UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE
CORSO DI LAUREA MAGISTRALE IN FISICA

Development of a monitoring system for the muon trigger of the ATLAS detector

Candidate: Lorenzo Sanfilippo
Enrolment no.: 954892

Internal supervisor: Dr. Lidia Dell'Asta
External supervisor: Dr. Antoine Marzin (CERN)

Academic year 2021–2022

For my mother and the memory of my father.

Abstract

The Muon-to-Central Trigger Processor Interface (MUCTPI) is a fundamental component of the Trigger and Data Acquisition (TDAQ) system of the ATLAS (A Toroidal LHC Apparatus) detector. ATLAS is a general-purpose particle physics experiment at the Large Hadron Collider (LHC) at CERN. The LHC collides protons at a rate of 40 MHz, but only a small fraction of the events will contain interesting characteristics for physics and should be recorded. For this reason the detector is equipped with the TDAQ system, which decides, in real time, whether to read out or discard the measurements of each observed interaction for offline analyses. The decision is based on the presence of defined signals in the detector, e.g. a muon exceeding a specific transverse momentum threshold. The ATLAS Trigger is based on a two-stage selection: The first is hardware-based with custom electronics, and the second is software-based with algorithms running on a large farm of off-the-shelf computers. The first stage is the Level-1 (L1) trigger and the second is the High-Level-Trigger (HLT). The L1 trigger processes information coming from the calorimeters, muon trigger chambers and forward detectors. The muon trigger detecting systems are based on Resistive Plate Chambers (RPC) in the central region, and Thin-Gap Chambers (TGC), small-strip Thin-Gap Chambers (sTGC), and Micromegas (MM) in the forward region of the detector. With the information coming from the muon system, the MUCTPI calculates muon candidate multiplicities for programmable transverse momentum thresholds. The MUCTPI then sends the muon multiplicity information to the Central Trigger Processor (CTP) and trigger objects to the L1 Topological processor (L1Topo). For every bunch-crossing, the CTP combines the information coming from the MUCTPI, the calorimeters, and the forward detectors to elaborate the L1 trigger decision. The output rate of the CTP is limited to 100 kHz.

During the Long Shutdown that preceded the LHC Run-3 begun in July 2022, the ATLAS detector was subject to a number of upgrades in order to improve its capabilities. These upgrades are known as of Phase-I upgrades. With higher rates and pile-up levels (additional pp interactions happening in the same bunch-crossing) with respect to its design values, the ATLAS TDAQ system needed to adapt to the detector's changes, with a number of planned upgrades. Among the TDAQ upgrades the MUCTPI is receiving a completely new design in terms of electronics and software. The 18 9U VME modules which composed the MUCTPI of the previous Run-1 and Run-2 have been replaced by a single ATCA blade featuring three state-of-the-art FPGAs and a System-on-chip (SoC). Two FPGAs are responsible for receiving the trigger information from the muon trigger sectors, doing overlap removal for duplicate muon candidates, calculating the transverse momentum threshold multiplicities, and sending the trigger objects to L1Topo. The third FPGA calculates trigger multiplicities and sends them to the CTP and trigger data to the DAQ system. The SoC is used for control, configuration, and monitoring of the hardware and operations of the MUCTPI. The high-level services provided by the SoC are based on the usage of a collection of common libraries non-specific to the detector, known as TDAQ online software. On the MUCTPI, a cross-compiled version of the TDAQ software allows the running of run applications in the ATLAS run control framework. The TDAQ software is used by the applications to realize the configuration, control and monitoring tasks on the MUCTPI.

The main aim of my thesis was to develop MUCTPI run control monitoring applications, which compose the MuctpiMonitoring package of the ATLAS TDAQ software. There are several objects being monitored: the rate of muon candidates coming from muon trigger sectors, the rate of multiplicities sent to the CTP, and the busy sources in the MUCTPI. The rate applications are also implemented for per-bunch rates. I also developed a separate application running on a host PC in the same ATLAS counting cavern where the MUCTPI is installed. This application publishes histograms based on per-bunch data. Developing these applications required the implementation of low-level software which provides an application programming interface (API) to the relevant FPGA firmware registers. Additionally I designed and implemented schemas and infrastructure software for the integration of the monitoring applications in the ATLAS run control system. Finally I developed a number of web monitoring pages in order to make the monitoring operations in the ATLAS Control Room possible for human users. Their features are presented in this work.

The software I wrote during the thesis work has been exercised during laboratory tests and the commissioning phase. Currently it is executed in the ATLAS Control Room for the LHC Run-3.

Acknowledgements

There are many persons I would like to thank, for their continuous help and support that I am receiving since the very first time I started working in the L1CT group. All the persons in the group are very nice, I passed a wonderful time at CERN.

First and foremost I want to express my gratitude to Dr. Lidia Dell'Asta and Dr. Antoine Marzin, for their knowledge, patience and attention towards me. Any time (multiple times per day) I had a problem or a question I was always promptly helped with the information needed to understand or to continue the work. I really appreciate their guidance.

I want to thank Dr. Thilo Pauly, he has always been very thoughtful and open for any of my needs, his help is invaluable. I enjoyed all the chat we had and I took his advices at heart.

A HUGE thanks goes to Dr. Aimilianos Koulouris and Dr. Yoav Afik. It did not pass any day at CERN without going in their office for questions, help or just a random chat. I will always remeber the motto *be strong*! I say thank you also to Dr. Ondrej Penc, he was always available for help and chat. Thanks to Dr. Patrick Czodrowski for being a very good office neighbor and . . . for the cakes!

Among the other students I met at CERN I would like to mention Daniele and Garvita, I really enjoyed hanging out with them and I am sure there will be time to meet in person soon. I thank also all my friends in Italy, because the fact that I left my town did not change our bonds. I am looking forward the next time we can make something together.

Thanks also to Morgan for her lovely support in any circumstance and because she's the best to me. Thanks to Cat Middleton for being always such a good assistant. In the end thanks to my mother, because she still tolerates me.

Contents

List of Figures

List of Tables

Introduction

At the Large Hadron Collider (LHC) beams of protons or ions are pushed near the speed of light to collide at unprecedented energies and luminosities to extend the frontiers of the research in physics. The LHC Run-3 operations, started in July 2022 and continuing until the end of 2025, for many aspects will be very different from those of Run-1 and Run-2 since the LHC performances during the last Long Shutdown 2 (LS2) have been upgraded, offering a considerable opportunity for new physics. The proton-proton collisions will be produced near LHC ultimate design at a centre-of-mass energy of $\sqrt{s} = 13.6\,\text{TeV}$, with an expected instantaneous luminosity of $2 \times 10^{34}\,\text{cm}^{-2}\text{s}^{-1}$.

In order to exploit and satisfy the requirements of LHC improvements the ATLAS (A Toroidal LHC ApparatuS) detector and its sub-systems have been upgraded. These upgrades are called Phase-I upgrades. The Trigger and Data Acquisition (TDAQ) needed to cope with event rates and pile-up (additional pp interactions happening in the same bunch-crossing) levels much greater than the values used for its original design, therefore it has gone through a number of upgrades [1].

The ATLAS Trigger system carries out event selection in two stages:

1. **Level-1 Trigger** – Hardware-based trigger made of custom electronics working with coarse-granularity information coming from the calorimeters, muon spectrometer and forward detectors.
2. **High-Level Trigger** – Software-based trigger operating on a large farm of computer processors.

This thesis work was completed during a 14-month technical studentship at CERN in the ATLAS Level-1 Central Trigger (L1CT) group. In particular the context of the work are the Phase-I upgrades of the new Muon-to-Central Trigger Processor Interface (MUCTPI). The MUCTPI is an electronic board responsible for calculating muon candidate multiplicities for programmable transverse momentum thresholds. After calculation, the MUCTPI then sends the muon multiplicity information to the Central Trigger Processor (CTP) and trigger objects to the L1 Topological processor (L1Topo). For every bunch-crossing, the CTP combines the information coming from the MUCTPI, the calorimeters, and the forward detectors to elaborate the L1 trigger decision with a maximum output rate of 100 kHz.

Depending on the physics event of interest, the final products might contain muons, which leave a very clean trace in the detector. This can be the signature that triggers the data acquisition system to record the event, if the muons have a specific multiplicity and pass a transverse momentum threshold. The input rate of the trigger is an important property to keep monitored, because the processes of interest for offline analyses occur at low rates, in the range of 0.1 to 10 Hz [2]. The rate information is also evaluated bunch by bunch which pass in the detector, referred to as “per-bunch”.

The work carried out in the L1CT group and described in this thesis consisted in the development of:

- Monitoring applications for
  - Rate of muons incoming from Sector Logic inputs.

  - Rate of muons sent to the CTP.
  - Per-bunch rate of muons incoming from Sector Logic inputs.
  - Per-bunch rate of muons sent to the CTP.
  - Busy state of the board.

- An auxiliary application which publishes histograms using the information of both the per-bunch applications.
- Web pages integrated in the ATLAS web monitoring system to display the information monitored by the applications.

Thesis outline

This thesis has been organized in the following structure:

1. **Chapter 1** – To begin, a brief introduction to the theoretical framework of the research done at CERN will be provided, followed by a basic introduction to the ATLAS experiment and its detector, with a brief explanation of each of the main sub-detectors. Then the ATLAS Trigger and Data Acquisition system will be introduced with a minimal quantitative motivation for its existence.
2. **Chapter 2** – This chapter will give an overview of the TDAQ operations and TDAQ software, explaining its main high-level services.
3. **Chapter 3** – This chapter is devoted to the MUCTPI. First its functionalities are discussed, then a description of its hardware. The firmware blocks involved in the operations of the monitoring applications are demonstrated and briefly discussed. The chapter concludes with the implementation of MUCTPI software integrated in the ATLAS run control framework, the commissioning of the MUCTPI, and the firmware/software validation techniques used.
4. **Chapter 4** – The purpose of the last chapter is to illustrate in more detail the monitoring applications and their working principles. The chapter closes with a short explanation of the web monitoring pages developed in the context of this thesis work.
5. **Chapter 5** – The last chapter contains the conclusions and a brief outlook for possible re-use of the code for the next LHC Run-4.

CHAPTER 1

The ATLAS experiment at the Large Hadron Collider

The purpose of this chapter is to give an overview of the ATLAS experiment. This is done by first introducing the theoretical framework of particle physics, the Standard Model. After that will be explained what is CERN, the Large Hadron Collider (LHC) and its main experiments. At this point will follow a discussion on the main sub-system composing the ATLAS detector. In the end a short example will motivate the presence of the Trigger and Data acquisition (TDAQ) system and the chapter will close with a brief explanation of the TDAQ system.

1.1 The Standard Model of particle physics

The Standard Model of particle physics is a theory which describes three out of the four known fundamental forces. It provides our current best understanding of electromagnetism, weak and strong interactions, while gravity at the moment is not contemplated. It has been shown by Glashow, Salam and Weinberg that above a certain scale, called Fermi scale (246 GeV), the electromagnetic and weak interaction merge into the electroweak interaction[1]. All the observed phenomena in nature can be reconducted to these basic interactions. For example the strong interaction holds together the nuclei of atoms, and the electroweak processes are responsible for the pp cycle which generates energy in stars.

The current formulation of the Standard Model can be dated back to the 1970s, upon experimental confirmation of the existence of quarks at the first SLAC experiment, where electron scattering showed that protons are not pointlike particles.

The formalism of the Standard Model is the Quantum Field Theory (QFT), which combines classical field theory, quantum mechanics and special relativity. In QFT particles are excitations of the fields whose mathematical nature is determined by the spin of the particles. The dynamics of a generic field ϕ_k in QFT is calculated in analogy to the Lagrangian formalism of classical mechanics, where the Langrangian is replaced by the Lagrangian density $\mathcal{L}$, which is a functional of the fields themselves and their derivatives. The application of the least action principle to an action built with the lagrangian density gives the Euler-Lagrange equations,

[1] For their contributions to the theory of the unified weak and electromagnetic interaction they were awarded the 1979 Nobel Prize in Physics.

$$\frac{\partial \mathcal{L}}{\partial \phi_k} - \partial_\mu \frac{\partial \mathcal{L}}{\partial(\partial_\mu \phi_k)} = 0 \quad . \tag{1.1}$$

The solution of the Euler-Lagrange system of equations describes the evolution and the dynamics of the field ϕ_k.

The Standard Model also provides a classification scheme for all the known particles, which are divided in two families of fundamental pointlike particles, fermions and bosons. Every particle also has their corresponding antiparticle, characterized by the same quantum numbers, but with opposite sign. Fig. 1.1 shows the elementary particles of the Standard Model (antiparticles are not included in the list).

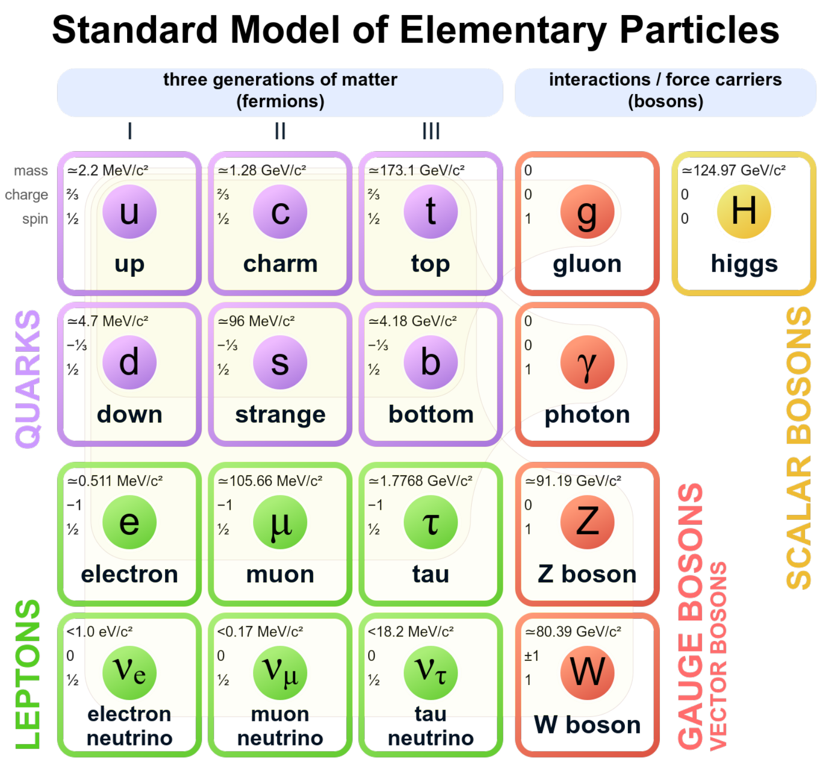

Figure 1.1: Standard model of elementary particles: the 12 fundamental fermions and 5 fundamental bosons. The masses of certain particles are subject to periodic reevaluation by the scientific community. Source: [3].

The bosons are integer spin particles, which obey the Bose-Einstein statistics and are the carriers of the interactions or give rise to particle properties, such as their mass. According to the Standard Model there are five different bosons:

- Four vector bosons with spin 1, acting as mediators of the fundamental forces.
  - Photon γ, carrier of the electromagnetic force.
  - Charged weak boson W, carrier of the charged weak interaction. It exist with two different electric charges.

  - Neutral weak boson Z, carrier of the neutral weak interaction.
  - Gluon g, carrier of the strong force. It exists with eight different color charges.
- One scalar boson with spin 0, the Higgs boson H, which is responsible for the Higgs mechanism which gives rise to the masses of elementary particles.

Fermions are half-integer spin particles which constitute matter and obey the Fermi-Dirac statistics. Fermions can be identified with six quarks and six leptons, which in turn are further divided in three generations of two particles. Each lepton generation contains a charged particle and its associated neutrino and differ only by their mass, while the quark generations are made by an up-like and a down-like quark, differing from each other by their flavor quantum numbers and their mass.

The fields describing particles are:

- Scalar fields: they describe spin-0 particles and can be real or complex. Scalar fields follow the Klein-Gordon lagrangian

$$\mathcal{L}_{\text{Klein-Gordon}} = (\partial^\mu \phi^\dagger)(\partial_\mu \phi) - m^2 \phi^\dagger \phi \quad . \tag{1.2}$$

- Vectorial fields: they describe spin-1 massive or massless particles. The lagrangian is the Maxwell lagrangian for massless particles, and the Proca lagrangian in case of massive fields. Before introducing the lagrangian it is necessary to define the field strength $F^{\mu\nu}$ of a field A^μ

$$F^{\mu\nu} = \partial^\mu A^\nu - \partial^\nu A^\mu \quad . \tag{1.3}$$

  In this way the Maxwell lagrangian is written as

$$\mathcal{L}_{\text{Maxwell}} = -\frac{1}{4} F^{\mu\nu} F_{\mu\nu} \quad , \tag{1.4}$$

  and the Proca lagrangian

$$\mathcal{L}_{\text{Proca}} = -\frac{1}{4} F^{\mu\nu} F_{\mu\nu} + \frac{1}{2} m^2 A^\mu A_\mu \quad . \tag{1.5}$$

- Fermionic fields: they describe spin-$\frac{1}{2}$ particles by the means of a four-component spinor. These fields satisfy the Dirac lagrangian

$$\mathcal{L}_{\text{Dirac}} = \bar{\psi}(i\gamma^\mu \partial_\mu - m)\psi \quad . \tag{1.6}$$

  The γ^μ ($\mu = 1, 2, 3, 4$) are the Dirac matrices, which satisfy $\{\gamma^\mu, \gamma^\nu\} = 2g^{\mu\nu}$, where $g^{\mu\nu}$ is the covariant metric. $\bar{\psi}$ is the conjugate spinor, defined as $\bar{\psi} = \psi^\dagger \gamma_0$.

The lagrangians shown from Eq. (1.2) to Eq. (1.6) are quadratic in the fields variables and therefore a substitution in Eq. (1.1) would yield linear equations of motion. In order to obtain an interaction between fields, it is required that the lagrangian contain higher monomial terms in the fields. In this way the new lagrangian would be given by a free field term $\mathcal{L}_0$ and an interaction term $\mathcal{L}_{\text{int}}$.

$$\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_{\text{int}} \quad . \tag{1.7}$$

Normally the resulting equations are not solvable in an analytical form, but their solution is obtained with perturbation theory and Feynman rules permit the calculation of matrix elements at any fixed order.

Lagrangian interaction terms between fields emerge naturally when imposing the symmetry on the fields in presence of a group of transformations. For example one can see this considering the Dirac free lagrangian of Eq. (1.6) and the transformations $e^{ie\alpha(x)}$ of the group U(1). The term $\bar{\psi}\gamma^\mu\partial_\mu\psi$ breaks the invariance, which can be fixed by

- Introducing a gauge vector field $A_\mu(x)$ which transforms as $A_\mu \to A'_\mu = A_\mu - \partial_\mu\alpha$.
- ∂_μ is replaced with the covariant derivative $D_\mu = \partial_\mu + ieA_\mu$.

The new lagrangian becomes

$$\begin{aligned} \mathcal{L} &= \bar{\psi}(i\gamma^\mu D_\mu - m)\psi \\ &= \bar{\psi}(i\gamma^\mu\partial_\mu - m)\psi - e(\bar{\psi}\gamma^\mu\psi)A_\mu \\ &= \mathcal{L}_{\text{Dirac}} + \mathcal{L}_{\text{int}} \end{aligned} \tag{1.8}$$

The Eq. (1.8) is not the correct lagrangian because it does not provide the correct equations if using the Eq. (1.1) for A_μ. The term missing is the free field for A_μ, which is given by the Maxwell lagrangian of Eq. (1.4), therefore in the end the QED lagrangian is

$$\mathcal{L}_{\text{QED}} = \bar{\psi}(i\gamma^\mu\partial_\mu - m)\psi - e(\bar{\psi}\gamma^\mu\psi)A_\mu - \frac{1}{4}F^{\mu\nu}F_{\mu\nu} \tag{1.9}$$

The lagrangian of the Standard Model is not just limited to the QED and therefore based on the group U(1), but the Standard Model is a non-abelian gauge theory based on the local symmetry of $\text{SU}(3) \times \text{SU}(2) \times \text{U}(1)$, where SU(3) is the symmetry group of the quantum chromodynamics and $\text{SU}(2) \times \text{U}(1)$ is the local symmetry group of the electroweak theory.

I has been demonstrated that the W and Z bosons are massive particles, but the gauge invariance requires that gauge bosons are massless particles. The puzzle was solved by the Brout-Englert-Higgs mechanism and the introduction of a scalar field and the characteristic potential of "mexican hat". The choice of one of the equivalent minimum values of the potential breaks the symmetry of the vacuum field, but preserves the symmetry of the lagrangian. This mechanism gives rise to the masses of the gauge vector bosons.

Concerning the masses of fermions, it is introduced in the same way a scalar field with the same potential. Since the mass term of the Dirac lagrangian $m\bar{\psi}\psi$ mixes the left and right components of the spinor, it is required to introduce a Yukawa term of the lagrangian. The couplings will represent the mass terms.

The Standard Model is a predictive theory; it has been extensively tested with precise measurements during the past 40 years and shown to be consistent. However there are still open questions that the Standard Model at the current state of the art is not able to give an answer. Here follows a brief and incomplete list of open questions in physics

- Do the forces really become unified at sufficiently high energy?

- Is there a particle associated with gravity?
- What is the nature of Dark Matter?
- Why there is an imbalance between baryonic and antibaryonic matter?
- How do neutrinos get their mass?

A possible explanation for the above-mentioned topics can be found within the realm of the physical theories called "Beyond the Standard Model" (BSM) [4]. One of the possible Standard Model extensions relies on the principle of supersymmetry (SUSY), which would imply every particle of the current model having a superpartner particle. Since these supersymmetric particles have not been observed yet they should be heavier than the sensitivity of previous experiments.

1.2 The Large Hadron Collider

The Conseil Européen pour la Recherche Nucléaire (CERN) is a particle phyics research centre in Geneva, Switzerland founded in 1954. One of the aims of the CERN laboratory is to provide the accelerators and infrastructure needed by international scientific collaborations for high-energy physics (HEP) research. The main accelerating machine currently used is the Large Hadron Collider [5]. The LHC is a two-ring superconducting proton-proton, ion-ion or proton-ion collider built between 1998 and 2008 in an underground tunnel about 100 m beneath the France-Switzerland border between the Jura mountains and lake Geneva. The tunnel has a circumference of roughly 27 km and it was formerly used to house the Large Electron-Positron collider (LEP) [6], which was decommissioned in 2000. A schematic layout diagram of the tunnel showing the locations of the four main experiments is displayed in Fig. 1.2.

The CERN Foundation has established that every new installed machine shall use the prior ones as injectors, boosting particles' energy gradually. Therefore the LHC is supplied with protons by a chain of other four accelerators: Linac4, Proton Synchrotron Booster (PSB), Proton Synchrotron (PS) and Super Proton Synchrotron (SPS). In 2020 during LS2 Linac4 replaced Linac2 as a source of proton beams for the LHC.

Linac4 accelerates H^- ion plasma to an energy up to 160 MeV. During the injection into the PS Booster the hydrogen is stripped from the electrons, leaving only protons. The PSB increases the energy to 1.4 GeV before injecting into the PS, which in turn prepares proton bunches and injects into the SPS at 26 GeV. The final step of injection into the LHC is done at 450 GeV. At this stage protons are transferred to two continuous vacuum beam pipes, one circulating clockwise and one counterclockwise; each beam is kept in orbit by dipole magnets and focused by quadrupole magnets. The Radio Frequency (RF) cavities are installed in a single facility along the ring. They are tuned to oscillate at 400 MHz and kick the beams up to their maximum energy in 20 min after injection.

A pictorial representation of the aforementioned chain is shown in Fig. 1.3 and other LHC design parameters relevant for further discussions can be found in Table 1.1.

