Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

FISICA

Ciclo 35

Settore Concorsuale: 02/A1 - FISICA SPERIMENTALE DELLE INTERAZIONI FONDAMENTALI

Settore Scientifico Disciplinare: FIS/01 - FISICA SPERIMENTALE

TEST VECTOR GENERATION FOR THE PHASE II ATLAS EVENT FILTER TRIGGER UPGRADE

Presentata da: Francesca Del Corso

Coordinatore Dottorato

Michele Cicoli

Supervisore

Alessandro Gabrielli

Esame finale anno 2023

We have seen that programming is an art,
because it requires knowledge, application, skill and ingenuity,
but above all for the beauty of the objects it produces.
— Donald E. Knuth [1973]

Abstract

In the near future, the LHC experiments will continue to be upgraded as the LHC luminosity will increase from the design 10^{34} to 7.5×10^{34}, with the HL-LHC project, to reach $3000 \times fb^{-1}$ of accumulated statistics. After the end of a period of data collection, CERN will face a long shutdown to improve overall performance by upgrading the experiments and implementing more advanced technologies and infrastructures. In particular, ATLAS will upgrade parts of the detector, the trigger, and the data acquisition system. It will also implement new strategies and algorithms for processing and transferring the data to the final storage. This PhD thesis presents a study of a new pattern recognition algorithm to be used in the trigger system, which is a software designed to provide the information necessary to select physical events from background data. The idea is to use the well-known Hough Transform mathematical formula as an algorithm for detecting particle trajectories. The effectiveness of the algorithm has already been validated in the past, independently of particle physics applications, to detect generic shapes in images. Here, a software emulation tool is proposed for the hardware implementation of the Hough Transform, to reconstruct the tracks in the ATLAS Trigger and Data Acquisition system. Until now, it has never been implemented on electronics in particle physics experiments, and as a hardware implementation it would provide overall latency benefits. A comparison between the simulated data and the physical system was performed on a Xilinx UltraScale+ FPGA device.

Contents

List of Figures

List of Tables

Introduction

The incoming High-Luminosity upgrade of the Large Hadron Collider (HL-LHC) at CERN in Geneva, Switzerland, will contribute to the study, with a significantly improved sensitivity, of known mechanisms expected from the theory of the Standard Model (SM), as well as new rarer processes which may be indicative of Physics beyond the SM. The ATLAS (A Toroidal LHC ApparatuS) Phase-II upgrade of the Trigger and Data Acquisition system (TDAQ) will enable the study of the mechanism of electroweak symmetry breaking through the properties of the Higgs boson, the search for new physics by studying rare processes of the SM, the search for new heavy states, measurements of the properties of any newly discovered particles. The HL-LHC is expected to start installation in 2026 and Run 5 is expected to start in 2032, reaching a maximum instantaneous luminosity of $\mathcal{L} = 7.5 \times 10^{34}\ cm^{-2}\ s^{-1}$, corresponding to about 200 inelastic proton-proton collisions per bunch crossing. The increase in luminosity will provide larger data sets. The new Inner Tracker (ITk) will be operational for more than 10 years, during which time ATLAS aims to accumulate a total data set of up to $4000 fb^{-1}$. Meeting these requirements will be a major challenge for the ATLAS TDAQ to fully exploit the physics potential of the HL-LHC. The trigger and the readout electronics currently in use will need to be upgraded, new strategies for data acquisition and processing, algorithms for data management and transmission to the final storage devices will be required.

Recently, new tracking capabilities have been investigated in High Energy Physics (HEP) applications using an advanced technique based on the Hough Transform (HT) on Field-Programmable Gate Array (FPGA) devices, characterised by a fixed low latency. HT is a well-known extraction technique mainly used in image analysis and digital processing; the advantage of using HT in FPGA is that the latency increases linearly with

respect to the quantity of input data, making it suitable for a large input data set.

Some hardware solutions based on Associative Memories (AM) Application Specific Integrated Circuits (ASICs) for pattern matching in the ATLAS Phase-II Hardware Tracking for the Trigger (HTT) upgrade have been investigated for years. HTT was the subsystem designed to provide fast hardware-based track reconstruction in the ATLAS TDAQ, later replaced by the Event Filter (EF) system. The team I worked with for my PhD was involved in investigating alternative solutions, in particular the possibility of implementing the HT in pattern recognition problems on a commodity device targeting the latest frontier FPGA. The Xilinx Ultrascale+ FPGAs were considered.

This thesis describes part of my three-year experience as a Ph.D. student at the University of Bologna (IT) related to the ATLAS experiment, and the ATLAS Qualification Task was also linked to this. The field of application is software development, as it involved the project of a software tool that emulates the FPGA-based firmware design planned by the team I work for.

The tool is based on the HT and can recognize patterns of straight lines that represent possible tracks of ionising particles in high-energy physics. It can distinguish between tracks in a noisy background, regardless of the amount of noise. In addition, the software tool is parameterised to be as generic as possible. The ability to detect straight lines may be adapted to other shape recognition tasks and may be useful in other future applications.

Simulations and performance comparisons are performed to study the system behaviour and demonstrate that the entire system could be well implemented on an FPGA device, avoiding the high cost of custom hardware implementation.

This thesis is divided into 5 chapters: Chapter 1 gives an introduction to CERN and the LHC; Chapter 2 describes the ATLAS detector and its infrastructure; Chapter 3 deals with the ATLAS Phase-II upgrade for the HL-LHC, with a focus on the TDAQ and the EF tracking decision process; Chapter 4 gives an overview of the HT algorithm and its implementation in the context of the charged particle track reconstruction from the ATLAS ITk detector for the HL-LHC; Chapter 5 presents my personal contributions to the EF tracking project. The final part contains a conclusion, Appendix A with an overview of the HTT project from which my study is born, Appendix B with the

developed code, a list of acronyms, acknowledgements, and a bibliography.

Chapter 1

CERN and LHC

The European Council for Nuclear Research (CERN), in French "Conseil Européen pour la Recherche Nucléaire", is a European research organization that operates the world's largest particle physics laboratory. Founded in 1952 on the Franco–Swiss border near Geneva, its particle accelerators and other high-energy physics research infrastructure are used by more than 7,000 scientists from over 60 countries to study the fundamental structure of the universe. Many experiments have been built at CERN in international collaborations, including the Large Hadron Collider (LHC), the largest scientific instrument ever designed and built for scientific research. Since 2010, it has been exploring the new high-energy frontier, with the aim of establishing a fundamental physics research organization in Europe.

1.1 Large Hadron Collider

The Large Hadron Collider (LHC) is a major worldwide collaborative scientific project, and many activities at CERN currently involve the LHC and its experiments. It began with the aim of designing a high energy physics collider capable of investigating the nature of electroweak symmetry breaking and the search for physics beyond the SM at the TeV scale. Its realization was approved by the CERN Council in December 1994 [1] and during its years of operation it achieved important results, such as the discovery of the Higgs boson in 2012 [2], whose properties are being studied continuously to confirm

the predictions of the SM and to search for new physics.

The LHC tunnel is located 100 m underground, near Geneva (Fig. 1.1). It is a 27.6 km circular ring previously occupied by the Large Electron–Positron Collider (LEP), which was shut down in November 2000. It consists of superconducting magnets with a series of accelerating structures to boost the energy of the particles along the way. Unlike previous particle-antiparticle colliders, in which both beams share the same beam pipe in a single ring, the LHC experiment is based on a proton-proton (pp) collision. The two counter-rotating proton beams are currently accelerated to a center-of-mass energy of $\sqrt{s} = 13\ TeV$, which will be increased to $\sqrt{s} = 14\ TeV$ in the next run upgrade. The LHC will also collide heavy ions, in particular lead nuclei, at 5.5 TeV per nucleon pair.

To reach these energies, the beam has to pass through several acceleration stages, as shown in Fig. 1.2.

Protons are created by stripping the electrons from hydrogen atoms taken from a bottle of hydrogen gas, then they are injected into the LINAC2 linear accelerator where their energy is increased to 50 MeV. In the circular Proton Synchrotron Booster (PSB) they are accelerated to 1.4 GeV and in the Proton Synchrotron (PS) they reach 25 GeV. This is followed by the Super Proton Synchrotron (SPS), where they reach an energy of 450 GeV. In the final step, the protons are transferred to the LHC where each beam is accelerated to 6.5 TeV. For the heavy ions, instead, a linear accelerator called LINAC3 takes the lead ions at an energy of 4.5 MeV/n and a Low-Energy Ion Ring (LEIR) accelerates them to 72 MeV/n. They then enter the SPS and follow the same path as the protons before entering the LHC, reaching an initial energy of 5.9 GeV/n and then of 177 GeV/n. In the final run, the LHC accelerates the lead ions to 1.38 TeV/n.

In a series of radio-frequency (RF) cavities, the protons are accelerated and grouped into bunches. This phenomenon occurs when a charged particle is accelerated in a circular collider, causing an emission of electromagnetic radiation and a corresponding loss of energy. The RF cavities focus the proton bunches along the beam pipe. The proton beams are kept on a circular trajectory by a set of 1232 superconducting dipole magnets made of copper-clad niobium-titanium cables (the superconductivity is essential to obtain the magnetic field needed to achieve the energy required at the center-of-mass of the collision), and a set of 858 quadrupole magnets arranged side by side and

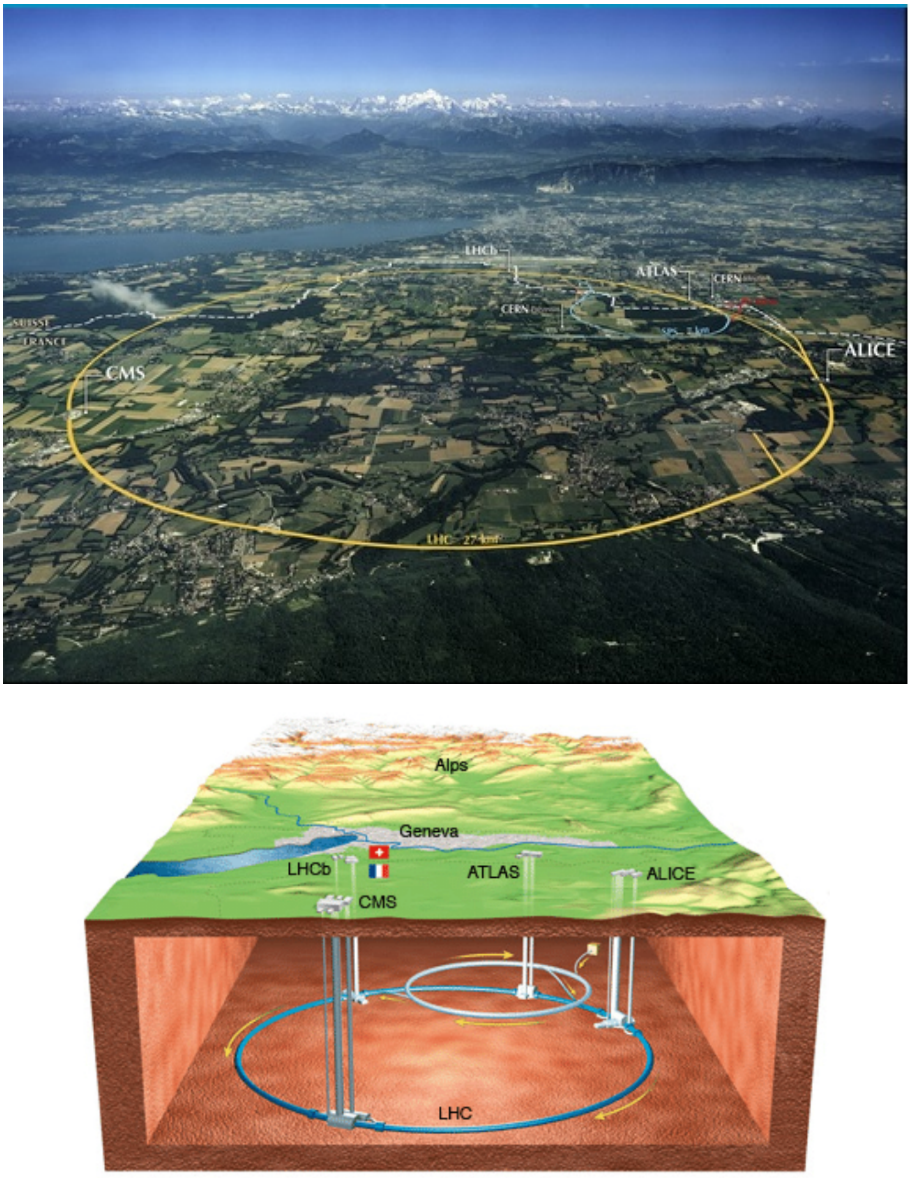

Fig. 1.1: The proton–proton collider LHC at CERN, Geneva.

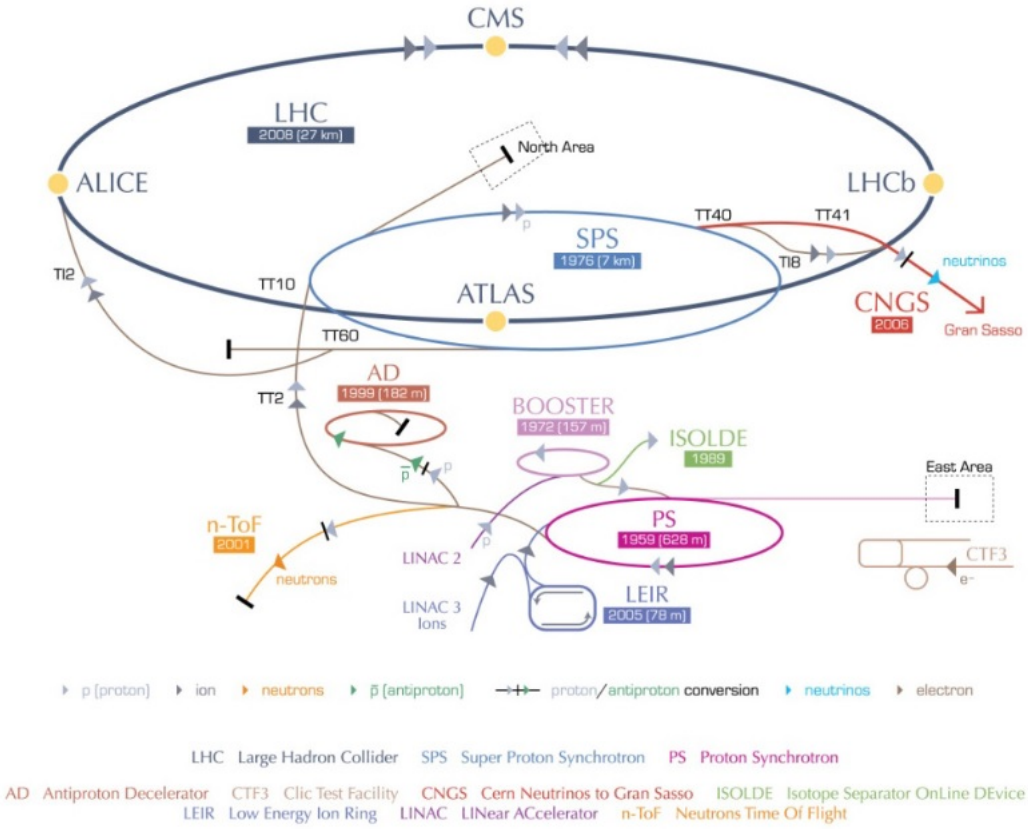

Fig. 1.2: CERN accelerator complex overview.

perpendicular to the poles to focus the beam pipe. Twin bore magnets, consisting of two sets of coils, are also used to support the LHC two-ring architecture. The system requires a low temperature of 2 K, which is achieved by using of liquid helium.

Under normal operating conditions, proton beams can circulate inside the LHC for many hours.

There are four major experiments along the collider, each studying particle collisions from a different angle, using different technologies and pursuing different physical goals.

- ATLAS (A Toroidal LHC ApparatuS) [3] is a multi-purpose experiment built to study pp (and lead-lead) collisions. My PhD thesis is related to this experiment, so a more detailed description will be given in the following chapters.

- CMS (Compact Muon Solenoid) [4] is a multi-purpose experiment; it studies a wide range of physics, including the search for the Higgs boson, extra dimensions, and particles that could make up dark matter. It has a length of 21 m, a diameter of 15 m, and weighs around 14,000 tonnes. Its scientific goals are the same as those pursued by the ATLAS experiment, but the technical solutions and magnet system design are different. The detector is built around a huge solenoid magnet generating a field of 4 T. This structure surrounds an all-silicon pixel and strip tracker, a lead-tungstate scintillating crystal electromagnetic calorimeter and a brass-scintillator sampling hadron calorimeter. The iron yoke of the flux return is equipped with four stations of muon detectors covering most of the 4π solid angle.

- LHCb (LHC-beauty) [5] is a special apparatus for pp collisions. It is dedicated to precision measurements of CP violation and rare decays of b-hadrons. It uses a series of subdetectors to detect mainly particles thrown forward in one direction by the collision. The first subdetector is mounted close to the collision point, and the others follow one after the other, over a length of 20 m. Starting from the interaction point, it is composed of a tracker, a ring imaging Cherenkov detector (RICH), other trackers, another RICH, an electromagnetic calorimeter, a hadronic calorimeter, and a muon detector.

- ALICE (A Large Ion Collider Experiment) [6] is a general-purpose, heavy-ion detector that studies the strong-interaction sector of the SM, the QCD (Quantum

Chromodynamics). Weighing 10,000 tons, the detector is 26 m long, 16 m high, and 16 m wide. It is designed to study strongly interacting matter and the quark-gluon plasma at extreme values of energy density and temperature in nucleus-nucleus collisions. The physics program of this experiment includes not only lead ion and proton collisions, but also lighter ion collisions, lower energy and dedicated proton-nucleus runs. It consists of 18 detectors surrounding the collision point including a time projection chamber, a transition radiation chamber, a "time of flight" detector, electromagnetic and hadronic calorimeters, and a muon spectrometer.

1.1.1 LHC parameters

The LHC was designed to achieve a peak luminosity of $10^{34}\ cm^{-2}\ s^{-1}$ and a center-of-mass energy of 14 TeV by injecting 2808 bunches of protons in each direction at a time. Each bunch consists of about 10^{11} protons and the distance between each bunch is 25 ns, giving a bunch-crossing rate of 40 MHz. This results in multiple pp collisions distributed over a few centimetres. The majority of the inelastic collisions are long-range collisions between the constituents (gluons and quarks) with low energy transfer.

The number of such inelastic pp collisions per bunch crossing, also called *pile-up*, is denoted by μ. This value is $\sim$ 15-50 along Runs 1 and Run 2, and $\sim$ 150-200 is the target for the HL-LHC, as presented in the next section 1.2.

The instantaneous luminosity $\mathcal{L}$, which expresses the collider performance, is defined as:

$$\mathcal{L} = f\frac{n_1 \cdot n_2}{4\pi \cdot \sigma_x \cdot \sigma_y} \tag{1.1}$$

where n_i is the number of particles in the bunches, f is the revolution frequency of the bunches and σ_x, σ_y are related to the transverse dimensions of the beam.

Based on the energy and density of the particles, $\mathcal{L}$ indicates the capability of the apparatus to generate physics events and it determines the number of searched events produced in a Run.

1.2 LHC roadmap

The latest updated LHC/HL-LHC schedule is shown in Fig. 1.3, and spans over many years including an ambitious series of future upgrades.

In the first operating period, Run 1 (2011-2012), the instantaneous luminosity achieved was $7.7 \times 10^{33}\ cm^{-2}s^{-1}$ and the proton center-of-mass energy ranged from 900 GeV up to 8 TeV.

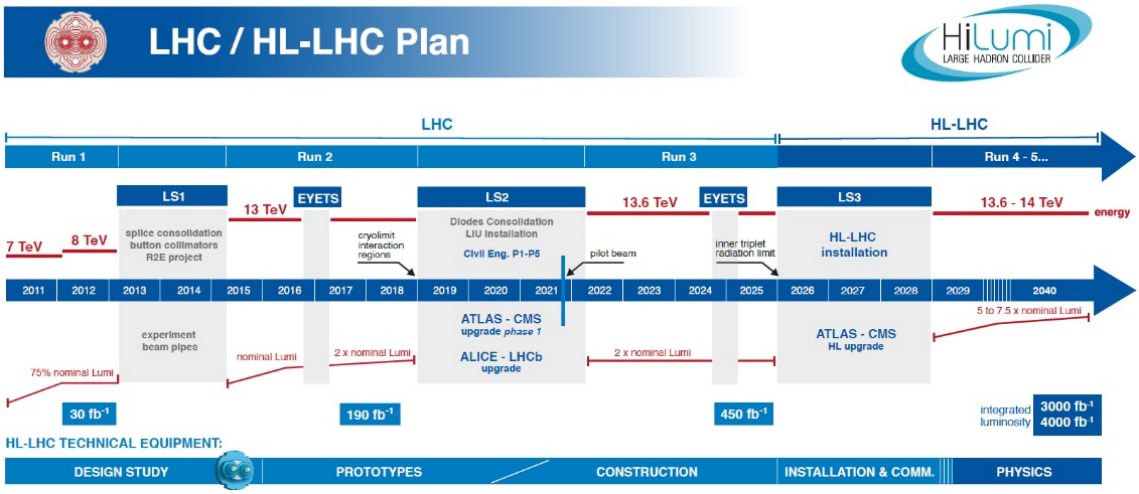

Fig. 1.3: Timeline for the LHC/HL-LHC scientific program, updated in February 2022. Runs are the periods of operation of the collider including data taking by the experiments, while Long Shutdowns (LS) represent periods of downtime for the upgrades to the accelerator and detectors.

The bunch crossing time was 50 ns, double with respect to the design specifications. The energy and $\mathcal{L}$ were very promising at that time: more than half of the target values were achieved. With these parameters, the Higgs boson was observed in 2012.

Run 1 was followed by the Long Shutdown 1, LS1 (2013-2014), during which the magnet splices were repaired and the collimation scheme was upgraded.

Since 3 June 2015, the LHC has been operating in Run 2 (2015-2018) at a center-of-mass energy of 13 TeV, gradually reaching a luminosity of $\mathcal{L} = 1 \times 10^{34}\ cm^{-2}\ s^{-1}$ on 26 June 2016. Despite the reduced number of bunches (about 2200 cf. 2800 nominal), a peak luminosity of up to $1.2 \times 10^{34}\ cm^{-2}\ s^{-1}$ was achieved due to the reduced emittance

of the injectors and a β^* value of 40 cm (cf. 55 cm nominal value) at the high luminosity interaction points. The total integrated luminosity at the end of the period was about 190 fb^{-1}. Run 2 was an extremely successful data acquisition period.

Long Shutdown 2, LS2, (2019-2021) followed Run 2, during which LINAC2 was upgraded to LINAC4 and new cryogenic facilities were subsequently installed to separate the cooling of the superconducting radio frequency modules from the magnet cooling circuit.

The current phase is the Run 3: from mid 2022, the LHC design parameters are expected to provide an instantaneous peak luminosity of $L \sim 2.2 \times 10^{34}\ cm^{-2}\ s^{-1}$ (Phase-I operation) and an integrated luminosity of $\sim 450\ fb^{-1}$. The end of this run is planned for the end of 2025.

After Run 3, the Long Shutdown 3 (2026-2028) will take place with a major upgrade of the LHC components.

Run 4 is expected to start at the end of 2028 and will reach a maximum peak luminosity of $L = 5 \times 10^{34}\ cm^{-2}\ s^{-1}$, corresponding to an average of $\langle\mu\rangle = 140$ simultaneous inelastic pp collisions per bunch crossing (pile-up) [28]. The integrated luminosity should be about 3000 fb^{-1}. Subsequently, Run 5 is expected to start in 2032 and reach $L \sim 7.5 \times 10^{34}\ cm^{-2}\ s^{-1}$, corresponding to $\langle\mu\rangle = 200$. The HL-LHC is expected to be active for the operations in the second half of 2026, collecting an order of magnitude more data than before. The instantaneous luminosity will increase significantly with the consequent growth of data sets, providing the opportunity to improve the sensitivity of current measurements and make entirely new ones. Fig. 1.4 shows the expected luminosity profile for HL-LHC. The ultimate goal is to provide a final integrated luminosity of up to 4000 fb^{-1} after a period of about 12 years.

