igh-energy **A**ccele**R**ator test facility (CHARM) [22] in two campaigns, both consisting of one week of preparation and one week of irradiation. During the first campaign, the Device Under Test was a single set of the SciFi front-end electronics, the so-called Readout Box (ROB); in the second campaign, two ROBs were tested. The objective of the tests was to measure and understand the effects of radiation on the on-detector FPGAs and to qualify the SciFi front-end as a whole for the radiation dose expected in the experiment, including a safety margin. The effects on the FPGAs that have been studied are:

- **Speed Degradation:** the propagation speed of electrical signals within the FPGA degrades with the increase of the radiation dose.
- **Leakage Current:** the power consumption varies due to an increase in leakage current that can be caused by the increase of the radiation dose.
- **FPGAs Reprogrammability:** the FPGAs lose their ability to be reprogrammed after absorbing a certain radiation dose.
- **Single Event Upsets (SEU):** a passing particle deposits charge at the base of a transistor, which may charge or discharge volatile memory cells as a consequence, causing so-called bit-flips.
- **Single Event Latch-ups (SEL):** an SEU can cause a short circuit in the chip, which can lead to overheating and damage to the device.

Both campaigns were successful, lasting a month in total and producing enough data for the studies. The results have shown that speed degradation, leakage current, and SELs will have a limited impact during the detector's lifetime. However, after sustaining a certain dose of radiation, the FPGAs will not be able to be programmed anymore, and this needs to be taken into account in the planning of the detector operations. The rate of SEUs can be mitigated by using triple modular redundancy and reduced to a level that does not affect the detector operation. The conclusion is that concerning the items above, the SciFi front-end electronics has been fully qualified for operation under the radiation dose expected for the LHCb experiment during run 3, within a factor of two of safety.

The mass-production SciFi modules and readout electronics were tested for the first time using a particle beam, a procedure commonly referred to as a "test beam." This is done to characterize the behavior of SciFi and determine key parameters for different operating settings in order to optimize the detector performance in all running scenarios. The test beam was a "slice test" of two full-width scintillating fiber modules coupled to the SciFi front-end electronics. The data acquisition was performed with the so-called "MiniDAQ2", a preliminary standalone version of the LHCb PCIe40 DAQ System. The test was performed in the North Area of the CERN Prevessin site, using the beamline H8A from the SPS accelerator. The main objective of the test beam was to understand how different parameters of the readout system can affect single-hit efficiency and resolution. The parameters studied were:

- Clusterization weighting coefficients;
- PACIFIC thresholds;
- PACIFIC shaper settings;
- Hit position along fibers;
- Maximum Cluster Width;
- Fractional bit precision.

An Analysis Framework was developed to analyze the test beam data, with the goal of providing all the necessary tools to perform the analysis of individual runs and to generate

the desired results using all runs of interest. The Framework was divided into 3 main parts: the Event Matching software that generates a lookup-table with matching events from the SciFi readout and the telescope readout, the Run Data Analysis software that reads the data and computes results for a specific run, and the Results Plotter that is a collection of python libraries and Jupyter Notebooks that use as input all the run-specific results of interest generated by the Run Data-Analysis and generate the final results.

A summary of all performance results is given in Table S1 for all test beam settings combinations.

	Parameters			Results	
Run	**Thresholds**	**Position**	**Shaper**	**Resolution**	**Efficiency**
30344	51525	SiPM	"August"	(69.70 ± 0.15) µm	(99.30 ± 0.01) %
30336	51525	SiPM	"February"	(68.20 ± 0.19) µm	(99.60 ± 0.01) %
30316	51525	Center	"August"	(69.50 ± 0.17) µm	(99.50 ± 0.01) %
30305	51525	Center	"February"	(68.90 ± 0.22) µm	(99.50 ± 0.01) %
30146	51525	Mirror	"August"	(68.70 ± 0.31) µm	(99.60 ± 0.02) %
30121	51525	Mirror	"February"	(68.30 ± 0.31) µm	(99.60 ± 0.02) %
30340	152545	SiPM	"August"	(66.00 ± 0.18) µm	(98.10 ± 0.03) %
30332	152545	SiPM	"February"	(65.90 ± 0.20) µm	(97.10 ± 0.03) %
30311	152545	Center	"August"	(66.10 ± 0.23) µm	(98.60 ± 0.03) %
30300	152545	Center	"February"	(66.10 ± 0.15) µm	(97.40 ± 0.02) %
30133	152545	Mirror	"August"	(65.60 ± 0.32) µm	(98.50 ± 0.04) %
30249	152545	Mirror	"February"	(65.50 ± 0.18) µm	(97.80 ± 0.03) %
30352	355595	SiPM	"August"	(63.20 ± 0.34) µm	(66.40 ± 0.14) %
30396	355595	SiPM	"February"	(62.30 ± 0.24) µm	(61.40 ± 0.10) %
30324	355595	Center	"August"	(61.40 ± 0.31) µm	(73.60 ± 0.13) %
30320	355595	Center	"February"	(61.40 ± 0.26) µm	(72.00 ± 0.12) %
30158	355595	Mirror	"February"	(61.00 ± 0.82) µm	(75.50 ± 0.36) %