1.3 Main experiments at CERN

Along the circumference of the LHC there are four crossing points where particles collide. In each of these is located one of the main experiments, each distinct and characterized by its own detector

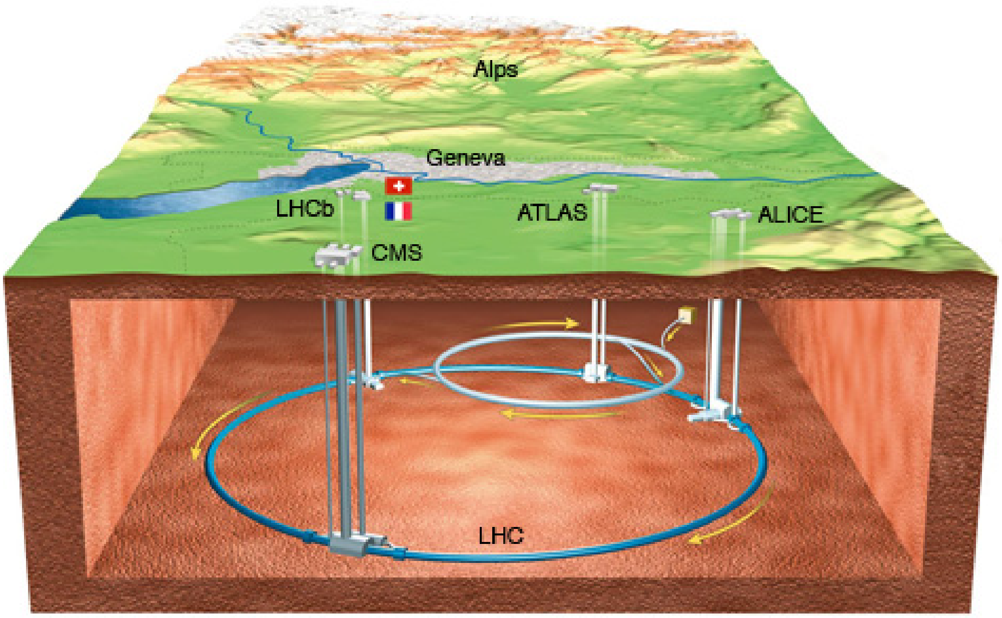

Figure 1.2: Layout diagram of LHC tunnel with its four main experiments.

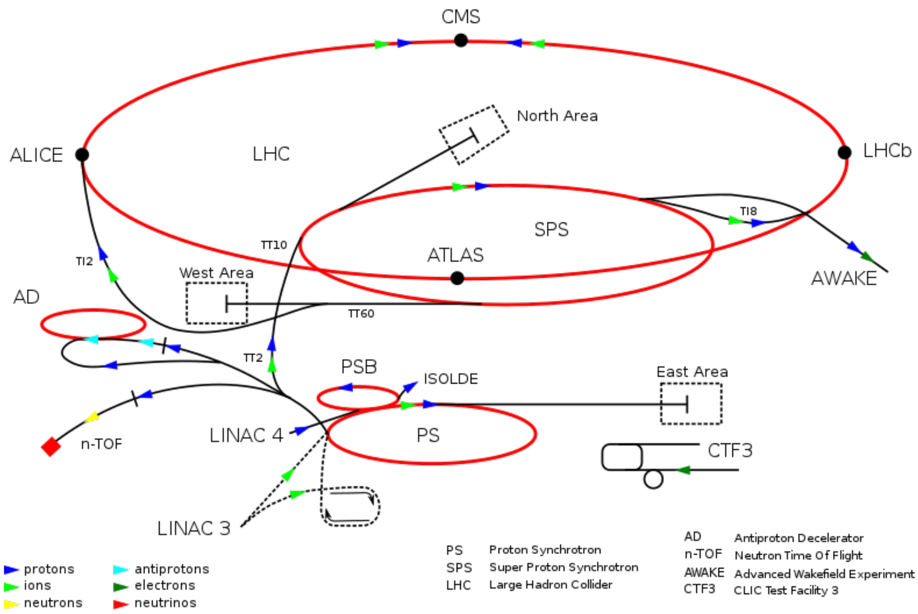

Figure 1.3: CERN's accelerator complex (not to scale). Source: [7].

Parameter	Value
RF frequency	400 MHz
Number of bunches	3 564
Filled bunches	2 808
Protons per-bunch	1.15×10^{11}
Bunch spacing	25 ns
Turns per second	11 245

Table 1.1: LHC design parameters. Source: [8].

system and physics research program. These are:

- **ATLAS**, **A** **T**oroidal **L**HC **A**pparatu**S**, a general purpose experiment designed to cover the widest possible range of physics at LHC. The ATLAS detector is a multi-layer type composed by concentric cylinders of material sensitive to different particles and it is equipped with magnets generating a field used for tracking and determining the charge sign of passing particles [8]. The location of the ATLAS detector along the LHC ring is called Point 1 (P1).
- **CMS**, **C**ompact **M**uon **S**olenoid, the other general purpose experiment, with the same scientific goals of ATLAS. The CMS detector has the same structure of concentrically wrapped layers as the ATLAS detector [9].
- **LHC*b***, **L**arge **H**adron **C**ollider ***b***eauty, an experiment specialized in the study of the slight matter/antimatter asymmetry present in interactions of B-mesons and to measure the parameters of the violation of the charge-parity symmetry (CP). The installed detector is not sourrounding the interaction zone, but it is optimized to work in the forward direction close to the beam axis, since it is expected to have an higher flux of b hadrons [10].
- **ALICE**, **A** **L**arge **I**on **C**ollider **E**xperiment, a detector with a similar layout ATLAS and CMS which is specialized in measuring and analyzing collisions between Pb nuclei[2]. It studies the physics of the quark-gluon plasma, an extremely dense and high temperature state of the matter where quarks and gluons are no longer confined inside hadrons. The study of this state of the matter is important because it is thought that at the very first instants of the universe, before the formation of hadrons, the state of the universe was a plasma of quarks and gluons [11].

1.4 ATLAS detector

The ATLAS detector is a multi-purpose particle detector[3] with a forward–backward symmetric cylindrical geometry designed to measure the broadest possible range of signals. It is a multi-layered instrument with nearly full coverage of the solid angle consisting in different detecting subsystems wrapped concentrically around the collision point to record trajectory, momentum and energy of particles. The cylindrical part of the detector is the barrel, and the flat coverings at each side are the end-caps. The ATLAS detector consists of an inner tracking detector surrounded by a thin superconducting

[2] Also p-p and p-Pb collisions are investigated.

[3] The description of the detector and its sub-systems is largely based on [8] unless stated otherwise.

solenoid, electromagnetic and hadronic calorimeters, and a muon spectrometer incorporating three large superconducting toroidal magnets. Each sub-detector is sensitive to different particles, whose passing through the layers leaves different traces allowing particle identification and precise measurements of momentum and energy . The inner detector measures the direction, momentum, and charge of electrically-charged particles with semiconductor-based sensors and transition radiation modules. The ATLAS calorimetry system exploits a liquid argon sampling calorimeter and a tile hadronic calorimeter with plastic scintillators to measure the energy of electrons/photons and hadrons. The calorimeter system also has triggering functions. The muon spectrometer is the detector's outermost layer. It features equipment with four different technologies (Resistive Plate Chambers, Thin-Gap Chambers, Monitored Drift Tubes and MicroMegas) and a system of toroidal superconducting magnets in the end-cap and barrel region to measure the momentum and track of muons and to trigger.

The ATLAS detector has dimensions of about 25 m in height and 46 m in length, while the overall weight is approximately 7×10^6 kg [8]. Fig. 1.4 shows the layout of the various sub-detector components which will be explained in more detail later in this section.

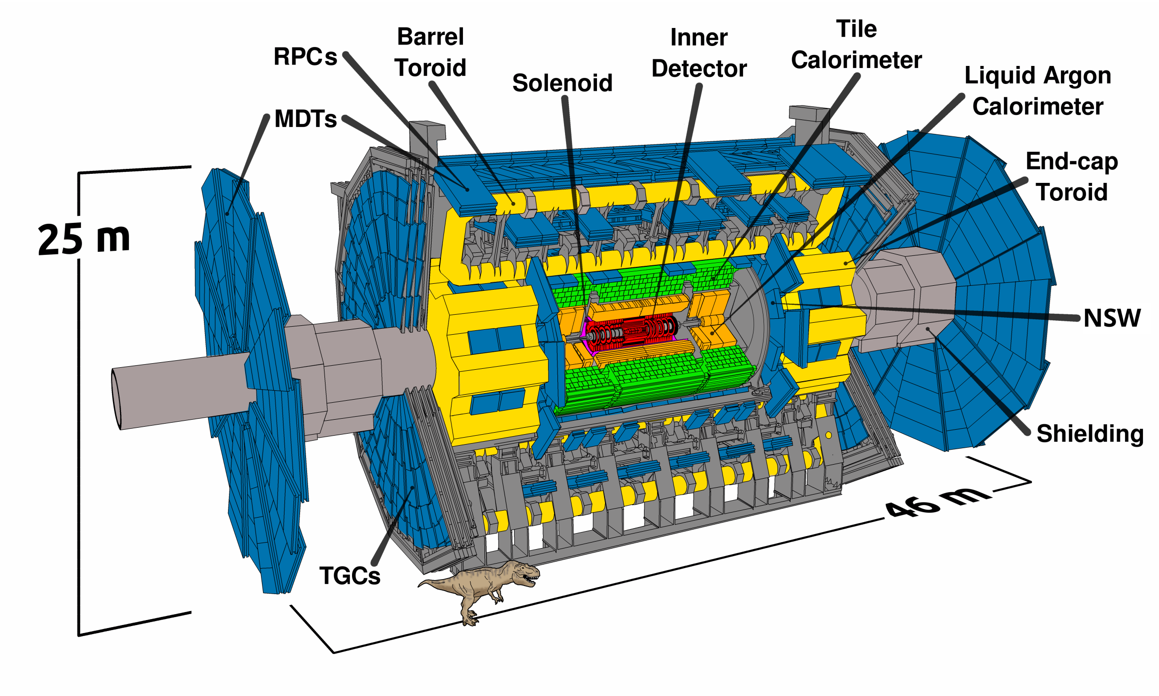

Figure 1.4: Longitudinal cut-away view of the ATLAS detector, T-Rex for scale. Source: [12].

The physics goals illustrated in Section 1.5 result in the following list of general requirements for the ATLAS detector [8]:

- Fast and radiation-hard electronics.
- High granularity in both space and time to cope with very high rate particle fluxes and possible overlapping events.
- High precision muon momentum measurements over a large range of momenta and exact charge determination.

- Excellent electromagnetic calorimetry for measurements and identification of photons and electrons, together with very good hadronic calorimetry for jets and missing transverse energy measurements.
- Good reconstruction of charged particle tracks and vertices at high instantaneous luminosity.
- Large pseudorapidity acceptance and as large as possible azimuthal coverage to minimize number of the particles escaping detection.

In order to keep improving the performance of the ATLAS detector and to achieve physics goals during the next Run-3, a number of upgrades have been planned and implemented. These upgrades are called Phase-I upgrades.

1.4.1 Coordinate system

The coordinate system defined at ATLAS is a right-handed system with origin in the center of the detector at the nominal interaction point, the positive x-axis pointing toward the centre of the LHC ring and the z-axis along the direction of the beams. The positive y-axis points upwards, slightly tilted with respect to the vertical due to a general tilt of the LHC tunnel.

In the transverse plane cylindrical coordinates (r,ϕ) are used, where $\phi \in [-\pi, \pi]$ is the azimuthal angle measured from the x-axis around the beam and r is the radial distance from the interaction point.

The polar angle $\theta \in [0, \pi)$ is the angle measured from the positive z-axis, but it is often expressed in terms of the pseudorapidity η, which is defined as

$$\eta = -\ln\left[\tan\left(\theta/2\right)\right] \quad . \tag{1.10}$$

When referring to regions of the detector close to the beam axis it is common to refer to “forward” and “backward” directions where $\eta \gg 1$ and $\eta \ll -1$ respectively. Furthermore the side of the detector where $z > 0$ is called side A and where $z < 0$ side C.

1.4.2 Magnet system

The magnetic field across the detector necessary for charge and momentum measurements is generated by a system of four superconducting magnets, consisting in a solenoid, a barrel toroid and two end-cap toroids. The cryogenic system provides both liquid and gaseous He at 4.5 K and 60 K for the cooling the superconducting magnets and for thermal shielding. The layout of the magnets is shown in Fig. 1.5.

The magnets’ technology and layout have been chosen to ensure large volumes and high fields. These are required for a good momentum resolution, since it scales with $1/B$ and $1/L^2$, where L is the tracking length [13].

The solenoid magnet is coaxial with the beam. It sorrounds the inner detector and precedes the calorimeter barrel, generating a 2 T axial magnetic field for inner tracking. The position of the solenoid magnet with respect to the calorimeter imposed a design constraint of reducing as much as possible the dead material, therefore it has been constructed in Al-stabilized NbTi with 4.5 cm thickness, which results in about 0.66 radiation lengths [8].

The end-cap toroid magnets can generate a field up to 3.5 T which fills the volume surrounding the calorimeters and the two end-cap regions. The end-cap magnets have a characteristic gear shape

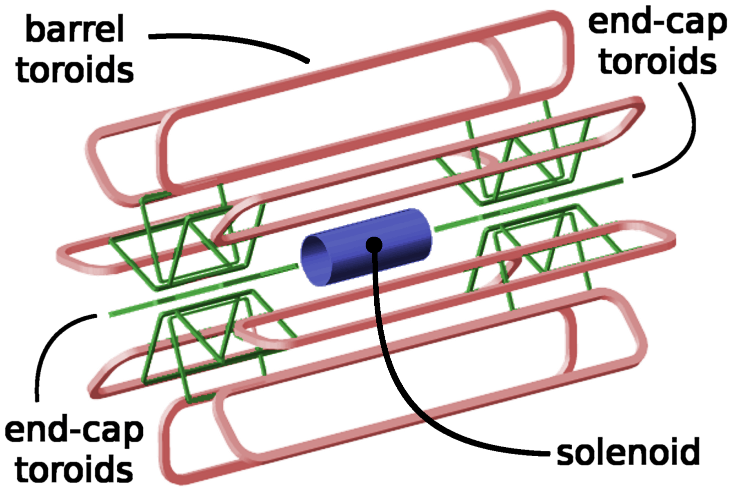

Figure 1.5: The ATLAS superconducting magnets. Source: [12].

to allow them to slot into the barrel toroid with a 22.5° angle between the "gear tooth" and the neighboring barrel, optimizing the overlap between the two systems [13].

The typical bending power provided by the magnetic field is 3 T m in the barrel and 6 T m in the end-caps. Due to the finite number of coils the field lines are not perfectly toroidal, but they show a regularly ripped profile, see Fig. 1.6. The bending power, shown in figure Fig. 1.7 given the non uniform magnetic field depends on the position, but also on the particle's initial direction, its momentum and its charge. The effect of fluctuations on these variables translates into a worse momentum resolution and this is most visible in the transition region between barrel toroid and end-cap toroid, where a considerable radial component is present [14].

1.4.3 Inner detector

The ATLAS inner detector (ID) is the part of the detector closest to the beam line, designed to work in a high-radiation environment. The ID is immersed in a 2 T magnetic field used to bend charged particles trajectories and provide high resolution measurements of momentum and position up to $|\eta| < 2.5$. The ID measurements play a crucial role in primary and secondary vertices reconstruction.

The ID consists of three different detecting layers: Pixel detector (PIX), semiconductor tracker (SCT) and transition radiation tracker (TRT). The assembly is compact, being contained in a cylinder with a diameter $d \approx 2.1$ m and a length $l \approx 6.2$ m. A cross-section view of the inner detector is shown in Fig. 1.8.

The pixel detector consists of four layers of 92 million silicon pixels in total, each of them provided with an electronic readout channel for high granularity. The innermost layer of the PIX is called Insertable B-Layer (IBL) and it was installed during the Long Shutdown 1 after Run-1 to improve performances in presence of large pile-up events. The pixels' size in the IBL is 50 µm × 250 µm in the $r\phi$ and z direction respectively, while in the other three layers it is 50 µm × 400 µm. The spatial resolution in the IBL is 8 µm azimuthally and 40 µm longitudinally, while for the PIX it is 10 µm and 115 µm respectively.

Moving away radially from the beam line, the next detecting equipment is the semiconductor tracker,

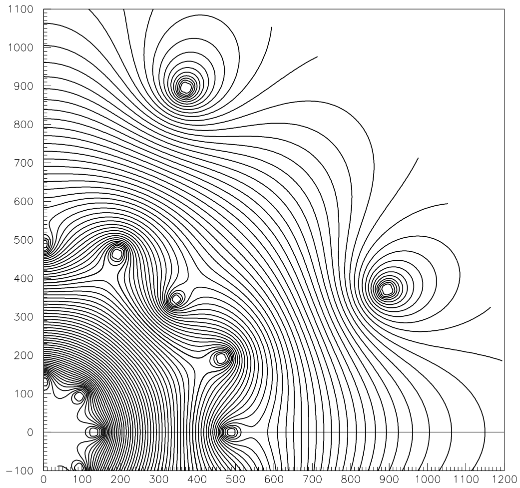

Figure 1.6: Magnetic field map in the transition region. The field lines are shown in a plane perpendicular to the beam axis and located in the middle of an end-cap toroid. The interval separating consecutive lines is 0.1 Tm. Individual barrel and end-cap coils are visible. The scales are in centimetres. Source: [14].

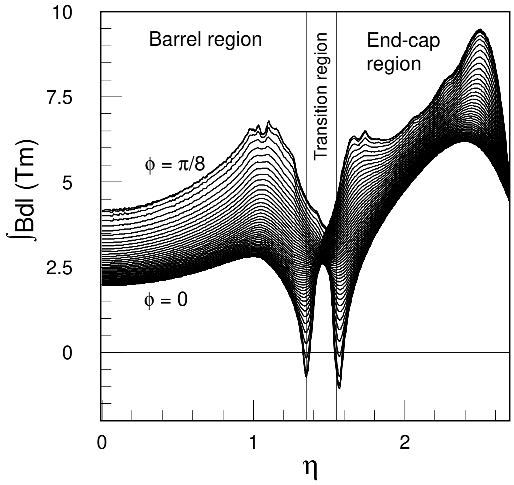

Figure 1.7: Toroid bending power $\int B\,\mathrm{d}l$ of the azimuthal field component, integrated between the first and last muon chamber, as a function of η. The curves correspond to azimuthal angles equally spaced between the barrel toroid and end-cap toroid coil planes. Source: [14].

which surrounds the Pixel detector. It is made of 4000 modules of 6×10^6 silicon sensor micro-strips displaced in 8 layers. The SCT spatial resolution is 17 µm × 570 µm for $(r\phi, z)$.

The inner detector's outermost part is the Transition Radiation Tracker, which is a drift-tubes system working in the proportional regime. It consists in 370 000 4 mm diameter straws kept at −1.5 kV with a central gold-plated tungsten grounded anode wire. Between layers of straws the free space is occupied by polypropylene acting as radiator. The gas mixture filling the tubes is composed of 70% of Xe for X-ray absorption with 27% CO_2 and 3% O_2 to increase electron drift velocity and photon quenching [8].

The TRT system provides on average 30 two-dimensional tracking points with an intrinsic resolution of 120 µm and a transition radiation signature which permits particle identification between electrons and pions [16]. The energy deposition of transition radiation photons due to electrons is about 8 to 10 keV, while the minimum ionizing particles such as pions deposit about 2 keV.

1.4.4 Calorimeter

The calorimeter is the sub-detector designed to measure energies of particles and jets by sampling the ionization signal from the induced particle showers. It can also provide position measurements by segmentation of the volume. The measure of the energy is also useful for the determination of missing energy $\not{E}_T$, carried by particles which escape detection, like neutrinos.

Depending on the particles measured, the calorimetry can be electromagnetic or hadronic. For the electromagnetic calorimetry electrons (positrons) and photons are measured through their electromagnetic interactions, while for the hadronic the measured particles are hadrons and the

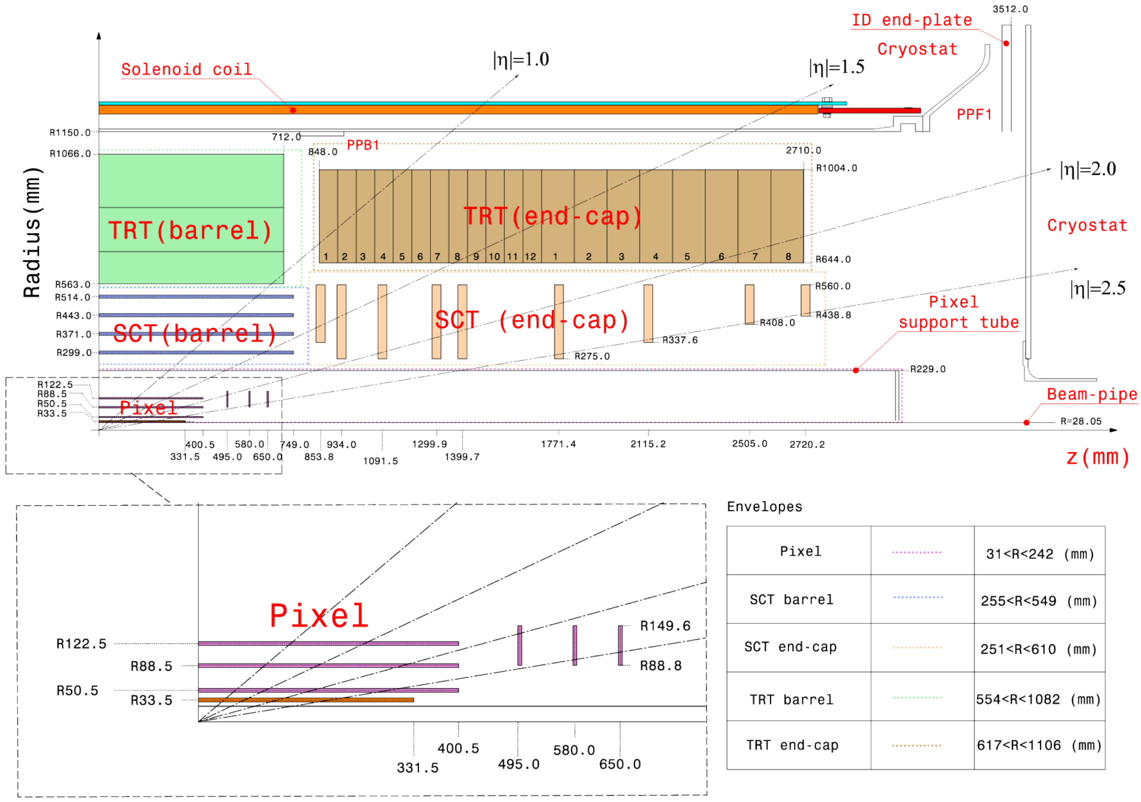

Figure 1.8: ATLAS inner detector in the r-z plane, where r is the radial distance from the z-axis (in the figure is shown only the $z > 0$ part). The detector is symmetrical around the beam axis and the perpendicular plane passing through the collision point (0,0). Source: [15].

dominant interaction is the strong interaction[4].

Calorimeters can be built exploiting either a single type of material or many alternating layers of active and passive material. In the first case they are referred to as homogeneous, while in the latter case they are called sampling calorimeters. Usually the absorber material is a high Z material with the purpose of developing the electromagnetic shower, since higher Z implies shorter radiation lengths X_0. The active material is responsible for generating the signal, generally associated with the ionization energy loss of secondary electrons. A typical choice of active material is a noble gas or a scintillator[5].

At the ATLAS detector there are two concentric calorimeters providing full ϕ-symmetry and coverage around the beam line. These are liquid Argon (LAr) calorimeters surrounded by the Tile calorimeter. The assembly of the calorimeters can be found in Fig. 1.9.

1.4.4.1 LAr calorimeter

The LAr calorimeter is a sampling calorimeter which uses liquid argon as active medium and Pb as absorber material. LAr has a good radiation-hardness and linear response over time, but at the cost

[4] An electromagnetic component is always present.