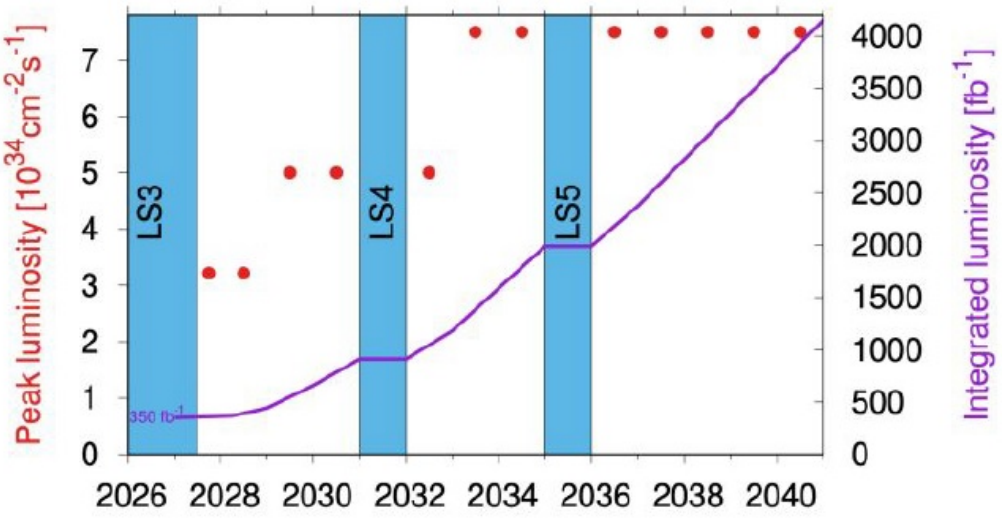

Fig. 1.4: Expected HL-LHC luminosity profile [28].

Chapter 2

The ATLAS experiment

ATLAS is a multi-purpose experiment with a large and rich program of studies, exploring a wide range of topics including rare decays and precision measurements of the Higgs boson, searches beyond the SM including SUSY, heavy flavour physics, and heavy-ion physics.

It is one of the largest scientific projects in history: more than 3000 physicists from over 175 institutes work together on the experiment.

2.1 ATLAS detector overview

Located 100 metres underground at CERN's Point 1 experimental site, the ATLAS detector has a cylindrical shape with a coverage of almost 4π in the solid angle, a length of 44 m, a diameter of 25 m and a total weight of about 7000 t.

The goals are to identify particles traveling through it and measure their momentum and energy, with exception for neutrinos. Due to the high luminosity of the beam, multiple simultaneous collisions happen in every bunch crossing, referred to as *pile-up*.

To investigate the properties of the collisions and reach the physics goals of the ATLAS community, the detector must satisfy some requirements:

- fine granularity: the detector have to distinguish different particle collisions in a high pile-up environment;

- geometric acceptance: the largest fraction of solid angle must be covered;
- fast timing: the time for signal shaping and propagation must be inferior to the bunch crossing separation;
- good energy resolution: particle energy have to be measured in a precise way in a wide range;
- tracking: the track of the charged particle has to be identified with a very precise spatial resolution for identifying the primary vertex and eventually secondary ones;
- fast and reliable electronics: all the electronics must be tolerant to high radiation and fast enough to deal with large data size;
- efficient trigger system: data must be selected in the most efficient way by the trigger system, to obtain optimal performance with the lowest possible rate.

In order to satisfy these requirements, the ATLAS detector is composed of several sub-detectors, each with a specific task.

The sub-detectors have a cylindrical geometry and are arranged radially in an "onion shape" from the Interaction Point (IP) outwards. The proton beams enter the detector from each side and collide in the centre at the IP.

The detector is built around two magnet systems and consists of a number of sub-detectors arranged in layers surrounding the IP. The Inner detector (ID) is immersed in a 2 T uniform magnetic field generated by a thin superconducting solenoid magnet; it identifies the tracks and measures the momentum of the charged particles. The ID is surrounded by the calorimeters, which stop and measure the energy of all particles, except muons and neutrinos. The outermost detector is the Muon Spectrometer (MS), which incorporates eight large superconducting toroidal magnets integrated into its design, used to detect and measure the momentum and the trajectory of muons.

The detector layout is presented in Fig. 2.1, with a frontal view in Fig. 2.2.

An overall summary of the resolution required to obtain the expected results for the relative pseudorapidity region is given in Table 2.1 for each of the ATLAS sub-detector. The pseudorapidity requirements for the trigger system associated with each sub-detector are also reported.

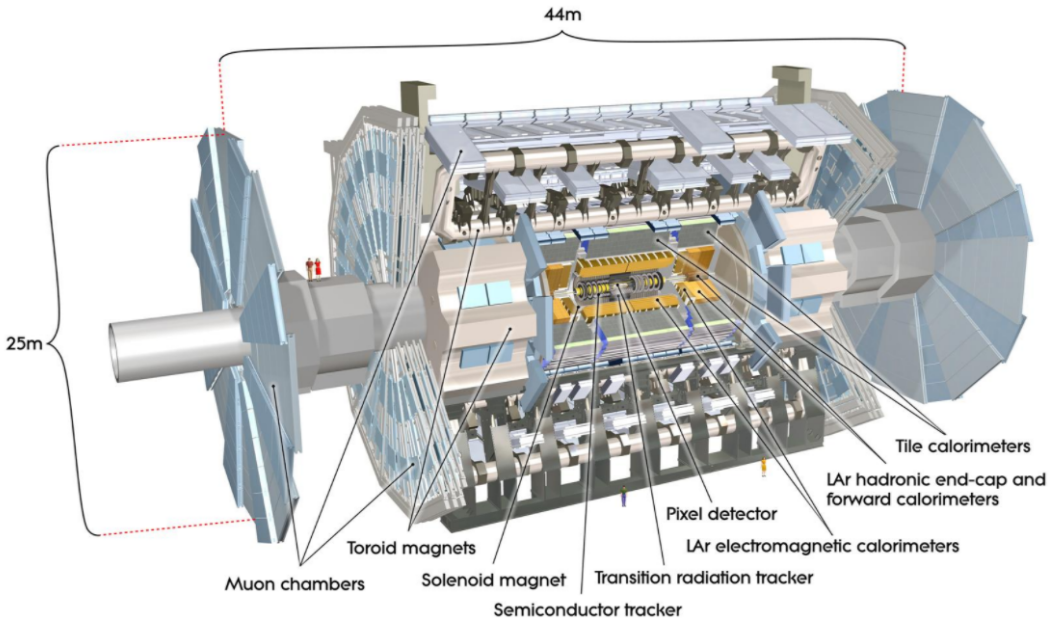

Fig. 2.1: ATLAS detector layout.

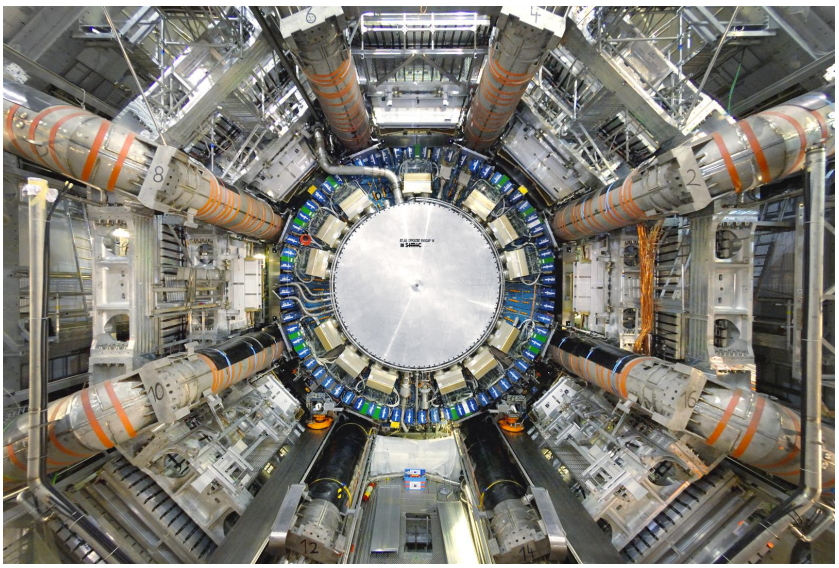

Fig. 2.2: ATLAS detector frontal view.

Detector component	Required resolution	η coverage Measurements (Trigger)
Tracking	$\sigma_{p_T}/p_T = 0.05\% p_T \oplus 1\%$	$\|\eta\| < 2.5$
EM calorimeter	$\sigma_E/E = 10\%\sqrt{E} \oplus 0.7\%$	$\|\eta\| < 3.2$ ($\|\eta\| < 2.5$)
Hadronic calorimeter barrel and end-cap forward	 $\sigma_E/E = 50\%\sqrt{E} \oplus 3\%$ $\sigma_E/E = 100\%\sqrt{E} \oplus 10\%$	 $\|\eta\| < 3.2$ ($\|\eta\| < 3.2$) $3.1 < \|\eta\| < 4.9$ ($3.1 < \|\eta\| < 4.9$)
Muon spectrometer	$\sigma_{p_T}/p_T = 10\%$ at $p_T = 1\ TeV$	$\|\eta\| < 2.7$ ($\|\eta\| < 2.4$)

Table 2.1: Summary of the ATLAS sub-detectors required resolutions and relative pseudorapidity regions. Units for E and p_T are in GeV.

2.2 The ATLAS Coordinate system

The ATLAS detector uses a right-hand coordinate system, with the origin at the nominal IP in the centre of the detector. As shown in Fig. 2.3, the x-axis points towards the centre of the LHC, the y-axis upwards, and the z-axis along the beam pipe. The positive z-axis defines the side-A of the detector, while the negative z-axis defines the side-C. The x-y plane is the *transverse plane* with respect to the z-axis; its transverse momentum p_T is null as the beam travels along the z-axis, meaning that there are some observables for the collision products that are conserved in the transverse plane, such as the transverse momentum p_T and the transverse energy E_t.

The transverse plane can also be described in polar coordinates (r,ϕ), where r is the distance from the IP and ϕ is the azimuthal angle around the z-axis, defined with respect to the positive x-axis. θ is the polar angle around the x-axis, defined with respect to the positive z-axis. The transverse momentum and the transverse energy are respectively defined as $p_T = p \sin\theta$ and $E_T = E \sin\theta$ in the x-y plane.

Instead of θ, in the LHC experiments it is widely used the *pseudorapidity* η, defined as a function of the angular position of the particle:

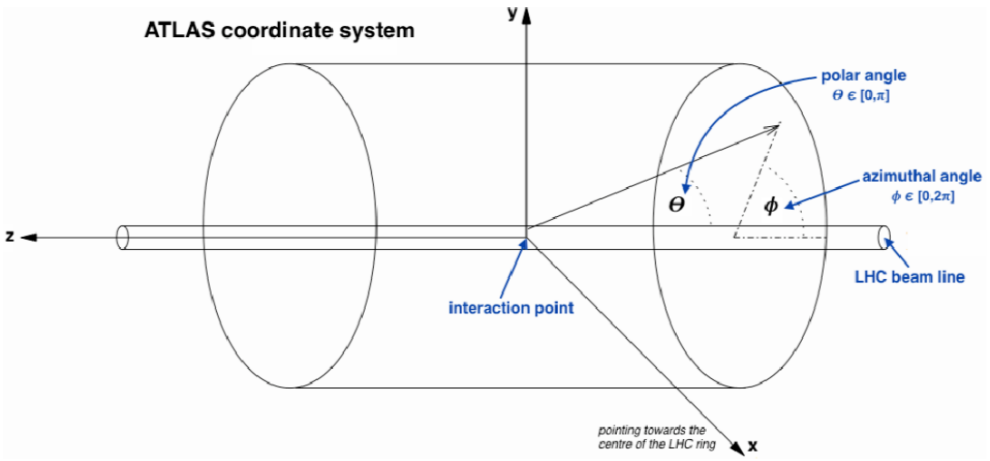

Fig. 2.3: ATLAS coordinate system.

$$\eta = -ln\ tan(\frac{\theta}{2}) \tag{2.1}$$

Considering massive objects such as jets, instead of pseudorapidity, it is used the *rapidity*, which is Lorentz-invariant for transformations along the z-axis. It is defined as:

$$y = \frac{1}{2} ln[\frac{E + p_L}{E - p_L}]. \tag{2.2}$$

where p_L is the linear momentum of the particle. The angular separation between two particles can be expressed in terms of their rapidity, using the Lorentz-invariant relationship:

$$\Delta R = \sqrt{\Delta y^2 + \Delta \phi^2}. \tag{2.3}$$

For relativistic boosted particles, where $m << p_T$, pseudorapidity becomes equal to rapidity ($\eta \approx y$), so the angular distance between two particles can be expressed in terms of angular quantities only:

$$\Delta R = \sqrt{\Delta \eta^2 + \Delta \phi^2}. \tag{2.4}$$

For this reason, the ATLAS coordinate system is often expressed in terms of the pseudorapidity η instead of the polar angle θ.

2.3 Detector composition

2.3.1 Inner Detector

The Inner Detector [8] is the closest to the beam line and the first detector to be passed by the particles after the pp interaction. This cylindrical apparatus is 6.2 m long with a diameter of 2.1 m and its pseudorapidity covers the range $|\eta| < 2.5$. Its main function is to measure the direction, momentum, and charge of the electrically-charged particles produced in each pp collision. This allows vertices to be reconstructed from primary interactions, in order to separate such vertices from those associated with pile-up, and to detect secondary vertices from particles with long lifetimes.

An overview of the ID is shown in Fig. 2.4, a system architecture based on a multi-layer cylindrical redundancy of three sub-detectors, each pointing towards the LHC beam pipe.

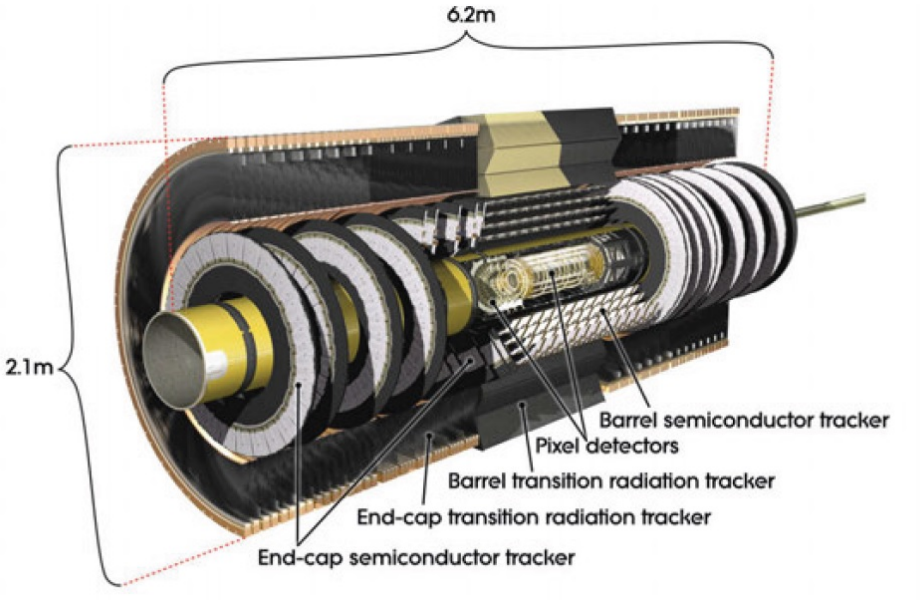

Fig. 2.4: Overview of the ATLAS Inner Detector, consisting of the Pixel Detector, Semi-Conductor Tracker and Transition Radiation Tracker.

The innermost component is the silicon Pixel Detector (PD) [9]; the middle part is the Semi-Conductor Tracker (SCT) [10], which is a silicon strip detector, and the outermost

component is the Transition Radiation Tracker (TRT) [11], consisting of straw-tube detectors. During 2014, a new detector, the Insertable Barrel Layer (IBL) [12], was added.

A section of PD, SCT and TRT is shown in Fig. 2.5.

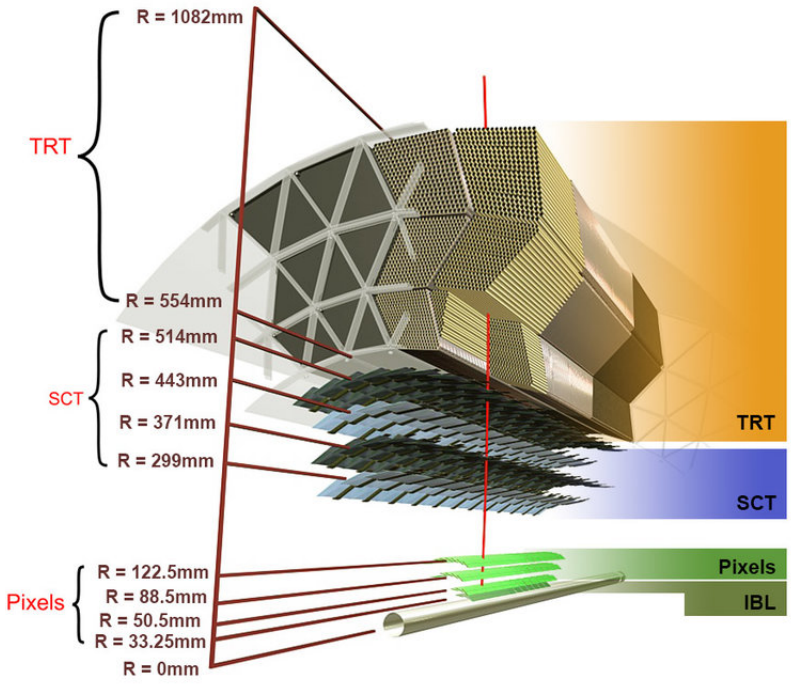

Fig. 2.5: Section of the ATLAS barrel Inner Detector, showing all the sub-detectors position with respect to the beam pipe.

The whole ID is placed inside a 2 T magnetic field generated by a superconducting solenoid that bends the trajectory of charged particles according to the Lorentz force:

$$\mathbf{F} = q\mathbf{E} + q\mathbf{v} \times \mathbf{B} \tag{2.5}$$

where q is the electric charge of the particle, $\mathbf{E}$ is the electric field, $\mathbf{v}$ is the velocity of the particle and $\mathbf{B}$ is the magnetic field. This allows the momentum of charged particles to be measured from the curvature of their trajectory.

A high-precision measurement of the particle tracks is performed by the *inside-out* and *outside-in* algorithms. The inside-out algorithm starts from Pixel seeds and adds hits moving away from the interaction point. It reconstructs most of the primary tracks. The outside-in algorithm reconstructs secondary particle tracks starting from segments reconstructed in the TRT, and extends them inwards by adding silicon hits.

The Pixel Detector

The PD is the most internal and precise detector and spans the radial region 33-150 mm. It is composed of four different semiconductor layers: IBL, B-Layer (B0), Layer-1 (L1) and Layer-2 (L2). IBL was introduced during LS1; it is the innermost layer, a four pixel layers located at a distance from the beam pipe of 33 mm, with 12 million of 50×250 μm^2 pixels covering the region $|\eta| < 3.03$. The others are located respectively at 50.5, 88.5 and 122.5 mm. The total amount of pixels is over 80 million, each with a typical size of 50×400 μm^2.

The Semi-Conductor Tracker

Together with the PD, the Semi-Conductor Tracker (SCT) is one of the most precise tracker of ATLAS; it spans the radial region 299-560 mm. It consists of four cylinders in the barrel region and two end-caps composed of nine disks each covering the region $|\eta| < 2.5$. SCT allows 17μm of resolution along the r-ϕ direction and 580 μm along the z-axis. This layout is used to reconstruct the z-position of hits in the SCT. The total number of strips in the SCT is approximately 12 million.

The Transition Radiation Tracker

The Transition Radiation Tracker (TRT) is the largest ID that surrounds the previous two. It is a combined tracking and electron identification gaseous detector. Tracking is carried out by drift tubes, while the interleaved radiators produce detectable X-rays when electrons traverse them. It spans the radial region 554-1082 mm. It consists of about 5×10^4 cylindrical straw tubes filled with a mixture of Xenon (70 %), CO_2 (27 %) and O_2 (3 %). In the barrel region, the straws are parallel to the beam axis, while in

the end-cap region they are perpendicular. The straws all together help to measure the particle momentum and to achieve a high tracking capability.

An overall summary of the characteristics of each ID sub-detector is reported in Table 2.2.

Detector	Hits/track	Element size	Hit resolution [μm]
Pixel, $\|\eta\| < 2.5$		50×400 μm^2 (B0,L1,L2)	
4 barrel layers	3		10 (r-ϕ), 115 (z)
2x3 end-cap disks		50×250 μm^2 (IBL)	10 (r-ϕ), 115 (r)
SCT, $\|\eta\| < 2.5$			
4 barrel layers	8	$50\mu m$	17 (r-ϕ), 580 (z)
2x9 end-cap disks			17 (r-ϕ), 580 (r)
TRT, $\|\eta\| < 2.0$			
73 barrel tubes	~ 30	d=4 mm, l=144 cm	130/straw
160 end-cap tubes		d=4 mm, l=37 cm	

Table 2.2: Inner Detector sub-components main characteristics.

2.3.2 Calorimeters

The ATLAS calorimeter system measures the energies of the surviving particles that pass through the Inner Detector, except neutrinos, and provides some information about their position and identity.

It also largely prevents any particles other than neutrinos and muons from entering the muon system.

Occasionally, a shower particle can leave the calorimeter and enter the ATLAS muon spectrometer, for example in the case of higher-energy initial particles. Such an event is called a calorimeter *punch-through* or *particle leakage* into the muon spectrometer, and the particles entering the MS are called *punch-through particles*.

The Figure 2.6 gives an overview of the ATLAS calorimeter system. It consists of two different categories of calorimeter, the ectromagnetic and the hadronic, because of the different nature of electronic and hadronic interactions and the requirements for resolution and pseudorapidity coverage.

The whole system covers pseudorapidity up to $\eta = 4.9$ and full coverage of ϕ.

Electromagnetic calorimeter

The Electromagnetic CALorimeter (ECAL) system absorbs and measures the energy of electromagnetic particles, i.e. electrons and photons. It is a *sampling* calorimeter, based on a Lead / Liquid-Argon (LAr) structure.

The Liquid Argon was chosen as the active material because it is relatively dense (no signal amplification is needed), the signal response is linear with the energy, it is stable with time and it is resistant to radiation, while lead was choosen as the passive material because of its good absorbing capacity.

The ECAL consists of an ElectroMagnetic Barrel (EMB), covering a pseudorapidity $|\eta| < 1.475$, and an ElectroMagnetic End-Cap (EMEC), covering $1.375 < |\eta| < 3.2$. When electrons or photons cross the ECAL, they produce an electromagnetic shower; this is very different from that one produced by hadrons. In this way, the calorimeter can distinguish between photons and neutral pions π^0.

In fact, the barrel is divided into three main regions along the z-axis; the first region, called the η-*strip* layer, is finely granulated to exploit the shower structure and is able to distinguish between photons and pions over a wide energy range ($\sim 5GeV$ - $\sim 5TeV$). The other regions also show a high granularity, which is extremely important for the reconstruction of the missing transverse energy.

The central region of the calorimeter is specifically designed to identify electrons and photons. It has a characteristic accordion structure, with a *honeycomb* pattern, to ensure that no particle escapes unchallenged.