Table S1: Overview of resolution and efficiency for each settings combination.

The clustering algorithm weighting coefficients affect drastically the resolution of the detector and should be adjusted according to the settings of the electronics, the PACIFIC threshold settings being the dominant factor.

The choice of PACIFIC thresholds affects directly both the resolution and the efficiency of the detector. The threshold settings are a trade-off: resolution gets better as one increases the thresholds while efficiency gets worse.

The choice of the PACIFIC shaper settings does not significantly affect resolution. The effect is mostly negligible (or limited to 2%.)

Concerning the hit position along fibers, the resolution is not significantly affected by it. While the efficiency is lower closer to the SiPMs, the extent of this can be neglected when tuning the detector parameters.

The resolution of 66 µm obtained with the final front-end and back-end electronics is comfortably fulfilling the detector specifications (removing the hardware limitations on the fractional bit precision, it was possible to push the SciFi resolution down to around 56 µm.)

Given that many of the detector configurations tested can achieve the detector requirements of an efficiency greater than 99% and a spatial resolution better than 70 µm, the SciFi detector is fully validated to operate at LHCb and deliver the expected performance. The SciFi detector has meanwhile been installed and is presently being commissioned. The configurations studied in this dissertation and the results obtained (see Table S1) provide a guide for the SciFi detector tuning during commissioning and beyond.

Samenvatting

Hoe kan het dat materie- en antimateriedeeltjes in gelijke getale wordt gemaakt en vernietigd, maar alles wat we zien om ons heen bestaat uit materie? Achterhalen wat deze balans heeft verstoord is een van de hoofddoelen van het Large Hadron Collider Beauty (LHCb) Experiment [4]. LHCb bevindt zich aan de Large Hadron Collider (LHC) [1], de belangrijkste deeltjesversneller bij CERN, 100 meter onder de grond en op de grenslijn tussen Frankrijk en Zwitserland. De LHC is een 27 km lange deeltjesversneller, en is ontworpen om botsingen teweeg te brengen tussen protonen, of tussen zware ionen. Dit gebeurt met een energie tot wel 14 TeV en met een luminositeit tot $10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$.

Naast een belangrijke vernieuwingsronde van de LHC tijdens de LS2 periode (tussen 2018 en 2021), onderging ook het LHCb experiment een vernieuwingsronde om het geschikt te maken voor werking bij hogere luminositeit. Deze toename in luminositeit, in de vorm van een toename van het aantal botsingen per bundelkruising, zal resulteren in een aanzienlijke stijging van het stralingsniveau, het bezettingsniveau van de detector en de hoeveelheid gegenereerde data. Om efficiënt gebruik te maken van de grote toename deze nieuw data, is het "Level 0" trigger systeem van het experiment aan de kant gezet; in plaats hiervan, worden de gegevens uitgelezen met een frequentie van 40 MHz. De stralingsniveaus vereisten de validatie van alle systeemcomponenten en het gebruik van geschikte elektronica. De geanticipeerde hoeveelheid data leidde tot het vervangen van alle elektronica voor de gegevensverwerving van het experiment en de front-end elektronica (FEE) van de subdetectoren. Omdat zulke FEE typisch geintegreerd is in de modules van de subdetectoren, was het noodzakelijk ook de Vertex Locator, Inner Tracker, Outer Tracker, en Trigger Tracker te vervangen. Bovenop de veranderingen in de FEE, vereiste het hogere bezettingsniveau in de Outer Tracker een volledige herziening van de gebruikte detectortechniek. Deze inzichten leidden tot de ontwikkeling van de "Scintillating Fiber Detector" (SciFi), die zowel de Outer Tracker en Inner Tracker verving.