[5] In the CMS detector the calorimeter is based on crystal scintillators in $PbWO_4$.

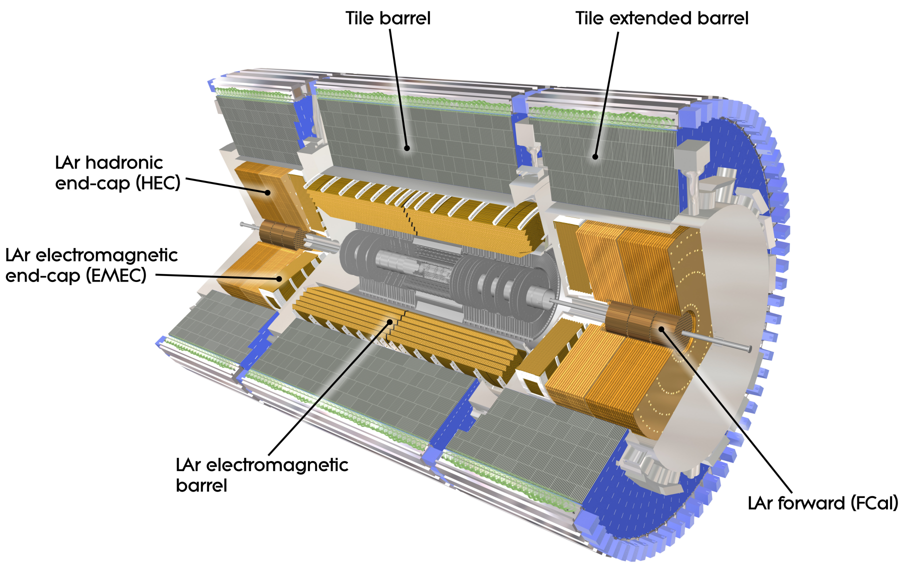

Figure 1.9: Cut-away view of the ATLAS calorimeter system. The LAr electromagnetic barrel is also referred to as EMB. Not labeled in the figure is the presampler calorimeter, used to correct for the energy lost by upstream electrons and photons in the region $0 < |\eta| < 1.8$. The presampler is also used to identify pions thanks to its fine granularity. Source: [17].

of enclosing the entire system in a cryostat working at 87 K, keeping the temperature constant and LAr free from impurities to prevent signal degradation [18]. As indicated in Fig. 1.9 it is made of a number of components:

- LAr electromagnetic barrel (EMB), covering $|\eta| < 1.475$.
- LAr electromagnetic end-caps (EMEC), covering $1.375 < |\eta| < 3.2$.
- LAr hadronic end-caps (HEC) covering the same region of EMEC.
- LAr forward (FCal), covering $3.1 < |\eta| < 4.9$.

In the EMB and EMECs the geometry exploited is accordion-like, where LAr fills the gap between two consecutive absorbers. The LAr gap is divided in two half gaps by the readout electrode, which is made in copper and also arranged in a honeycomb-like structure. The barrel is segmented in three longitudinal compartments – strips, middle and back – while each end-cap is divided in two concentric wheels. In the EMB all layers together correspond to more than 22 X_0, while in EMECs more than 24 X_0. Fig. 1.10 shows a pictorial representation of the accordion geometry in a barrel module.

HECs are located behind EMECs, they are composed of two independent wheels, each of those divided in two segments in depth by flat copper absorber plates. The wheels closest to the interaction point are built from 25 mm plates, while those further away use 50 mm plates. The resulting longitudinal containment is 10 λ.

FCal completes the almost full η hermiticity of the calorimeter, with a benefit of reducing the radiation background levels on the muon spectrometer. Therefore in order to withstand intense

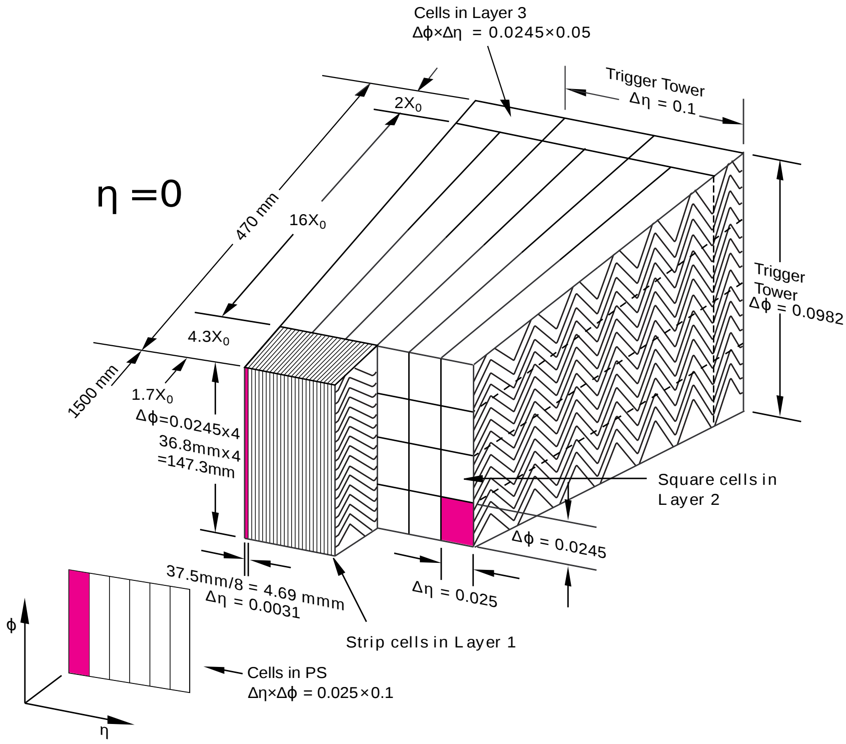

Figure 1.10: Sketch of a barrel module where the different layers are clearly visible with the ganging of electrodes in ϕ. The granularity in η and ϕ of the cells of each of the three layers and of the trigger towers is also shown. Source: [8].

radiation levels and neutron fluxes, the absorber is made in copper and tungsten. Each end-cap is composed of three modules; the closest to the interactions in copper is optimized for electromagnetic measurements, while the other two modules in tungsten measure the energy of hadronic interactions.

1.4.4.2 Tile calorimeter

The Tile calorimeter is located directly over the EMB envelope, a barrel and two extensions cover the pseudorapidity region $0 < |\eta| < 1.7$. The barrel and extensions are divided azimuthally in 64 modules in total, while radially the division is in three segments. At $\eta = 0$ the interaction length λ of the system is 9.7.

The Tile calorimeter is a sampling calorimeter which uses plastic scintillating tiles as active material and iron plates as passive material, the structure of one of the ϕ wedges with its signal collection is shown in Fig. 1.11. The scintillation light is transmitted to photomultiplier tubes through wavelength shifters to improve the coupling. The resulting signal from photomultipliers is measured and digitized every 25 ns before being sent to the off-detector data acquisition system.

1.4.5 Muon spectrometer

The muon spectrometer is the outermost sub-detector at ATLAS, it is used to identify and measure the momenta of charged particles in the pseudorapidity range $|\eta| < 2.7$ and to trigger up to $|\eta| < 2.4$.

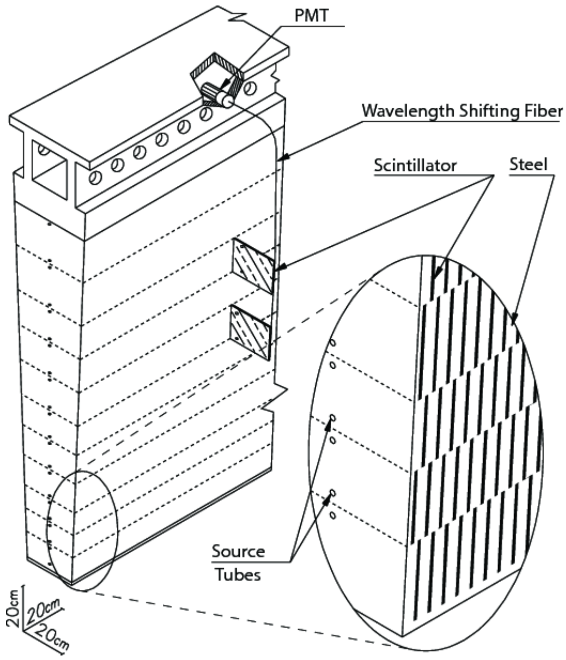

Figure 1.11: Mechanical assembly and readout scheme of the Tile Calorimeter, corresponding to a ϕ wedge. Source: [19].

The momentum measurement is based on the measurement of the sagitta in the tracks deflected by the magnetic field generated by the toroids previously discussed in subsection Section 1.4.2. The momentum resolution varies from 3 % at 100 GeV to about 10 % at 1 TeV [20].

The muon spectrometer takes advantage of different detection units to serve tracking and trigger functions. The high-precision tracking in the barrel region is done by monitored drift tubes (MDT), while in the end-cap region by both MDTs and Micromegas (MM); an estimate of muons p_T is taken, evaluating the degree of deviation from an infinite momentum muon hit pattern. The trigger functionalities are provided by resistive plate chambers (RPC) in the barrel and thin gap chambers (TGC) in the end-caps; for high-p_T muons is requested a coincidence in all the detecting layers, while with low-p_T muons the coincidence can be requested on a subset of the detecting layers.

The detection units in the barrel are arranged in three concentric cylindrical surfaces around the beam line. Following the eight-fold ϕ symmetry induced by the barrel toroids, the cylinders are divided into 16 sectors: 8 large (or long) and 8 small (or short). In the end-caps the detectors are arranged in multiple planes: the group of chambers on the outermost layer is the “Big Wheel” and the group innermost is the “Small Wheel”. It is common to refer to the layers of each sector, where detectors have similar characteristics, as “stations”.

During the Long Shutdown preceding Run-3 the muon spectrometer received upgrades to improve its tracking and trigger capabilities: a New Small Wheel (NSW) were installed in each end-cap, and new RPCs and sMDT in the region $1.0 < |\eta| < 1.3$.

A cut-away view of the spectrometer can be found in Fig. 1.12 and the nomenclature of barrel stations is shown in Fig. 1.13.

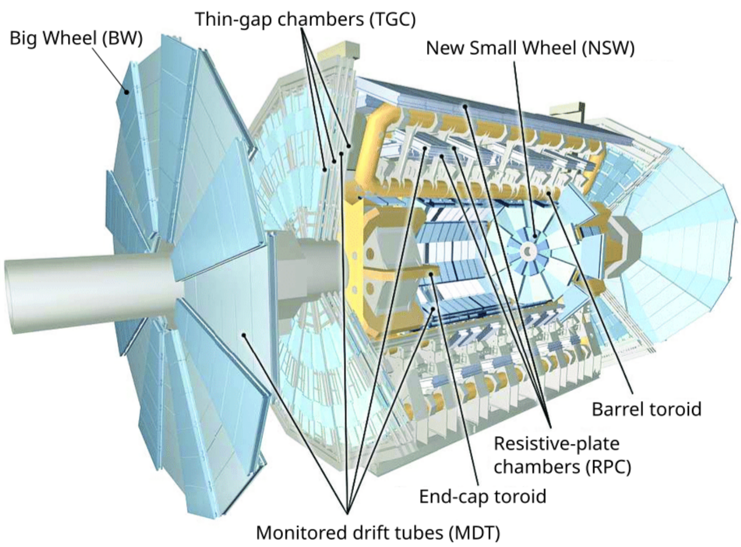

Figure 1.12: Cut-away view of the ATLAS muon system. Source: [8]

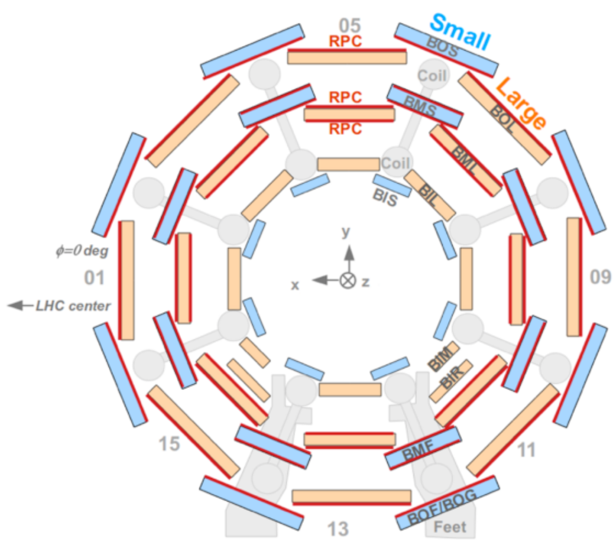

Figure 1.13: Cross-sectional view of the barrel muon spectrometer perpendicular to the beam axis (non-bending plane). The MDT chambers in the small sectors are shown in light blue, the MDT chambers in the large sectors are shown in orange and the RPC chambers in red. The supporting feet and the eight coils are also visible in grey. Source: [21]

.

1.4.5.1 RPC

The resistive plate chambers (RPCs) are gaseous parallel-plate detectors providing muon triggers and up to six position measurements along the muon's track with a space-time resolution of 2 cm × 2 ns. RPCs detectors cover the pseudorapidity range $0 < |\eta| < 1.05$ and are disposed in three concentric cylindrical doublet layers for large and small azimuthal sectors. The doublet layers are referred to as "middle confirm layer" (RPC1), "middle pivot layer" (RPC2) and "outer confirm layer" (RPC3) [22].

The basic unit is composed by two parallel high-resistive bakelite plates, which limits the charge that can be produced during ionization and therefore allows high-rate RPCs operations. Between the two plates a nominal voltage of 9.6 kV is applied; the two plates are coated with a graphite paint in order to have a uniform distribution of the voltage. The plates have a 2 mm gap and are filled with a gas mixture of $C_2H_2F_4$ at 94.7%, C_4H_{10} at 5% and SF_6 at 0.3%. This mixture helps reduce the diffusion of the charge and limits the avalanches' growth [22].

Each RPC can measure η and ϕ coordinates using copper strips placed on the external sides of the electrodes. Muon sagittae are measured with η strips perpendicular to the bending (r,z) plane. The possible regimes of RPCs operations are:

- streamer – strong signals where no amplification is required and maximum detection frequency is about 100 Hz cm^{-2}.
- avalanche – weak signals that require amplification and maximum detection frequency is about 1 000 Hz cm^{-2}. ATLAS RPCs work in this regime.

A sketch of an RPC cut-away view is shown in Fig. 1.14.

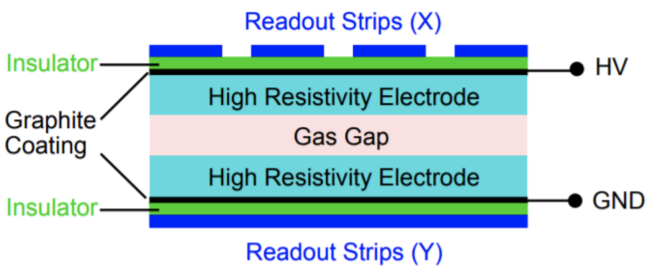

Figure 1.14: Example of RPC unit. Source: [23].

1.4.5.2 TGC

TGCs have a similar design of multiwire proportional chambers (MWPCs): the passing particle ionizes the gas and the localized cascade of electrons is then collected to the nearest wire, resulting in a signal proportional to the ionization. The TGCs operate with a gas mixture of 55% CO_2 and 45% n-pentane, the latter increases the ionization while absorbing the photons in the avalanche, providing high gain without sparks. The chamber timing resolution is 4 ns, which is suitable for the trigger. TGCs are also used to measure the second azimuthal coordinate to complement the measurement of the MDTs in the bending (radial) direction [8] [14]. TGCs are mounted in many planes and in total they cover the pseudorapidity range $1.05 < |\eta| < 2.7$.

The NSWs installed for Phase-I upgrades take advantage of small-strip TGCs (sTGC), which are an improvement of the TGCs providing a better resolution at higher luminosities [24].

1.4.5.3 MDT

Monitored drift tubes are based on 30 mm aluminum tubes, where the grounded wall acts as a cathode and the central gold-plated W-Re 50 µm wire as anode kept at high voltage for collection of ionization secondary electrons. Tubes are filled with a gas mixture of 93% Ar, 7% CO_2 at 3 bar absolute pressure for reducing the fluctuation of ionization and reduced diffusion [14].

Due to the large tube diameter, the maximum drifting time for a MDT can be about 700 ns as shown in Fig. 1.15. Among the Phase-I upgrades, is the installation of small monitored drift tubes, which are functionally the same object as MDTs but have a 15 mm diameter, permit a smaller maximum drift time and improved tracking due to a greater number of tubes packed in unit volume.

A tube layer is a group of tubes which are on the same plane of an MDT chamber. When multiple tube layers are put together they form a multi-layer. Two multi-layers, separated by a mechanical spacer, form the MDT chamber itself. In the innermost layer of the spectrometer, a multi-layer is a stack of four tube layers, while everywhere else in the spectrometer is a stack of three layers. The layout of an MDT chamber is shown in figure Fig. 1.16. The readout of MDT chambers is in general on the same side, the general rule is that in even-numbered sectors the readout is at the large ϕ end, while for odd-numbered sectors at the small ϕ end [25].

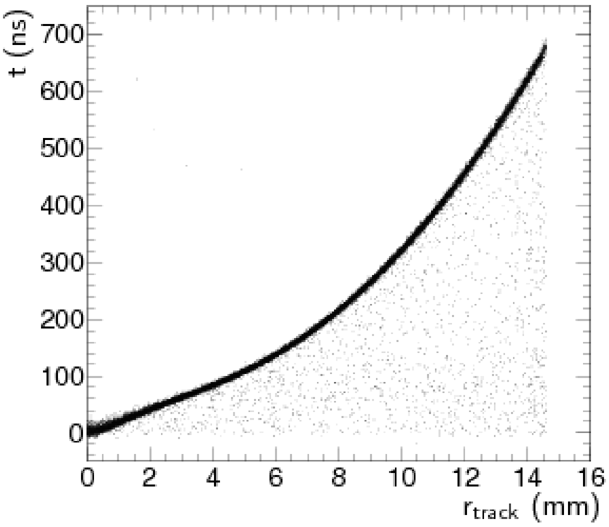

Figure 1.15: Electrons‘ drift times t in MDT caused by crossing muons as a function of the muons’ distance r_{track} from the anode. Source: [26].

1.4.5.4 New Small Wheel

The new small wheel improves triggering by reducing fake triggers from background hits, and tracking capabilities in the high rate environment. It has been been installed in the innermost station covering the pseudorapidity $1.3 < |\eta| < 2.7$ region.

The NSW implements Micromegas (MM) and small thin gap chambers (sTGC). Micromegas are parallel plate gaseous detectors where the drifting region is separate from the amplification region; this allows high rates (15 kHz cm^{-2}) because ions are not found in the drifting region. sTGC are an improvement of TGC,

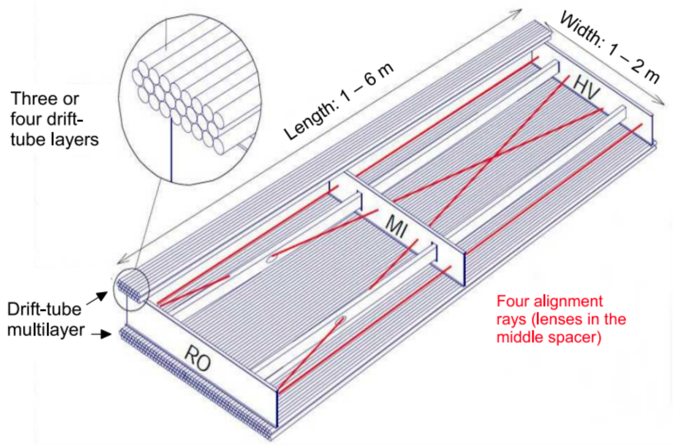

Figure 1.16: Mechanical structure of a MDT chamber. An aluminium space frame carries two multi-layers of three or four drift tube layers. An alignment system allows for monitoring of the internal geometry of the chamber. RO and HV designate the location of the readout electronics and high voltage supplies, respectively. Source: [8].

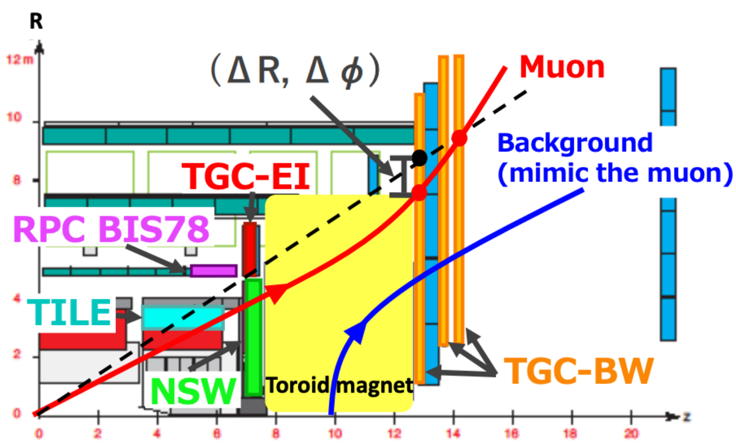

Figure 1.17: Schematic view of the muon spectrometer. Fake muon trigger in blue and good muon trigger in red. Source: [27].

Both Run-1 and Run-2 data taking has been affected by fake muon triggers, generated by low-p_T charged particles produced far from the interaction point like protons coming from beam background, therefore with the NSW these background sources can be suppressed by requiring additional coincidences. A sketch in Fig. 1.17 shows a schematic view of a fake muon trigger.

1.4.6 Forward detectors

The ATLAS detector system is completed with sets of small detecting systems built to provide good coverage in the very forward region along the beam line. The forward detectors are located at different distances from the interaction point. In order of increasing distance there are:

- BCM (Beam Conditions Monitor), at about ±2 m from the interaction point, within the Pixel detector. It measures time-of-flight and pulse height to distinguish anomalous events resulting from lost beam particles from those normally occurring in pp collisions. The BCM also provides a measurement of bunch-by-bunch luminosities in ATLAS by counting in-time and out-of-time collisions [28].
- LUCID (LUminosity measurement using Cerenkov Integrating Detector), at ±17 m from the interaction point. Its main purpose is to detect inelastic p-p scattering for integrated luminosity measurements and to provide online monitoring of the instantaneous luminosity and beam conditions.
- ZDC (Zero-Degree Calorimeter), at ±140 m from the interaction point. Its main purpose is to detect forward neutrons with $\eta > 8.3$ in heavy-ion collisions.
- ALFA (Absolute Luminosity for ATLAS), at ±170 m from the interaction point. Its main purpose is to measure elastic pp scattering for accurate track reconstruction of the proton beam during luminosity measurements.
- BPTX (Beam Pick-up Timing system), at ±175 m from the interaction point along the beam pipe. BPTX signals are used both in the trigger system and for LHC beam monitoring. The BPTX signals are discriminated to provide a Level-1 trigger when a bunch passes through ATLAS. The BPTX detectors are also used by a stand-alone monitoring system for the timing signals and LHC bunches [29].
- AFP (ATLAS Forward Proton), at ±240 m from the interaction point. Its main purpose is to measure hard pp scattering, enabling the observation and measurement of a range of processes where one or both protons remain intact which otherwise would be difficult or impossible to study.

Among the functionalities of forward detectors there are trigger capabilities.

1.5 Trigger motivation

The trigger systems are used to select events potentially interesting for physics analyses. One of the main rate limitations is the bandwidth to transfer data and the finite storage capacity. The following considerations want to give a quantitative idea of the problems caused by these limitations.

At the LHC the processes of interest for the offline physics analysis are often very rare. Fig. 1.18 shows a plot containing the cross sections for selected Standard Model processes [6]. As illustrated in

[6] During LHC Run-2 reached a peak instantaneous luminosity of about $2.06 \times 10^{34}\ \text{cm}^{-2}\text{s}^{-1}$, therefore about an order of magnitude higher.