To keep the argon in liquid form, the calorimeter is maintained at -184°C. Specially-designed, vacuum-sealed cable cylinders carry the electronic signals from the cold liquid argon to the warm area where the readout electronics are located.

The resolution achievable in the barrel and end-cap region is:

$$\frac{\sigma_E}{E} = \frac{9.4\%}{\sqrt{E(GeV)}} \oplus 0.1\% \tag{2.6}$$

where the first term is the stochastic term and the second is the constant term.

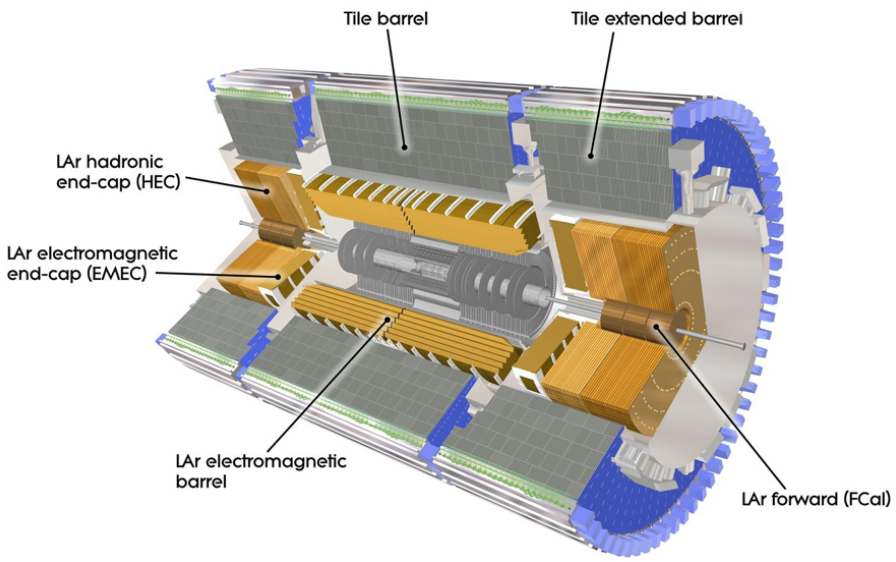

Fig. 2.6: Overview of the calorimeters.

Hadronic calorimeter

The Hadronic CALorimeter (HCAL) absorbs and measures the energy and missing momentum of hadrons produced in pp collisions or formed from secondary decays, and jets.

It is composed of the Hadronic Tile Calorimeter (HTC), the Hadronic End-Caps Calorimeter (HEC) and the Forward CALorimeter (FCAL); the first one is a scintillator-tile calorimeter, the others are both LAr calorimeters.

The HTC surrounds the LAr calorimeter and measures the energy of hadronic particles that do not deposit all of their energy in the LAr Calorimeter.

The HTC is divided into a *Tile barrel* covering a pseudorapidity region $|\eta| < 1.0$, and two smaller tile barrels called *Tile Extended barrels* covering $0.8 < |\eta| < 1.7$. It is a sampling calorimeter, and it consists of layers of steel as absorber and about 420,000 plastic scintillating tiles as active material working in synchrony. Weighing almost 2,900 tonnes, it is the heaviest part of the ATLAS experiment.

When particles hit the steel layers, they create a shower of new particles. The plastic scintillators in turn produce photons, which are converted into an electric current whose

intensity is proportional to the energy of the original particle.

The energy measurement resolution of the HTC combined with the Electromagnetic Calorimeter to isolated charged pions is:

$$\frac{\sigma_E}{E} = \frac{52\%}{\sqrt{E(GeV)}} \oplus 3\% \tag{2.7}$$

The HEC is also a sampling calorimeter using copper as absorber (passive material) and LAr (as the ECAL) as active material. It covers the region covers $1.5 < |\eta| < 3.2$. The energy resolution for charged pions is:

$$\frac{\sigma_E}{E} = \frac{71\%}{\sqrt{E(GeV)}} \oplus 1.5\% \tag{2.8}$$

The FCAL is designed to provide very high pseudorapidity coverage for both hadronic and electromagnetic calorimeters, $3.1 < |\eta| < 4.9$. It is a sampling calorimeter (2.6 radiation lengths) using LAr as the active material, and copper and tungsten as the absorber. Its measured energy resolution is:

$$\frac{\sigma_E}{E} = \frac{94\%}{\sqrt{E(GeV)}} \oplus 7.5\% \tag{2.9}$$

2.3.3 Muon Spectrometer

The Muon Spectrometer is the outermost part of ATLAS. It triggers and measures the momentum of charged particles, mainly muons, which are not stopped by the calorimeters. An overview is shown in Fig. 2.7.

The MS is divided into barrel and end-cap regions, where toroidal magnets are placed to provide the magnetic field needed to bend the trajectory of the muons and measure their momentum. It consists of four components: Monitored Drift Tubes (MDT), Cathode Strip Chambers (CSC), Resistive Plate Chambers (RPC), and Thin Gap Chambers

(TGC). Different technologies are used in different η regions and for different purposes. They are all gas detectors.

Muons with an energy below the threshold, $p_T > 3\ GeV/c$, are completely absorbed before reaching the MS, so they cannot be identified.

The detectors cover pseudorapidity range $|\eta| < 2.7$, while the trigger operates in the region $|\eta| < 2.4$. The p_T resolution is below 20% up to 1 TeV.

Regarding the *punch-through particles*, their effect on the muon spectrometer varies considerably depending on their type, energy, charge, and position: from having no effect on generation of clean particle track signatures, to full tracks that have to be reconstructed and are accidentally misinterpreted as primary muon tracks by the MS reconstruction. These tracks are further referred to as *fake primary muon* tracks, since they can be used as misidentified muons in physics analyses.

Monitored Drift Tubes and Cathode Strip Chambers

Monitored Drift Tubes (MDT) and Cathode Strip Chambers (CSC) measure the muon momentum. MDT chambers are drift chambers with two multilayer drift tubes, focused on the precise measurement of the r-z coordinate in the barrel region. The hit position of the particle can be reconstructed by measuring the drift time in individual tubes. CSCs are multi-wire chambers with strip cathodes for the measurement of muon momentum in the range of $1.0 < |\eta| < 2.7$. The CSC wires consist of parallel anodes which are perpendicular to 1 mm large strips of opposite polarity. They are placed close to the beam pipe in the innermost layer of the end-cap.

During the upgrades for Run 3, the New Small Wheel (NSW) muon detector replaced the previous end-cap CSC detector. It was composed of a Small strip Thin Gap Chambers (sTGC) and Micromegas wedges, to improve rate capability and performance, reducing the acceptance of good muon tracking and the high rate of false high-p_T muon triggers. The NSW detector consists of 2 wheels, each with 8 sectors in the front and back side, for a total of 32 sectors. It is based on a micro-mesh gas structure.

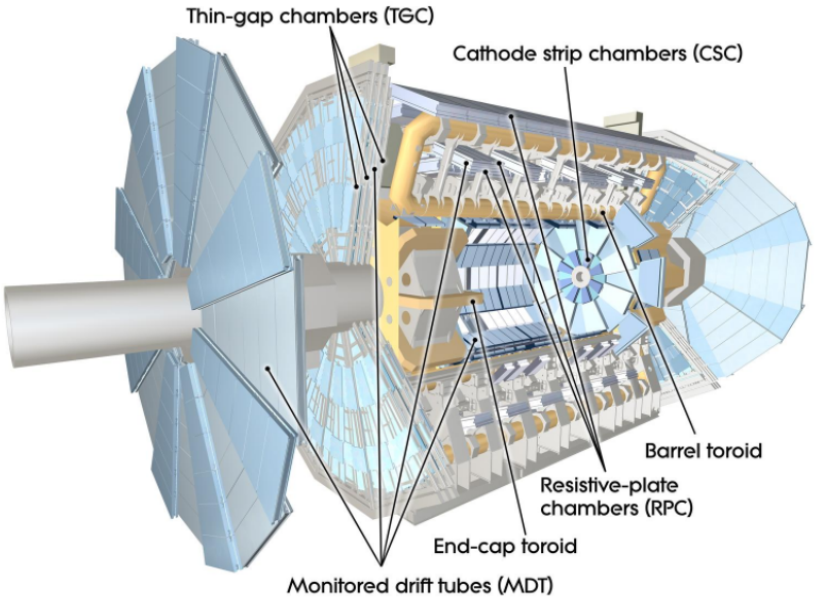

Fig. 2.7: ATLAS muon system overview.

Thin Gap Chambers and Resistive Plate Chambers

TGCs and Resistive Plate Chambers (RPCs) provide the online trigger. TGC in the end-cap region is a very thin multi-wire chamber: the spatial resolution is 4 mm in the radial direction and 5 mm in the ϕ coordinate. The anode-cathode spacing is smaller than the anode-anode spacing, leading to a drift time lower than 20 ns. TGCs are also used to improve the measurements along the ϕ coordinate obtained from the precision chambers. RPCs are part of the barrel and cover the region $|\eta| < 1.05$, while TGCs are part of the end-cap and cover the region $1.05 < |\eta| < 2.7$, where the region $1.05 < |\eta| < 2.4$ is used for triggering.

2.3.4 Magnetic System

By bending the trajectories of charged particles, ATLAS can measure their momentum and charge. This is done using two different types of superconducting magnet systems: solenoidal and toroidal. The ATLAS magnet system is composed of three main sections: a central solenoid magnet, a barrel toroid and two end-cap toroids. It is cooled to approximately 4.5 K (–268°C) in order to provide the necessary strong magnetic fields.

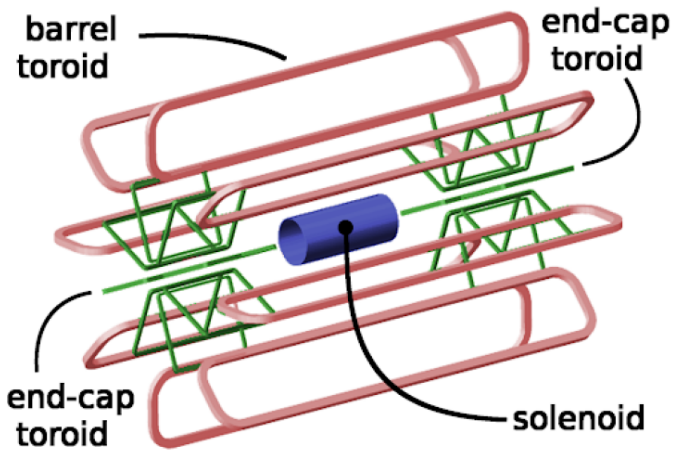

Fig. 2.8: Overview of the ATLAS magnet system.

The central solenoid surrounds the inner detector at the core of the experiment, while three large air-core toroids, one barrel and two end-cap, generate the magnetic field in the MS. Fig. 2.8 shows an overview of the magnet system.

The CS is 5.3 m long, 2.4 m in diameter, 4.5 cm thick and weighs 5 t. A cryostat, shared with the electromagnetic calorimeter barrel, maintains an operating temperature of 4.5 K. 2 T is the magnetic field provided, pointing in the positive z-axis direction. This is achieved by embedding over 9 km of niobium-titanium superconducting wires in strengthened, pure aluminium strips to minimise possible interactions between the magnet and the particles being studied.

The barrel toroid consists of 8 flat superconducting race-track coils, each 25.3 m long and 20.1 m in outer diameter. The 8 separate coils in the toroid are held in place by 16 support rings. Its total weight is 830 t. It generates a magnetic field of 4 T and are cooled down to 4.7 K by liquid helium.

The two magnetic end-cap toroids are positioned at either end of the experiment, and provide the required 4 T magnetic field over a radial span of 1.5 m to 5 m and operate at a working point temperature of 4.7 K. Each end-cap toroid has an axial length of 5.0 m, an external diameter of 10.7 m and a weight of 240 t. The coil system of the end cap toroid is rotated at an angle of 22.5° with respect to the barrel toroidal coil. This creates a radial overlap between the two coil systems and optimises the bending performance.

The most important parameters for momentum measurements are the bending power field and the total transverse deflection of the particle from its initial path.

2.3.5 Forward detectors

The ATLAS forward region is covered by a number of small sub-detectors: LUCID (Luminosity measurement using Cherenkov Integrating Detector) [13], ZDC (Zero-Degree Calorimeter) [14], AFP (ATLAS Forward Proton) [15] and ALFA (Absolute Luminosity for ATLAS) [16]. Figure 2.9 shows their position along the beam line.

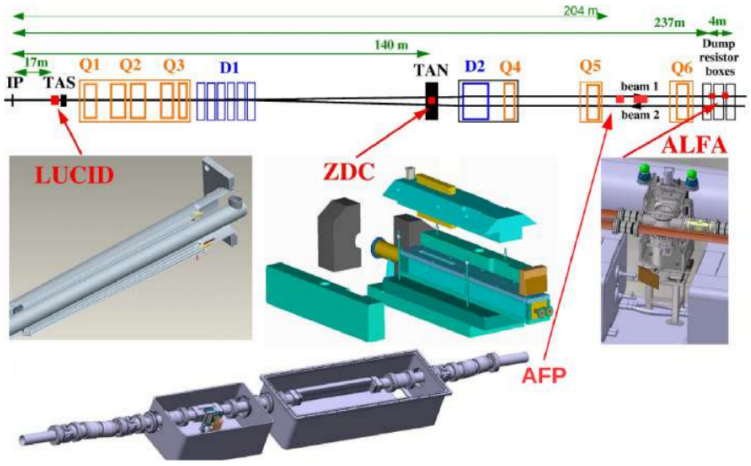

Fig. 2.9: ATLAS Forward Detector infrastructure.

LUCID

Luminosity measurement using the Cherenkov Integrating Detector (LUCID) is a Cherenkov counter that measures the online relative luminosity by detecting inelastic pp collisions. Two LUCID detectors are placed symmetrically in the two forward regions at a distance of 17 m from the interaction point. Each consists of 16 photomultiplier tubes and 4 quartz fiber bundles. In the quartz window and in the fiber bundles, Cherenkov light is emitted by the charged particles and detected by the photomultiplier tubes.

ALFA

The Absolute Luminosity For ATLAS (ALFA) is the most distant detector, located 237 m from the interaction point, on both sides of ATLAS. ALFA consists of square-shaped scintillating fibers placed in Roman pots up to 1 mm from the beam pipe. It measures the elastic pp scattering amplitude at small angles and uses the optical theorem to determine the cross section, which is then used to calculate the absolute luminosity. Special beam conditions are used to measure the luminosity, since the nominal beam divergence is less than the 3 μrad required to measure the forward elastic-scattering amplitude.

ZDC

The Zero-Degree Calorimeter (ZDC) detects neutrons in a very forward region, in both pp and heavy-ion collisions at $|\eta| < 8.3$. It is located at 140 m on either side of ATLAS and consists of an electromagnetic module (about 29 radiation lengths thick) and three hadronic modules, made of tungsten with an embedded matrix of quartz rods, attached to photomultiplier tubes.

AFP

The aim of ATLAS Forward Proton (AFP) is to measure the transfer momentum and energy loss of protons emitted from the collision point in a very forward region. Two AFP detectors are placed along the beam line, at 204 m and 217 m, containing a 3D silicon tracker and a time-of-flight detector.

2.4 Trigger and Data Acquisition

The ATLAS TDAQ [17] system is an essential component of the experiment as it is responsible for deciding whether or not to keep an event from a given bunch-crossing interaction for future study.

By design, the ATLAS Trigger is multi-level; from Run 2 onwards, there are two levels: the hardware-based Level-1 trigger (L1) and the software-based High-Level Trigger (HLT), as shown in Fig. 2.10.

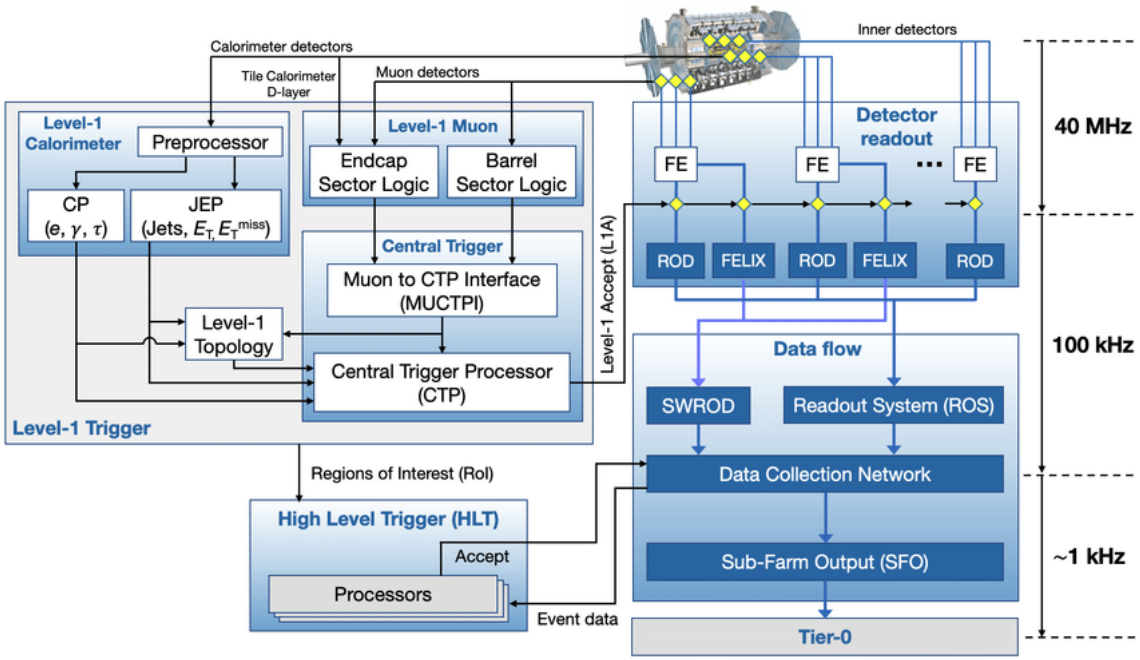

Fig. 2.10: Schematic overview of the ATLAS TDAQ system in Run 3.

The L1 trigger decision consists of the Central Trigger Processor (CTP), which receives inputs from the L1 calorimeter (L1Calo) and L1 muon (L1Muon) triggers, and several other subsystems such as the Minimum Bias Trigger Scintillators (MBTS), the LUCID Cherenkov counter, and the Zero-Degree calorimeter (ZDC). The CPT is also responsible for the application of the preventive dead-time. It limits the minimum time between two consecutive L1 accepts ($simple\ dead-time$) to avoid overlapping readout windows and restricts the number of L1 allowed in a given number of bunch crossings

(*complex dead − time* to avoid overflowing front-end buffers.

Event data move from the ATLAS on-detector electronics to the front-end buffers at the bunch crossing rate of 40 MHz, with each bunch colliding every 25 ns. The L1 trigger system has an average trigger acceptance rate of 100 kHz at the output and identifies the Regions of Interest (RoI) in the detector used by the HLT. L1 maximum readout latency is of 2.5 μs.

After the L1, the data are readout from the front-end electronics of the detector in the ReadOut Drivers (RODs), which perform fragment building and associated error detection, data checking, transformation, and monitoring. The data are then buffered in a Read-Out System (ROS) until requested by the HLT.

The HLT runs on a computer farm of about 40K processing units, located in a room close to the detector and connected to the read-out system by high-speed optical links. The HLT performs both *regional* reconstruction in the RoIs identified by the L1 and *global* event reconstructions, depending on L1 requests.

The events selected by the HLT are stored on permanent storage at an output rate of 1.5 kHz. Given an average size of 1MB per event during nominal operation data, data are recorded at approximately 1.5 GB/s.

After the acceptance by the HLT, the events are transferred to local storage at the experimental site and exported to the Tier-0 facility at the CERN computing centre for offline reconstruction.

Chapter 3

The ATLAS Phase-II upgrade

The current phase is Run 3, scheduled to end in late 2025, followed by Long Shutdown 3 (LS3) in 2026-2028, during which several components will be upgraded to allow the LHC to operate smoothly at an instantaneous luminosity above its initial nominal value. The HL-LHC is expected to start operation at the beginning of 2026, eventually reaching a peak instantaneous luminosity of $L = 7.5 \times 10^{34}\ cm^{-2}\ s^{-1}$, corresponding to about 200 inelastic pp collisions per bunch crossing, and delivering more than ten times the integrated luminosity of the LHC Runs 1-3 combined (up to $4000 fb^{-1}$).

In order to adapt to the new conditions of high pile-up without losing the acceptance of the physics, ATLAS will have to upgrade most of its detectors and the TDAQ architecture during the so-called Phase-II.

The ATLAS collaboration firstly gave a description of the initial plan for the Phase-II upgrade of the detector in the Letter of Intent (LoI) in 2012. Since then, the collaboration has been improving and refining the initial proposals in a series of Technical Design Reports (TDR), one for each detector system separately.

In general terms, the upgrade will increase the bandwidth and performance of the trigger and readout electronics, to send the maximum amount of data off-detector to permanent storage, and to run the sophisticated trigger algorithms required to cope with the increased data rates in the off-detector trigger firmware and software rather than in the on-detector trigger hardware.

3.1 Upgrade proposals

The HL-LHC will provide an extremely challenging environment for the ATLAS experiment, with the aim of extending the search for physics beyond the SM and improving the precision of the SM measurements.

All the modifications required to maintain this runtime will be carried out during the Long Shutdown 3 (LS3), during which sensors, hardware, firmware, software, and strategies will be updated. The most important of these will be the ITk and the TDAQ systems, due to radiation damage (for Run 0-3) and the need for higher granularity in order to cope with the HL environment. The ITk will reconstruct tracks up to $|\eta| < 4$, and a new High Granularity Timing Detector will cover the forward region $\eta > 4.0$ helping reconstructing interactions vertexes.

Other changes will be made to the Calorimeters (both the Liquid Argon and the Tile one) and in the MS. Some of the sub-detectors will be completely replaced, while others will only have their readout electronics replaced.

3.1.1 Inner Tracker

In the Phase-II, the current ID will be completely replaced with a new all-silicon ITk detector. A core scheme of ATLAS ITk is built through a software simulation (Fig. 3.1).

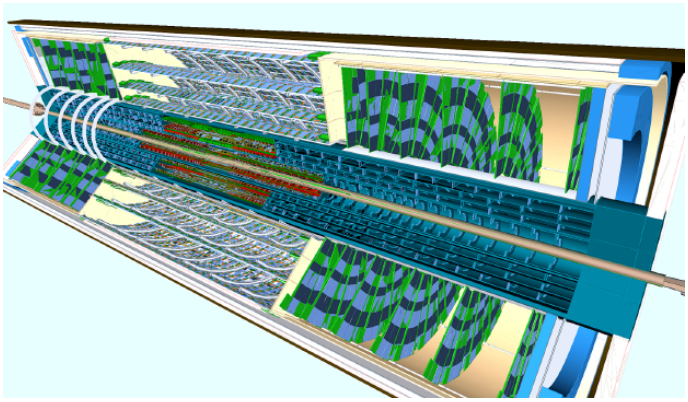

Fig. 3.1: Scheme of the new ATLAS ITk detector built by simulations.

The ITk consists of two subsystems: a Pixel Detector surrounded by a Strip Detector.

The Pixel Detector consists of five flat barrel layers close to the beam line and several inclined or vertical ring-shaped end-cap disks, extending the coverage to $|\eta| = 4$.

The Strip Detector has four strip double-module layers in the barrel region and six end-cap disks, covering the pseudorapidity range up to $|\eta| = 2.7$. The innermost two strip double-layers, called the short-strips, have a strip length of 24.1 mm while the outermost two, called the long-strips, have a length of 48.2 mm. All strips are 75.5 μm wide. As with the old SCT, the double layers of the new strip detector are rotated relative to each other around the radial axis, at an angle of 52 mrad, to improve the longitudinal resolution.

A detailed description of the ITk can be found in the Technical Design report (TDR) of the pixel [20] and the strip detector [21].