De SciFi is nu een van de twee meest belangrijke subdetectoren voor de spoorreconstructie in de LHCb-spectrometer. De verwachting van de subdetector is dat deze zorgt voor het reconstrueren van signalen van geladen deeltjes met een efficiëntie van meer dan 99%, alsmede een plaatselijke resolutie beter dan 70 µm. De rol van de SciFi is om deze cruciale signalen te verschaffen om een accurate reconstructie van de sporen van deze geladen deeltjes door de detector mogelijk te maken, met in het bijzonder een precieze inschatting van hun impuls en een goede inschatting van het pad van deze deeltjes door de andere detectoren.

Om zeker te zijn dat de front-end elektronica van de SciFi klaar is voor het gebruik in het experiment werden tests uitgevoerd bij de **C**ERN **H**igh-energy **A**ccele**R**ator (CHARM) faciliteit [22]. Dit gebeurde in twee sessies, beide bestaande uit één week voorbereiding

en één week bestraling. In de eerste sessie is een enkele set front-end elektronica van de SciFi, de zogenaamde Readout Box (ROB), uitvoerig getest. In de tweede campagne werden twee ROBs samen getest. Het doel van de tests was in de eerste plaats om de effecten van straling op de gebruikte FPGAs te bestuderen. In de tweede plaats gaf dit de kans om de SciFi front-end als geheel te valideren voor de verwachte stralingsdosis, samen met een veiligheidsmarge.

De effecten met betrekking tot de FPGAs die zijn bestudeerd, zijn onder andere:

- **Snelheidsdegradatie:** de snelheid van elektrische signalen binnen de FPGA verslechtert naarmate de stralingsdosis toeneemt;
- **Lekstroom:** het energieverbruik varieert als gevolg van een toename van de lekstroom die kan worden veroorzaakt door de toename van de stralingsdosis;
- **Hetprogrammeren van FPGAs:** de FPGAs verliezen hun vermogen om te worden hergeprogrammeerd na het absorberen van een bepaalde dosis straling;
- **Single Event Upsets (SEU):** een passerend deeltje deponeert lading aan de basis van een transistor, wat geheugencellen kan opladen of ontladen als gevolg, wat zogenaamde bit-flips veroorzaakt;
- **Single Event Latch-ups (SEL):** een eerdergenoemde SEU kan een kortsluiting in de chip veroorzaken, wat kan leiden tot oververhitting en schade aan het apparaat.

Beide sessies, die in totaal een maand duurden, waren succesvol, en leverden voldoende gegevens op voor de studies. De resultaten hebben aangetoond dat snelheidsdegradatie, lekstroom en SELs tijdens de levensduur van de detector een beperkte impact zullen hebben. Echter, na het ondergaan van een bepaalde stralingsdosis, zullen de FPGAs niet meer programmeerbaar zijn, en dit moet in overweging worden genomen bij de planning van de detectoroperaties. De snelheid van SEU's kan worden verminderd door gebruik te maken van "triple modular redundancy" en teruggebracht tot een niveau dat de werking van de detector niet beïnvloedt. Wat betreft de bovengenoemde zaken, is de conclusie dat de front-end elektronica van de SciFi volledig geschikt is voor gebruik in het LHCb-experiment tijdens run 3, met een veiligheidsmarge voor de stralingsdosis van een factor twee.

SciFi-modules die in grote getale zijn geproduceerd, samen met de bijhorende uitleese-lektronica, ondergingen hun eerste tests waarin ze werden blootgesteld aan een deeltjes-bundel. Er wordt vaak naar dir soort testen gerefereerd met de naam "testbundel". Deze tests waren nuttig voor de optimalisatie van de parameters van de elektronica. De test-bundel was een "slice test" van twee volledige SciFi detector modules gekoppeld aan de SciFi front-end elektronica. De gegevensverzameling werd uitgevoerd met de zogenaamde "MiniDAQ2", een voorlopige zelfstandige versie van het LHCb PCIe40 DAQ-systeem. De test werd uitgevoerd op het gebied van CERN in Prevessin, met behulp van de H8A deeltjesbundel van de SPS-versneller. Het belangrijkste doel van de testbundel was om te begrijpen hoe verschillende parameters van het uitleessysteem de efficiëntie en resolutie van de detector kunnen beïnvloeden.

De bestudeerde parameters waren:

- Coëfficiënten voor clusterisatiegewicht;
- PACIFIC-drempels;
- PACIFIC-vorminstellingen;
- Raakpositie langs vezels;
- Maximale clusterbreedte;
- Fractie van bitprecisie.