Fig. 1.18 the number of useful interactions is small, in particular for those events with a small cross section. The quantity that express the ability of an accelerator to provide the required rate for an event with a cross section σ is the luminosity $\mathcal{L}$. In fact the rate of events dN/dt is given by

$$\frac{dN}{dt} = \sigma \mathcal{L} \quad . \tag{1.11}$$

The analytical expression of the luminosity is

$$\mathcal{L} = \frac{N_1 N_2 f N_b}{4\pi\sigma_x\sigma_y} W \quad , \tag{1.12}$$

where N_1 and N_2 are the number of particles per colliding bunch, f is the revolution frequency, N_b is the number of bunches, σ_x and σ_y are the standard deviation of the beam density profiles. W is a correction factor that takes into account operation conditions like beams colliding at a crossing angle and beam offsets.

By integrating the luminosity on a certain time interval, one can retrieve the number of events of interest that happened in that time. That is referred to as integrated luminosity $\mathcal{L}_{\text{int}}$ and it is measured in pb^{-1},

$$\mathcal{L}_{\text{int}} = \int_0^t dt' \, \mathcal{L}(t') \quad . \tag{1.13}$$

One can consider from Fig. 1.18 the production of a 125 GeV Higgs, which has cross section $\sigma_H \approx \sigma_{ggH} + \sigma_{WH} + \sigma_{VBF} \approx 50\,\text{pb}$ at $\sqrt{s} = 13\,\text{TeV}$, about 12 orders of magnitude smaller than the total cross section for proton-proton interactions. Higgs events are particularly important for the Standard Model, its detection in 2012 [30] [31] completed the Standard Model particles observed confirming a prediction that was not experimentally proved since 1964.

The expected production rate for Run-2 calculated with (1.11) is

$$\frac{dN_H}{dt} = \sigma_H \cdot \mathcal{L} = 50\,\text{pb} \cdot 2.06 \times 10^{34}\,\text{cm}^{-2}\text{s}^{-1} \approx 1\,\text{s}^{-1} \quad , \tag{1.14}$$

which means that only one Higgs boson per second has been produced on average, therefore about 40 millions of events are necessary for one Higgs. If the average size on disk of every event is 1.5 megabyte of memory then about 60 terabyte of space per second are needed and then still the Higgs would need to be found among the data. At the moment no such system capable of recording data at this bandwidth exists and in any case this would be a waste of resources. This is one reason that motivates the Trigger and Data Acquisition System at the ATLAS experiment, but more in general for any particle physics experiment.

This example showed the general importance of the trigger for physics research. In particular the trigger of muons is a fundamental part of the trigger; many processes studied at the LHC require high-p_T muons with a precise multiplicity. The presence of muons in final states (like in dimuon decays of the Higgs [32]) is in general a clean signature which makes the identification and reconstruction easier.

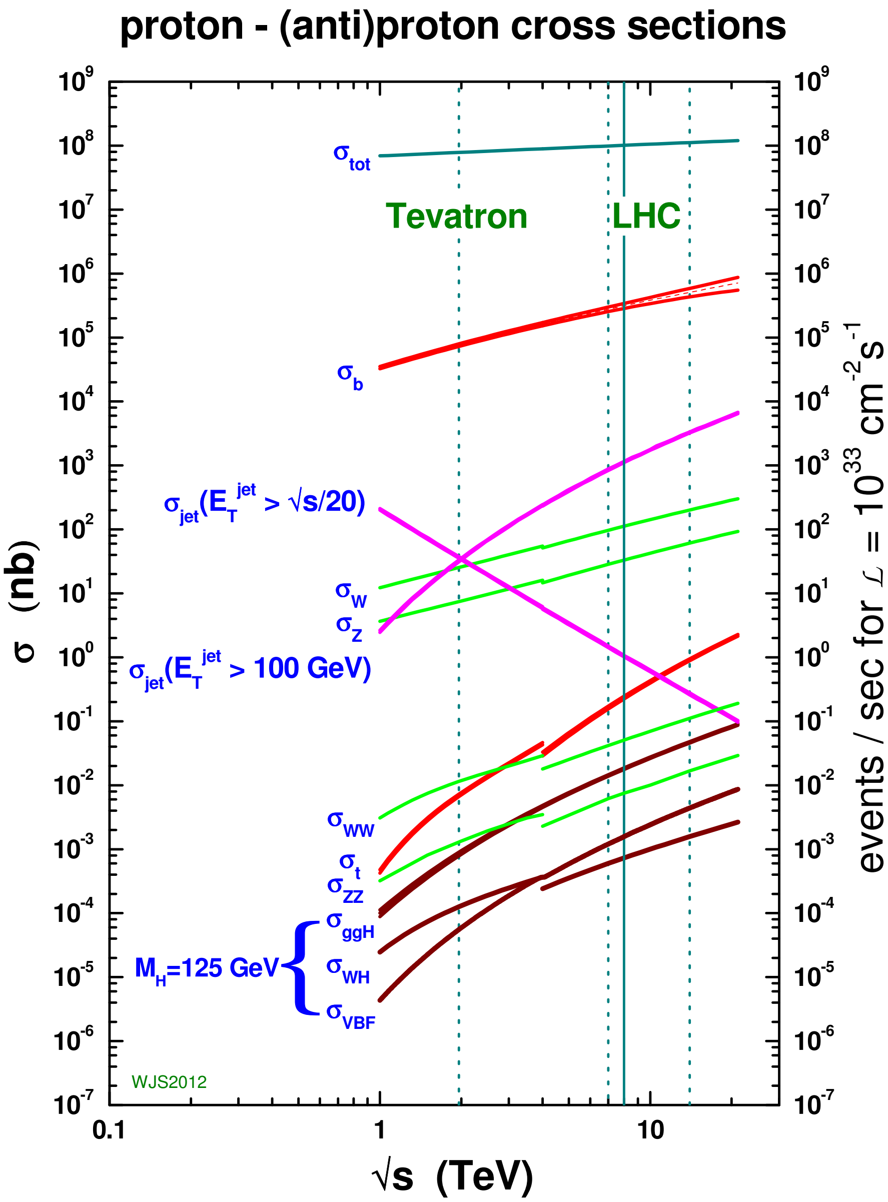

Figure 1.18: Cross section (left) and production rate (right) for various physics channels in function of centre of mass energy $\sqrt{s}$. Source: [33].

1.6 Trigger and data acquisition

The birth of trigger in particle physics experiments can be dated back to the early 1930s. Beckett and Occhialini in 1931 worked on a cloud chamber controlled by a trigger system which was implemented with two Geiger-Müller tubes positioned at the opposite sides of the chamber. A circuit detected the coincidence of the pulses generated by the passing ionizing particle and therefore triggered the camera to record the event [34].

Nowadays in particle physics Trigger and Data Acquisition refers to a collection of systems which decides, in real time[7], whether to read out or discard the measurements corresponding to each observed interaction and then converts the data in a suitable format to save on a permanent storage for offline analysis.

1.6.1 ATLAS trigger and data acquisition

The ATLAS trigger and data acquisition system is based on a two-stage system of online event selection, which means that the selection takes place in real time. Events passing the first selection will be compared with stricter and additional criteria in the second stage. The collision events are generated at the bunch-crossing rate of 40 MHz, while the rate of recording on permanent storage is 1 kHz.

The first stage in the ATLAS trigger system is the Level-1 (L1) trigger. It is built with custom electronics in order to cope with the high input rates. The characteristic signatures of events that can trigger the data acquisition are high-p_T muons or electrons, as well as considerable electromagnetic energy deposit or missing transverse energy. The decision is made by processing the event information retrieved from the muon spectrometer and the calorimeter. The L1 sub-systems responsible for receiving signals from the detector's front-end electronics and providing the first elaboration are: the L1 calorimeter trigger (L1Calo) and the L1 muon trigger (L1Muon). The selection is based on the satisfaction of energy or momentum thresholds, the multiplicity of objects involved and geometric relationships between objects in case of multiple input. The L1 acceptance decision signal (L1A) is taken in the Central Trigger system, which is composed by the Central Trigger Processor (CTP), the Muon-to-Central Trigger Processor (MUCTPI) and the topological processor L1Topo. The CTP collects information from MUCTPI, L1Topo and L1Calo and compares it against 512 programmable menu items, corresponding to the definitions of the physics signatures the experiment is triggering on. If the conditions are satisfied the CTP emits the L1A, which propagates to all the sub-detectors, readout systems, and trigger components by means of the Trigger and Timing Control (TTC) system [35].

The available time to elaborate and distribute the trigger decision, the latency time, is fixed to 2.5 μs and the maximum output frequency of accepted events shall not be greater than 100 kHz. It is important to keep the latency time as small as possible because data is kept stored in pipeline memories realized with custom electronics and with limited memory.

A scheme of the ATLAS trigger and the DAQ system of LHC Run-3 can be found in Fig. 1.19.

After the L1 trigger, the second stage is the High-level Trigger (HLT). The HLT is software based and it is implemented on a computer farm. This permits the application of more complex algorithms. In fact when data is passed to the HLT it is coupled with their Regions-of-Interest (ROI) information,

[7] Also referred to as *online*.

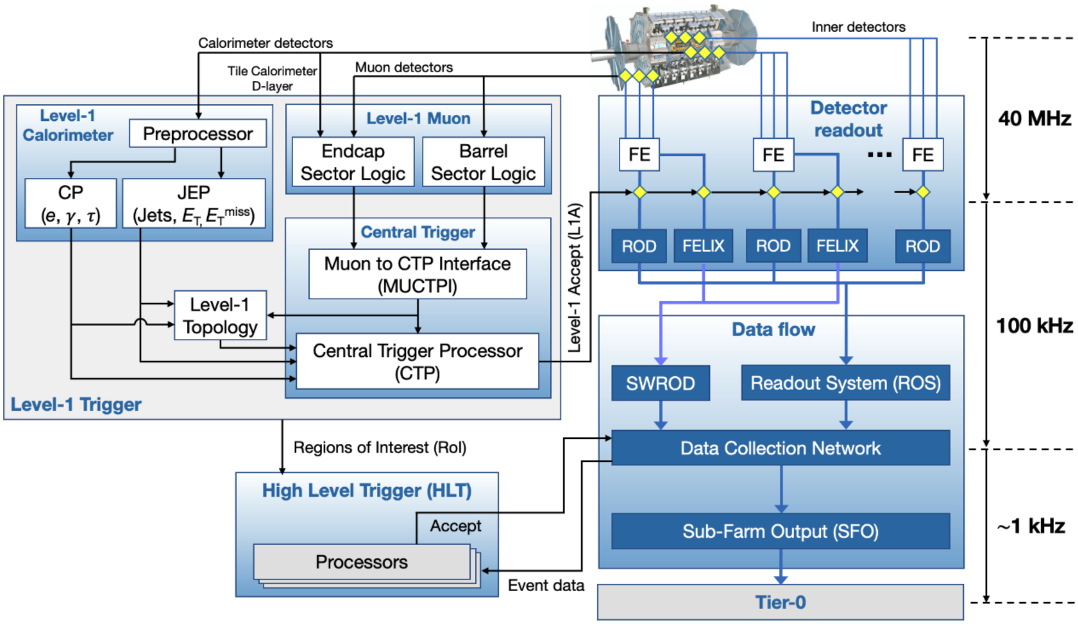

Figure 1.19: Schematic diagram of the TDAQ system for Run 3. Source: [36].

which contains the coordinates η, ϕ and additional information regarding the type of feature identified and the criteria passed, which for example can be a threshold. The output rate of the HLT to record data on disk is 1 kHz.

In the ATLAS trigger, and in general in the trigger of most experiments, it is common to use FPGAs (Field Programmable Gate Arrays) to process information and for control, configuration and monitoring tasks. FPGAs are semiconductor devices that are based around a matrix of configurable logic blocks connected via programmable interconnections. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing. It is common to refer to the specific configuration of the logic implemented on the FPGA as the firmware of the FPGA. The crucial advantages of FPGAs are that their firmware can be designed to process data in parallel and they are reprogrammable devices.

Tasks which are not related to processing the information passing through the trigger path are usually performed by integrated circuits such as by System-on-Chip (SoC). System-on-Chip are integrated circuits which feature most or all the components of a computer and a FPGA. More in detail a SoC features an ARM-based processing unit, semiconductor memory (ROM, RAM, EEPROM, flash memory), I/O interfaces and a FPGA. The FPGA provides communication with the processing FPGAs that may be included in the system.

1.6.1.1 L1 calorimeter trigger

The L1Calo trigger system receives input signals from the LAr calorimeter and the tile calorimeter. L1Calo searches for the identification of events containing objects which leave their traces mainly in the calorimeter, such as photons, electrons, taus and jets. The inputs to L1Calo trigger are signals coming from a set of 7168 trigger towers, formed by projection of calorimeter cells. The trigger

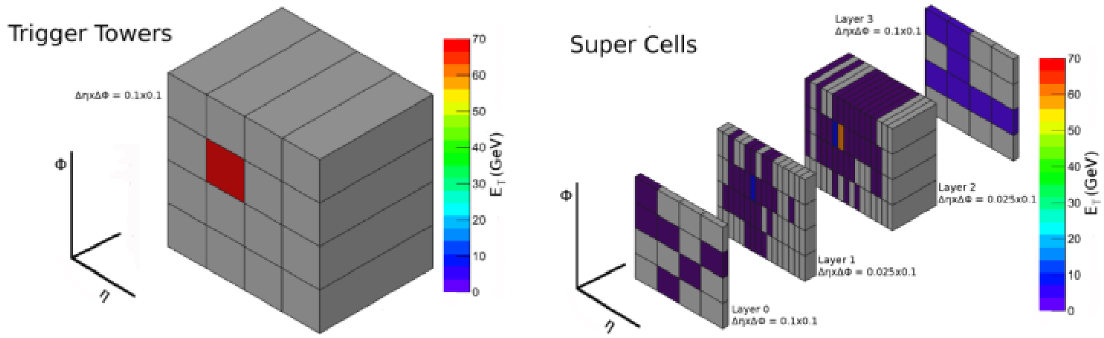

Figure 1.20: On the left a visual description of a $E_T = 70$ GeV electron shower using trigger towers, while on the right the same electron shower described with SuperCells. The higher granularity allows a more precise description. Source: [1].

towers have a granularity which varies from $\Delta\eta \times \Delta\phi = 0.1 \times 0.1$ in the barrel, to 0.4×0.4 in the forward calorimeters. Trigger towers are formed independently in the electromagnetic and hadronic calorimeters. Trigger algorithms are fed with the sum of transverse energies over a group of towers and can determine e.g. EM or hadronic clusters, which are then used for the identification of photons, electrons, taus and jets [8].

The input signals take advantage of three Feature EXtractors (FEXs) sub-systems introduced for the LHC Run-3 to find characteristic signatures and patterns in the energy depositions in order to improve the identification of particles. The eFEX (electromagnetic FEX) is optimized for electrons, tau and photons; the jFEX (jet FEX) identifies jets and large-area taus and calculates energy sums including missing transverse energy; while the gFEX (global FEX) calculates global event variables [37]. Each of the FEXs consist of a system of FPGAs to process data and a separate SoC for control purposes and installed on ATCA (Advanced Telecommunications Computing Architecture) modules.

With the Phase-I upgrades, the L1Calo system has been provided with input digital optical signals. This allows the granularity of each electromagnetic trigger tower to be increased with the use of the SuperCells [1]. Each tower will contain four layers of SuperCells providing a total of 10 SuperCells, which improve L1Calo algorithms with higher granularity, higher resolution and shower shape information resulting in better performance for the discrimination and identification of particles. Fig. 1.20 shows the improvement that one can obtain by the use of SuperCells with an electron shower of $E_T = 70$ GeV.

The output of L1Calo consist of a trigger signal of 104 bits containing calorimeter trigger signatures sent to the CTP. This output signal contains information about multiplicities of candidate photons, electrons, taus, hadrons and jets, flags for the total transverse energy, missing transverse energy and jets transverse energy. In addition FEXs information under form of trigger objects (TOBs) is sent to L1Topo.

1.6.1.2 L1 muon trigger

The L1 muon trigger (L1Muon) receives data coming from RPC and TGC chambers in the muon spectrometer. Muon candidates are identified by requiring spatial and temporal coincidence of hits in the trigger chambers crossed by the muon path.

In the barrel section of the muon spectrometer, the trigger of a low-p_T muon requires a coincidence

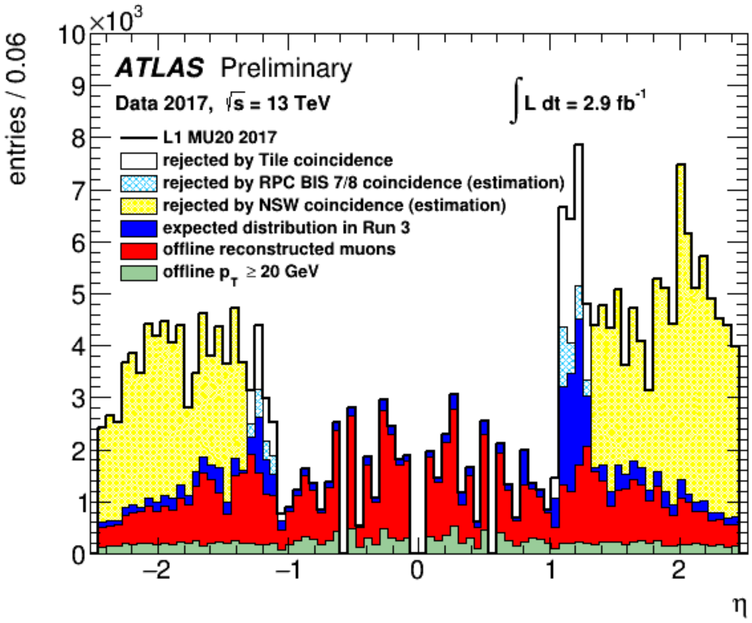

Figure 1.21: The pseudorapidity distribution of muons passing the L1 selection with $p_T > 20$ GeV from 2017 Run-2 data. The yellow histogram shows the region which is expected to be rejected by the New Small Wheel during Run-3. It is remarkable that the blue histogram of triggered muons is almost superimposed with the red histogram of good muons after offline reconstruction. Source: [27].

in the two innermost RPC stations, with three programmable p_T thresholds. For a high-p_T muon however, the coincidence is required in three RPC stations, which also can be set on three programmable p_T thresholds.

In the end-caps in order to estimate the p_T a coincidence of three hits is required in the TGCs of the Big Wheel (BW). For the Run-3 there are 15 available p_T thresholds. In order to reduce the contamination coming from fake muons, an additional coincidence is required between the BW with inner detectors in the muon spectrometer, such as the New Small Wheel, or the TGC in the end-cap inner station or the outermost layer of the Tile calorimeter. It is estimated that the expected performance for triggering muons with $p_T > 20$ GeV will be $\sim 45\%$ less contaminated by muons with p_T below threshold; see Fig. 1.21 to assess the impact of fake muon rejection on the muon reconstruction.

The estimation of muons' p_T is based on the degree of deviation δR, $\delta\phi$ from the hit pattern expected for a muon with infinite momentum. The deviations and the ROI for each candidate are sent to the Sector Logic boards, which then calculate the p_T with the use of fully programmable Look-up Tables (LUTs) for specific ROIs. The two muons with the highest p_T are selected and their p_T values, ROI, and a quality flag for coincidences in the muon stations – for a total of 64 bits per bunch-crossing – are sent to the MUCTPI through optical links. The ATLAS muon spectrometer is covered with a total of 208 Sector Logics (104 for each side of the detector), which elaborate data coming from the 64 RPCs and 144 TGCs [1].

1.6.1.3 L1 Trigger

The ATLAS L1 trigger consists in the following elements:

- **MUCTPI**

  The MUCTPI is an electronic system which is used to elaborate the output coming from all the Sector Logic boards and send muon threshold multiplicities and quality flags for various combinations of p_T. The MUCTPI also interfaces with L1Topo for the transmission of full muon ROI granularity using optical fibers. The information coming from the Sector Logics is received and processed by two FPGAs (one for each side of the detector). The results are then combined by a third FPGA which sends the multiplicities to the CTP. For control, configuration and monitoring purposes the MUCTPI has a SoC.

  The MUCTPI received a major upgrade during the Long Shutdown 2 before Run-3. The first reason for the upgrade was that a full redesign of the system was required to provide full granularity ROI information for each bunch-crossing to L1Topo. The second reason, but not less important, was that the Sector Logics were also redesigned using high-speed optical output links. The smaller physical dimensions of the optical modules, combined with Moore's law permitting manufacturers to produce FPGAs with higher densities of blocks on the same chip, made it possible to replace 18 9U VME modules with a single ATCA blade.

  When the MUCTPI receives the L1A signal from the CTP, it outputs a list of muon candidates to the DAQ system, while it sends ROI data to the HLT to seed the muon processing.

- **L1Topo**

  L1Topo is a system which applies topological algorithms based on E_T, p_T, η and ϕ requirements. Kinematic criteria between objects improve the rejection of background events, while preserving high acceptance of signal events. L1Topo receives inputs from the FEXs in L1Calo and from the MUCTPI. The output of L1Topo is directed to the CTP. For Run-3 L1Topo is realized with three boards, each containing two FPGAs for algorithm processing, I/O interfaces with optical links, and a SoC.

- **CTP**

  The Central Trigger Processor is the most important component of the Central Trigger as it is responsible for accepting or rejecting data. If the trigger menu conditions are satisfied, the CTP emits the L1A signal, enabling the readout chain.

  The CTP receives inputs from L1Calo, MUCTPI, L1Topo and forward detectors, which in general provide data arriving not at the same time. The CTP aligns and synchronizes the inputs with FPGAs, allowing the correct BCID (Bunch Crossing Identification). Once the trigger input is ready, it is compared with 512 trigger items, each of those corresponding to a combination of one or more requirements applied to the input. An example of a trigger item could be `2EM25I`, standing for 2 isolated "electromagnetic" objects, thus electrons, with $p_T > 25\,\text{GeV}$. This trigger item can be the first stage of the selection necessary for the reconstruction of decays of the type $Z \rightarrow e^+e^-$, which are a common background for Higgs boson researches. The CTP permits the application of a mask to the items for the selection of the trigger items and the application of a pre-scaling factor to control the rate of events. A pre-scale factor of N would mean that only one event in N is accepted.

The L1A signal is generated if the logical OR of all the masked and prescaled trigger items returns true. At the generation of the L1A signal, the CTP forms a 8-bit word identifying the kind of event (physics, L1A calibration), and can be used to select options in the event data processing in the front-end electronics and readout chain. Once a L1A accept signal is emitted, the CTP might receive busy signals from the sub-detectors to throttle the generation of a consecutive L1A, in order to inhibit new L1As until the busy condition set by the sub-detectors is cleared.

The CTP is also responsible for the application of a deadtime between two consecutive L1A to be within constraints on detector readout latency, which corresponds to a maximum latency of 2.5 µs.

The hardware implementation of the CTP consists of 6 modules housed in a 9U VMEbus crate.

- **TTC**

  The Trigger and Timing Control system allows the distribution of trigger and timing signals to the readout electronics. It keeps the experiment in sync and propagates the L1A signal to all the sub-detectors enabling the readout chain.

  A basic concept related to the TTC system in the ATLAS experiment is partitioning. Partitioning refers to the ability to operate subsets of the ATLAS detector in parallel and disjointly, thus facilitating the concurrent commissioning and operation of subsets of the detector. Two or more partitions can be operated together as a single partition. This process of combining partitions gives rise to a fully integrated detector description. Each partition contains a controller and a local trigger interface, responsible for receiving signals from the CTP and injecting them into the TTC of a sub-detector front end TTC partition.

  TTC partitions have been designed in such a way they have two working modes: global or local. In the global mode TTC signals come from the CTP and data is read out through the ATLAS data acquisition system. The local mode permits to run standalone partitions with private TTC signals; data is not part of the main ATLAS run. In the following any interface between sub-detector front-end components and the TTC system will be understood as an interface with the local trigger interface of the given sub-detector.