The ITk planned layout is shown in Fig. 3.2: the vertical axis represents the radius from the beam pipe, the horizontal axis the z-axis parallel to it; in red we have a five-layer Pixel detector surrounded by the Strip detector in blue. Only the positions of the active sensors are shown.

The general requirement is to provide a more accurate reconstruction and to reduce the multiple scattering than the current ID, this can be achieved using an i*nclined layout*, as it reduces the amount of material that a particle, exiting the Interaction Point, will pass through.

The ITk detector is designed to measure the transverse momentum and direction of all the charged particles, those emerging from the primary interaction vertexes (i.e. those associated with p-p interactions with high momentum transfer) and those emerging from the pile-up vertexes (i.e. the ones with low momentum transferred). As the detector must provide an environment where the integrated radiation dose is ten times higher than the previous LHC conditions, the radiation tolerance of the inner technology should reach a resistance of 9.9 MGy, taking into account the instantaneous luminosity and the pile-up at which the HL-LHC will operate.

New technologies are being used to ensure that the system is able to survive in this high radiation environment. The 65 nm CMOS front-end technology has been targeted at a radiation tolerance of 5 MGy for 4000 fb^{-1} of the total absorbed dose, with a limit of almost 10 MGy for 2000 fb^{-1} required for the inner replaceable layers.

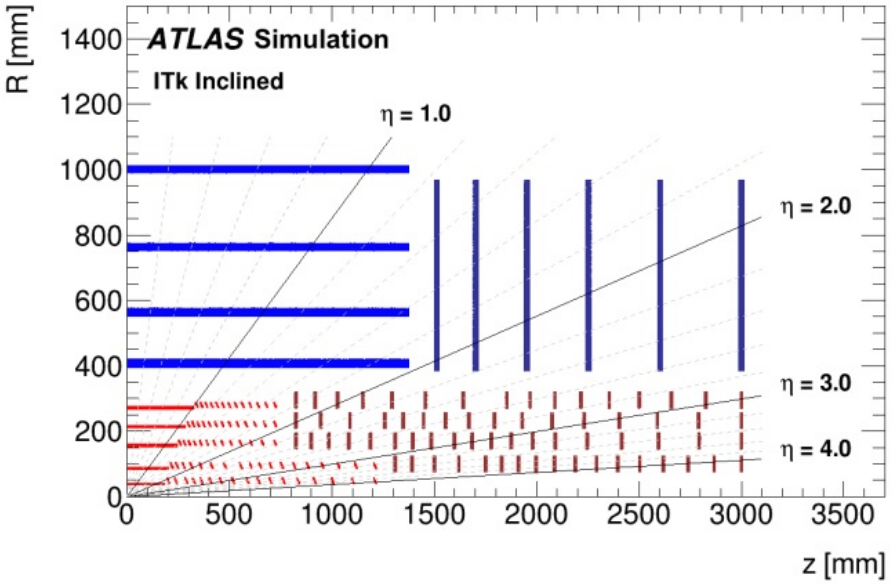

Fig. 3.2: A schematic view of the ITk Layout [20]. A quarter of the detector is represented, where the active elements of the barrel and end-cap Strip Detector are shown in blue, for the Pixel Detector the sensors are shown in red for the barrel layers and in dark red for the end-cap rings. The pseudorapidity coverage is up to $|\eta| = 4$.

3.1.2 High Granularity Timing Detector

The High Granularity Timing Detector (HGTD) [23] is a new architecture proposed to improve the identification of the interaction vertexes in the forward region through a time measurement. A secondary purpouse is to increases the precision of luminosity measurements. A 3D view of it and where it will be expected to be placed is shown in Fig. 3.3.

Its strategic position gives the possibility to measure both online luminosity bunch-per-bunch during HL-LHC running and enhance the high precision sampling of the integrated luminosity. The HGTD will increase the ITk spatial and time performance with a 30 ps time resolution for the minimum ionizing particle going through the innermost detector. The HGTD detection region will cover the range of $2.4 < |\eta| < 4.0$.

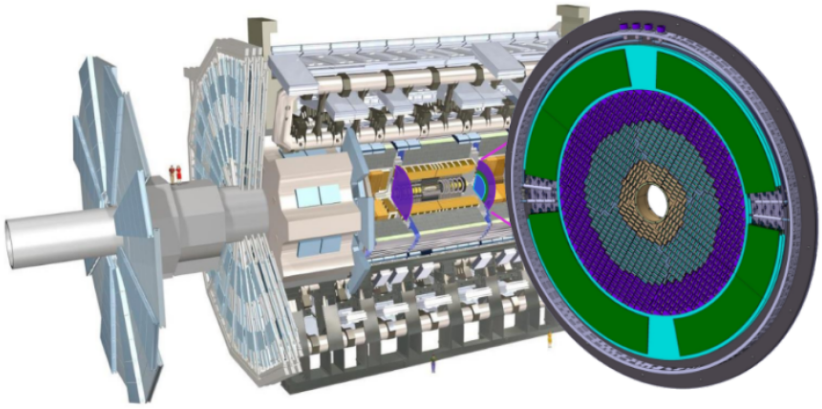

Fig. 3.3: 3D view of the new HGTD detector and its position in the future ATLAS structure.

3.1.3 Calorimeter

The current electronics of the ATLAS Liquid Argon Calorimeter (LAr) is not compatible with Run 4, which provides increased latency and trigger rate. Morover, the radiation hardness requirements are above its original design (1 kGy and $2.7 \times 10^{13} neq/cm^2$).

To satisfy these requirements, the readout electronics (front-end and back-end) will be full replaced [24], while Run 3 upgraded boards will continue to be used.

They will be able to send full granularity digital data at 40 MHz to back-end; improved algorithms will deal with overlapping events deriving from increased pile-up. On FPGA based electronics, AI algorithms will be applied for measuring the energy.

Instead, in the central region of the hadronic calorimeter, the Tile Calorimeter (TileCal) [25] will have the same position and goal as in section 2.3.2.

In-detector and off-detector electronics will be fully replaced to improve the radiation tolerance and the performances at high pile-up.

Fig. 3.4 shows the scheme and position of the Tile Calorimeter in the ATLAS Phase-II detector. The Long Barrel (LB) and the Extended Barrel (EB) are both divided in A and C. The subdetector will capture roughly 30 % of the jet energy and, similarly to the previous runs, and it will be of crucial relevance in jet and missing energy measurements, jet substructure, electron isolation, and triggering.

The TileCal is built with lead absorbers and 460000 plastic scintillator plates and

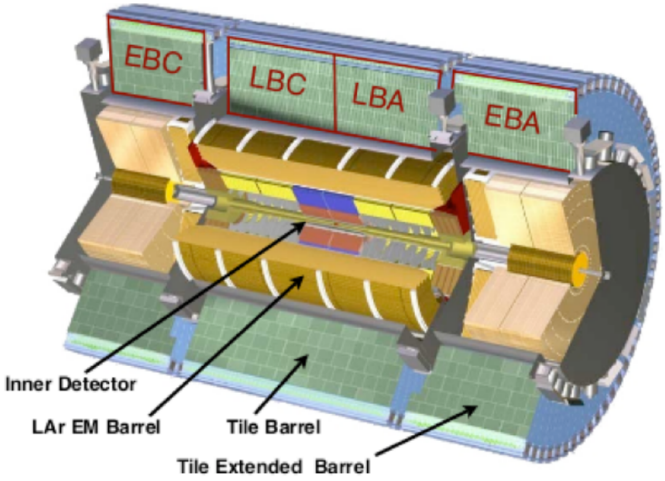

Fig. 3.4: Calorimeter system schema in the ATLAS Phase-II detector.

read out by wavelength-shifting fibers. The fibers are bundled in cells and read out by photomultiplier tubes, which extract data by the 4670 cells two at a time. The detector is separated in three sectors, "A", "BC" and "D" with respectively 1.4, 3.9 and 1.8 interaction lengths at $\eta = 0$. The granularity of $\eta \times \phi$ is approximately 0.1 x 0.1.

3.1.4 Muon Spectrometer

During Phase-II, the MS [26] will be significantly upgraded in terms of performances and precision in the tracking and momentum resolution, in order to cope with the operational conditions at the HL-LHC in Run 4 and beyond.

The main modifications will involve the barrel and end-cap detectors. A large fraction of the front-end and on- and off-detector readout and trigger electronics for the RPC, TGC, and MDT chambers will be replaced. The Level-0 trigger (see definition below) of the muons will use the MDT chambers to increase the p_T resolution. Additional RPC chambers will be installed in the inner region of the barrel ($|\eta| < 1$) to increase the geometrical coverage in the barrel and increse the trigger performances reducing the trigger fake rate.

3.2 Trigger and Data Acquisition

The HL-LHC upgrade represents a significant challenge for the ATLAS TDAQ system. To fully exploit the physics potential of the HL-LHC, the new trigger system will have to improve efficiency in selecting events of interest. In Run 4, considering the significant increase in luminosity up to $3000/4000 fb^{-1}$, exceptional performance from the trigger and data acquisition system will required to maintain a high maximum rate (a factor of 10 higher trigger rates compared to Run 3) and longer latency.

In a similar way to what has happened with most of the components of the ATLAS detector, the TDAQ system upgrade comes through different evolution designs and technologies stages, as documented in [27] and [28].

3.2.1 The architecture

The ATLAS TDAQ architecture is composed of three main systems: the Level-0 trigger system, the Data AcQuisition (DAQ) system, and the EF system (see Fig. 3.5).

It consists of a single Level-0 hardware trigger with a detector readout rate of 1 MHz and a maximum latency of $10\mu s$. The Level-0 trigger decision is made using calorimeter and muon information. Additional processors are added to implement sophisticated offline-like algorithms to provide additional background rejection. The EF system further selects events based on a commodity processor farm to reduce the event rate to 10kHz. Events selected by the EF system are then transferred for permanent storage.

The hardware-based Level-0 (L0) trigger system is composed of the Level-0 Calorimeter Trigger (L0Calo) and the Level-0 Muon Trigger (L0Muon) subsystems, the Global Trigger, the Muon CTP Interface (MUCTPI) and the Central Trigger Processors (CTP), as shown in Fig. 3.6.

The L0Calo sub-system is composed of the Phase-I electron Feature EXtractor (eFEX), jet Feature EXtractor (jFEX) and global Feature EXtractor (gFEX) complemented by a new forward Feature Extractor (fFEX) to reconstruct forward jets and electrons. The L0Muon sub-system includes the Barrel region of RPC and the End-cap Sector Logic processors of TGC, the NSW and the MDT Trigger processor.

The Global Trigger uses the high-granularity calorimeter information to perform

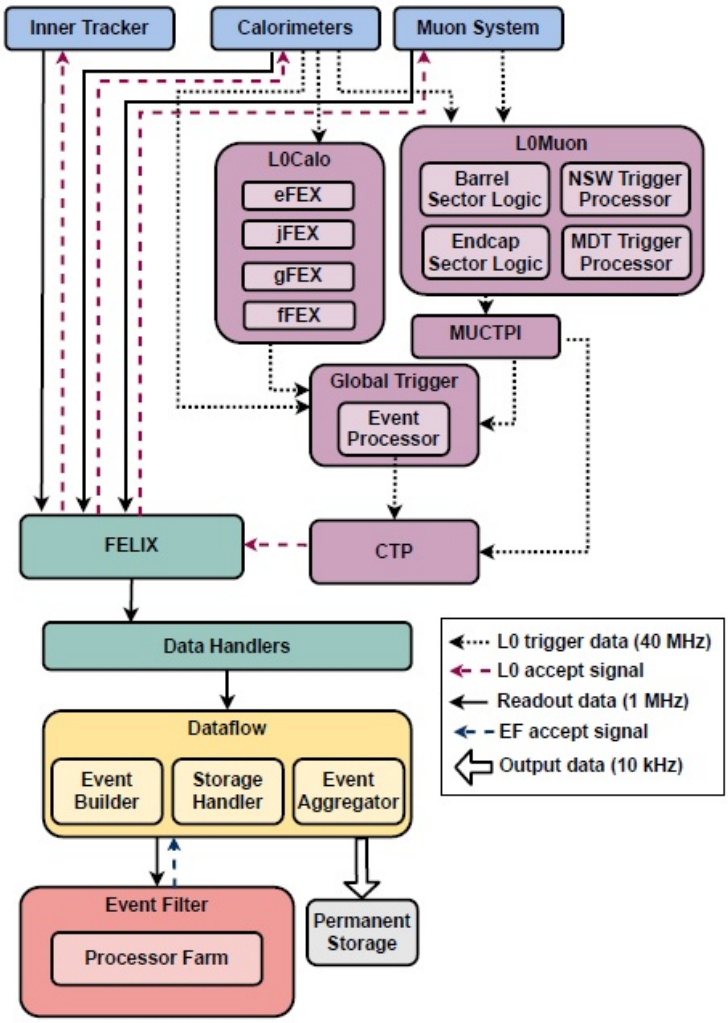

Fig. 3.5: The ATLAS TDAQ Phase-II architecture [28]. The black dotted arrows indicate the Level-0 dataflow from the detector systems to the Level-0 trigger system at 40 MHz, which must identify physics objects and calculate event-level physics quantities within 10 μs. The red dashed arrows indicates the result of the Level-0 trigger decision transmitted to the detectors. The trigger and detector data are transmitted through the DAQ system at 1 MHz, as shown by the black solid arrows. Direct connections between each Level-0 trigger component and the Readout system are suppressed for simplicity. The EF system reduces the event rate to 10 kHz; the selected events are transferred for permanent storage.

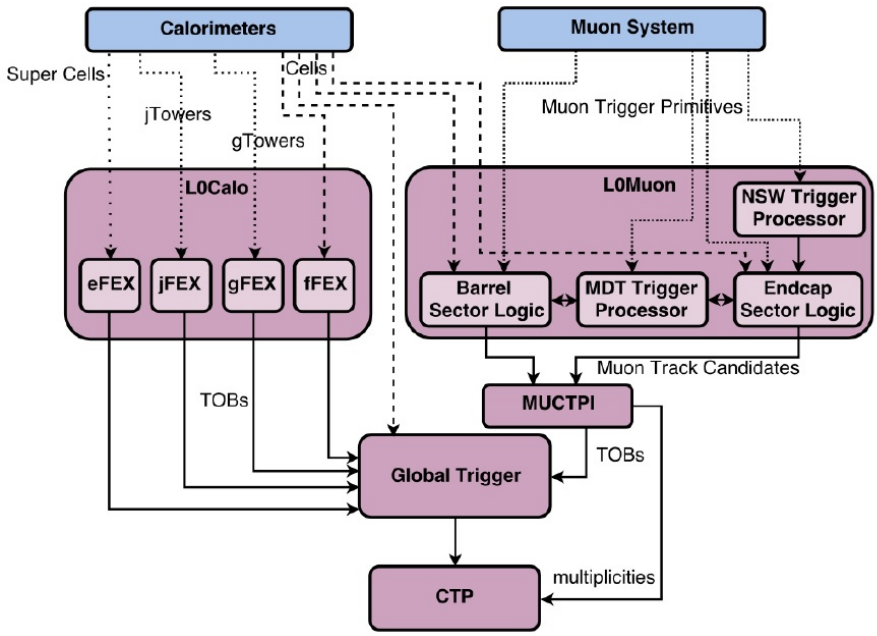

Fig. 3.6: ATLAS TDAQ Phase-II Level-0 trigger system.

offline-like algorithms, refining the L0Calo and L0Muon objects, calculate event-level quantities, and executes topological algorithms. The CTP forms trigger decisions based on information received from the Global Trigger and other sources, drives the Timing and Control (TTC) network to start the readout detectors process, applies prescale factors, and introduces dead time when necessary to avoid readout and front-end saturation.

The Level-0 trigger decision output is sent over custom point-to-point serial links at 1 MHz to the Readout system.

The Readout system is made up of Front-End Link eXchange (FELIX) cards, which provides an interface between the detector-specific custom point-to-point serial links and the commodity multi-gigabit data network downstream. Data are then passed to the Data Handlers, where detector-specific processing (such as formatting and/or monitoring) are implemented before buffering data in the Dataflow subsystem.

The Readout system is designed to accept 1 MHz event rate from the Level-0 hardware trigger, for a total bandwidth of 5.2 TB/s.

The Event Filter consists of a large processor farm with CPUs, GPUs, FPGA, capable of dealing with the 1 MHz input rate, and designed for fast software-based track

reconstruction.

Finally, the events selected by the EF system are transferred to permanent storage of the ATLAS offline computing system. The raw output event size is expected to be about 6 MB, while the total trigger output is feasible to be 10 kHz; thus, the total bandwidth out of the system is 60 GB/s.

3.2.2 The EF tracking decision process

The improved resource-efficiency in software tracking and heterogeneous computing systems, the design choices for the ITk and the decision to drop the need to design a low-latency custom-based track trigger for L1, a second stage of trigger selection introduced to reconstruct tracks and to reduce the readout rate to an acceptable level (< 1 MHz), have led the ATLAS collaboration to review and modify the EF tracking architecture, as described in the ATLAS TDAQ TDR Phase-II upgrade [28].

The design of the ITk has been significantly refined. In particular, the five-layer pixel system was extensively revised, to simplify the detector layout and construction and to optimize the ring design for tracking performance and to minimize the CPU required for reconstruction.

At the same time, a novel approach for track reconstruction for high pile-up events has been developed that takes full advantage of this new ITk design. Thanks to the latest reconstruction software tracking and the updated ITk layout, the time to reconstruct an event is 8 times less than that expected from the ATLAS TDAQ TDR system [27].

Data centers are moving towards a heterogeneous system where a CPU is no longer the primary compute unit for many workloads; they are integrating different types of computational units such as multi-core CPUs, GPUs, ASICs and FPGAs, to perform computations faster (and therefore with lower latency) and to achieve higher performance with lower power consumption to meet cooling, power and rack space limitations.

Three different alternative architectures for EF tracking at HL-LHC have been considered:

- Software-only: optimised dedicated software for the alternative EF tracking using the EF CPU farm.

- Heterogeneous commodity: mixed commodity hardware consisting of processors and accelerators (CPU/GPU/FPGA); the HT implementation on FPGA devices is a possible option.

- Custom: simplified hardware design obtained by using FPGAs as an alternative to AM pattern banks for pattern recognition, and re-optimization after removing the previous constraints on the HTT design as a consequence of upstream changes (ITk readout and L1 option cancellation).

The physical goals remain the same as documented in [27]. A combination of regional tracking at 1 MHz and full scan tracking at 150 kHz is used. The latter has been increased to conservatively cover for potential requirements for missing transverse momentum, particle flow reconstruction, large-radius tracking (LRT) for long-lived particle (LLPs), b-physics trigger and Trigger Level Object analysis (TLAs).

In December 2020, two task forces were created: one for the optimized Custom architecture and one for the Heterogeneous commodity architecture. The already existing organization for the Software-only architecture continued to develop its option. Each task force produced a technical solution to prove the feasibility of its specific approach.

The custom-based solution had no clear competitive advantage over a commercial solution. On the contrary, it carried a significantly higher risk, which is inherent in all custom developments and systems.

Speaking of cost, the cost of the FPGA-based heterogeneous commodity system is lower than that of the software-only equivalent but with potentially large uncertainties. The use of FPGAs as server accelerators in data centers is relatively new compared to their 30-year history in the electronics industry. For server use, this technology is packaged as acceleration cards that plug into a PCIe slot on the motherboard of a server. Such commercial accelerators are widely available. In general, their use benefits the CPU-based applications: latency decreases due to the high level of parallelism and buffering capabilities inherent in their architectures, so the EF Tracking system is not latency-limited. Power consumption is also reduced. Although the CPU-based EF is comfortably within the overall power budget, the use of accelerators can increase the flexibility to boost up the EF farm.

However, processing time is still an issue. The presence of large buffers does not affect the amount of data that must be processed in the unit of time, it clearly simplified the trade-off between server processing speed and farm size, i.e. the slower the EF node processes an event, the more nodes should be active to absorb the 10 kHz of events accepted at L0.

An independent ATLAS committee reviewed the reports of these task forces and recommended that a commercial solution for the EF tracking should be pursued, including the use of accelerators (FPGAs and/or GPUs) as an optimization to mitigate risks related to cost and power. The decision was based on technical feasibility, estimated tracking performance, operational procedures, opportunities for improvement, risks and resource requirements.

An ambitious programme is underway to monitor and evaluate commodity computing technologies and to further develop and optimize efficient algorithms for commodity platforms. A variety of high-performance accelerator technologies, system architectures, and implementation languages have been investigated.

Tracking demonstrators on various types of commodity hardware will be evaluated to verify that they meet all the necessary specifications. The final ET tracking system will be ready (implemented and installed) for Run 4.

In March 2022, the new baseline for the EF tracking project was delivered [28], amending the previous one.

The final Phase-II TDAQ architecture baseline is shown in Fig. 3.5. The black dotted arrows indicate the Level-0 dataflow from the detector systems to the Level-0 trigger system at 40 MHz, which must identify physics objects and calculate event-level physics quantities within 10 µs. The result of the Level-0 trigger decision (L0A) is sent to the detectors as indicated by the red dashed arrows. The resulting trigger data and detector data are transmitted through the DAQ system at 1 MHz, as shown by the black solid arrows. Direct connections between each Level-0 trigger component and the Readout system are suppressed for simplicity. Events that are selected by the EF trigger decision are transferred to the CERN farm for permanent storage.

Chapter 4

The Hough Transform

The HT has been investigated in the context of track identification for charged particles traversing the ATLAS ITk detector for the HL-LHC.

For the Phase-II ATLAS TDAQ upgrade the HT algorithm was implemented on an FPGA device as an alternative to the AM ASICs solution for the pattern recognition in the HTT system. After the decision to move to a heterogeneous commodity architecture relying on commercial hardware for the EF tracking, an FPGA-based demonstrator is being evaluated for the ITk pattern recognition stage using the HT as a possible choice for future implementation.

The electronic group to which I belong participated in the proposal of the heterogeneous commodity task force in the HTT project. The R&D activities included the development of a firmware design on a hardware demonstrator, used as a proof of concept, and a software capable of testing the performance of the firmware. The development of this software is part of my PhD activity, emulating the HT firmware implementation.

4.1 HT overview

The HT was originally developed by Paul V. C. Hough for particle physics to detect particle tracks in photographic plates from bubble chambers in the late 1950s [30]. Since then, the method has been generalized and is now widely used in many fields, especially in computer vision for automated shape and feature recognition. It has been optimized

in several variants for pattern recognition on digital images to detect features such as linear segments (especially straight lines), circumferences, and in general any shape with a parametric representation [32]. Today, it is so popular that Google Scholar returns about 230.000 results for the search term "Hough Transform".

The original idea is to map each point of a two-dimensional (2D) plane, such as a pixel, to a set of points (e.g. a line) in a parameter space, namely the Hough space.

Let (x_i, y_i) be a point in the xy-plane and consider the general line equation:

$$y_i = ax_i + b \tag{4.1}$$

There are an infinite number of lines passing through (x_i, y_i) and all of them satisfy the equation 4.1 as the values of a and b vary. However, writing this equation as:

$$b = -x_i a + y_i \tag{4.2}$$

and considering the a-b plane (the *parameter space*) gives the equation of a single line for a fixed point $(x_i,\ y_i)$. Furthermore, a second point $(x_j,\ y_j)$ also has a single line associated with it in the parameter space, which intersects the line associated with $(x_i,\ y_i)$ at a certain point $(a^{'},\ b^{'})$, in the parameter space, where $a^{'}$ is the *slope* and $b^{'}$ is the *offset* (intercept) of the line containing both $(x_i,\ y_i)$ and $(x_j,\ y_j)$ in the x-y plane, assuming the lines are not parallel. In fact, *all* points on this line (collinearity of these points) in the x-y plane have lines in the parameter space that intersect at $(a^{'},\ b^{'})$. Fig.4.1 illustrates these concepts.