Er is een analyseframework ontwikkeld om de gegevens van de testbundel te analyseren, met als doel alle benodigde tools te bieden om de analyse van individuele runs uit te voeren en de gewenste resultaten te genereren met behulp van alle relevante runs. Het framework is verdeeld in 3 hoofdonderdelen: de Event Matching software die een zoektabel genereert met overeenkomende events van de SciFi readout en de telescoop readout, de Run Data Analysis software die de gegevens leest en resultaten berekent voor een specifieke run, en de Results Plotter die een verzameling Python bibliotheken en Jupyter Notebooks is die als invoer alle run-specifieke resultaten van belang gebruiken die zijn gegenereerd door de Run Data Analysis en de uiteindelijke resultaten genereren.

Een samenvatting van de prestatie resultaten is te vinden in Tabel S1 voor alle combinaties van testbundelinstellingen.

	Parameters			Resultaten	
Run	**Drempelwaarde**	**Positie**	**Shaper**	**Resolutie**	**Efficiëntie**
30344	51525	SiPM	"August"	(69.70 ± 0.15) µm	(99.30 ± 0.01) %
30336	51525	SiPM	"February"	(68.20 ± 0.19) µm	(99.60 ± 0.01) %
30316	51525	Center	"August"	(69.50 ± 0.17) µm	(99.50 ± 0.01) %
30305	51525	Center	"February"	(68.90 ± 0.22) µm	(99.50 ± 0.01) %
30146	51525	Mirror	"August"	(68.70 ± 0.31) µm	(99.60 ± 0.02) %
30121	51525	Mirror	"February"	(68.30 ± 0.31) µm	(99.60 ± 0.02) %
30340	152545	SiPM	"August"	(66.00 ± 0.18) µm	(98.10 ± 0.03) %
30332	152545	SiPM	"February"	(65.90 ± 0.20) µm	(97.10 ± 0.03) %
30311	152545	Center	"August"	(66.10 ± 0.23) µm	(98.60 ± 0.03) %
30300	152545	Center	"February"	(66.10 ± 0.15) µm	(97.40 ± 0.02) %
30133	152545	Mirror	"August"	(65.60 ± 0.32) µm	(98.50 ± 0.04) %
30249	152545	Mirror	"February"	(65.50 ± 0.18) µm	(97.80 ± 0.03) %
30352	355595	SiPM	"August"	(63.20 ± 0.34) µm	(66.40 ± 0.14) %
30396	355595	SiPM	"February"	(62.30 ± 0.24) µm	(61.40 ± 0.10) %
30324	355595	Center	"August"	(61.40 ± 0.31) µm	(73.60 ± 0.13) %
30320	355595	Center	"February"	(61.40 ± 0.26) µm	(72.00 ± 0.12) %
30158	355595	Mirror	"February"	(61.00 ± 0.82) µm	(75.50 ± 0.36) %

Table S1: Overzicht van de resolutie en efficiëntie voor elke combinatie van instellingen.

De gewichten in de clusteringsalgoritmes hebben een drastisch effect op de resolutie van de detector en moeten worden aangepast aan de instellingen van de elektronica, waarbij de PACIFIC drempelwaarden de dominante factor zijn.

De keuze van PACIFIC drempelwaarden heeft directe invloed op zowel de resolutie als de efficiëntie van de detector. De drempelwaarden zijn een compromis: de resolutie verbetert naarmate de drempelwaarden worden verhoogd, terwijl de efficiëntie verslechtert.

De keuze van PACIFIC shaper instellingen heeft geen significante invloed op de resolutie. Het effect is grotendeels verwaarloosbaar (of beperkt tot 2%).

De resolutie wordt niet significant beïnvloed door de positie van de hits langs de fibers. Hoewel de efficiëntie dichter bij de SiPM's lager is, kan het effect hiervan verwaarloosbaar worden gemaakt door het afstemmen van de detectorparameters.

De resolutie van 66 µm, verkregen met de uiteindelijke front-end en back-end elektronica, voldoet ruimschoots aan de specificaties van de detector. Daarnaast was het mogelijk om de SciFi resolutie zelfs te verlagen tot ongeveer 56 µm door het verwijderen van de beperkingen voor de fractionele bit precisie.

Aangezien veel van de geteste detectorconfiguraties kunnen voldoen aan de eisen van een efficiëntie van meer dan 99% en een resolutie in de plaatsbepaling van deeltjes van beneden de 70 µm, is de SciFi-detector volledig geschikt bevonden om gebruikt te worden bij LHCb en de verwachte prestaties te leveren. De SciFi detector is inmiddels geïnstalleerd en wordt momenteel in bedrijf gesteld en gevalideerd. De in dit proefschrift bestudeerde configuraties en de behaalde resultaten (zie Tabel S1) bieden een leidraad voor het afstemmen van de SciFi detector tijdens de inbedrijfstelling en daarna.