  The ATLAS TTC system has different interfaces with the various sub-systems:

  - LHC machine interface

    The TTC is interfaced with the LHC to receive the BC clock (bunch crossing) of 40 MHz and the ORBIT signal, corresponding to the period taken by one bunch to do one turn around the LHC.

  - CTP interface

    The CTP inputs in the TTC system the L1A signal, which is always coupled with the identification 8-bit word. The L1A signal and the word will be broadcasted to all the sub-detectors.

  - Sub-detectors interfaces

    The TTC system is interfaced with the sub-detector interface (ALTI) that forwards the TTC signal from the CTP to the sub-systems (L1A, Trigger Type, BCR ECR) and transmit the busy signal and calibration requests from the sub-systems to the CTP.

The TTC hardware implementation varies from element to element in the network.

- Forward detectors

  Forward detector information, for example from the BPTX detector is used in the monitoring of the timing of bunches.

1.6.1.4 Readout system

After CTP emits the L1A signal and it reaches the TTC partitions of the sub-detectors, the readout starts. The pipelined data is transferred to Readout Drivers (RODs), which gather event data on each sub-detector. RODs have a specific implementation from system to system in ATLAS. Consequently the data is transferred through a number of Readout Links (ROLs) to the Readout Buffers (ROBs), where they are temporarily stored until requested again by the HLT.

The readout system (ROS) has been upgraded during Phase-I, but not entirely, part of it will still work using ROD, ROL and ROS, but a fraction of detectors will take advantage of the new readout system, called FELIX.

In the new system the detector's front-end electronics are now interfaced with the Software Read-Out Drivers (SW RODs) through the FrontEnd Link eXchange (FELIX). FELIX is an interface between the TTC system, the data acquisition (DAQ) and detector control. It is implemented as a commercial server with two PCIe cards and it serves the function of a router via a commodity switched network, connecting the serial links from front-end ASICs and FPGAs to the DAQ and processing components.

CHAPTER **2**

ATLAS TDAQ online software

2.1 TDAQ system operations

Before introducing the TDAQ online software, whose libraries are massively used by the core of the monitoring applications developed for the MUCTPI, it is necessary to explain operational concepts related to the description of the ATLAS detector in the TDAQ system.

At the highest abstraction level, the LHC, the detector, and the TDAQ system can be described by the means of a Finite State Machine (FSM). At any time the state of the system defines the possible operations, or transitions to another state, that can be applied. The main states that are used to define TDAQ operations are:

- **Initial**

  The Initial state is the first state where it is possible to send direct commands to the TDAQ control elements. In the Initial state all the sub-components of the TDAQ system are started and idle. The transitions allowed from this state can bring only to a condition suitable for data-taking. The TDAQ system can be reverted to this state at any time, causing the re-initialisation of the components included in the TDAQ system. For example this operation can be useful during tests to include or exclude single applications from the TDAQ system.

- **Configured**

  After initialisation, the transition brings the FSM in the Configured state, where the TDAQ system is ready to start a data-taking run. During the transition from Initial to Configured, it is common that for a part of the TDAQ system configuration routines are executed. These will take care of the configuration of the TDAQ applications, for example defining parameters characteristic of the run.

- **Running**

  The TDAQ system in the Running state is taking data.

The states of the ATLAS detector and the LHC machine will not be discussed here because they are not needed for the description of the monitoring applications. The interested reader can find more on this topic in [38].

The architecture of the TDAQ system has a modular structure that can be divided in many ways, each of those with a definition corresponding to a specific function. The following list includes the definitions of the TDAQ sub-systems.

- **TTC partition**: A TTC partition represents a part of the TTC system containing a local trigger interface which can run independently of the CTP.
- **TDAQ resource**: A TDAQ resource is a part of the system which can be masked and unmasked during shutdown. An example of resources can be the input links of the MUCTPI and the CTP, as well as the busy connections.
- **TDAQ (standalone) partition**: A TDAQ partition is a subset of the ATLAS detector with testing purposes. A TDAQ partition corresponds to one or more TTC partitions. An example of a TDAQ partition can be the MUCTPI partition.
- **TDAQ segment**: A TDAQ segment is the set of TDAQ system elements that can be configured and controlled independently from the rest of the TDAQ system. A TDAQ segment can be dynamically included or excluded into a TDAQ partition on shutdown. During the partition run segments can only be killed. An example of TDAQ segment is the MUCTPI segment, included in the MUCTPI partition and including itself the monitoring applications.

During data-taking at the ATLAS experiment the partition used by the experiment is the `ATLAS` partition, where all the sub-systems will end up being the elements of many segments included in the partition and the full TDAQ functionalities are available. To test the MUCTPI monitoring applications, it was possible to run a custom standalone partition which included a single segment containing information of the monitoring applications.

It is possible to run and control a TDAQ partition with the help of a graphical user interface for its control. The application which does this is called IGUI and it permits rc commands to be sent and makes transitions between the states of the FSM with the click of a button. The IGUI also permits the receipt of information from the Error Reporting Service (see Section 2.2.5) of the applications loaded into the partition's segments. Fig. 2.1 shows the main screen of the IGUI.

2.2 TDAQ Online Software

The TDAQ online software is a collection of common libraries, non-specific to each sub-system[1] which is used to provide high-level services such as configurations, control, and monitoring of the TDAQ system [38]. It is currently composed of more than 200 packages which are available for ATLAS users in the form of regular software releases [40].

The online software is used by all the TDAQ sub-systems and it provides the infrastructure so they can work together, as well as the interface between the user and the TDAQ system. Control routines support the initialisation and shutdown of the TDAQ system, providing control commands distribution, synchronization, error handling and system verification. Configuration services are designed in such a way that they can hold a large number of parameters describing system topology, hardware components, software components and running modes. During data-taking runs, the online

[1] But still specific to the ATLAS detector

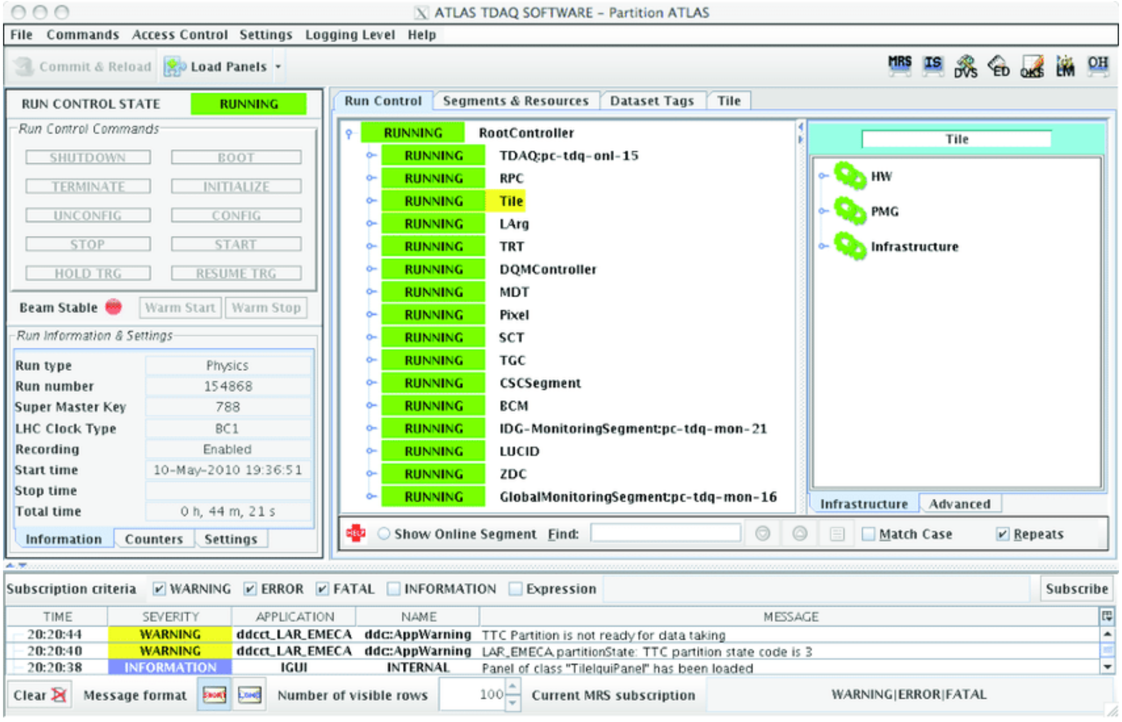

Figure 2.1: Partition's IGUI where many applications are running. Source: [39].

software grants access to information sampled by monitoring tasks, statistics, histograms, error and alert messages sent by individual applications integrated in the TDAQ system [38].

The online software for the most part is written in C++, although some of the DAQ software uses Java, and Python is the standard choice to implement complex scripting tasks. In the ATLAS experiment all the software packages are available on the DevOps platform GitLab [41], which is also used for version control, the practice of tracking and managing changes to software code. The software used to build ATLAS TDAQ projects is CMake [42]. CMake is an open-source family of tools designed to build and test software packages. CMake is used to control the software compilation process using configuration files, independent from platform and compiler, which generate native makefiles and workspaces that can be used in the compiler environment. In general each package is provided with its configuration file `CMakeLists.txt`, which contains all the rules needed to compile source files to build an executable file.

The online software services are based on intermediate layers which contain all the information shared between processes. The communication system between different processes is managed by the Inter Process Communication (IPC), which in turn takes advantage of the CORBA (Common Object Request Broker Architecture) standard.

CORBA is a standard architecture for distributed object systems which is designed to allow distributed objects to interoperate in a heterogeneous environment. The IPC is a software package which wraps around CORBA providing an API for the core CORBA facilities, an API for the OMG (Object Management Group) Naming Service and caching for remote object references [43]. IPC and CORBA are present on all the machines, otherwise the transfer of data between machines would not be possible.

The IPC provides a definition of TDAQ partition in the context of a programming language with the IPCPartition class. It provides a set of functions for accessing objects registered within a certain partition. The IPCPartition implementation is based on the OMG Naming Servics, which provides the mechanism to locate the binding of the CORBA object reference with a name. Any application can interact with the Naming Service to obtain the reference of a certain object by providing its binding name. The Naming Service API is wrapped with the IPCPartition and the IPCNamedObject, which provide the methods to bind/unbind an object to a name in a partition and to retrieve the reference of a remote object [43].

In the following subsections is a brief description of the basic services involved in the development of a TDAQ application. These services were fundamental for the realization of the MUCTPI monitoring applications.

2.2.1 Information Service

The Information Service (IS) is a monitoring service based on the CORBA standard and provided with a C++ API. It is used to share user defined information between TDAQ applications in a distributed environment [44]. The architecture of the IS framework is shown in Fig. 2.2.

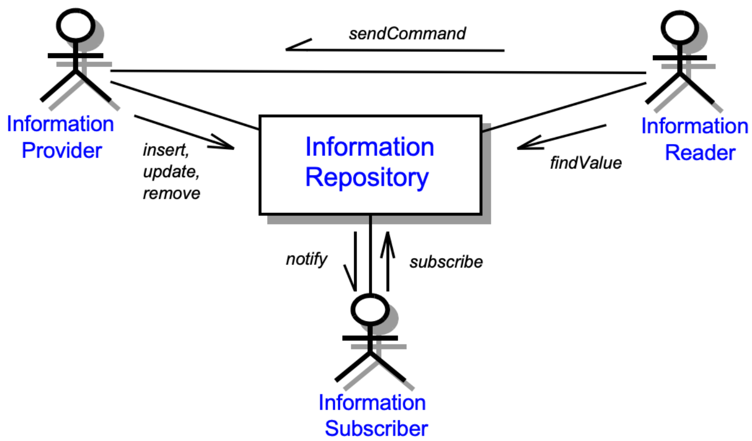

Figure 2.2: Information Service's architecture schema. Source: [44].

The container dedicated to shared information is the IS repository and its content can be created, modified or deleted by an Information Provider. The Information Reader can get information values from the IS repository by sending requests to it and can send commands to any of the running Providers. The Information Subscriber can subscribe to the IS repository for updates about the changes, receiving notifications every time the Provider interacts with the repository. In general an application can implement any combination of the possible interactions described.

The IS repository is implemented by processes called IS Servers, which are accessed by applications through their unique Process IDentifier (PID).

In order to provide the desired interactions with an IS server a three level object model is used:

- **Level 1** – Description of the information as meta-classes in an XML file, called IS schema.
- **Level 2** – Meta-classes definitions are mapped in C++ classes.

- **Level 3** – IS information are Level 2 classes instances.

All the MUCTPI monitoring information has been described with proper XML IS schema files. Each monitoring application instance has its IS object counterpart, which has the counter objects which contain the monitoring information to be published under the MUCTPI server.

2.2.2 Object Kernel Support

The OKS (Object Kernel Support) is a library which supports in-memory object management. It is used for the static description of the TDAQ system and it is based on two types of files: schema and data files. Schema files contain the definitions of the classes of the objects that are going to be described and the data files contain the instances of the objects, based on the definitions found in the schema. Both OKS schemas and data files are XML files, therefore it is possible to think of the OKS as a textual database, although it does not implement the functionalities of a database.

The definition of classes in the schema files can take advantage of the common OOP concepts like inheritance. Other than basic data types like int, char, etc. in the classes it also is possible to define vectors of basic types of other classes. The OKS files can be used by CORBA to define the objects which will be exchanged by processes. For example partitions, segments, and monitoring applications are described with OKS schemas, but more in general every element in the TDAQ system will be described by its dedicated schema file.

2.2.3 Online Histogramming Service

The Online Histogramming Service (OH) allows histograms generated by a number of providers to be retrieved, which can be for example user monitoring applications. It implements full support for graphs, but the system is not responsible for creating and filling the histograms, which can be created and filled using ROOT. The OH working principle is similar to IS: histograms are published in a server and they can be accessed through a receiver interface.

2.2.4 WebIS

A web interface, called WebIS, grants access to any object and histogram in the ATLAS IS Servers, therefore it makes it possible to create JavaScript based web pages to show updated information from IS and OH.

Every operation with WebIS is defined by a specific URL and retrieves information in XML format, except for histogram images where a PNG image is returned.

2.2.5 Error Reporting System

The Error Reporting System (ERS) provides a service which allows developers to throw, track and address errors in TDAQ applications. The ERS API interface is provided in C++ and it is built around three main concepts: the Issue class and the Input/Output streams [45].

2.2.5.1 Output stream

The output stream is an interface that, depending on the level of severity of the issue, can send messages to the stream. The implemented levels of severity are LOG, INFO, WARNING, ERROR, FATAL and DEBUG. The output stream can be configured at a software level to be directed to a well defined stream and can be eventually followed by a list of relevant parameters. For example one can have:

```
TDAQ_ERS_ERROR=''stderr, filter(Muon), abort''
```

which means that all the errors will be redirected to the standard error stream and those issues coming from the Muon subsystem will result in aborting the application.

2.2.5.2 Input stream

For inter-process issues, ERS provides developers the possibility to define a corresponding input stream for each output stream. The input stream will receive all the issues directed to their output counterpart.

2.2.5.3 Issue class

The `Issue` class is the main class used to report problems. It inherits directly from `std::exception` and it is abstract. This permits the developer to define a specific Issue by inheriting from `ers::Issue` or adopt one already existing in a different package, furthermore the compatibility with the standard library allows to catch Issues like a `std::exception`.

2.2.6 PBEAST Grafana dashboards

The IS service described in Section 2.2.1 is a fundamental part of the online services used by TDAQ applications. IS allows the publication of monitoring data on dedicated servers and it is used by all the sub-systems and sub-detectors in the ATLAS experiment.

The Persistent Back-End for the ATLAS Information System (PBEAST) [46] service stores all raw monitoring data for the lifetime of the ATLAS experiment and provides programming and graphical interfaces to access them, including Grafana dashboards [47].

Grafana is a multi-platform open-source analytics and interactive visualization web application. It provides the possibility to create web panels containing charts and graphs when connected to supported data sources. For TDAQ operations the data source is PBEAST.

CHAPTER 3

Muon-to-central trigger processor interface

3.1 Functionalities

As anticipated in Section 1.6.1, the MUCTPI is an electronic board part of the Central Trigger operating in between L1Muon and the CTP, and also interfaced with L1Topo. Its main functionalities can be summarized as follows:

- Receive and combine muon candidates from the 64 RPC and 144 TGC trigger sectors.
- Avoid double counting of single muons detected in overlapping muon sectors.
- Calculate and send muon multiplicities of programmed p_T thresholds and quality flags to the CTP.
- Sort muon candidates by p_T, encode η and ϕ coordinates and send Muon Trigger Objects (TOBs) to L1Topo.
- Send muon candidates, thresholds multiplicities and TOBs to DAQ for triggered events.
- Provide trigger monitoring facilities like muon input rate counters, per-bunch histograms and event monitoring.

3.2 Hardware

The MUCTPI implementation for the LHC Run-3 consists of a single ATCA blade. ATCA is a specification designed to provide the requirements for carrier grade communications equipment and it was selected as a hardware platform for the upgrade of the back-end electronics for the ATLAS experiment. The ATCA base specification defines the frame and shelf form factors, core backplane fabric connectivity, power, cooling, management interfaces, and electromechanical specifications of the boards. The use of this standard enabled the replacement of the bulky 18 9U VME boards used as MUCTPI in the previous LHC Run-1 and Run-2. The ATCA specification permitted improved cooling, power supply and board real-estate.

The MUCTPI takes advantage of three large state-of-the-art Xilinx FPGAs and a Xilinx SoC. Three prototypes of MUCTPI have been built and the fourth is in production. The configuration described in this chapter is the MUCTPI v3, although currently it is installed the MUCTPI v1. The reason is that is planned a replacement of MUCTPI v1 with MUCTPI v4, which has the same components of v3 but small modifications on the PCB [48]. The various MUCTPI versions feature different models of FPGAs and SoC, but they provide the same functionalities. The block diagram showing the interfaces can be found in Fig. 3.1.

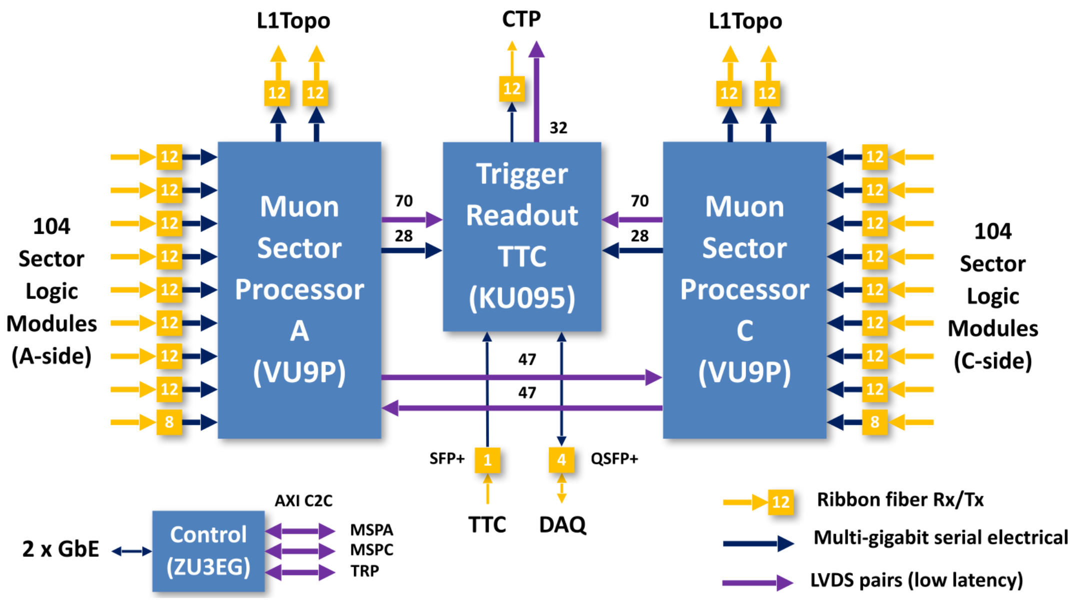

Figure 3.1: MUCTPI v3 block diagram. Source: [49].

Two of the FPGAs, called Muon Sector Processors (MSPs), are Xilinx Virtex Ultrascale (VU160). The MSPs receive in total 208 optical links, one for each Sector Logic in the L1Muon system, and are organized in such a way that one MSP receives the 104 optical links from the side A (MSPA) and the other MSP the half coming from the side C of the detector (MSPC). Both MSPs are designed to synchronize the muon sector data, provide overlap handling by suppression of the double counting of single muons from overlapping sectors, send muon (η,ϕ) coordinates and p_T (TOBs) to L1Topo and provide monitoring capabilities on rates and per-bunch histograms for each sector [49]. The Sector Logic coverage of each side of the detector is organized with 32 barrel, 48 end-cap, and 24 forward overlapping sectors. An example of overlap between sectors is shown in Fig. 3.2.

The other FPGA is a Xilinx Kintex Ultrascale (KU095), called Trigger Readout Processor (TRP), it merges information coming from MSPA and MSPC, calculates and sends muon multiplicities to the CTP over an optical link running at 6.4 Gb/s. The TRP is also responsible for receiving, decoding and distributing the TTC information, output muon candidate, and thresholds multiplicities. The TRP outputs TOBs to the DAQ system on a different path.

The SoC installed on the MUCTPI is used for board configuration, control and monitoring of the hardware. The programmable logic part of the SoC for compatibility reasons with the other FPGAs is also a Xilinx FPGA, model Zynq Ultrascale+ MPSoC (ZU3EG) and it provides read and write access to registers and memories of the other processing FPGAs [51]. The processor system of the SoC

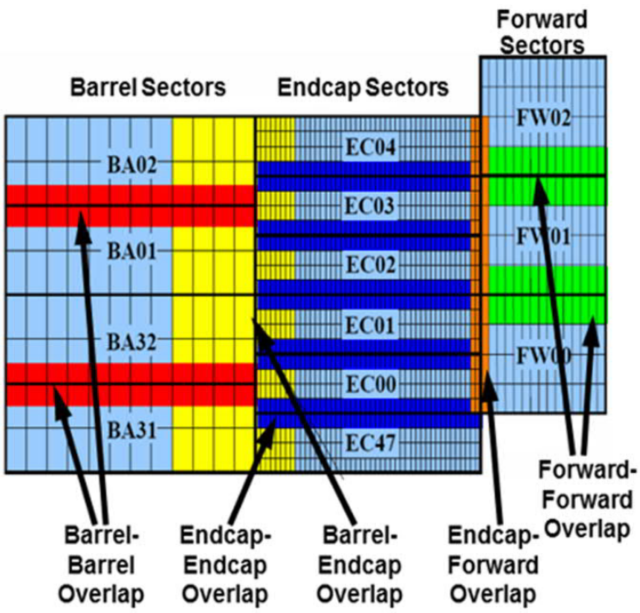

Figure 3.2: Example of overlapping sectors for barrel, end-cap and forward sectors. Source: [50].

is equipped with a quad-core 64-bit ARM Cortex-A53 processor[1] with 4GByte DDR4 memory, 2 Gigabit Ethernet interfaces to communicate with the ATLAS run control system and LVDS links to communicate with the hardware for configuration and control. Registers and memories are mapped into the processing system part of the SoC using AXI (Advanced eXtensible Interface) protocol and can be accessed using `/dev/uio` files through C++ code for single read/write functions. The operating system running on the SoC is CentOS Linux. It has been chosen because it is widely accepted at CERN, it is available for aarch64 architecture, and it is well supported and updated by the CentOS community.

The monitoring applications object of this thesis are run control applications and run on the SoC (except the per-bunch OH apps, which run on a separate PC, see Section 4.5), therefore the SoC needs to run TDAQ software and this is possible because the TDAQ software is cross-compiled using the `gcc` compiler for arm7 and aarch64. The arm7 version is built from source, while the aarch64 version is the one available on the Worldwide LHC Computing Grid (WLCG), more detail on this topic is given in Section 3.4. Fig. 3.3 shows the role of the SoC on the MUCTPI.