However, there is a problem when a, the slope of a line, approaches infinity as the line approaches the vertical direction. To get around this, the *normal representation* of a line is used:

$$\rho = x\cos\theta + y\sin\theta \tag{4.3}$$

Fig.4.2 illustrates the geometric interpretation of the parameters θ and ρ [37]. Each sinusoidal curve represents the family of lines passing through a given point (x_k, y_k) in the x-y plane. The intersection point $(\rho^{'}, \theta^{'})$ corresponds to the line that passes through both (x_i, y_i) and (x_j, y_j).

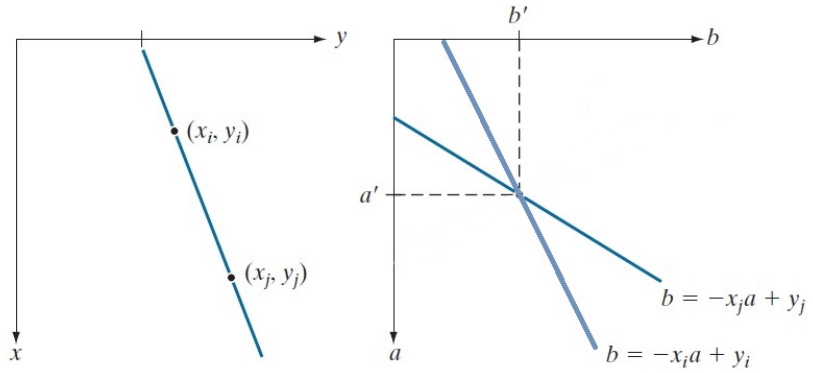

Fig. 4.1: (a) x-y plane. (b) Parameter space.

The ρ-θ parameter space is subdivided into the so-called *accumulator cells*. The accumulator corresponds to a two-dimensional 2D histogram, in which the cell at coordinates (i, j) corresponds to the square associated with parameters-space coordinates (ρ_i, θ_j). A voting mechanism is performed: first, these cells are set to 0; then, for each $(x_k,\ y_k)$ in the xy-plane, θ is varied in all allowed θ-axis subdivisions, and the corresponding ρ values are calculated according to the formula $\rho = x_k \cos\theta + y_k \sin\theta$.

The resulting ρ values are then rounded to the nearest allowed cell value along the ρ-axis, and the corresponding cell value in the accumulator is incremented by one unit (the so-called *votes*). At the end of the procedure, a value of K in a cell A(i,j) means that K points in the x-y plane lie on the line $\rho_i = x_k \cos\theta_j + y_k \sin\theta_j$.

The coordinates of the cells in the accumulator with the most votes (global maximum value) correspond to the candidate features (slope and offset of the lines in the x-y plane). In other words, local maxima in the accumulator indicate the parameters of the most prominent lines in the input image. Potential candidates can also be found most easily by applying a *threshold*, i.e. values equal to or greater than some fixed percentage of the global maximum value.

The number of subdivisions (bins) in the ρ-θ plane is very important in the trade-off between collinearity accuracy, data storage space, and processing speed.

The number of computations is linear in n, the number of points in the x-y plane. With respect to a straight line detection, 2 parameters define a line: slope and offset. Therefore, a 2D parameter space is required for straight line detection. If N is the size

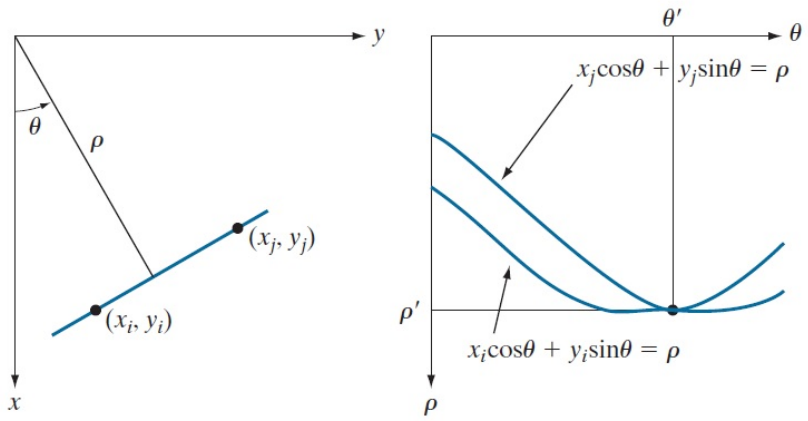

Fig. 4.2: (a) (ρ,θ) parametrization of a line in the x-y plane. (b) Sinusoidal curves in the ρ-θ plane

of each dimension of the parameter space, an order of $O(N^2)$ is required to store the parameter space. In addition, it takes time of order $O(N^2)$ to vote for each point and to search in the accumulator array.

The HT is applicable to any equation of the form $g(\mathbf{v},\mathbf{c}) = 0$, where $\mathbf{v}$ is a vector of coordinates and $\mathbf{c}$ is a vector of coefficients. For example, the points lying on a circumference:

$$(x - c_1)^2 + (y - c_2)^2 = c_3^2. \tag{4.4}$$

can be determined by using the approach just described. The three parameters c_1, c_2, c_3 result in a 3D parameter space with cubic cells. Of course, the HT implementation is more expensive in terms of computer memory and time.

Fig.4.3 shows an HT example of circle identification with known radius; in this case, the search is reduced to 2D. Each point in the x-y plane (left) belonging to a circumference of radius R, generates a circumference in the parameter space (right) of radius R. The intersection of these circumferences gives the coordinates of the centre of the circumference to which the initial points belong.

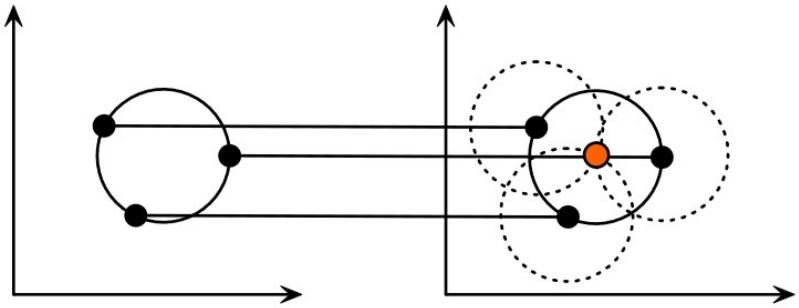

Fig. 4.3: Example of HT for circumference identification.

4.2 HT for particle tracking

A charged particle describes a helicoidal track in the ITk due to the solenoidal magnetic field that is present into the detector, so a circular arc is traced out in the transverse x-y plane. This track is described by five parameters (d_0, z_0, ϕ, θ, q/p_T) and a reference point, the average position of the pp interactions (beam spot position) [29]. The parameters are described as follow:

- d_0: the transverse impact parameter, defined as the distance of the closest approach in the transverse plane
- z_0: the longitudinal impact parameter, defined as the z-position of the track's closest approach to the beam pipe
- ϕ: the azimuthal angle of the track momentum at the reference point
- θ: the polar angle of the track momentum at the reference point
- q/p_T: the electric charge q, in units of elementary charge, divided by p_T, the transverse momentum

The track parameters are illustrated in Fig. 4.4, except q/p_T which is proportional to the radius of curvature of the track.

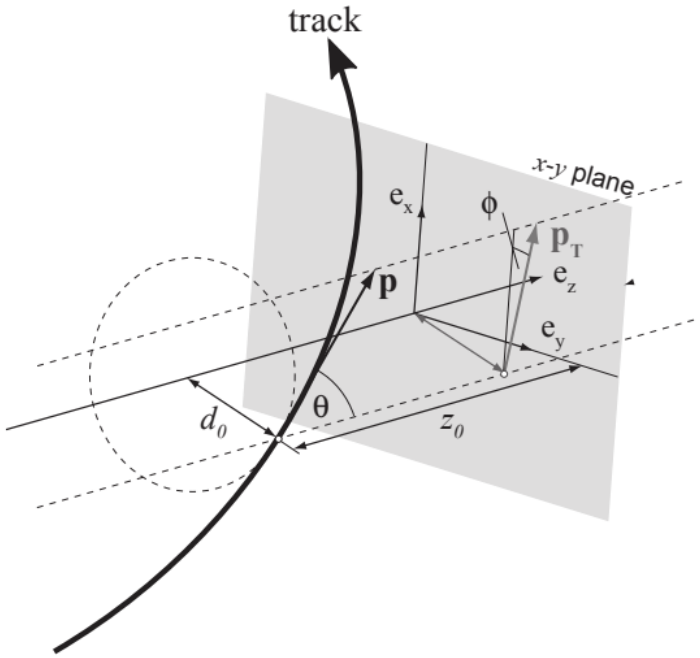

Fig. 4.4: Helix parameters used to describe particle tracks [29].

In the presence of any electromagnetic field, charged particles are subject to the Lorentz force:

$$\vec{F} = q\vec{E} + q\vec{v} \times \vec{B}, \tag{4.5}$$

where q is the electric charge of the particle, E is the electric field, v is the speed of the particle and B is the magnetic field. The ITk electric field can be considered negligible and the magnetic field along the z direction is uniform ($\vec{B} = B\hat{z}$). Choosing cylindrical coordinates so that $\vec{v} = v\hat{\phi}$, the Lorentz force then becomes:

$$\vec{F} = qvB\hat{r}. \tag{4.6}$$

Hence, if the momentum of the particle remains constant, its trajectory is a circumference and the force can be described as a radial acceleration:

$$\vec{F} = \frac{p_T v}{r}\hat{r} \tag{4.7}$$

where r is the circumference radius and p_T is the transverse component of the relativistic momentum of the particle.

Substituting Eq.4.6 into Eq.4.7:

$$p_T = qBr \tag{4.8}$$

where p_T is in $Kg{\cdot}m/\mathrm{s}$ and q in *Coulomb*. Using units of elementary charge e and GeV/c:

$$p_T = \frac{cqeBr}{e} \cdot 10^{-9} = cqBr \cdot 10^{-9} \approx 0.3qBr \tag{4.9}$$

where $c \approx 3 \times 10^8 ms^{-1}$, p_T is in GeV/c, q in e, B in T and r in m.

The equation above is the relationship between the transverse momentum of a particle and the radius of the circumference it describes in a uniform magnetic field.

The HT algorithm can be adapted to find a curved track with some approximation.

If the interaction vertex is constrained to the origin of the detector coordinate system, the track through a point (the hit cluster) with polar coordinates (r, ϕ) can be described as:

$$\frac{qA}{p_T} = \frac{sin(\phi_0 - \phi)}{r} \approx \frac{\phi_0 - \phi}{r} \tag{4.10}$$

where q is the sign of the electric charge, r is the radius in mm, A is the constant factor for a 2 T magnetic field ($A = 3 \times 10^{-4}\ GeVc^{-1}\ mm^{-1}e^{-1}$), p_T is the transverse momentum of the particle and ϕ_0 is the azimuthal angle of the particle track at the origin.

If we are interested in a small ϕ_0 region, the equation can be simplified using $sin(\phi_0 - \phi) \approx (\phi_0\text{-}\phi)$.

In order to include additional z-information due to the spread of the beam spot, the 2D scan can be repeated in *slices* along the z-direction.

4.2.1 HT implementation

The HT was implemented as a pattern recognition step in the analyses of the clustered hits coming from the ATLAS ITk detector, to identify combinations of hits associated to particle tracks.

Interest in HT has increased in recent years due to the advantages in massively parallel, high-throughput computing derived from GPUs and FPGAs [34] [35]. Some HT operations are very resource-intensive, so the possibility to use high speed hardware and optimizations are crucial for its possible adoption in the final configuration.

The basic concept is that each hit cluster on a curved track defined by (r, ϕ) in a transverse x-y plane is transformed into a straight line (see eq.4.10) in the qA/p_T-ϕ_0 plane; this parameter space is commonly referred to as the *accumulator*.

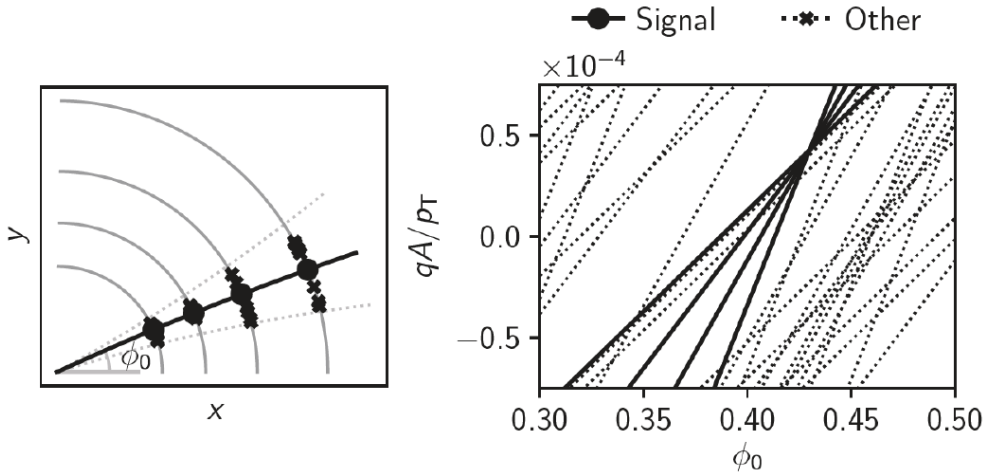

Fig. 4.5: Left: one quadrant of the ITk transverse plane. Right: Hough parameter space.

Fig. 4.5 illustrates what happened when the HT is applied to several clustered hits in the ATLAS ITk detector. The figure on the left represents one quadrant of the x-y transverse plane with a signal track. The layers of the tracker, where the hits are registered, are drown in gray. In each layer it is possible to distinguish the so-called *clusters*: black dots are clusters along the signal track, while black crosses are clusters not associated with it in the range $0.3 < \phi_0 < 0.5$.

Hit clusters are considered instead of a single hit: a particle passing through a pixel/strip detector generates multiple adjacent hits, therefore the centre of the clus-

ter of the hits provides a better estimate of the position of the point where the particle crossed the detector layer.

Clusters belonging to the same track are transformed into straight lines through the HT, which intersect at the same point in the HT parameter space. The coordinates of this point correspond to the parameters (slope and offset) of the reconstructed candidate track. Clusters not coming from any track, or belonging to tracks outside the considered parameter ranges, form randomly crossing lines in the accumulator.

The whole set of operating tracking layers must be considered, and the candidate tracks have to be found with a match with all of these.

4.2.2 The accumulator

The accumulator has a central role in the HT. The main operations performed by the HT include accumulator creation, initialization, filling and selection of track candidates.

The accumulator is implemented as a two-dimensional (2D) histogram for a given binning in qA/p_T and ϕ_0 and a certain $\eta \times \phi$ region. For each qA/p_T bin, all ϕ_0 values consistent with a point in the qA/p_T range are filled. For complete $\eta \times \phi$ coverage ($|\eta| < 4.0$), a certain number of regions must be defined across the detector; for example, if $\eta \times \phi = 0.2 \times 0.2$, 1280 regions must be considered.

The nominal number of tracker layers used is 8, although this number can be optimized in the future. With a reduced number of layers the combinatorics are smaller and a pre-selection can be applied more quickly. In the central barrel, these layers include the outermost pixel layer, the inner side of the first strip layer, and both sides of the remaining strip layers. This choice is motivated by occupancy: only one pixel layer is used, the one with the lowest occupancy, and then fewer hits to readout, resulting in faster readout and fewer combinations generated. The remaining layers are from the strips, that are larger and have a low occupancy because they are a bit far from the interaction region.

The accumulator can contain different information depending on the needs. In a simple case, the accumulator is filled by entering the cluster coordinates in the equation 4.10 and sweeping ϕ_0 over the range defined by RoI to obtain the coordinate qA/p_T. In this case, each bin of the accumulator carries the number of layers containing clusters:

it is filled for each hit from the choosen layers so that when at least 8 clusters are found in the same bin, this is an indication of a good 8-hit candidate track. Finding a feasible candidate track (called *road*) results in applying a threshold to that number.

In other implementations [36], it contains an 8-bit number (every digit is a *layer bit*) to keep track of which of the eight layers have been hit in the parameter space that it spans, and a list of all clusters passing through that particular bin.

It is possible to implement other algorithms for the HT to obtain better signal efficiency and at the same time background suppression, but keeping in mind that the HT has to be implemented in hardware, the easier scheme is to choose.

After the accumulator has been filled, some criteria are applied to select good track candidates and discard unwanted tracks, for example, bins with less than 7 layers hits can be discarded; the surviving bins are track candidates.

At the end of all the selection steps, the candidate track is used as seed of a Kalman Filter (KF) [28], so that all the hits selected and the hits on the remaining layers are fit in a KF. The track-finding process for which the HT is studied, is very useful to reduce the number of combinations to pass to the track fitter, so it reduces the load of the KF algorithm which otherwise explodes.

An example of the classical HT implementation for a single muon event is shown in Fig. 4.6. The clusters belong to the whole set of tracking layers used. The yellow area represents the region of the parameter space where the highest number of hits are described by the same curve, i.e. it represents the best set of parameters for the candidate track. The intensity of the colour indicates that the given set of clustered hits probably comes from the same charged particle track. The single muon candidate is reconstructed at $q/p_T = 0.52$ and $\phi_0 = 0.43$ in a single z-slice.

However, with the inclusion of pileup it is more difficult to distinguish the muon signal from the background, as shown in Fig. 4.7, where the same muon has been embedded in minimum bias events with 200 pp interactions. Cluster combinations can also be observed in the same image.

Applying a threshold decision to find good track candidates is harder in this case due to the huge number of overlapped clusters. The pp collisions in ATLAS at the HL-LHC are expected to be spread uniformly over 300 mm along the beam line. The HT

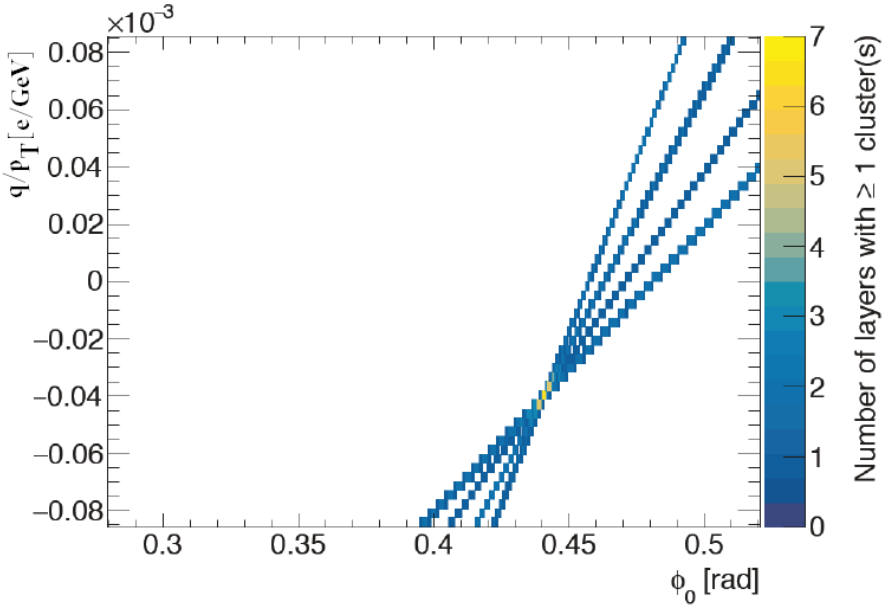

Fig. 4.6: Accumulator filled with clusters from a single muon event without pileup.

implementation looks at the projection in the transverse plan, so it is hard to separate tracks with similar p_T and ϕ_0 that originate from different points along the beam line.

Z-slicing

The situation depicted in the previous section gets better if the RoI is splitted into smaller slices along the z-axis. This splitting technique improves the efficiency in finding good track candidate and rejecting unwanted clusters, reducing the number of clusters to be processed by the HT and the combinatorial background, and thus reducing the number of potential roads.

Two approaches can be used to separate tracks along the z-axis: the first is to parameterize the track in ϕ_0 as well as z. In this case, another dimension is added to the accumulator, with the result of squaring the number of computations.

The second approach is to first find track candidates in the transverse plane, than search for lines in the r-z plane. This additional step increases the execution time by about 30% for single muon embedded in 200 minimum bias events, proportional to the number of hits passing the first stage, because the clusters are read out in a serial way.

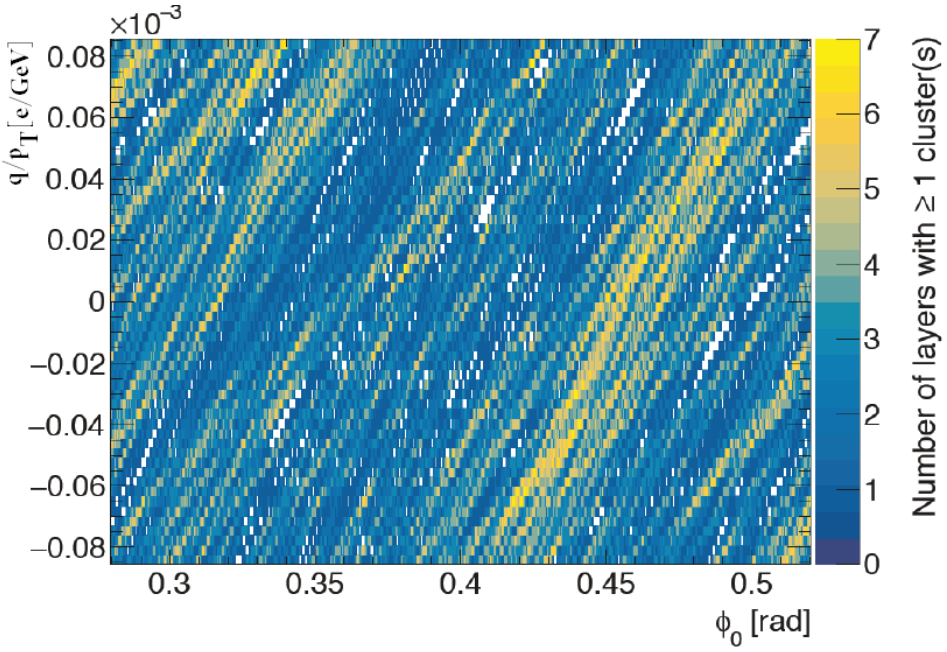

Fig. 4.7: Accumulator filled with clusters from a single muon event with pileup.

In addition, the low resolution of the strip layers in z and the cost of filling separate accumulators for each slice, by sorting the clusters according to their z coordinate, must be taken into account. Nearby splits overlap slightly along z-axis, so the same hits/clusters can appear in several accumulators.

It was determined [28] that slicing along the z-axis at a radius of 562 mm (the so-called *key-layer*) reduced the duplication compared to slicing in z at a radius of 0 in the central η region; in this case, six slices were chosen (Fig. 4.8.

Fig. 4.9 shows the result of splitting the accumulator of Fig. 4.7 into four slices; here, it is easier to extract candidate patterns (associated with muons) from the background.

The above description easily fits into an FPGA. The same cluster may appear in several accumulators due to overlapping of nearby splits; this fact should be taken into account.

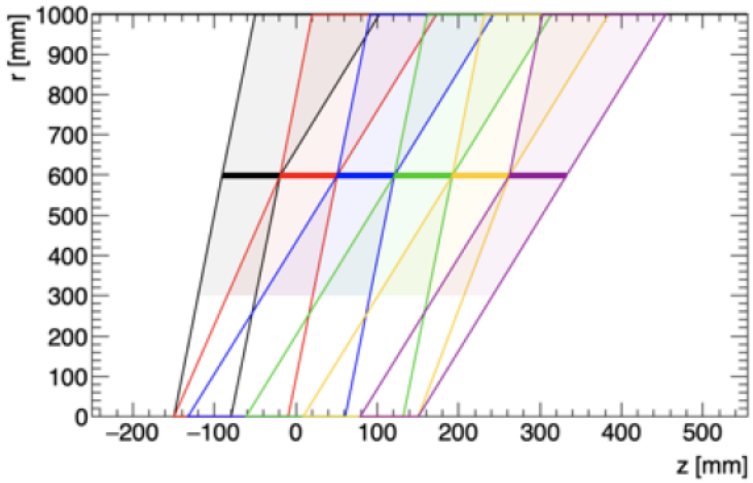

Fig. 4.8: RoI is sliced up along z-axis and nearby splits overlap slightly. Best *key-layer*, the layer in which the overlapping is minimized, appears to be the outer short strip layer, in case of 6 layers.