3.3 MUCTPI registers and low-level software

The firmware for the MUCTPI processing FPGAs (MSPA, MSPC and TRP) is developed using the Xilinx Vivado design suite [51]. All the registers and their bit fields, as well as memories and FIFOs are described in a XML file. The XML files are processed by pre-compiler programs, called `HWCompiler` and `L1CTHardwareCompiler`, whose outputs are C++ classes with read and write methods for each register or memory in the XML map. This process provides the basic elements to implement an application programming interface (API) to the FPGA firmware, which is known as low-level software.

[1] The MUCTPI v1 and v2 feature a drifferent SoC with a 32-bit architecture.

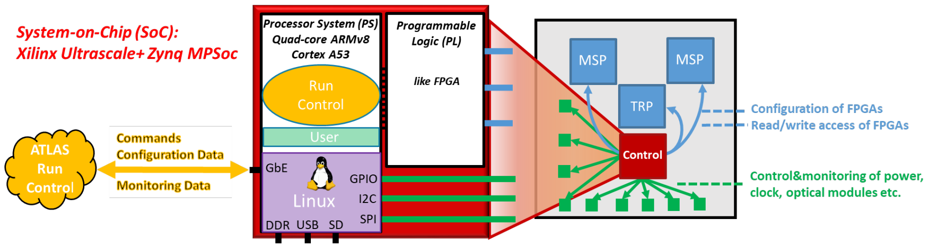

Figure 3.3: Role of the SoC on the MUCTPI. Source: [49].

The low-level software is heavily used in every monitoring application object of this thesis and part of the work done during the Technical Studentship was implementing a portion of the low-level software for the MSP and TRP. All the low-level software functions have been included and implemented in the headers and source files of the package `MuctpiModule`.

The use of low-level software in any monitoring application is fundamental because data needs to be read periodically. The FPGA firmware also contains many configuration registers, which are also accessed by the monitoring applications either during the Configuration transition between Initial and Configured state or in the Running state of the FSM.

The low-level software and the access to FPGAs registers is also fundamental for debug and test purposes. In fact the MUCTPI has been largely tested with a "menu" program running on the SoC and accessed from the workstations with `ssh`. The menu program is a text-driven program showing the list of available low-level functions and providing access to them for evaluating their values in the terminal. Part of the Technical Studentship' work has been implementing generic menu functions, but also functions to mimic the behavior of the reading routines of the monitoring applications for their debugging.

In Section 3.3.1 and Section 3.3.2 are described the firmware listings for the two MSPs and the TRP respectively. In both MSPs and TRP firmware, the counters monitor the number of muons. In order to obtain the rate counter in Hz, the value is multiplied by the orbit frequency value in Hz (see Table 1.1) and divided by the turn number. The firmware blocks described are also per-bunch. The per-bunch monitoring is important for timing, e.g. of Sector Logic inputs, and monitor the activity in earlier and/or later bunch crossings compared to the triggered one.

3.3.1 MSP

3.3.1.1 Sector Logic input

The block of firmware for the MSPs contains the relevant information for the monitoring of the input rates of candidates coming from each Sector Logic of the detector. The monitoring is counter-based and each sector has four different counters that can be configured to monitor different p_T thresholds by using the low-level functions that permit acting on the `PtSelect` register. Fig. 3.4 shows the interested firmware block.

It is important to mention that the reading process of counter-based information is not immediate, but requests the execution of multiple low-level functions to be executed correctly. For example in the case of the counters in the PTM block, the sequence of actions to take is the following. The other monitoring activities execute similar procedures.

1. Counters are disabled — necessary because counters cannot be read if still active and increasing.
2. Reading of the counter values.
3. Clear the counter values — necessary because the counters are read to calculate their *rate* in the past monitoring period.
4. Enable the counters — necessary for next reading.

```
<block name="PTM">
  <block name="SLR"                 addr="0x013C0000" multiple="3" offset="0x01000000">
  <block name="SCT"                 multiple="40" offset="0x20">
    <register name="PtSelect"       addr="0x00000000" multiple="4" offset="4" mask="0x00007FFF"/>
    <register name="CounterValue"   addr="0x00000010" multiple="4" offset="4"/>
  </block>
  </block>
  <memory name="BcidLut"            addr="0x023F0000" size="0x200"/>
  <register name="GlobalStatus"     addr="0x000BFD10" modf="R">
    <field name="TurnCount"         mask="0x00FFFFFF" form="N"/>
    <field name="TurnCountOverflow" mask="0x01000000" form="B"/>
    <field name="CounterOverflow"   mask="0x02000000" form="B"/>
    <field name="ClearBusy"         mask="0x04000000" form="B"/>
    <field name="CountersEnabled"   mask="0x08000000" form="B"/>
    <field name="CopyBusy"  mask="0x10000000" form="B"/>
  </register>
  <register name="GlobalControl"    addr="0x000BFD14" >
    <field name="Clear"             mask="0x00000001" form="B"/>
    <field name="Enable"            mask="0x00000002" form="B"/>
    <field name="Copy"              mask="0x00000004" form="B"/>
    <field name="BcidOffset"        mask="0x0000FFF0" form="N"/>
  </register>
</block>
```

Figure 3.4: Firmware block for the Sector Logic input (PTM).

3.3.1.2 Sector Logic per-bunch monitoring

The per-bunch monitoring information is also counter-based. Each sector is monitored with one counter containing the Sector Logic input values for each of the 3564 bunches present in an orbit. The firmware offers the possibility to mask the desired p_T value to monitor. Fig. 3.5 shows the interested firmware block.

```
<block name="PBM" multiple="40" offset="256">
  <register name="Status"                     addr="0x00000060" modf="R">
    <field name="CounterActive"               mask="0x00000001" form="B"/>
    <field name="ClearBusy"                   mask="0x00000002" form="B"/>
    <field name="TurnCount"                   mask="0x0FFFFFF0" form="N"/>
  </register>
  <register name="Control"                    addr="0x00000064">
    <field name="Enable"                      mask="0x00000001" form="B"/>
    <field name="Clear"                       mask="0x00000002" form="B"/>
    <field name="PtSelect"                    mask="0x0007FFF0" form="N"/>
  </register>
</block>

<block name="SL_PBM_MEM"    addr="0x01300000" multiple="3" offset="0x01000000">
  <memory name="CounterMemory"   addr="0x00000000" size="0x4000" multiple="40" offset="0x4000" mask="0x000FFFFF"/>
</block>
```

Figure 3.5: Firmware block for the Sector Logic per-bunch input (PBM). The information on the histograms is in a separate block.

3.3.2 TRP

3.3.2.1 Multiplicity combiner

The multiplicity combiner register allows the output of the TRP directed to the CTP to be monitored by using 32 different programmable multiplicity thresholds and for each of those three different counters are available. The counters can be configured to the desired multiplicity to monitor through the use of three bits. The values of the mask register can be mapped to a number in $[0,7]$, corresponding to: exactly 0 muons, exactly 1 muon, . . . , ≥ 7 muons. Fig. 3.6 shows the interested firmware block.

```
<block name="MLT">
  <register name="CounterControl"              addr="0x00060180">
    <field name="Clear"                        mask="0x00000001" form="B"/>
    <field name="Enable"                       mask="0x00000002" form="B"/>
    <field name="Copy"                         mask="0x00000004" form="B"/>
    <field name="BcidOffset"                   mask="0x0000FFF0" form="N"/>
  </register>
  <register name="CounterStatus"               addr="0x00060184" modf="R">
    <field name="TurnCount"                    mask="0x00FFFFFF" form="N"/>
    <field name="TurnCountOverflow"            mask="0x01000000" form="B"/>
    <field name="CounterOverflow"              mask="0x02000000" form="B"/>
    <field name="ClearBusy"                    mask="0x04000000" form="B"/>
    <field name="CountersEnabled"              mask="0x08000000" form="B"/>
    <field name="CopyBusy"                     mask="0x10000000" form="B"/>
  </register>
  <register name="CounterMultiplicitySelect"   addr="0x00060190" multiple="24" offset="4">
    <field name="MultiplicityMask"             mask="0x0000007F" multiple="4"  offset="7" form="N"/>
  </register>
  <register name="CounterValue"                addr="0x00060200" multiple="96" offset="4" modf="R"/>
  <register name="WidthControl"                addr="0x000603F0" multiple="2"  offset="4">
     <field name="Width"                       mask="0x00000003" multiple="16" offset="2" form="N"/>
  </register>
  <register name="OutputControl"               addr="0x00060400" multiple="16" offset="4">
    <field  name="BitSelect"                   mask="0x0000003F" multiple="4"  offset="8" form="N" />
  </register>
  <memory name="CounterBcidLut"                addr="0x02000000" size="0x0200" mask="0xFFFFFFFF"/>
</block>
```

Figure 3.6: Firmware block for the multiplicity combiner output to the CTP (MLT).

3.3.2.2 Multiplicity combiner per-bunch monitoring

The multiplicity combiner also offers the possibility to monitor the counter values for each bunch crossing. 32 counters are available for the monitoring and each of those can be configured to the desired multiplicity. Fig. 3.7 shows the relevant firmware block.

```
<block name="PBM"                      addr="0x03000000">
  <register name="Control"             addr="0x00004000">
    <field name="Enable"               mask="0x00000001" form="B"/>
    <field name="Clear"                mask="0x00000002" form="B"/>
    <field name="BcidOffset"           mask="0x0000fff0" form="N"/>
  </register>
  <register name="Status"              addr="0x00004004" modf="R">
    <field name="CounterActive"        mask="0x00000001" form="B"/>
    <field name="ClearBusy"            mask="0x00000002" form="B"/>
  </register>
  <register name="HistogramSelect"     addr="0x00004008" mask="0x0000001f"/>
  <register name="TurnCount"           addr="0x0000400c" mask="0x00ffffff" modf="R"/>
  <register name="MultiplicitySelect"  addr="0x00004020" multiple="8" offset="4">
    <field  name="Mask"                mask="0x0000007F" multiple="4" offset="7" form="N"/>
  </register>
  <memory   name="Histogram"           addr="0x00000000" size="0x000037B0" mask="0x0003ffff" modf="R"/>
</block>
```

Figure 3.7: Firmware block for the multiplicity combiner per-bunch output to the CTP (PBM).

3.3.3 Busy

This part of the firmware provides the structure for the monitoring of busy sources on the MUCTPI, e.g. the FIFO DAQ output, or the MSPs link input FIFO. The monitoring is based on the FIFO (First In First Out) queueing structure. The FIFO level (number of data words for each busy source) status determines the percentage of the busyness state of the source. At every iteration if the FIFO level passes a watermark it is cleared. Fig. 3.8 shows the relevant firmware block.

```
<block name="BSY"                           addr="0x02800000">
  <register name="Status"                   addr="0x00004000" modf="R">
    <field name="MonResultBusy"             mask="0x00000001" form="B"/>
    <field name="MspDaqBusy"                mask="0x00000002" form="B"/>
    <field name="MspMonDeramdomizerBusy"    mask="0x00000004" form="B" multiple="2" offset="1"/>
    <field name="TrpDaqBusy"                mask="0x00000010" form="B"/>
    <field name="MspL1aDeramdomizerBusy"    mask="0x00000020" form="B" multiple="2" offset="1"/>
    <field name="OnDemandBusy"              mask="0x00000080" form="B"/>
    <field name="CombinedBusy"              mask="0x00000100" form="B"/>
    <field name="DaqInputFifoBusy"          mask="0x00000200" form="B" multiple="2" offset="1"/>
    <field name="DaqOutputFifoBusy"         mask="0x00000800" form="B"/>
    <field name="DaqLinkFullFlag"           mask="0x00001000" form="B"/>
    <field name="DaqMultFifoBusy"           mask="0x00002000" form="B"/>
  </register>
  <register name="Control"                  addr="0x00004004">
    <field name="ClearFifo"                 mask="0x00000001" form="B" modf="W"/>
    <field name="WriteStrobe"               mask="0x00000002" form="B" modf="W"/>
    <field name="WriteMode"                 mask="0x00000004">
      <value name="MANUAL"                  data="0x00000000"/>
      <value name="AUTOMATIC"               data="0x00000004"/>
    </field>
    <field name="ReadMode"                  mask="0x00000008">
      <value name="FIRST"                   data="0x00000000"/>
      <value name="LAST"                    data="0x00000008"/>
    </field>
  </register>
  <register name="WriteCounter"             addr="0x00004008">
    <field name="MonResultBusy"             mask="0x00000001" form="B"/>
    <field name="MspDaqBusy"                mask="0x00000002" form="B"/>
    <field name="MspMonDeramdomizerBusy"    mask="0x00000004" form="B" multiple="2" offset="1"/>
    <field name="TrpDaqBusy"                mask="0x00000010" form="B"/>
    <field name="MspL1aDeramdomizerBusy"    mask="0x00000020" form="B" multiple="2" offset="1"/>
    <field name="OnDemandBusy"              mask="0x00000080" form="B"/>
    <field name="CombinedBusy"              mask="0x00000100" form="B"/>
    <field name="DaqInputFifoBusy"          mask="0x00000200" form="B" multiple="2" offset="1"/>
    <field name="DaqOutputFifoBusy"         mask="0x00000800" form="B"/>
    <field name="DaqLinkFullFlag"           mask="0x00001000" form="B"/>
    <field name="DaqMultFifoBusy"           mask="0x00002000" form="B"/>
    <field name="Interval"                  mask="0x00004000" form="B"/>
  </register>
  <register name="ClearCounter"             addr="0x0000400c" modf="W">
    <field name="MonResultBusy"             mask="0x00000001" form="B"/>
    <field name="MspDaqBusy"                mask="0x00000002" form="B"/>
    <field name="MspMonDeramdomizerBusy"    mask="0x00000004" form="B" multiple="2" offset="1"/>
    <field name="TrpDaqBusy"                mask="0x00000010" form="B"/>
    <field name="MspL1aDeramdomizerBusy"    mask="0x00000020" form="B" multiple="2" offset="1"/>
    <field name="OnDemandBusy"              mask="0x00000080" form="B"/>
    <field name="CombinedBusy"              mask="0x00000100" form="B"/>
    <field name="DaqInputFifoBusy"          mask="0x00000200" form="B" multiple="2" offset="1"/>
    <field name="DaqOutputFifoBusy"         mask="0x00000800" form="B"/>
    <field name="DaqLinkFullFlag"           mask="0x00001000" form="B"/>
    <field name="DaqMultFifoBusy"           mask="0x00002000" form="B"/>
    <field name="Interval"                  mask="0x00004000" form="B"/>
  </register>
  <register name="WriteInterval"            addr="0x00004010" mask="0x00ffffff"/>
  <register name="FifoWatermark"            addr="0x00004014" mask="0x00000fff"/>
  <register name="FifoStatus"               addr="0x00004018" modf="R">
    <field name="Level"                     mask="0x00000fff" form="N"/>
    <field name="Full"                      mask="0x00001000" form="B"/>
  </register>
  <fifo name="BusyFifo"                     addr="0x00000000" size="0x4000" mask="0x00ffffff" modf="R"/>
</block>
```

Figure 3.8: Firmware block for the busy sources (BSY).

3.4 MUCTPI integration

The ATLAS run control provides the framework to impart control commands to the MUCTPI, load configuration data and perform online monitoring. In order to allow user applications to access MUCTPI hardware and to integrate them in the ATLAS run control system it is necessary to integrate the MUCTPI SoC in the ATLAS Control Network (ATCN) and the TDAQ software.

The SoC is booted using boot files provided by Xilinx and it runs a Xilinx PetaLinux operating system[2]. However CERN IT and ATLAS sysadmins security policy at P1 do not allow this operating system to run in the ATCN for security reasons [52]. The solution adopted is to isolate the SoC and the MUCTPI behind a gateway PC connected via GigaBit Ethernet. The operating system installed on the gateway PC is CentOS/aarch64, and its root file system is mounted on the SoC with NFS (Network File System). This configuration provides a full operating system with basic functionalities for login, scripting, mounting other file systems and networking (ssh, bash, python tcl, NFS, ifconfig, . . .). Other packages can be added to the installation. Fig. 3.9 shows a schema of the architecture of the MUCTPI using a cross-installed file system.

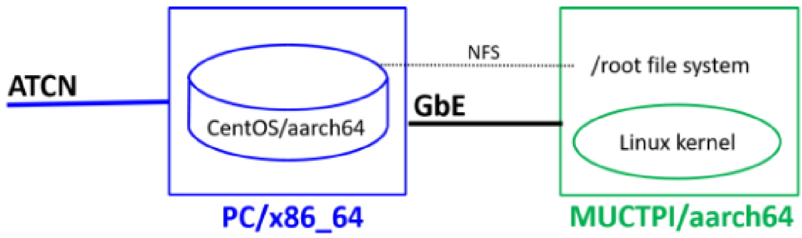

Figure 3.9: Software architecture of the MUCTPI using a cross-installed file system. Source: [52].

The cross-compilation of TDAQ software is done with the `gcc` compiler for aarch64 and the relevant SoC architecture. This allows the development of run control applications that can be executed on the SoC of the MUCTPI. Fig. 3.10 shows a schema representing the architecture of the MUCTPI software.

3.5 MUCTPI commissioning

All the MUCTPI prototypes are fully qualified to be used in the experiment for the LHC Run-3. Currently the MUCTPI v1 is the configuration which is installed in the USA15 cavern. As mentioned in Section 3.2, the MUCTPI v1 will be replaced with the v4 for most of Run-3 [48]. The USA15 cavern is adjacent to the ATLAS detector's cavern, called UX15. See Fig. 3.11 and Fig. 3.12 for a scheme of the caverns underground complex of the ATLAS experiment and a MUCTPI ATCA blade prototype of the same kind which has been installed, respectively.

The MUCTPI is installed in the USA15 cavern since July 2021, but the full population including all the optical links coming from the trigger sectors, electrical connections, networking and patch panel connections was established in the end of August 2021 [49]. See Fig. 3.13 for a reference to the full populated MUCTPI installed in the ATLAS USA15 cavern.

[2] It is also possible to run the Yocto/OpenEmbedded framework.

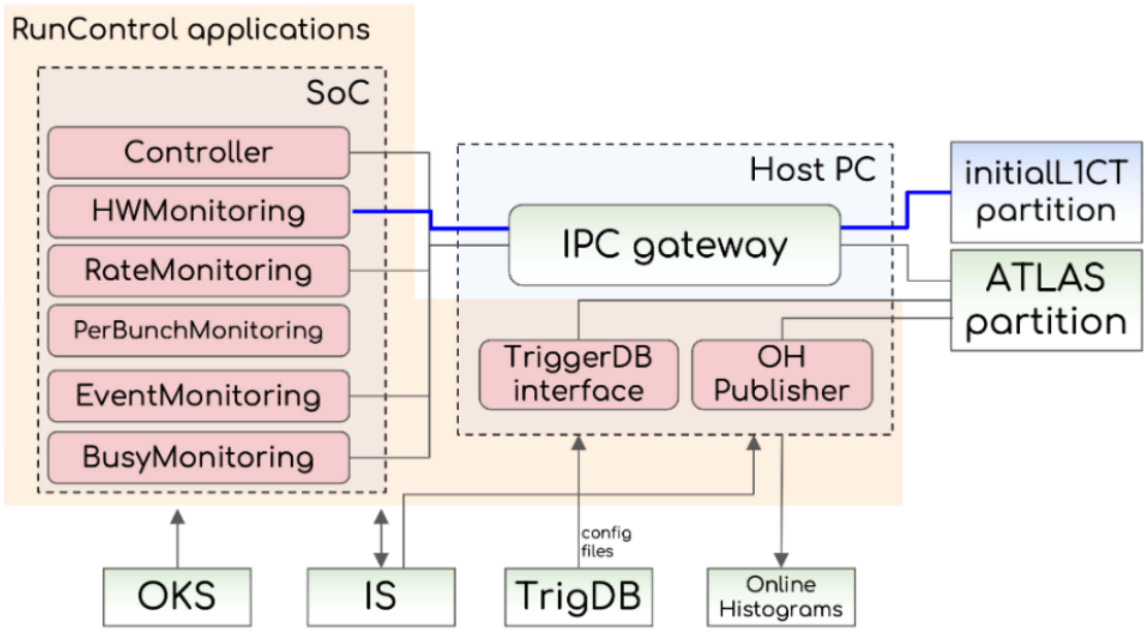

Figure 3.10: Schema of the MUCTPI software with the monitoring applications and the high-level services. In the schema it is also shown the TrigDB service, which provides configuration data. The `initialL1CT` partition is a dedicated partition for the L1CT applications which have to be always running. It is used to run the hardware monitoring application in order to monitor the hardware state of the MUCTPI even when the ATLAS partition is not running. Source: [53].

3.5.1 Firmware and software validation

The validation of the MUCTPI firmware concerning the path of trigger and DAQ relies heavily on Python scripts. It is based on the comparison of MUCTPI output and the output obtained with a simulation. This process takes advantage of playback and snapshot memories installed on the MUCTPI. Playback memories are used for generation of signals that the MUCTPI will process, instead of reading the input from optical links. Playback memories can also be used in the TRP to send TTC patterns and Sector Logic or multiplicities patterns. Snapshot memories have a complementary use, since they capture input data for diagnostic and monitoring purposes. One example describing the process of firmware validation can be summarized with the following steps [49]:

1. A data generator generates files containing RPC and TGC muon data with characteristic patterns.
2. Muon generator files are loaded into MUCTPI playback memories and the inputs are also fed into a bit-wise simulation of the hardware.
3. Snapshot memories record the outputs from the MUCTPI's firmware.
4. MUCTPI's output is compared with the output obtained from the simulation.

Debug activities on the low-level software are done with the help of a menu program. Tests on the MUCTPI software are performed by running a standalone TDAQ partition where one can assess the values read from the FPGAs populating `IS` objects in the ATLAS run control system.

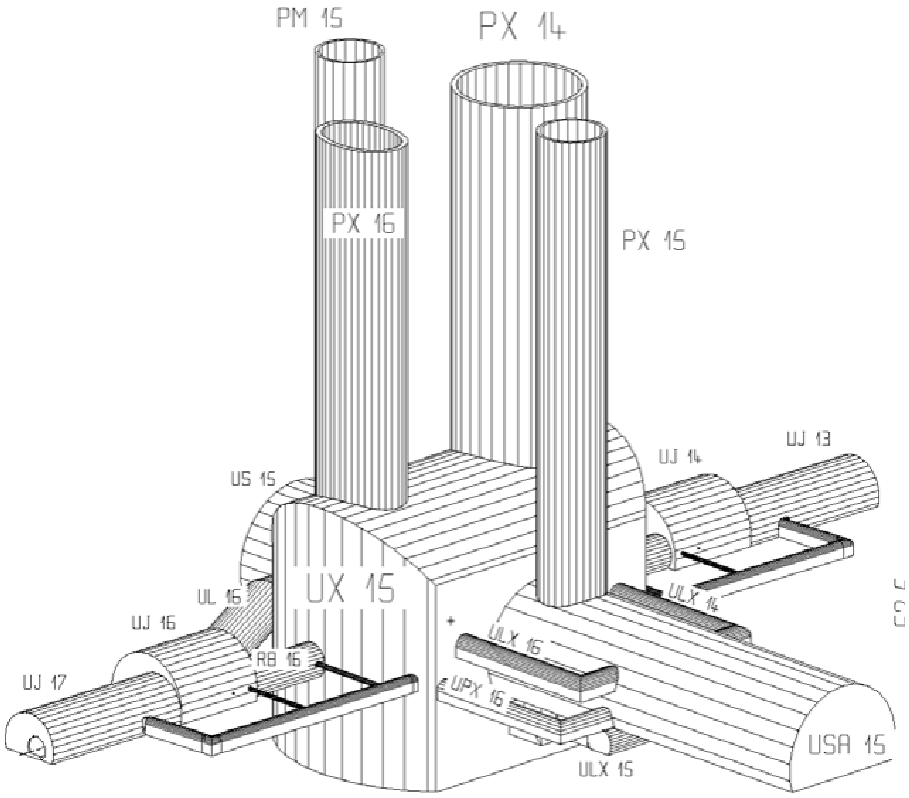

Figure 3.11: The caverns, access shafts and service tunnels surrounding the ATLAS experimental cavern UX15. Source: [54].