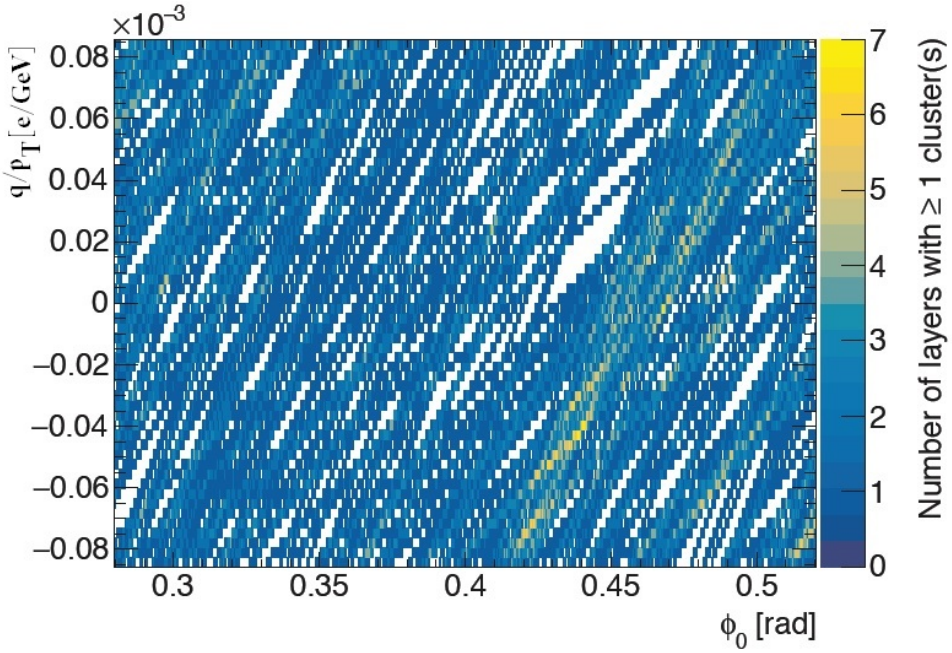

Fig. 4.9: One out of four z-slices of the accumulator of Fig. 4.7 .

Bins selection criteria

More sophisticated bin selection schemes in the accumulator can lead to significant improvements in signal efficiency and background suppression.

The simplest bin selection is the single bin with associated clusters belonging to different layers whose number is below a certain threshold. Applying a threshold to a set of adjacent bins is an interesting criterion for cross-checking the presence of clusters in them. In particular, it has been shown to be effective when considering 5 neighbouring bins along constant q/p_T: the central bin must have 8 clusters, the left and right bins at least 7, the left-left and right-right at least 6, and all the clusters for a given bin must belong to different layers. For example, if three clusters come from the layers 0, 1 and 2 and three clusters all come from layer 4, the total number of clusters will be four and not six, because the two clusters coming from layer 4 after the first one are not taken into account.

Although five bins are considered, only the ϕ_0 and q/p_T values related to the central bin are selected.

Fig. 4.10 shows the accumulator and 5 adjacent bins. Each bin stores the number of different layers that have been hit by the clusters coming through that bin.

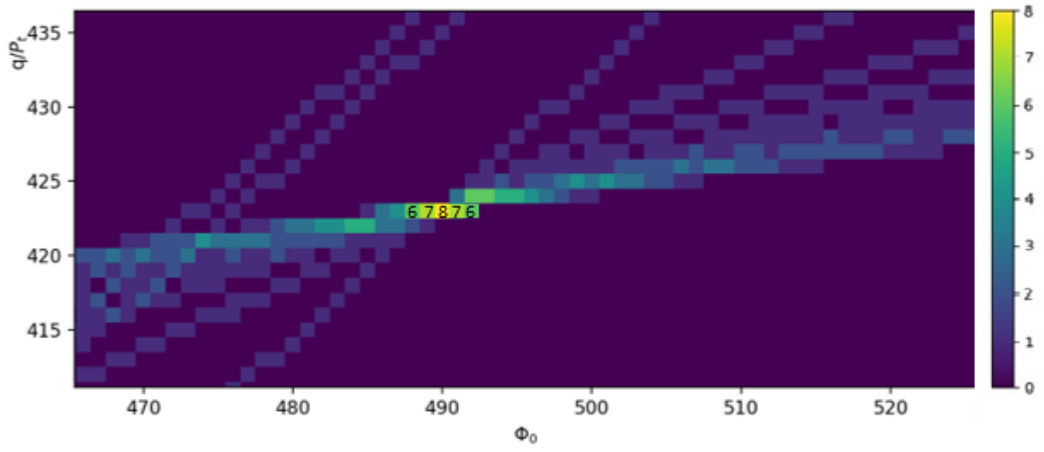

Fig. 4.10: Plot of 5 adjacent bins in a zoomed part of a 600x1100 accumulator for 160 (r, ϕ_0) single muon values, showing the 6-7-8-7-6 sequence.

4.2.3 HT tuning and optimization

High track efficiency and as much as possible suppression of low-p_T background can be obtained by tuning some parameters in the HT algorithm, such as:

- the number of bins in q/p_T
- the number of bins in ϕ_0
- the number of slices along the z-axis
- selection of layers to use
- hit padding
- hit extension
- accumulator threshold
- methods for track candidate selection (*stub-finding, space point formation*)

The optimal number of bins is strongly related to the geometry of the detector, its resolution, and the layer configuration used, while the optimal number of z-slices is mostly affected by the pileup conditions.

Hit padding is the technique in which extra bins are added beyond the nominal ranges to extend the accumulator and not to lose efficiency near the edges. Fig. 4.11 (left) shows padding bins drown in gray.

Hit extension is the technique where a certain number of extra bins on either side in the ϕ_0 direction are filled in the accumulator, to maintain high efficiency, especially for non-prompt tracks (Fig. 4.11 (right)).

A hit extension equal to 0 means that only the bin in the accumulator from Eq. 4.10 is filled; a hit extension of k requires that a certain bin is filled as well as the k adjacent bins on each side along ϕ_0 and at the same q/p_T value.

k value may be different for each accumulator. In one of the first implementation the following configuration has been chosen: a hit extension equal to 2 in the only used pixel

layer, 1 in the first used strip layer, and 0 elsewhere, as well as a hit padding of 6 (2) in the ϕ_0 q/p_T direction of the accumulator image.

Additional strategies for fake rejection, duplicate removal, and fitting are required to lower the number of these roads (and thus track candidates) to a manageable quantity for the next precision fit step.

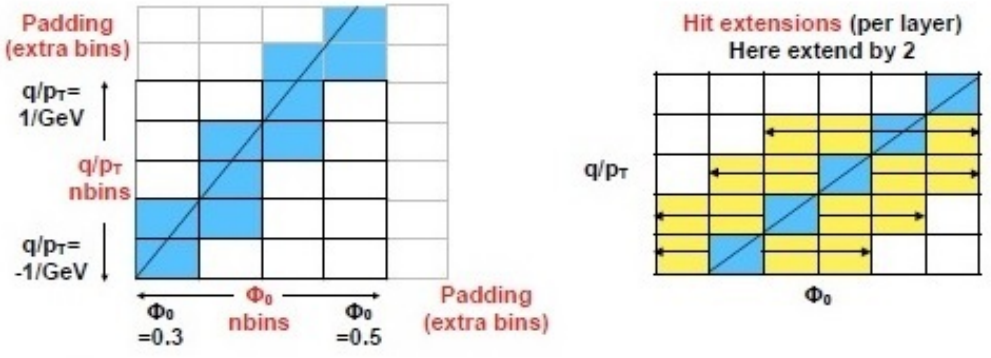

Fig. 4.11: HT optimization options Hit padding (left), Hit extension (right).

Chapter 5

A software development tool for the HT

Two years ago, the working group I belong to started to develop a firmware design for the Event Filter of the Phase-II ATLAS TDAQ, and a hardware demonstrator on FPGA using the HT algorithm as a pattern recognition strategy. The aim was to evaluate an alternative solution to the Associative Memories (AM) ASICs.

In this context, my personal contribution concerns the development of a software tool that emulates the firmware based on the HT algorithm and compatible with the ATLAS TDAQ system. The aim is to test the performance of the firmware and the capability of the hardware demonstrator to perform pattern recognition with competitive performance with respect to AM-based and software-only implementations. To achieve these results, simulated physics data are produced and encoded in an appropriate format; these data sets are referred to as *Test Vectors* (TV).

The development of the HT software tool was part of my ATLAS Qualification Task, which regarded the TDAQ Test Vector generation compatible with the ongoing Phase-II Hardware Trigger Upgrade scenarios.

The results obtained are published in the articles [41] and [42] and reported in the poster session at the TWEPP 2022 [43] and ACAT 2022 conferences [44].

5.1 The HT Model

In this study, a simplified version of the HT is used. Small ϕ_0 regions are the RoI, and under this condition $sin(\phi_0 - \phi) \approx (\phi_0 - \phi)$, so the HT formula becomes eq. 4.10 with $r = \sqrt{x^2 + y^2}$ and $\phi = \arctan(y/x)$. Only the orthogonal tracks with respect to the beam line are considered, they represent the particle paths to be recognized.

The HT implementation on FPGA offers some advantages, first of all low and fixed *latency*, defined as the number of clock periods from when the entire event is loaded to when the first set of clusters belonging to the first road is sent out. The latency time increases linearly with the number of input data, while the growth trend is greater as the input data increases with a combinatorial algorithm. In addition, the HT is much more tolerant of missing data that do not perfectly match with the default straight line, due to the tunable resolution of the accumulator bins. As a result, using the HT on an FPGA is expensive when considering a limited input data set, while it is still convenient when considering a large input data set due to its improved performance. In addition, compared to other similar devices such as CPUs and GPUs, FPGA components can support the implementation of solutions characterized by low power budget and high data rate.

5.2 Initial Development

This work started with the development of a general framework, a parametric software tool for low-latency applications geared towards FPGA implementations, based on the HT for the recognition of straight lines patterns.

In high energy physics applications, straight lines represent possible overlapping tracks of charged particles with a certain amount of undesired data as background.

Here, background noise is imagined of any origin, composed of traditional white noise or unwanted physical data coming from particle collisions, and the HT implementation is seen as a reasonable solution for separating useful information from the background.

The TV are generated by the tool itself, compatible as much as possible with the physical data and the firmware constraints, with the goal to extract track candidates and comparing the results with the ones obtained from the firmware.

5.2.1 Development Environment and Parameters

The HT general framework is initially developed as a stand-alone software tool using Python version 3 in the Anaconda environment [47]. This Integrated Development Environment (IDE) is one of that recommended by the ATLAS collaboration, providing a complete set of Python libraries for scientific computing and numerical recipes. Python by itself provides modules to read, extract and deal easily with ROOT data, the official ATLAS data format.

The tool is developed on Windows and Linux (Ubuntu) operating systems, but it can run on any other platform (Mac, other Linux operating systems), due to the portability of the code.

The software tool is parameterised with respect to several variables, some of which are based on the ATLAS ITk inclined geometry 3.2. The most important are:

- number of layers
- number of clusters per layer
- number of roads
- number of events
- % of fakes
- accumulator dimensions (q/p_T and ϕ_0 bin numbers)
- ϕ_0 in $[0.3, 0.5]$ rad
- q/p_T in $[-1, 1]$ GeV^{-1}
- accumulator threshold
- hit extension

All parameters are configured in a JavaScript Object Notation (JSON) [50] configuration file, which can be easily changed as required without affecting the code, resulting in a flexible and reusable system.

5.2.2 The Development Tool blocks

The Development Tool consists of 5 main blocks (Fig. 5.1):

- *Data Generation*
- *Data Extraction*
- *Accumulator Creation*
- *Road Extraction*
- *Statistics*

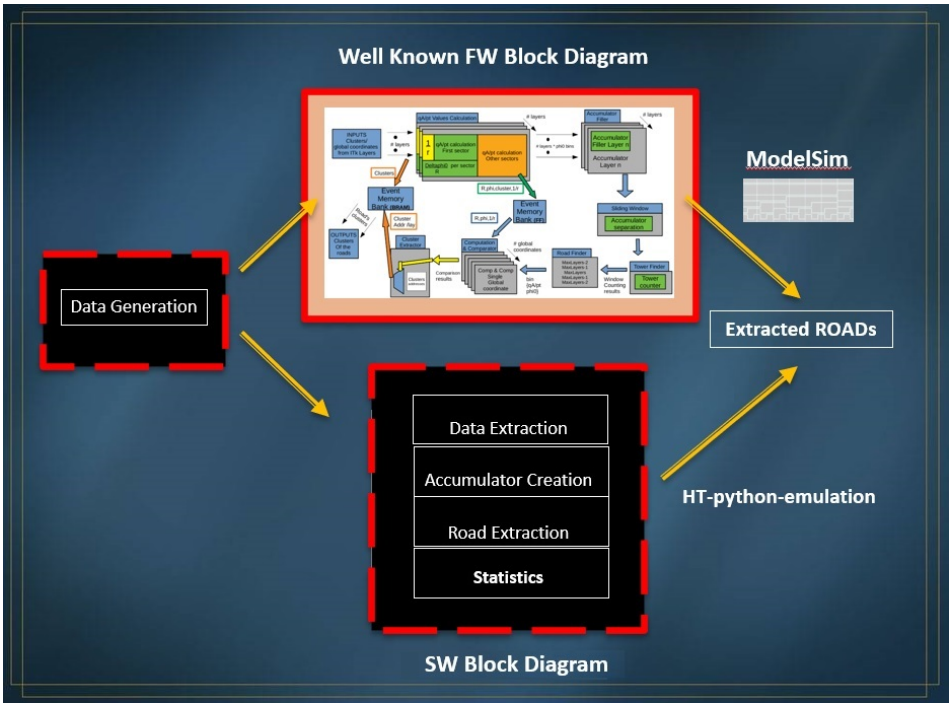

Fig. 5.1: SW and FW logic block diagrams.

In the *Data Generation* block, the (r, ϕ) pairs that identify the dummy data are generated randomly but according to the HTT TV data format. r ranges from 300 mm to 1000 mm and covers part of the Pixel and Strip detectors belonging to the 8 layers, following the ITk geometry. For example, if 10 roads are generated and each of them contains 8 layers with 2 clusters each, the dummy data are composed of 10x2x8=160 clusters. The case of fixed r is not considered.

Fake roads mimic the background data, here called *noise* as above; they are randomly selected in the ϕ_0 and q/p_T ranges, to simulate reasonable data. If no fake roads are generated, in the previous example there are 160 initial clusters. If 40% of the initial clusters are dummy data, 266 dummy input data must be generated. Fake data belonging to fake roads (i.e. roads other than the expected ones), are taken into account or not (the so-called *white noise*), as well as fake data generated outside the RoI. This will be discussed in detail in the data analysis section 5.2.3.

These data are written to a file and then given as input to both the Data Extraction block of the software tool and to a physical system implemented on an FPGA hardware device, to compare the output results and validate the emulation process.

In the *Data Extraction* block, input clusters are read and prepared for the next step.

In the *Accumulator Creation* block, a 3D accumulator is filled by applying the HT formula. Each (r, ϕ) pair of data input defines a straight line in the Hough space, recording in each vector of the cells touched by the line the layer position to which the input cluster belongs. Each cell of the accumulator is a vector with a 0/1 value per each layer, initialized to 0. If the value of the layer vector position is 0, it is set to 1, if it is already set to 1, it does not change. The filling process for the entire Input Set is named *Forward Process*.

An example of a 3D histogram representation of a 64x1200 accumulator with a unique 8 value (all the layers fired) that identifies a single road is shown in Fig 5.2. The plot is binned on x-axis along ϕ_0 bins and on y-axis along qA/p_T bins.

In this picture, the accumulator can be imagined as a set of individual towers, each consisting of a maximum of 8 elements, and containing the information of whether that tower has been touched by at least one straight line per different element.

For example, if a cluster belonging to the 6^{th} layer gives rise to a straight line crossing

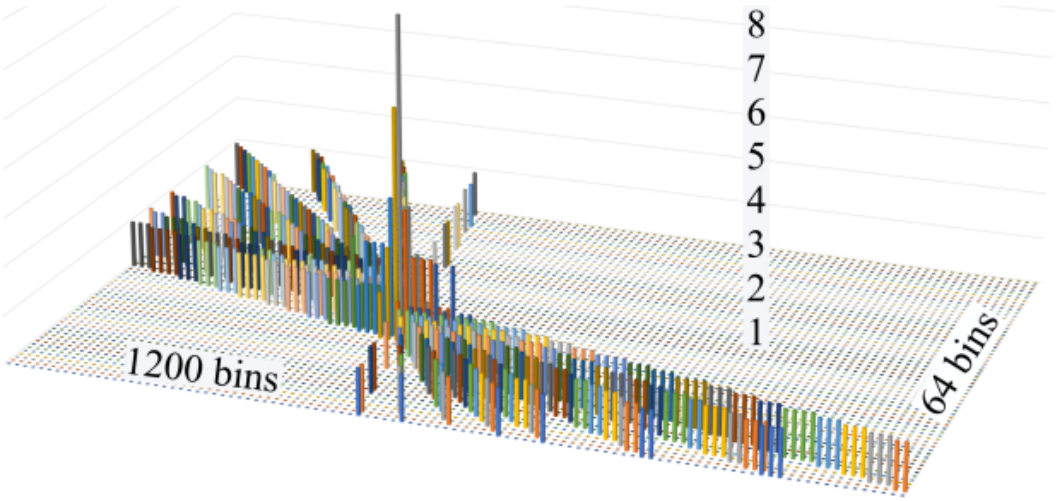

Fig. 5.2: 3D plot accumulator for a single extracted road.

the n^{th} row and the m^{th} column of a 64x1200 accumulator, then the tower addressed at the n^{th} row of the 64 qA/p_T and at the m^{th} column of the 1200 (ϕ_0) is updated with a 1 in its 6^{th} position. This is repeated for all the towers touched by the line. A 3D view of the accumulator with a track candidate is shown in Fig. 5.3.

On an FPGA device, this process is performed in parallel, within each clock cycle, for each ϕ_0 bin value and for 8 inputs, which requires a large amount of electronic resources to fit in.

In a second step, an *annotated heatmap accumulator* is constructed: each cell contains the number of the fired layers. The bins of the cells containing a value equal to or greater than the threshold parameter are good offset and slope candidates for potential roads, and are taken apart.

In the *Road Extraction* block, the (ϕ_0, q/p_T) pairs belonging to the road candidates are extracted from the accumulator and compared with the original ones. A road is considered found if the differences between the two pairs of values are less than the bin dimensions for ϕ_0 and q/p_T, respectively.

The information about how the input set is generated is available, and it gives a measure of the quality of the implementation of the HT algorithm. In fact, if the number

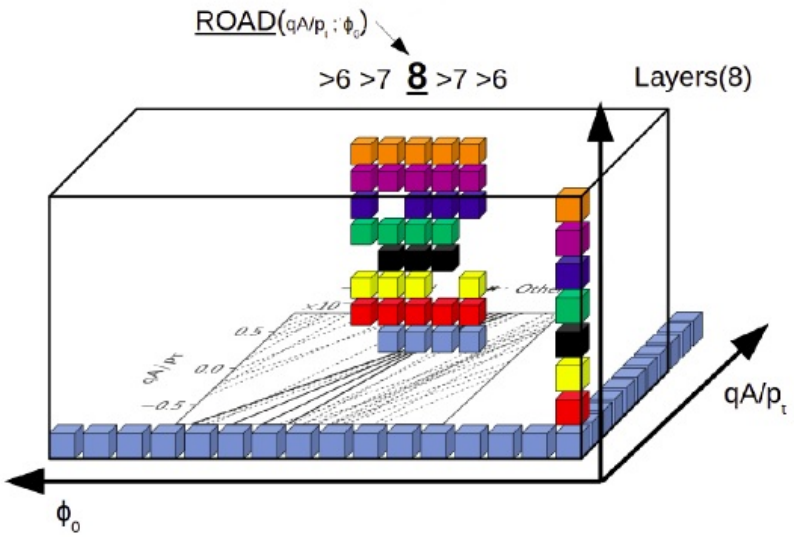

Fig. 5.3: 3D view of an accumulator with a track candidate.

of the roads found is equal to the original number of roads, then the HT algorithm is well implemented. Of course, in a real case the number of the original roads is unknown, so all the extracted roads that satisfy a predefined rule must be kept aside and selected by further subsequent cross-checking processes.

The input data set is then scanned back to identify which (r, ϕ) initial clusters are associated with the extracted candidate roads. This task requires the HT formula to be applied in a reverse mode (*Backward Process*). These clusters are kept aside, there is a high probability that they belong to the original input tracks.

Cluster extraction is done at the end of the accumulator creation rather than during its construction, to mimic the firmware behaviour; in fact, on an FPGA device this task can be done in a parallel pipeline for the entire input set. In firmware, it is too costly to use memories to compute and store clusters data each time a candidate (ϕ_0, q/p_T) pair is encountered during the accumulator fill process.

It may happen that background data leads to a fake road being recognised as a good one; further subsequent analysis with other techniques (second order fit performed outside the FPGA) will be able to distinguish the truth tracks. The more the accumulator is filled, the higher is the probability of recognizing fake roads as good roads.

The Accumulator Density % is another calculated parameter: it is equal to the ratio

between the total accumulator filling and the maximum accumulator capacity.

In the *Statistics* block, all the information from the previous steps is collected and plotted in 3D. The total number of clusters, fake clusters, percentage fake clusters, initial roads, extracted roads, found roads, and percentage accumulator density are the most important parameters involved.

5.2.3 Data Analysis

Input data sets consisting of a given number of (r, ϕ), with or without background data, are generated. First, the capability of the system to detect and reconstruct the original input data without unwanted data is tested. Then, by adding randomly distributed (r, ϕ) pairs, the system's effectiveness in identifying real input data superimposed on a background is evaluated.

A 600x1100 accumulator is considered with 8 layers, 2 clusters per layer (for a total of 16 clusters per road), and a certain number of roads (5, 10, 20, 30, 40, 50). In this hypothesis, the maximum accumulator capacity is given by 600x1100x8.

The top plot of Fig. 5.4 shows the accumulator with 10 roads and an Input Set consisting of 8 inputs, each providing 2 pairs of (r, ϕ), resulting in 10x8x2=160 input clusters. No random input data is given. Therefore, ideally there should be 10 $(\phi_0, q/p_T)$ pair values in the Parameter Space corresponding to recognized candidate tracks.

The bottom plot of the figure shows the same accumulator with the 10 roads plus 640 random input data pairs (80% of the input data is noise), for a total of 800 input sets. These pairs also create lines in the Hough space but, since they do not belong to lines in the Cartesian space, they do not form cross points here. An accumulator density of 25% is obtained.

In this case, 15 roads are extracted with a threshold equal to 8. The extracted roads are potential candidate roads. Unwanted data creates fake roads that do not belong to any particle tracks, and must be eliminated in further subsequent analysis.

A 600x1100 accumulator plot is shown in Fig. 5.5 with an input set of about 5000 clusters, 10 roads and a background noise of about 97%. Here, the number of the extracted roads and clusters explodes due to the high noise percentage, making reconstruction of the original roads quite impossible.