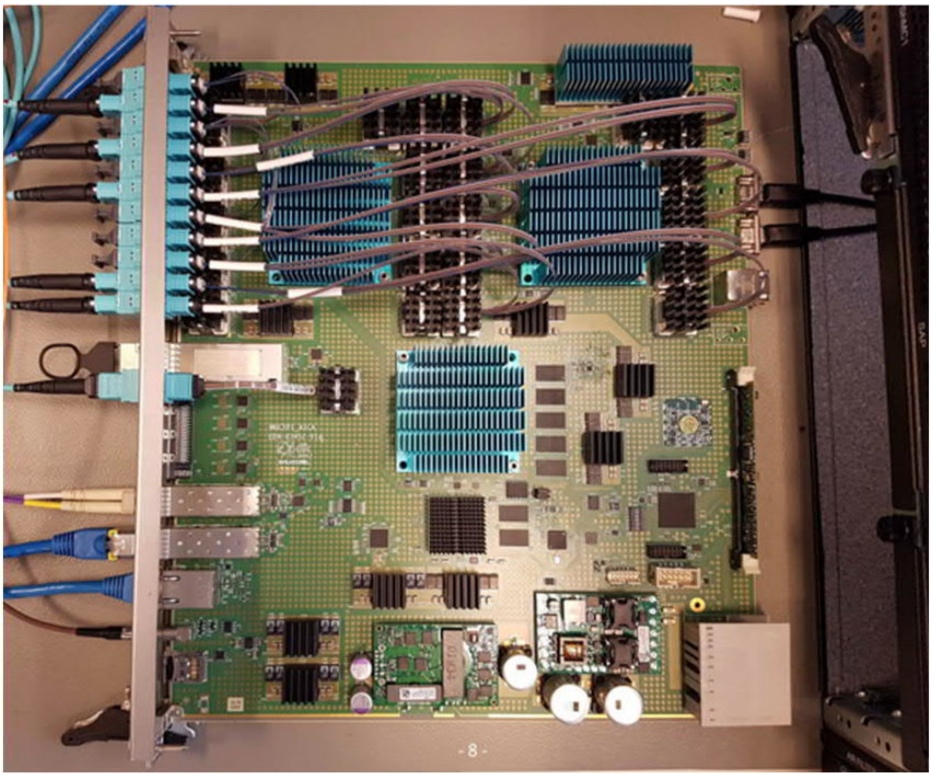

Figure 3.12: MUCTPI ATCA blade prototype. Source: [50].

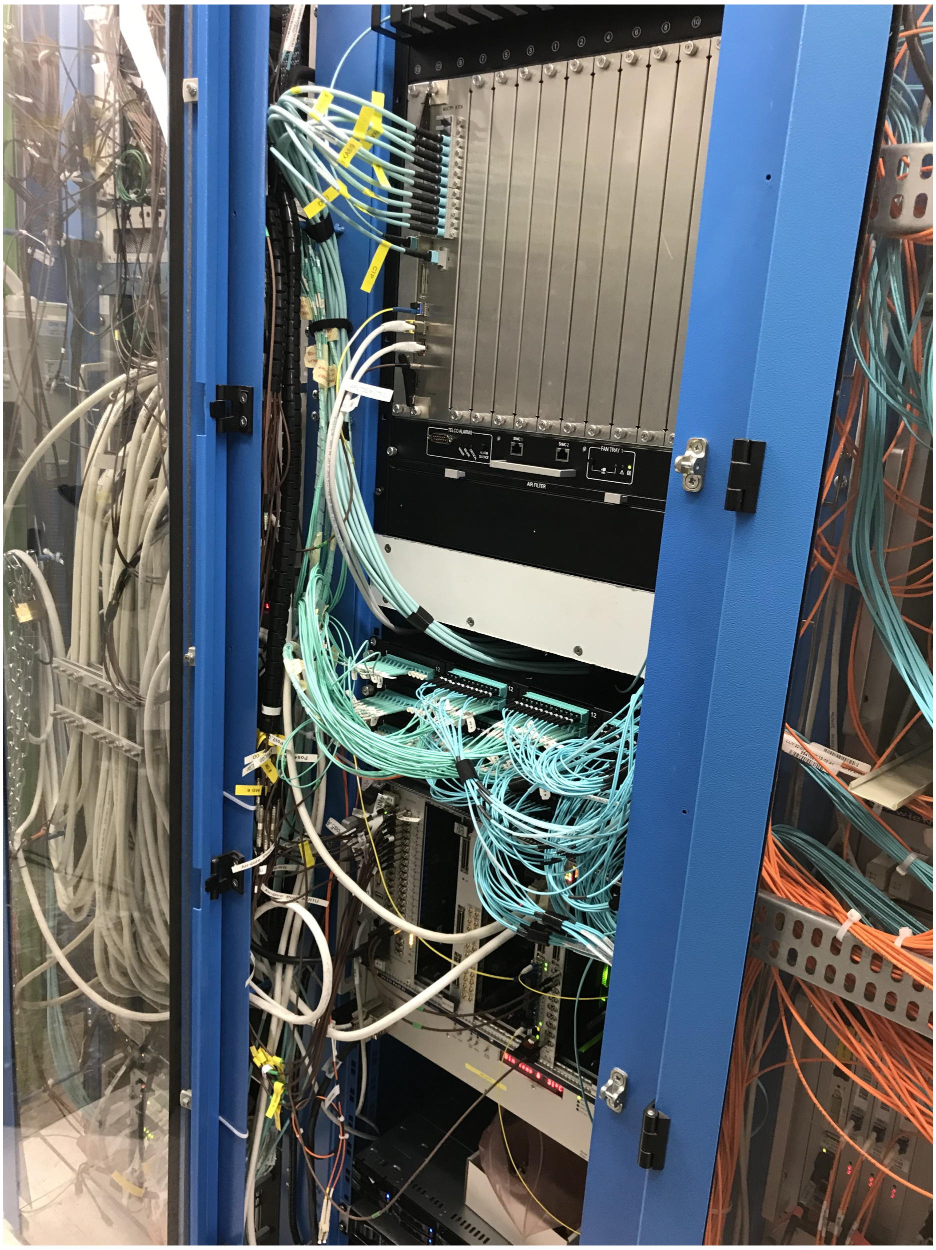

Figure 3.13: MUCTPI fully populated and connected in the ATCA chassis in USA15. All of the input fibers from the muon Sector Logic boards and the output fibers to L1Topo, CTP and the DAQ system are shown. Source: [48].

CHAPTER 4

Online monitoring applications

The monitoring applications developed during the Technical Studentship belong to the same package, called MuctpiMonitoring. The structure of the package and of the applications are derived from the CtpMonitoringHW package, which was developed for the monitoring of the CTP. All the applications are based on the same structure, except for those that manage the publications of histograms in the `OH` service.

The MuctpiMonitoring package also includes other two monitoring applications that were not developed directly by the author, but by other members of the L1CT team. These applications concern the hardware monitoring of the board, displaying voltages, temperatures, power supply currents, optical receiver powers and event monitoring, which samples events for online data validation.

The MuctpiMonitoring package is not a standalone package and it needs some dependencies to compile correctly and function. The dependencies are expressed in terms of linked libraries in the `CMakeLists.txt` file. Some of the dependencies are part of the TDAQ software release and are satisfied by sourcing the cross-compiled TDAQ release with a script file. Other dependencies are the packages which are not part of the TDAQ release, in particular MuctpiModule, MuctpiIS and muctpi-oks.

MuctpiModule provides the API required to use all the low-level software functions in the user applications running on the MUCTPI. MuctpiIS is the package containing all the `IS` schema files required to create the C++ classes of the `IS` objects relevant to the MUCTPI during compilation. muctpi-oks contain all the relevant `OKS` schema files for the MUCTPI. As anticipated in Section 2.2.2 the XML files describe the objects that will be shared between processes using IPC. From the `OKS` description files during compilation time are generated C++ classes which are used in the applications for configuration of parameters. For example in the XML description of the applications is defined the `IS` server that shall be used to publish data and the probeInterval, which is the interval in seconds reciprocal of the monitoring frequency.

The generated executable files will be then deployed to the MUCTPI installed at P1, where the required `OKS` and `IS` schema files for the run control will be included in two dedicated packages for P1 operations.

Now follows an explanation of the general purpose behind every monitoring application of the package, however the code itself will not be discussed here. Any reader interested in the study of the MUCTPI online monitoring applications and in possession of a valid CERN computing account can find the relevant git package in [55].

The general structure of the monitoring applications is based on a controller object which owns a reference to each of the relevant monitoring tasks and executes their methods periodically. The typical monitoring call of a task will require the use of low-level software to read the monitoring data from the FPGA and then to update and publish an `IS` object to the relevant server. In general every application and every counter object has its own `IS` object. Fig. 4.1 shows a screenshot of the `IS` objects published in the relevant server for the MUCTPI online monitoring applications. Section 4.2 to Section 4.5 will give an explanation of each monitoring application developed during the Technical Studentship.

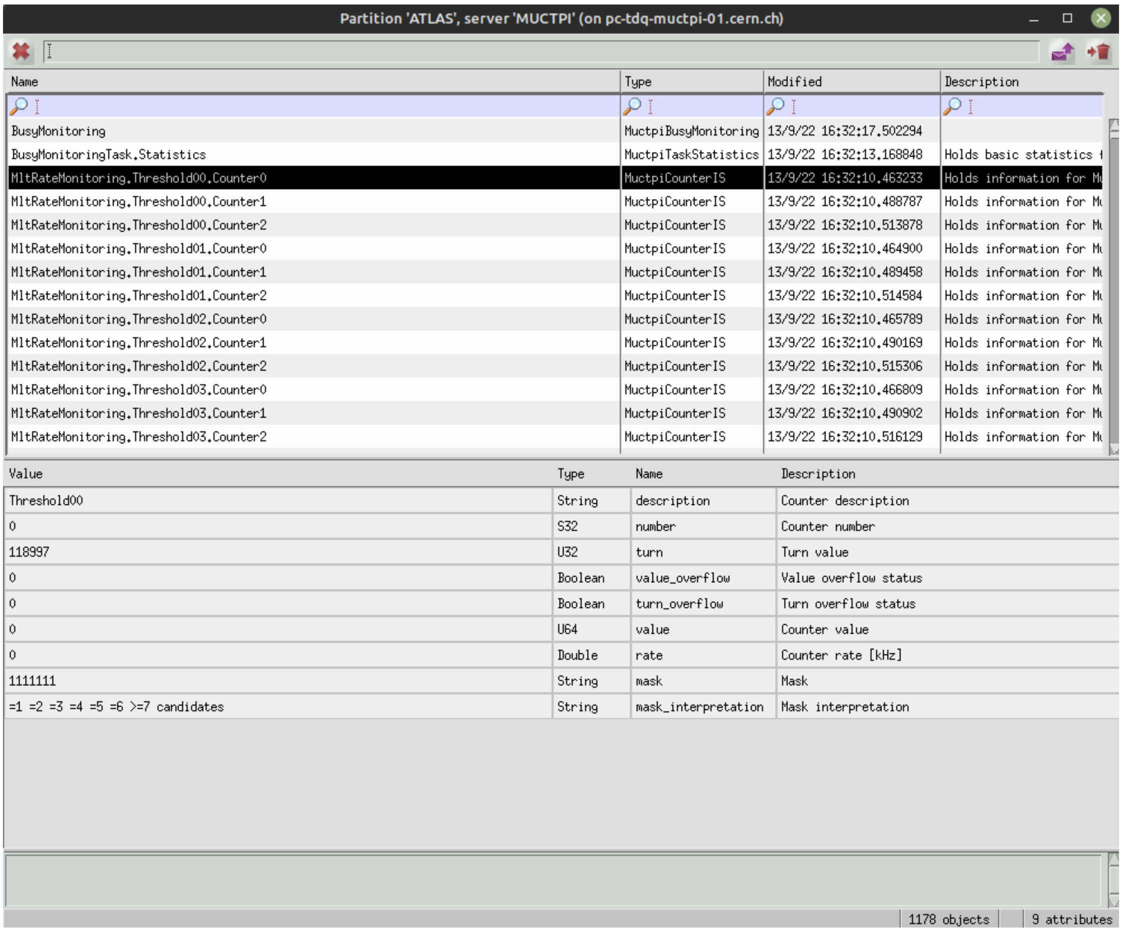

Figure 4.1: Part of MUCTPI `IS` server content running inside ATLAS partition. The highlighed object is a counter used by the monitoring application for responsible to monitor the TRP multiplicity output to the CTP for a given programmable p_T threshold.

The online monitoring activities in the ATLAS Control Room (ACR) are not based on a periodic check in the `IS` monitor. A dedicated package called `WebIS` permits precise HTTP requests for `IS` objects and implementation of web monitoring pages, which are updated with the same cadence of the run control application and have a more readable interface compared to the `IS` monitoring of Fig. 4.1. Section 4.6 contains an explanation of the structure of the web monitoring pages developed during the Technical Studentship.

The web monitoring based on `WebIS` is an instantaneous monitoring which displays the monitoring values at a given time. A second system called Grafana offers the possibility to plot the history of the monitoring values through the use of dashboards. This approach is complementary to the instantaneous monitoring and it is adopted for each object in the MUCTPI IS server.

All the monitoring applications developed are fully integrated in the FSM of the ATLAS partition. A screenshot taken from the IGUI where a TDAQ partition with the MUCTPI segment and the monitoring applications are running can be found in Fig. 4.2.

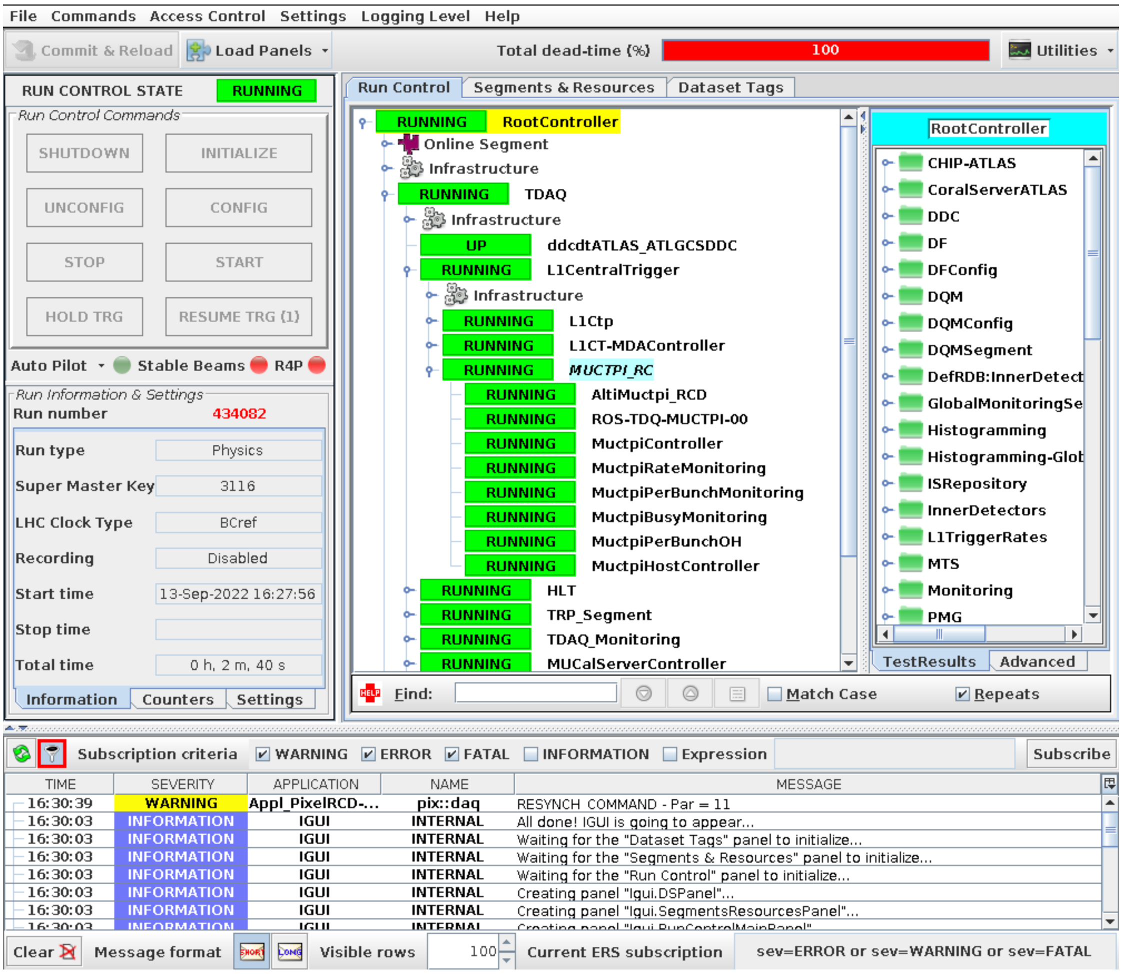

Figure 4.2: MUCTPI monitoring applications running in a IGUI.

4.1 Sector Logic input rate monitoring

The Sector Logic input rate monitoring application monitors the group of four rate counters for each Sector Logic implemented in the firmware shown in Section 3.3.1.1 and their rate. Each counter is published in the relevant IS server. Fig. 4.3 shows an example of Grafana dashboard where one can assess the history of counter rate values for a given Sector Logic.

For this application an additional test was conducted to evaluate the impact of different monitoring

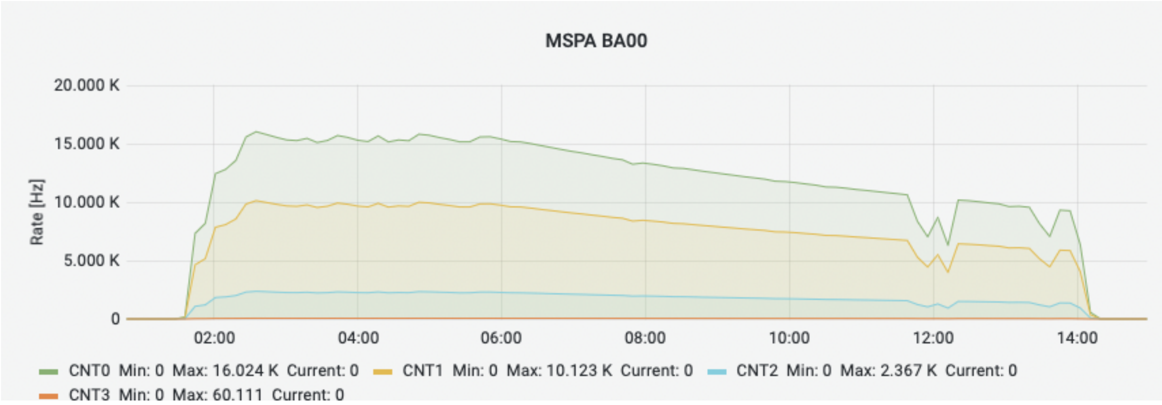

Figure 4.3: Value history for the four counters monitoring the sector MSPA BA00. The screenshot is taken from a Grafana dashboard.

intervals to the load on the SoC processor. The monitoring intervals were chosen between 0.1 and 10 seconds, and for each different monitoring interval the load[1] has been evaluated averaging the output of the Linux command `uptime` over 10 different runs of 15 minutes. Fig. 4.4 shows that the chosen monitoring frequence of 2 seconds is a good choice, since the load spikes when the interval is less than 0.5 seconds.

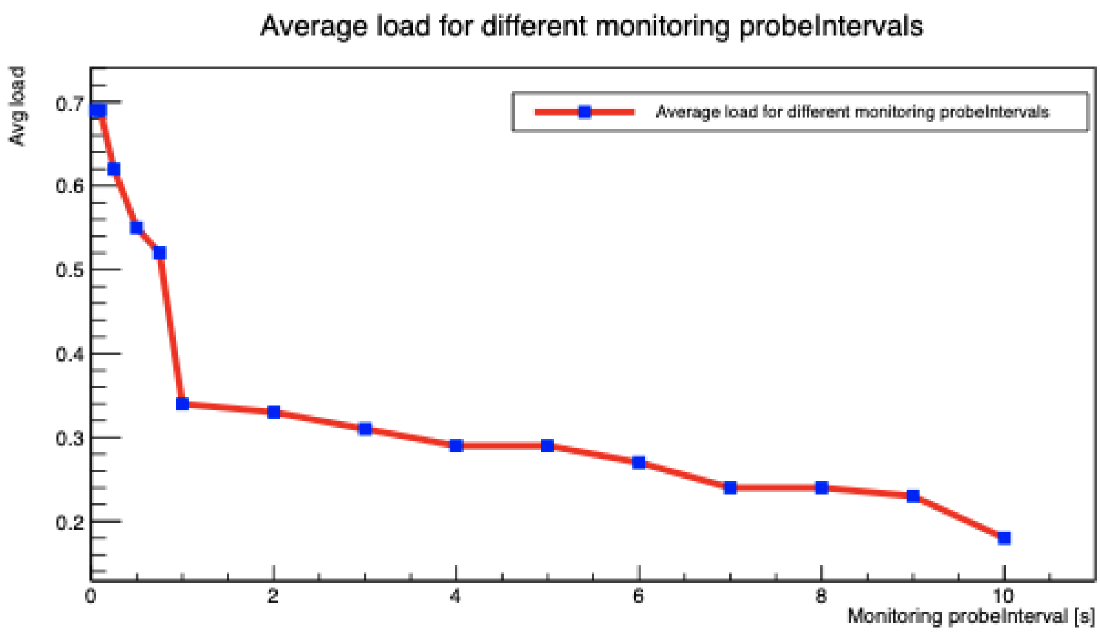

Figure 4.4: Average load on the SoC processor with different monitoring intervals. One should consider that the SoC of MUCTPI v1 has a dual-core processor and therefore the load should be downscaled by at least a factor of 2. For MUCTPI v4 the load will be even reduced by 4, since it is a quad-core system.

[1] The other monitoring applications were also running on the SoC.

4.2 Multiplicity rate monitoring

This application monitors the counters implemented in the firmware shown in Section 3.3.2.1 and their rate. The multiplicity rate monitoring application takes care of the monitoring of three counters for each of 32 multiplicities in the Trigger Processor. It is possible to mask the desired multiplicity for each counter. Every counter is a published object in the relevant `IS` server. Fig. 4.5 shows the plot of the counter rates for a given multiplicity threshold.

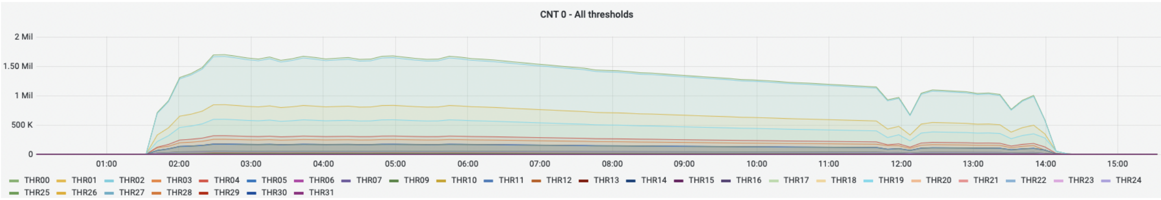

Figure 4.5: Value history for the first counter for each of the multiplicities threshold available. The screenshot is taken from a Grafana dashboard.

4.3 Per-Bunch rate monitoring

Two Per-Bunch rate monitoring applications have been developed, both for Multiplicity and Sector Logic. These applications monitor the rate of the counters per-bunch publishing their 3564 values. The counters published by this application are also used by the Per-bunch OH application of Section 4.5. Per-bunch applications are used to do the timing-in of the Sector Logic inputs.