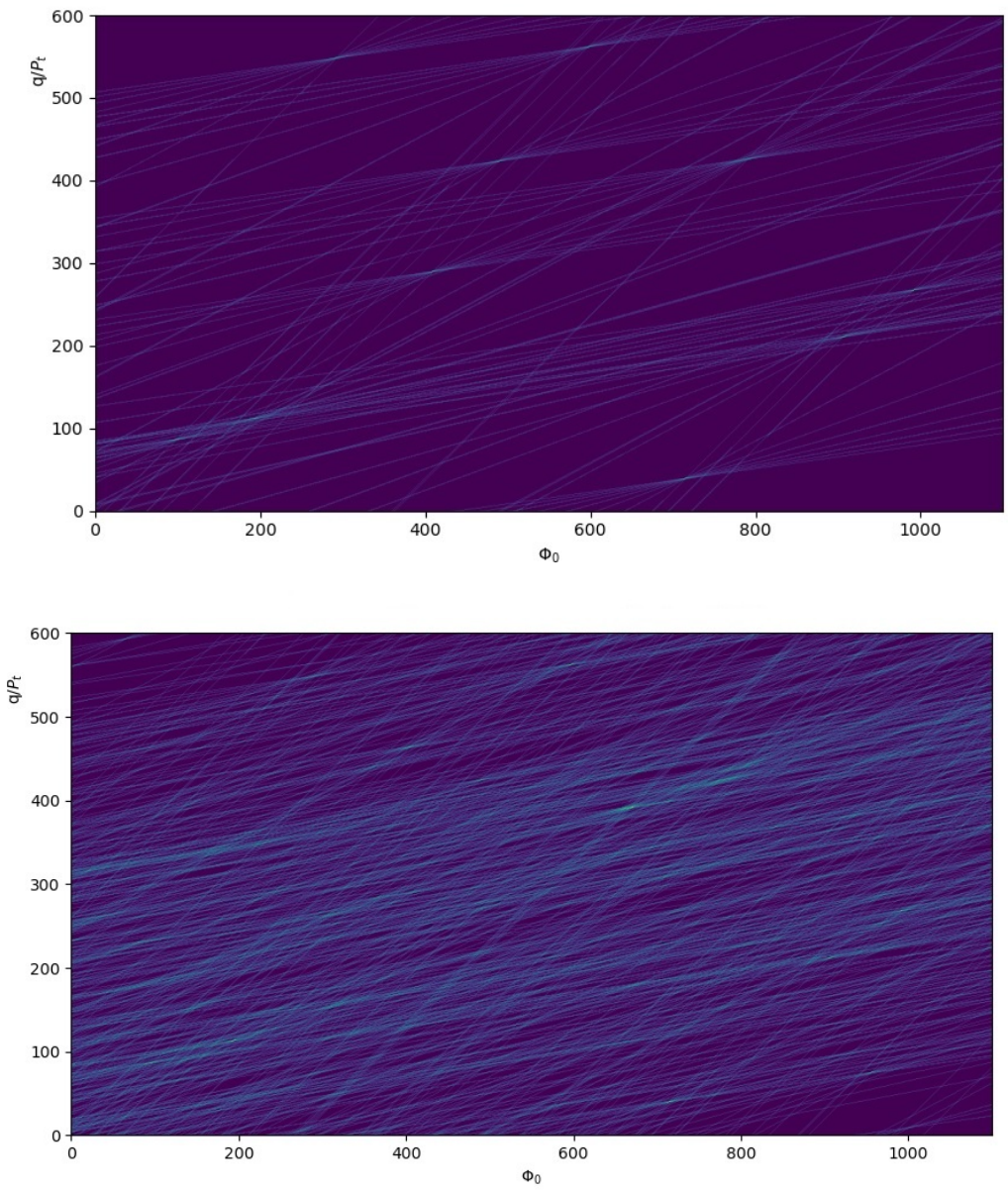

Fig. 5.4: 600x1100 accumulator plots with 10 roads, without noise (top) and with 80% noise added (bottom).

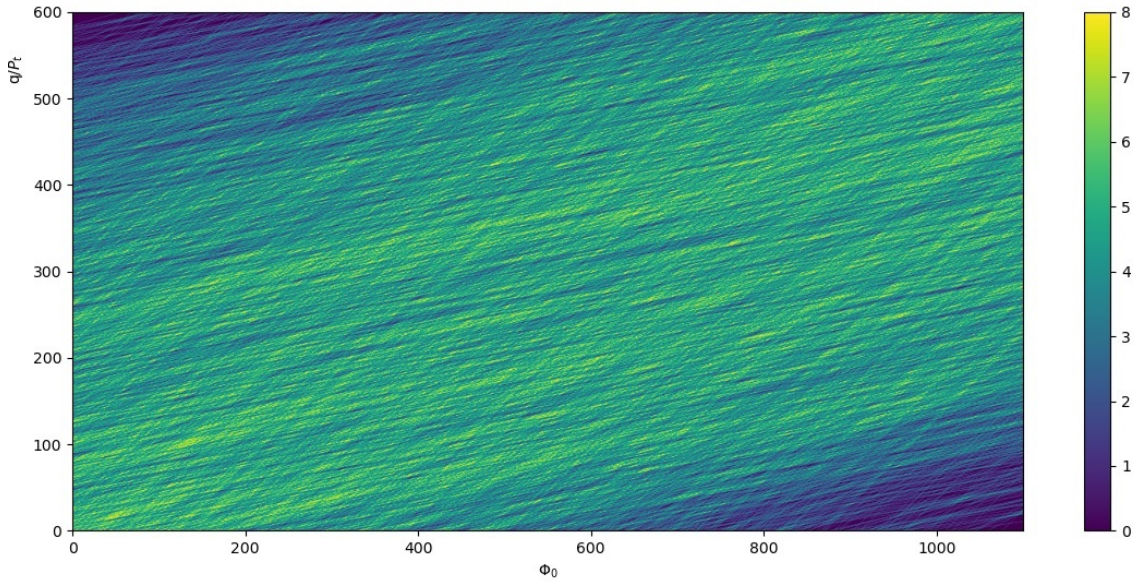

Fig. 5.5: 600x1100 accumulator plot with 10 roads and >90% noise.

A summary of the results of the software analysis is shown in Table 5.1, which lists the input sets, the clusters, the generated roads, the extracted roads, the % of background data used in the simulations, and the accumulator density in terms of the percentage of filling compared to the total capacity. For example, the first row shows an input set consisting of 80 clusters, which is the result of 5x2x8, reflecting the decision to generate 5 roads with 2 clusters belonging to each of the 8 layers. The system extracts 5 candidate roads without unwanted background data, so the system achieves a success rate of 100% and an accumulator density of 1%. The first Input Set column is filled with different data set B, C, D and E. Varying the percentage of additional background data is expected to increase the accumulator density and the number of extracted roads. This is especially true when the percentage of noise exceeds the 75-80%. The initial roads are all recognized (as found roads), which gives an indication of the efficiency of the algorithm.

Input Set	Clusters	Generated Roads	Extracted Roads	Noise data %	Accumulator Density %
B1	80	5	5	0	1
B2	100	5	5	20	1
B3	133	5	6	40	3
B4	200	5	6	60	6
B5	400	5	7	80	14
B6	800	5	23	90	23
C1	160	10	16	0	2
C2	200	10	16	20	3
C3	266	10	18	40	5
C4	400	10	18	60	1
C5	800	10	39	80	25
C6	1600	10	1082	90	41
D1	320	20	28	0	3
D2	400	20	28	20	5
D3	533	20	30	40	10
D4	800	20	42	60	22
D5	1600	20	1221	80	43
D6	3200	20	25797	90	64
E1	640	40	59	0	6
E2	800	40	61	20	10
E3	1066	40	72	40	20
E4	1600	40	566	60	38
E5	3200	40	36054	80	67
E6	6400	40	222745	90	85

Table 5.1: 600x1100 accumulator software analysis results.

Fig. 5.6 show two 3D plots of the number of clusters, extracted roads and % noise data in the first, and the number of clusters, extracted roads and accumulator density percentage in the second.

It is clear that the number of the extracted roads increases with the combination of noise percentage and input data pairs: if they are simultaneously high, the number of extracted roads increases significantly, increasing the processing time and latency in the firmware implementation. These simulations can be used to set a limit on the percentage of acceptable noise, depending on the application. This ensures that the overall latency

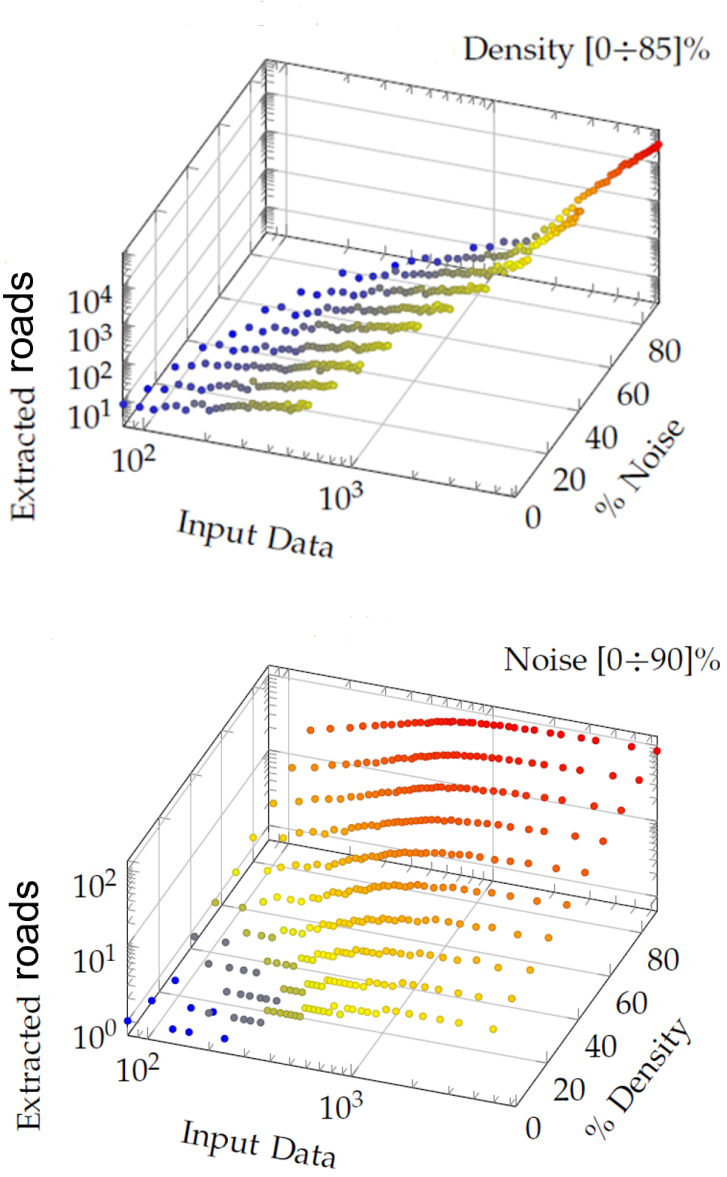

Fig. 5.6: 600x1100 accumulator 3D plots for HT noise (top) and density (bottom) analysis.

of the process does not increase excessively.

An important parameter to consider is the *granularity*. It represents the size of the bins, and describes the quality of the digitised lines that fill the accumulator. The higher the binning, the finer and more accurate is the identification of the line at the expense of the size of the accumulator.

The number of bins for ϕ_0 and q/p_T are important parameters to consider when constructing an accumulator and setting a threshold to extract candidate roads. Even if computed in floating point, they are discretised to find the corresponding accumulator bins to fill, so precision is lost. If a more accuracy is required to improve road identification, a finer binning must be chosen, and this requires more resources in the FPGA as the overall complexity of the accumulator increases. A balance between these two aspects should always be kept in mind.

Many tests are run to fine-tune the number of ϕ_0 and q/p_T bins and thus the granularity, and also to see if there are any correlations between the way the fake data are generated and the number of roads extracted. It is tested how much the quality of the extraction process depends on the way the noise is generated, e.g. taking random noise in the RoI rather than in the whole 1st ITk quadrant, or all random rather than belonging to fake roads.

Different accumulator sizes are tested: 500x1000, 600x1100, and 600x1500. Fig. 5.7 shows the results obtained by analysing the 600x1500 accumulator; similar results are obtained with the others. Dummy roads are generated in the RoI defined as ϕ_0 in [0.3, 0.5] rad and q/p_T in [-1,1] GeV^{-1} (top), and in the first ITk quadrant (bottom). The number of extracted roads, together with the fractional accumulator density, explodes as the total number of clusters increases, while if the ϕ_0 and q/p_T values for the dummy roads are taken anywhere in the first ITk quadrant, the number of extracted roads is less than 70 and the % accumulator density does not exceed 8%, regardless of whether fake clusters belong to dummy roads or not.

Another well-tested technique for finding candidate roads is to search for a *6-7-8-7-6 sequence*. It consists of looking at five adjacent bins along ϕ_0, given the same q/p_T value, in the accumulator whose value must be greater than or equal to the corresponding value in the sequence. Due to the rounding of ϕ_0 and q/p_T values necessary to identify an accu-

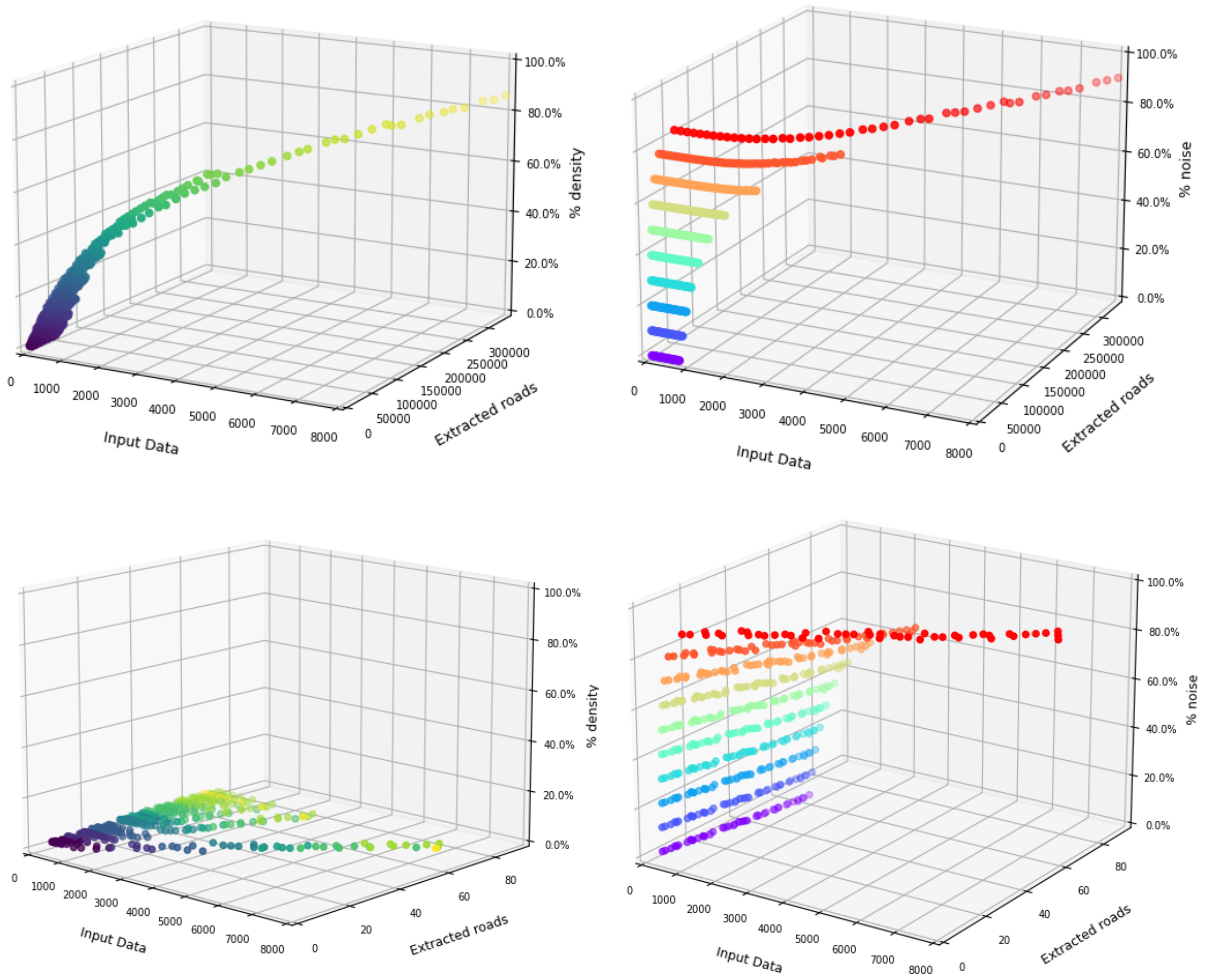

Fig. 5.7: 600x1500 accumulator test results with dummy roads in RoI (top), in the whole 1st ITk quadrant (bottom).

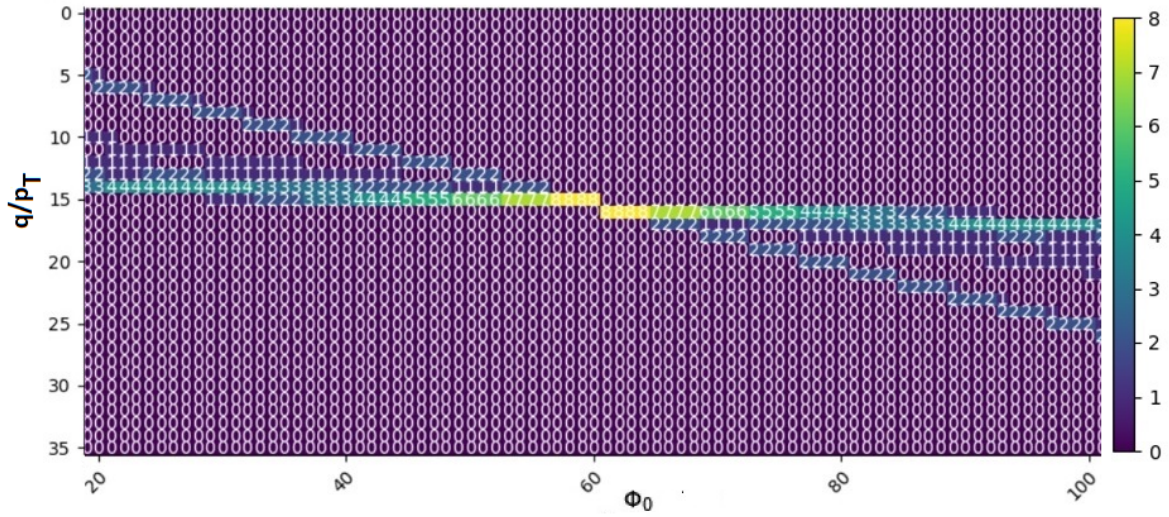

Fig. 5.8: Example of a portion of a 64x1200 annotated heatmap accumulator. Each cell contains a number and has a colour. Yellow cells indicate that 8 layers have been hit and their value is 8. This representation is useful for visually identifying 6-7-8-7-6 sequences.

mulator cell, adjacent bins can accumulate the same clusters and satisfy almost the same threshold condition. This enforces the selection and removes more fake combinations, so it can be well used to reduce the number of extracted roads.

In Fig. 5.8, part of a 64x1200 accumulator (called *mini accumulator*) is plotted as an annotated hitmap [51], i.e. an accumulator with a number and a colour in each cell indicating how many layers were hit. In this example, 4 candidate roads are extracted by selecting the sequences 7-7-8-8-8, 7-8-8-8-8, 8-8-8-8-7 and 8-8-8-7-7. Their offset and slope are equal to the ϕ_0 and q/p_T values with respect to the central value of the sequence. Instead, applying a threshold value equal to 8, 8 candidate roads are selected (all the cells of the accumulator with 8 layers hit), while with a threshold value of 7, 16 roads are found (all the accumulator cells with a value of 7 or 8).

An application of this technique can be seen in Fig. 5.3.

5.2.4 Compatibility with the Firmware

An optimized VHDL firmware design compatible with the software implementation is developed and verified on a Xilinx Ultrascale+ FPGA accelerator card with the VU9P device on board, the Felix-II, a custom card originally designed for the ATLAS data acquisition and triggering applications at CERN.

A schema of the firmware implementation is shown in Fig. 5.9.

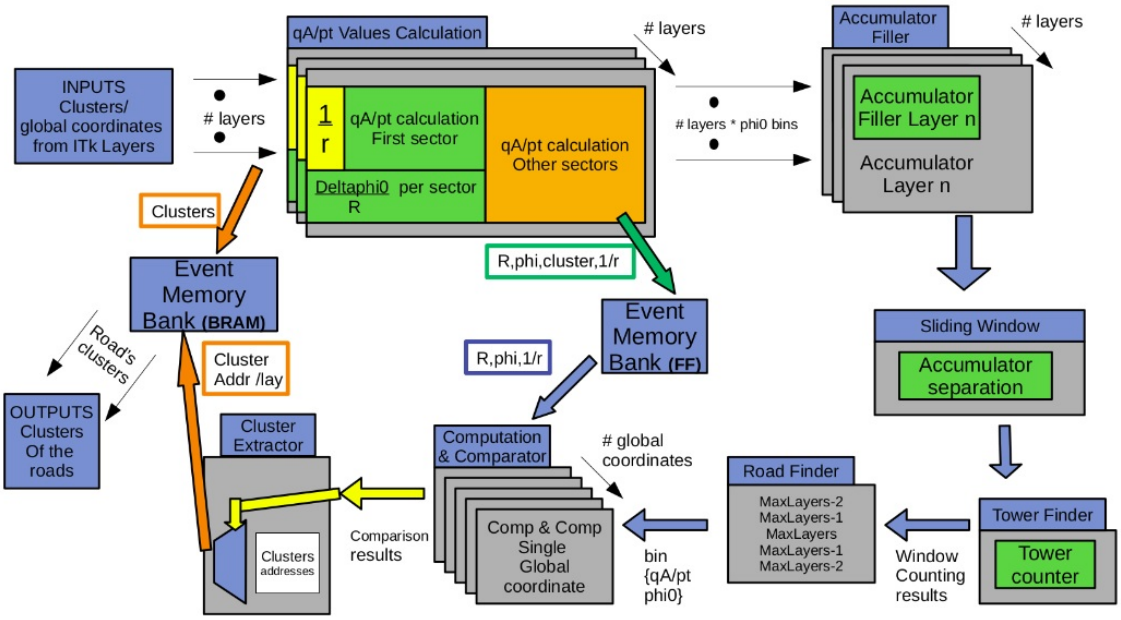

Fig. 5.9: Firmware implementation schema on FPGA.

In the firmware, the total processing time associated with an event depends on the total number of input data, the latency, the total number of candidate roads within a given event, and the clock frequency.

The software emulator follows the system described in the firmware step by step, regardless of the optimization of the operations and the processing time. Of course, the software does not take into account the internal latencies and other timing templates of the electronics.

The two independent simulations share the same input set (see Fig. 5.10).

The Table 5.2 shows the results obtained using a reduced input data set of the one

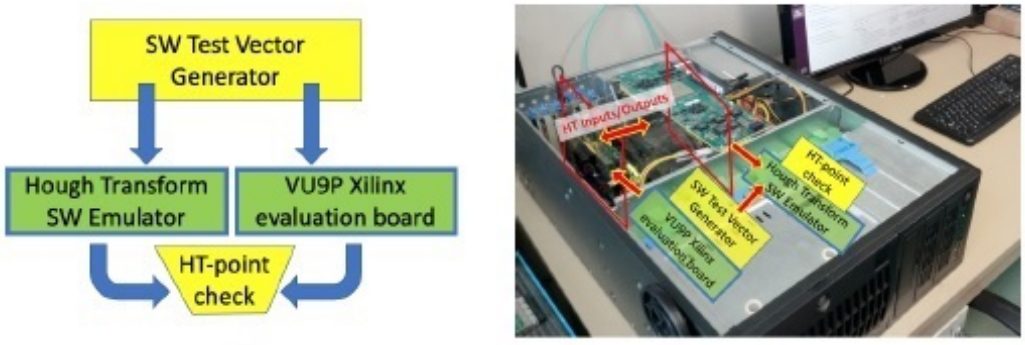

Fig. 5.10: SW validation logic blocks (left), HW validation (right).

used in 5.2.3, about 1k input clusters, and a 280x280 accumulator with 8 in high, the size of which has been adapted to be implemented on the FPGA device.