4.4 Busy monitoring

The Busy monitoring application monitors various busy sources on the MUCTPI as described in Section 3.3.3, publishing objects containing the busy fractions in the `IS` server. In Fig. 4.6 is shown a screenshot of the relevant web monitoring page.

4.5 Per-Bunch OH

This application publishes in `OH` the histograms related to the per-bunch counters for both Multiplicity and Sector Logic rate monitoring. It is used to do the per-bunch timing-in of the Sector Logic inputs. This application is also based on a controller and a task executed periodically, but it does not publish any `IS` information and it does not need to run on the MUCTPI SoC. The Per-bunch OH application runs on a host pc in USA15. The histograms are built and filled using the `ROOT` methods. An example of histogram produced with the Sector Logic per-bunch application is shown in Fig. 4.7.

4.6 Web Monitoring

Web monitoring is used for instantaneous monitoring of data. For this reason new MUCTPI monitoring pages have been implemented on the base of existing web pages, in particular for Sector Logic,

Busy source	Busy fraction [%]
Combined busy sent to LEMO output	0
Software controlled busy on demand	0
Combined DAQ busy for the two MSPs	0
Combined monitoring busy from MSP A/C	100
MSP A monitoring path derandomizer	100
MSP C monitoring path derandomizer	100
MSP A L1A path derandomizer	0
MSP C L1A path derandomizer	0
MSP A link input FIFO	0
MSP C link input FIFO	0
TRP DAQ path busy result	0
DAQ output FIFO	0
DAQ S-LINK link	0
DAQ multiplicity FIFO	0

Figure 4.6: Screenshot of the busy web monitoring with all the busy sources and their relevant busyness fractions.

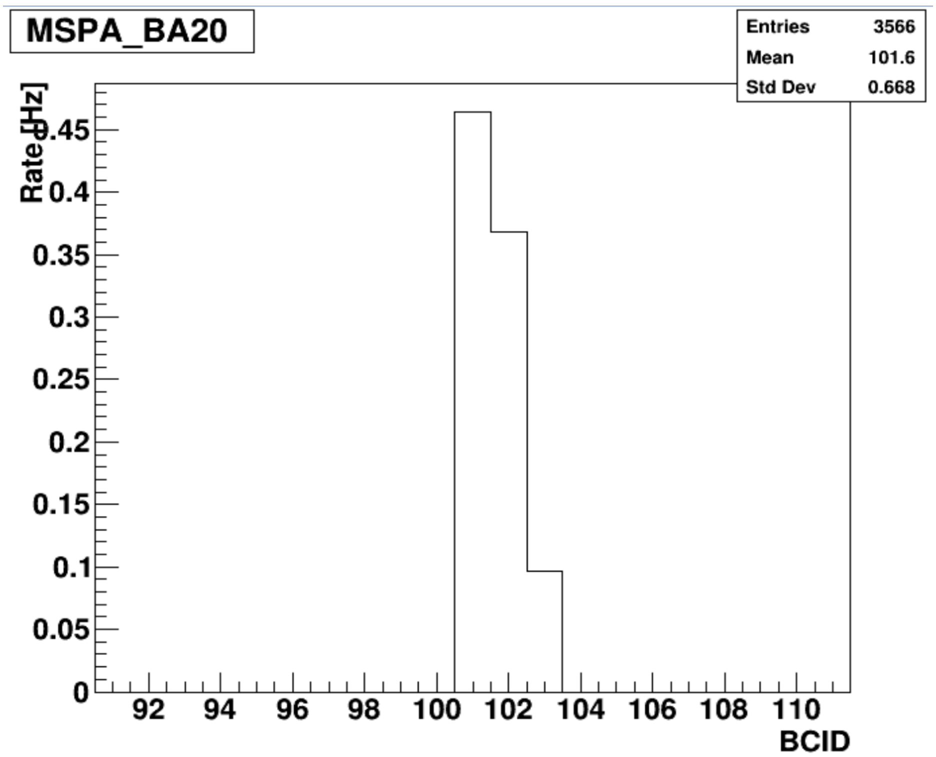

Figure 4.7: `OH` histogram for the sector MSPA BA20.

multiplicity and busy monitoring. Every monitoring page is provided with a table where it is possible to read the values of the monitored counters, but also filter and sort them. The `WebIS` service can be used with JavaScript code in a webpage to perform HTTP requests of `IS` or `OH` objects, the response is a XML file containing all the information contained in the `IS` object and a png image of the selected histogram. Fig. 4.8 shows an excerpt of a screenshot of the multiplicity rates table.

		Counter 1		Counter 2		Counter 3	
Threshold #	Threshold name	Rate [Hz]	Multiplicity	Rate [Hz]	Multiplicity	Rate [Hz]	Multiplicity
0	Threshold00	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7
1	Threshold01	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7
2	Threshold02	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7
3	Threshold03	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7
4	Threshold04	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7
5	Threshold05	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7
6	Threshold06	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7
7	Threshold07	14.0	=1 =2 =3 =4 =5 =6 >=7	14.0	=1 =2 =3 =4 =5 =6 >=7	14.0	=1 =2 =3 =4 =5 =6 >=7
8	Threshold08	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7
9	Threshold09	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7	40079000.0	=1 =2 =3 =4 =5 =6 >=7
10	Threshold10		=1 =2 =3 =4 =5 =6 >=7		=1 =2 =3 =4 =5 =6 >=7		=1 =2 =3 =4 =5 =6 >=7
11	Threshold11	0.094	=1 =2 =3 =4 =5 =6 >=7	0.094	=1 =2 =3 =4 =5 =6 >=7	0.094	=1 =2 =3 =4 =5 =6 >=7
12	Threshold12		=1 =2 =3 =4 =5 =6 >=7		=1 =2 =3 =4 =5 =6 >=7		=0
13	Threshold13		=0		=0		=0
14	Threshold14		=0		=0		=0
15	Threshold15		=0		=0		=0

Figure 4.8: Screenshot of part of the multiplicity rates table, only the first 16 multiplicity thresholds are shown. All the three counters are shown in their relevant columns. The specific p_T value of the thresholds was not defined. The screenshot was taken when the LHC was not running and therefore the values shown are due to cosmic muons and noise.

The Sector Logic web monitoring page has been provided with a dynamic drawing of the sectors covering the ATLAS detector. The JavaScript code reads periodically all the values from the first counter of each sector and paints the drawing according to a programmable temperature scale. An example of usage of this web application was during a special LHC run with a 900 GeV stable beam that happened during the week from May 31st to June 6th 2022. Horizontal muons were produced by the LHC moving collimators in the beam pipe. The MUCTPI was required to be configured to trigger on the TGC Sector Logic inputs from only one side at a time, as shown in the screenshots of Fig. 4.9 and Fig. 4.10.

The Sector Logic monitoring page has also been provided with a 2D histogram as a combination of all the per-bunch histograms. On the y axis it is shown the sector number from 0 to 207, considering BA, EC and FW sectors from side A and then side C. On x axis the bunch number and the rate value

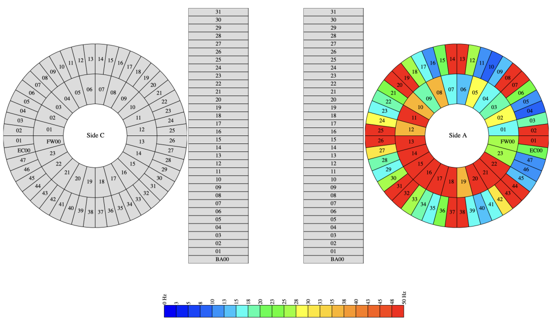

Figure 4.9: Screenshot of the web monitoring showing the rates with a temperature scale for the end-cap and forward sectors of side A.

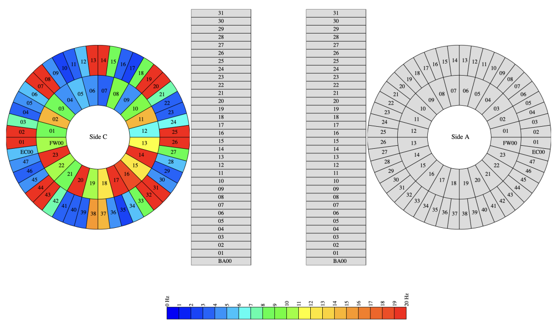

Figure 4.10: Screenshot of the web monitoring showing the rates with a temperature scale for the end-cap and forward sectors of side C.

is expressed in terms of a color map. Fig. 4.11 shows an example of histogram of this kind.

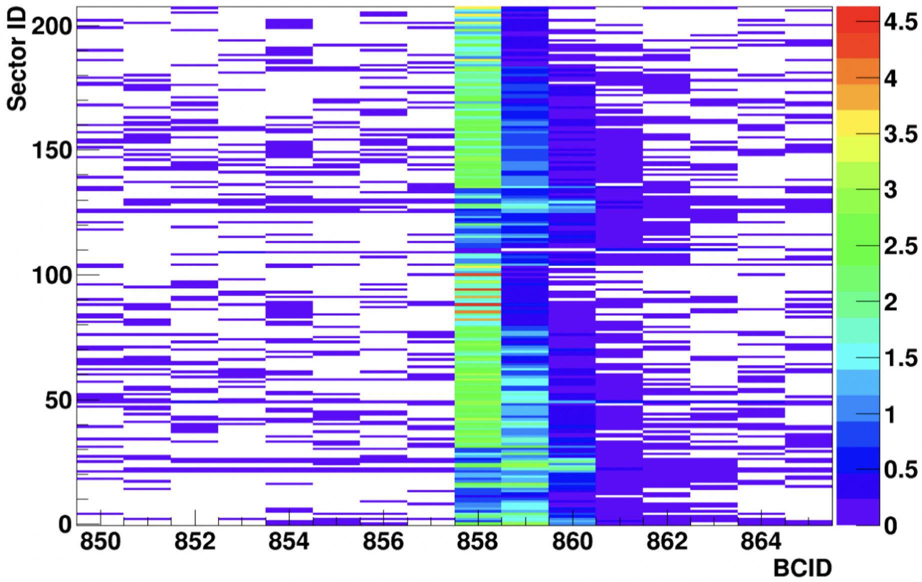

Figure 4.11: 2D per-bunch Sector Logic histograms.

CHAPTER 5

Conclusions

This work explained the major Phase-I upgrades on the MUCTPI, which is an electronic board in the ATLAS L1-Trigger that was completely re-designed to move from 18 9U VME modules to a single ATCA blade. The MUCTPI has a crucial role in the L1-Trigger, being the component that sends the multiplicity of muons per event for various programmable transverse momentum thresholds to the CTP. The MUCTPI upgrade required the development of new hardware, firmware and software.

This thesis focused on the development of monitoring applications for the MUCTPI, explaining how they are based on the low-level software and how the basic high-level services offered by the TDAQ online software provide the infrastructure to integrate the applications in the ATLAS run control framework. A number of integration tests have been performed on the software using the IGUI and a menu program. The beginning of the Run-3 on the 5^{th} July 2022 demonstrated the succesful execution of the applications and of the web monitoring pages in the ATLAS Control Room.

For the next High Luminosity LHC Run-4 starting in 2029 the TDAQ system will undergo a number of other upgrades, also known as Phase-II upgrades, in order to cope with the tenfold increase of data due to the high luminosity. This will require a fundamental upgrade of the ATLAS TDAQ system currently in place for LHC Run-3. For the Phase-II upgrades for the MUCTPI, it is planned to continue to use the same hardware of the Phase-I upgrade, but with new firmware which can take into account the new requirements. In order to handle the increased bandwidth requirements, two MUCTPI blades might be used, one per detector hemisphere [36]. The fact that the hardware of Run-3 might also be reused also for Run-4 means it may be possible to reuse the code of the monitoring applications explained in this thesis as a starting point for future applications.

Bibliography

[1] ATLAS Collaboration,
Technical Design Report for the Phase-I Upgrade of the ATLAS TDAQ System, tech. rep., 2013,
URL: https://cds.cern.ch/record/1602235 (cit. on pp. 1, 27, 28).

[2] L. Dell'Asta, *The ATLAS muon and tau triggers*, tech. rep., CERN, 2013,
URL: https://cds.cern.ch/record/1604500 (cit. on p. 1).

[3] Wikimedia Commons, *File:Standard Model of Elementary Particles.svg — Wikimedia Commons, the free media repository*, [Online; accessed 1-March-2022], 2022,
URL: https://commons.wikimedia.org/w/index.php?title=File:Standard_Model_of_Elementary_Particles.svg (cit. on p. 4).

[4] J. Ellis, *Physics Beyond the Standard Model*, Nuclear Physics A **827** (2009) 187c,
ISSN: 0375-9474, URL: http://dx.doi.org/10.1016/j.nuclphysa.2009.05.034
(cit. on p. 7).

[5] L. R. Evans and P. Bryant, *LHC Machine*, JINST **3** (2008) S08001. 164 p, This report is an abridged version of the LHC Design Report (CERN-2004-003),
URL: http://cds.cern.ch/record/1129806 (cit. on p. 7).

[6] S. Myers, *The LEP Collider, from design to approval and commissioning*,
Geneva: CERN, 1991, URL: http://cds.cern.ch/record/226776 (cit. on p. 7).

[7] Wikimedia Commons,
File:Cern-accelerator-complex.svg — Wikimedia Commons, the free media repository,
[Online; accessed 1-March-2022], 2021,
URL: https://commons.wikimedia.org/w/index.php?title=File:Cern-accelerator-complex.svg (cit. on p. 8).

[8] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*,
JINST **3** (2008) S08003. 437 p, URL: https://cds.cern.ch/record/1129811
(cit. on pp. 9–11, 13, 16, 18, 19, 21, 27).

[9] CMS Collaboration,
The CMS experiment at the CERN LHC. The Compact Muon Solenoid experiment,
JINST **3** (2008) S08004. 531 p, URL: https://cds.cern.ch/record/1129810
(cit. on p. 9).

[10] LHCb Collaboration, *The LHCb Detector at the LHC*, JINST **3** (2008) S08005,
URL: https://cds.cern.ch/record/1129809 (cit. on p. 9).

[11] ALICE Collaboration,
The ALICE experiment at the CERN LHC. A Large Ion Collider Experiment,
JINST **3** (2008) S08002. 259 p, URL: https://cds.cern.ch/record/1129812
(cit. on p. 9).

[12] J.J. Goodson, *Search for Supersymmetry in States with Large Missing Transverse Momentum and Three Leptons including a Z-Boson*, Presented 17 Apr 2012,
PhD thesis: Stony Brook University, 2012 (cit. on pp. 10, 12).

[13] M. Schott, G. Duckeck and M. Obermaier, *Study of the Sagitta Resolution of MDT-Chambers*, tech. rep., CERN, 2006, URL: http://cds.cern.ch/record/934494 (cit. on pp. 11, 12).

[14] ATLAS Collaboration, *ATLAS muon spectrometer: Technical Design Report*, tech. rep., 1997, URL: https://cds.cern.ch/record/331068 (cit. on pp. 12, 13, 19, 20).

[15] ATLAS Collaboration, *Study of the material of the ATLAS inner detector for Run 2 of the LHC*, Journal of Instrumentation **12** (2017) P12009, ISSN: 1748-0221,
URL: http://dx.doi.org/10.1088/1748-0221/12/12/P12009 (cit. on p. 14).

[16] A. Vogel,
ATLAS Transition Radiation Tracker (TRT): Straw Tube Gaseous Detectors at High Rates,
tech. rep., CERN, 2013, URL: https://cds.cern.ch/record/1537991 (cit. on p. 13).

[17] J. Pequenao, "Computer Generated image of the ATLAS calorimeter", 2008,
URL: https://cds.cern.ch/record/1095927 (cit. on p. 15).

[18] J. del Peso, *Design and Performance of the ATLAS LAr Calorimeter*, tech. rep., CERN, 2011, URL: http://cds.cern.ch/record/1385899 (cit. on p. 15).

[19] B. Sotto-Maior Peralva, *Calibration and Performance of the ATLAS Tile Calorimeter*,
tech. rep., 2013, arXiv: 1305.0550, URL: https://cds.cern.ch/record/1545496
(cit. on p. 17).

[20] A. Salvucci, *Measurement of muon momentum resolution of the ATLAS detector*,
EPJ Web of Conferences **28** (2012) (cit. on p. 17).

[21] ATLAS Collaboration, *Standalone Vertex Finding in the ATLAS Muon Spectrometer*,
JINST **9** (2013) P02001. 22 p, arXiv: 1311.7070,
URL: https://cds.cern.ch/record/1631701 (cit. on p. 18).

[22] ATLAS Collaboration,
Performance of the ATLAS RPC detector and Level-1 muon barrel trigger at $\sqrt{s} = 13$ *TeV. Performance of the ATLAS RPC detector and Level-1 muon barrel trigger at* $\sqrt{s} = 13$ *TeV*,
JINST **16** (2021) P07029. 63 p, arXiv: 2103.01029,
URL: https://cds.cern.ch/record/2753039 (cit. on p. 19).

[23] F. Mei, *Studio delle prestazioni attese di un rivelatore 'Resistive Plate Chamber' (RPC) tramite simulazione col software Garfield*,
MA thesis: Alma Mater Studiorum - Bologna University, 2021 (cit. on p. 19).

[24] E. Perez Codina,
Small-Strip Thin Gap Chambers for the Muon Spectrometer Upgrade of the ATLAS Experiment,
tech. rep., CERN, 2015, URL: https://cds.cern.ch/record/2027546 (cit. on p. 19).

[25] J. Wotschack,
ATLAS Muon Chamber Construction Parameters for CSC, MDT, and RPC chambers, tech. rep., Back-up document for the ATLAS Detector Paper: CERN, 2008,
URL: `https://cds.cern.ch/record/1099400` (cit. on p. 20).

[26] J. Dubbert et al., *Performance of the ATLAS Muon Drift-Tube Chambers at High Background Rates and in Magnetic Fields*, tech. rep., 2016, arXiv: `1604.01598`,
URL: `https://cds.cern.ch/record/2144441` (cit. on p. 20).

[27] A. Mizukami, *ATLAS Level-1 Endcap Muon Trigger for Run 3*, tech. rep., CERN, 2020,
URL: `http://cds.cern.ch/record/2709553` (cit. on pp. 21, 28).

[28] V. Cindro et al., *The ATLAS Beam Conditions Monitor*,
Journal of Instrumentation **3** (2008) P02004,
URL: `https://doi.org/10.1088/1748-0221/3/02/p02004` (cit. on p. 22).

[29] C. Ohm and T. Pauly, *The ATLAS beam pick-up based timing system*,
Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **623** (2010) 558, ISSN: 0168-9002,
URL: `https://www.sciencedirect.com/science/article/pii/S0168900210006339`
(cit. on p. 22).

[30] ATLAS Collaboration, *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, Physics Letters B **716** (2012) 1,
ISSN: 0370-2693 (cit. on p. 23).

[31] CMS Collaboration,
Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC,
Physics Letters B **716** (2012) 30, ISSN: 0370-2693 (cit. on p. 23).

[32] ATLAS Collaboration,
A search for the dimuon decay of the Standard Model Higgs boson with the ATLAS detector,
Physics Letters B **812** (2021) 135980, ISSN: 0370-2693,
URL: `http://dx.doi.org/10.1016/j.physletb.2020.135980` (cit. on p. 23).

[33] W. Stirling, private communication (cit. on p. 24).

[34] M. Leone and N. Robotti, *P M S Blackett, G Occhialini and the invention of the counter-controlled cloud chamber (1931–32)*, European Journal of Physics **29** (2008) 177
(cit. on p. 25).

[35] P. Borrego-Amaral et al., *The ATLAS Level-1 Central Trigger System 012*, tech. rep., CERN, 2004, URL: `https://cds.cern.ch/record/795015` (cit. on p. 25).

[36] ATLAS Collaboration,
Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System, tech. rep., CERN, 2017, URL: `https://cds.cern.ch/record/2285584` (cit. on pp. 26, 59).

[37] M. Weirich et al., *Development of the jet Feature EXtractor (jFEX) for the ATLAS Level 1 calorimeter trigger upgrade at the LHC*, tech. rep., CERN, 2017,
URL: `https://cds.cern.ch/record/2289210` (cit. on p. 27).

[38] ATLAS Collaboration,
ATLAS high-level trigger, data-acquisition and controls: Technical Design Report, tech. rep., 2003, URL: https://cds.cern.ch/record/616089 (cit. on pp. 33–35).

[39] C. Solans Sanchez, J. Ferrer and E. Higon-Rodriguez,
Implementation of the ROD Crate DAQ Software for the ATLAS Tile Calorimeter and a Search for a MSSM Higgs Boson decaying into Tau pairs, (2010) (cit. on p. 35).

[40] A. Kazarov, M. Caprini, R. Hauser and I. Soloviev,
Software releases management in the trigger and data acquisition of ATLAS experiment, tech. rep., CERN, 2011, URL: http://cds.cern.ch/record/1337261 (cit. on p. 34).

[41] *GitLab website*, [Online; accessed 1-Sept-2022], URL: https://about.gitlab.com/ (cit. on p. 35).

[42] *CMake website*, [Online; accessed 1-Sept-2022], URL: https://cmake.org/ (cit. on p. 35).

[43] S. Kolos et al., *Experience with CORBA communication middleware in the ATLAS DAQ.*, (2005) 6 p, URL: https://cds.cern.ch/record/796515 (cit. on pp. 35, 36).

[44] S. Kolos, *Information Service user's guide*, 2011, URL: https://atlas-tdaq-monitoring.web.cern.ch/IS/doc/userguide/is-usersguide.pdf (cit. on p. 36).

[45] S. Kolos, A. Kazarov and L. Papaevgeniou, *The Error Reporting in the ATLAS TDAQ System*, tech. rep., CERN, 2014, URL: https://cds.cern.ch/record/1950302 (cit. on p. 37).

[46] I. Soloviev, G. Avolio, A. Kazymov and M. E. Vasile,
ATLAS Operational Monitoring Data Archival and Visualization, tech. rep., CERN, 2020, URL: https://cds.cern.ch/record/2710109 (cit. on p. 38).

[47] *Grafana website*, [Online; accessed 1-Sept-2022], URL: https://grafana.com/ (cit. on p. 38).

[48] A. Marzin, private communication (cit. on pp. 40, 46, 50).

[49] S. Perrella et al., *Integration and commissioning of the ATLAS Muon-to-Central-Trigger-Processor Interface for Run-3*, Journal of Instrumentation **17** (2022) C04006, URL: https://doi.org/10.1088/1748-0221/17/04/c04006 (cit. on pp. 40, 42, 46, 47).

[50] R. Spiwoks et al., *CentOS Linux for the ATLAS MUCTPI Upgrade*, tech. rep., Presented at IEEE RT 2020, 12-23 October 2020: CERN, 2021, arXiv: 2010.08105, URL: https://cds.cern.ch/record/2741337 (cit. on pp. 41, 49).

[51] R. Spiwoks et al., *Continuous Integration for the Software and the Firmware of the new ATLAS Muon-Central-Trigger-Processor Interface (MUCTPI)*, tech. rep., CERN, 2022, URL: https://cds.cern.ch/record/2823852 (cit. on pp. 40, 41).

[52] R. Spiwoks et al., *CentOS Linux for the ATLAS MUCTPI Upgrade*, tech. rep., Presented at IEEE RT 2020, 12-23 October 2020: CERN, 2021, arXiv: 2010.08105, URL: https://cds.cern.ch/record/2741337 (cit. on p. 46).

[53] Y. Afik and A. T. Collaboration,
The new Muon-to-Central-Trigger-Processor Interface at ATLAS, (2022), URL: https://cds.cern.ch/record/2823861 (cit. on p. 47).

[54] I. Dawson and V. Hedberg, *Radiation in the USA15 cavern in ATLAS*, tech. rep., CERN, 2004, URL: http://cds.cern.ch/record/734133 (cit. on p. 48).

[55] L1CT group, *MuctpiMonitoring git package*, [Online; accessed 1-June-2022], URL: https://gitlab.cern.ch/atlas-l1ct/MuctpiMonitoring (cit. on p. 51).