Input Set	Clusters	Generated roads	Extracted roads	Procressing Time (ns)
A1	448	20	32	1052
A2	528	25	42	1220
A3	608	30	55	1444
A4	688	35	73	1652
A5	768	40	93	1932
A6	848	45	118	2244
A7	928	50	168	2940
A8	1008	55	220	3548

Table 5.2: Summary results for a 280x280 accumulator implementation on a VU9P FPGA device and a reduced input data set.

The *Clusters* column shows the number of input data (r, ϕ) pairs with a fixed amount of background noise; the *Generated Roads* column shows the number of initial roads; the *Extracted Roads* column tells how many input lines are potential candidate lines in the Hough space. The values shown here may be larger than those in the corresponding *Generated Roads* column, because noise can create fake roads that do not belong to any

input line and are eliminated as much as possible by a further process outside the HT. The extracted roads represent potential physical tracks that need to be confirmed by the next processes, so it is important not to lose any of them in the background noise. If more roads are detected and extracted than expected, they will be eliminated by a further cross-check using additional layers than those used by the HT process. The *Processing Time* is estimated in the last column, using a reference clock period of 4 ns (250 MHz).

In this example, *Generated Roads* consists of 2x8=16 (r,ϕ) pairs, e.g. the penultimate row of Tab. 5.2 refers to 50 initial roads covering 50x16=800 pairs of input data from 928 of the column *Clusters*, resulting in 128 pairs of background noise data. All the Input Sets contain a common set of 128 noise pairs (r, ϕ) and use 8 parallel inputs (simulating the ITk layers).

Under these conditions, the software and firmware implementations shows to be compatible confirming the feasibility of the HT implementation on the FPGA. The software simulation is much faster than the firmware simulation: the software processes each event in 1/2 second, including data extraction, while the firmware takes over 10 seconds (from 15 to 20 seconds), a factor of 5 more. Therefore, SW can speed up the development of FW.

Other accumulator sizes are tested on the same FPGA device with similar results. Again, the two independent simulations shared the same input set.

The firmware design is characterized by a 250 MHz clock signal, a reduced 216x216 accumulator and a total number of clusters in the order of 1000. The generated input clusters are dummy and contain a given number of roads, each consisting of 2x8 clusters. Again, all the events contain a common bunch of 128 noisy clusters (r, ϕ) and refer to 8 input layers simulating the Pixel and Strip layers.

The results of the simulations are reported in [42].

5.3 ATLAS TV data

Further measurements are performed using another version of the development tool, consisting of five blocks:

- *Data Extraction*
- *Accumulator Creation*
- *Road Extraction*
- *Cluster Comparison*
- *Statistics*

The *Data Generation* block is missing because the input samples are given by the ATLAS collaboration, generated by the Monte Carlo simulations of the ATLAS detector based on the GEometry ANd Tracking (Geant4) detector simulation framework. Instead, there is a new block, the *Cluster Comparison*, in which the activated clusters belonging to the extracted roads are compared with those contained in the initial input set.

The used samples are:

- single muon events: 1k, 10k
- single muon events with pileup $\mu = 200$: 10k

taken from the file:
singlemu_invPtFlat1_10k_wrap.root and
user.martyniu.24590899.EXT0._000003.httsim_rawhits_wrap.root
and belong to the RoI defined by ϕ_0 in [0.3, 0.5], η in [0.1, 0.3], q/p_T in [-1.0, 1.0]
These are the same data sets used for the TDAQ-TDR amendment studies [28].

The first tested input data set is 1k single muon clusters belonging to 8 layers and 6 independent events. 64 roads are extracted, some of which are recognized by more than one event.

The corresponding Cartesian and Hough spaces are shown in Fig. 5.11 and Fig. 5.12.

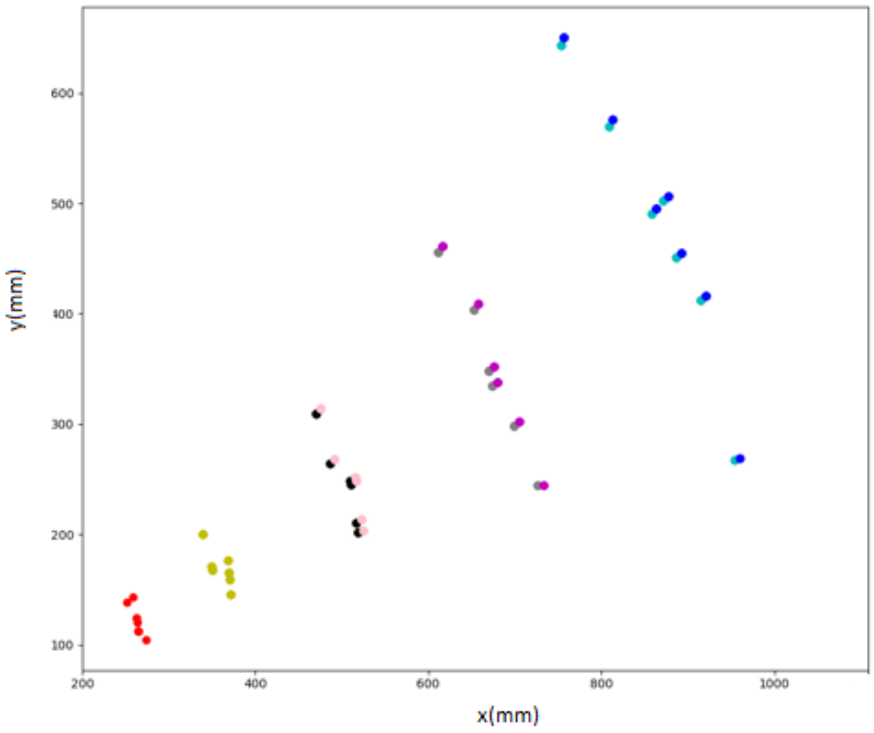

Fig. 5.11: *x-y* Cartesian space for about 1k single muon clusters belonging to 8 layers and 6 events.

In the Cartesian Space representation, the different color highlights the separated layers of the ITk in the central barrel, composed by the outermost pixel layer, the inner side of the first strip layer, and both sides of the other strip layers. The (x,y) couples, whose values are stored in the input data file, are transformed into the corresponding polar (r,ϕ) pairs and digitized respectively with 12 and 16 bits.

Fig. 5.13 shows a 220x230 accumulator applied to this data set.

Compared to the previous one, this version is closer to the firmware representation: all the mathematical operations are integer converted (divisions, sums of float numbers), and the r_b and ϕ_b *bitwise conversion* is implemented for each r and ϕ input values:

$$r_b = \frac{r}{1100} * 2^{12}, \phi_b = \frac{\phi}{2\pi} * 2^{16} \tag{5.1}$$

where 1100 represents in mm the maximum radius value considered.

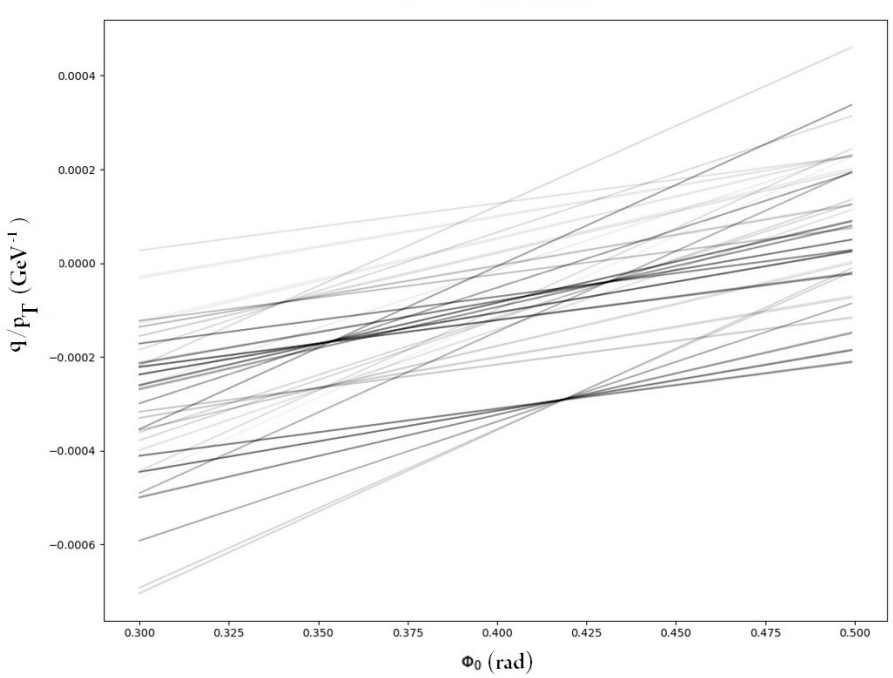

Fig. 5.12: q/p_T-ϕ_0 Hough space for 856 (r, ϕ) single muon clusters.

For convenience, a conversion factor $K = 2^{16} * 1100/(2^{12} * 2\pi) = 2802$ is used to pass from q/p_T in float to q/p_T bitwise.

For firmware compatibility, the *pipelines*, the *sectors* and the *Hit Extension* are firmware techniques implemented in software.

The *pipelines* is used in firmware to reduce the logic distance between two clock driven components, by inserting registers (flip flop components) within a path. In this case, it creates a clock domain separation along with ϕ_0 and q/p_T .

The *sectors* is a firmware technique for drawing monotonic lines; it is used for the accumulator construction to minimize the FPGA resources needed for its construction. It significantly reduces the number of multiplications, and thus the number of Digital Signal Processings (DSPs) used in the firmware. The accumulator is divided into n sectors along ϕ_0 (see Fig. 5.14); for the first sector, the q/p_T values are calculated using the HT formula 4.10, for the others, the q/p_T values are calculated adding to the corresponding one of the first sector a step factor value of $\Delta\phi_{0,l}/r$, where l represents the number of sectors and $\Delta\phi_{0,l}$ is a constant for each sector and represents the ϕ_0 value

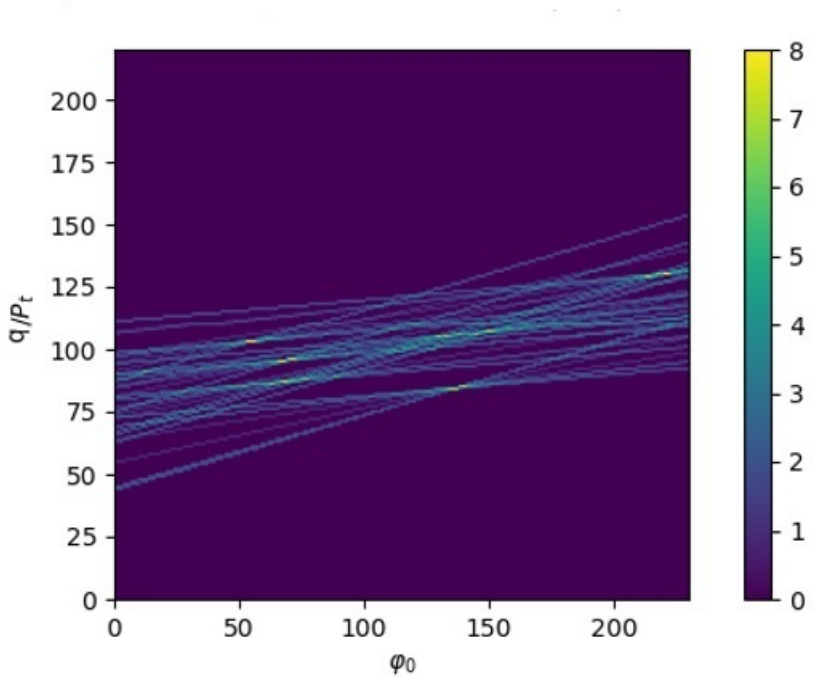

Fig. 5.13: 220x230 accumulator plot for 856 (r, ϕ) single muon clusters.

covered by all the bins of that sector.

The *Hit Extension* is the technique where a certain number of extra bins on both sides along the ϕ_0-axis are filled in the accumulator, to maintain high efficiency. The technique can also be applied along q/p_T and other adjacent bins, such as 1 or 2 on the left, 1 or 2 on the right, 1 or 2 above, 1 or 2 below, forming a kind of cross.

Fig. 5.15 shows the application of the technique to the same ATLAS data set, with 856 hits, using a 400x300 accumulator: right plots show the application of the technique to the ϕ_0 bins, left plots without the technique. In the right plots the straight lines are more prominent. The technique is applied by looping the HT formula over ϕ_0 (top plots) and over q/p_T) (bottom plots).

The use of this technique increases the number of extracted roads because more bins are fired into the accumulator, minimizing the loss of good tracks.

Another simulation sample used in our first tests consists of 10k single muon events with pileup. The polar coordinate r values range from 300 mm to 1000 mm (Fig. 5.16) and span across 8 layers: the red ones belong to the P5 (the outermost pixel layer),

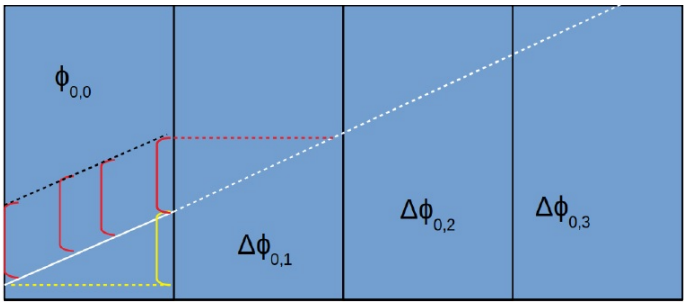

Fig. 5.14: Logic representation of the sector method applied to the accumulator construction. The dotted white line in the first sector is translated in the other sectors using a step factor of $\Delta\phi_{0,l}$, where l is the number of the sector.

the green ones to the S1 (the innermost strip layer), the pink and the black ones to S2 (top/bottom 2nd strip layer), the magenta and the grey ones to S3 (top/bottom 3rd strip layer) and the light blue and blue to S4 (top/bottom 4th strip layer).

The RoI spans in the range [0.3, 0.5] for ϕ_0 and [-1.0, 1.0] for q/p_T.

Using the HT formula, the accumulator is filled, and candidate roads are extracted and kept aside. By re-analysing all the input data, the clusters belonging to the extracted roads are also found out and kept aside. They are called *activated clusters*.

From a firmware point of view, it is too expensive to use memory to store the clusters belonging to the layers that are fired during the accumulator filling; the software has to mimic the firmware behaviour, after the accumulator is filled the tool reanalyses all the input data to extract the clusters belonging to the extracted roads.

The use of *int* instead of *float* in all the mathematical operations and the *bitwise* conversion implemented for r and *phi* input values to mimic the hardware/firmware behaviour have resulted in a lower track recognition capability, so a long and still ongoing work of fine-tuning the parameters is started, to arrive at the best compromise between efficiency in pattern recognition and feasibility of firmware implementation.

Another data sets containing different sizes of single muon events without and with pileup 200 are analysed; the slicing technique is performed, with 6 and 19 z-slices. The hit extension technique is implemented and tested to see if it can provide some improvement. Reduced accumulator sizes are analysed, in particular 216x32 and 216x64 with a threshold equal to 7 (which means that the accumulator cell must have a value of 7 or 8

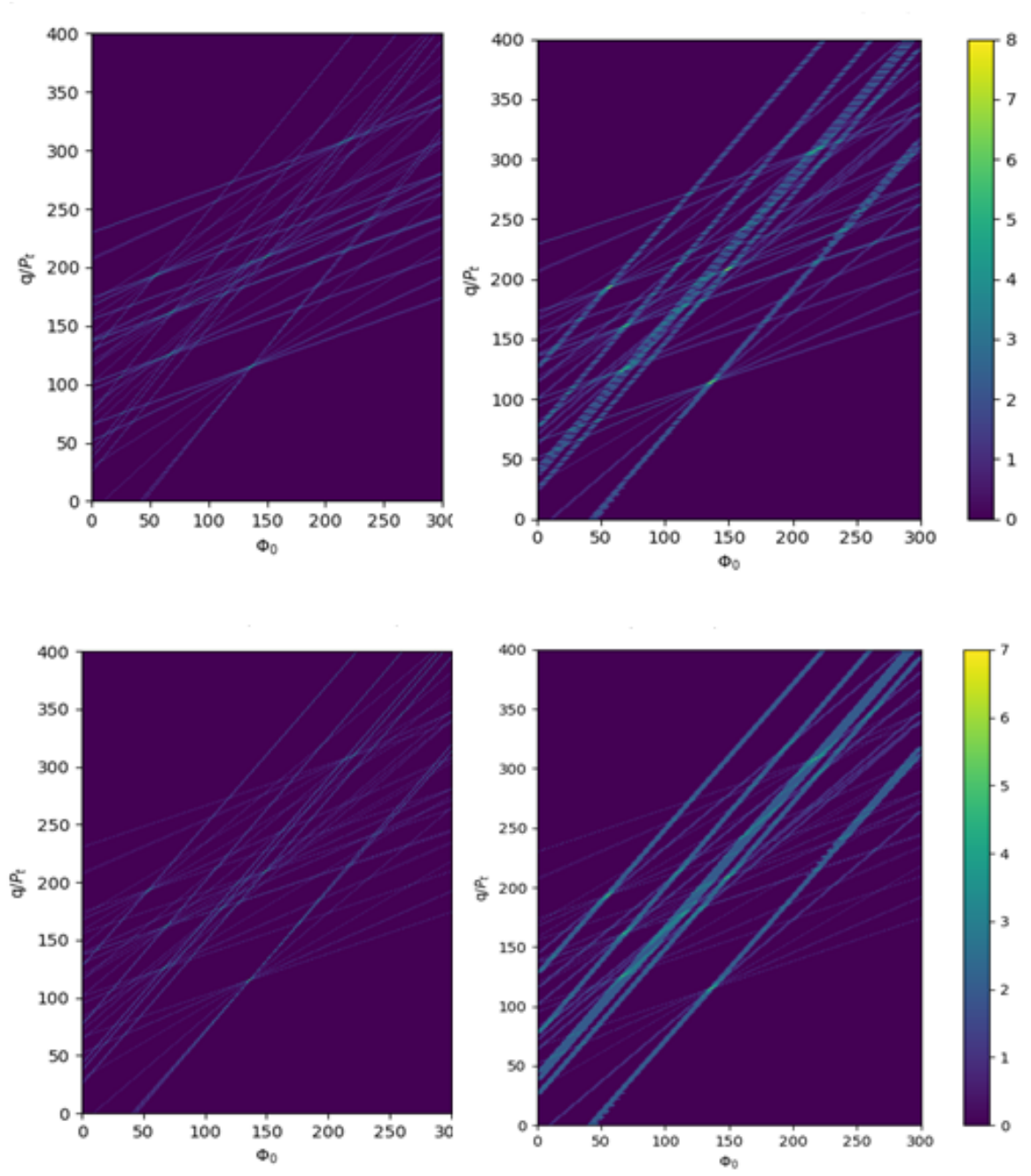

Fig. 5.15: 400x300 accumulator plots for 856 (r, ϕ) single muon clusters. On the right, the Hit Extension technique is applied in the two plots, while the technique is not applied in the left plots. In the top two plots, the HT formula is looped over ϕ_0, while in the bottom two plots the loop is over q/p_T).

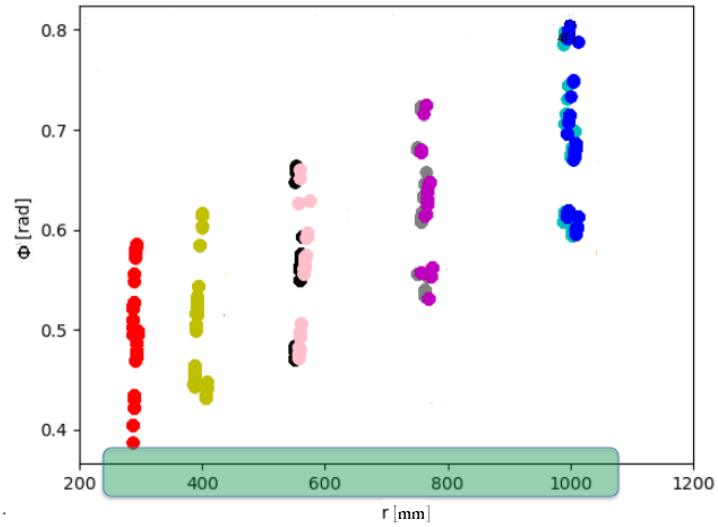

Fig. 5.16: 10k single muon (r, ϕ) cluster sample representation. The red points belong to P5 (the outermost pixel layer), the green to the S1 (the innermost strip layer), pink and black to S2 (top/bottom 2nd strip layer), magenta and grey to S3 (top/bottom 3rd strip layer), and the light blue and blue to S4 (top/bottom 4th strip layer).

in case of 8 layers).

Unlike the previous tests, only the strip layers are used. The system is flexible, and different choices can be easily implemented.

System performance evaluations are performed to test how many original single muon tracks the system could detect, to have a measure of the pattern recognition efficiency, the so-called *track recognition capability*, and the background rejection capability.

Efficiency is defined here as:

$$\epsilon_{m/M} = \frac{N_{m/M}}{N_{Tot}} \tag{5.2}$$

where $N_{m/M}$ is the number of events with at least one extracted road containing m over M hits generated by a true original particle (muon), and N_{Tot} is the total number of events.

The goal is to have as high an efficiency as possible, preferably above 99%, and as few cluster combinations as possible.

The minimum number of clusters required is 7 (the threshold value), so $\epsilon_{7/8}$ and $\epsilon_{8/8}$ can be searched.

All combinations of clusters that pass this selection will have to be fitted by the track fit (Kalman filter), so this number should be as small as possible.

The number of possible cluster combinations N is calculated by taking the product of the number of clusters $n_{b,l}$ in each layer l and taking the sum over all the selected bins b:

$$N = \sum_{b}(\prod_{l} n_{b,l}) \tag{5.3}$$

Other parameters to consider are:

- number of the *extracted roads* per event (average);
- number of the *activated clusters* per event (average);
- number of *events without extracted roads.*

A lot of tuning is done to find out the best accumulator sizes to fit into the hardware.

Starting with a 216x32 accumulator, which is shown to have good ϕ_0 and q/p_T bins sizes in hardware tests, the size of the accumulator is varied slightly, to see how the results might be affected. 180x28, 200x30, 240x34 and 260x36 are the accumulator sizes tested. Fig. 5.17 and Fig. 5.18 show the plots for efficiency, extracted roads, and activated roads on a sample of 10k single muon events, with their statistical errors. The tests are done for 6 and 19 z-slicing, according to the values chosen in [28].

The smaller the accumulator in terms of number of bins, the less granularity it has (it has wide meshes), consequently more roads are detected and more clusters are extracted compared to an accumulator like 260x36 with a finer binning, because the ranges of ϕ_0 and q/p_T do not change ([-1,+1] GeV^{-1} for q/p_T and [0.3-0.5] rad for ϕ_0). So, the larger the accumulator, the less efficient it is, following the definition of efficiency given before. Finer lines are examined and there is less chance of more lines crossing. Comparing the results using the two different z-slicing, the test have put on evidence that using the 19 z-slicing is preferable to the 6 z-slicing in terms of extracted roads and activated clusters (they are fewer in number) and the efficiency is only slightly lower.

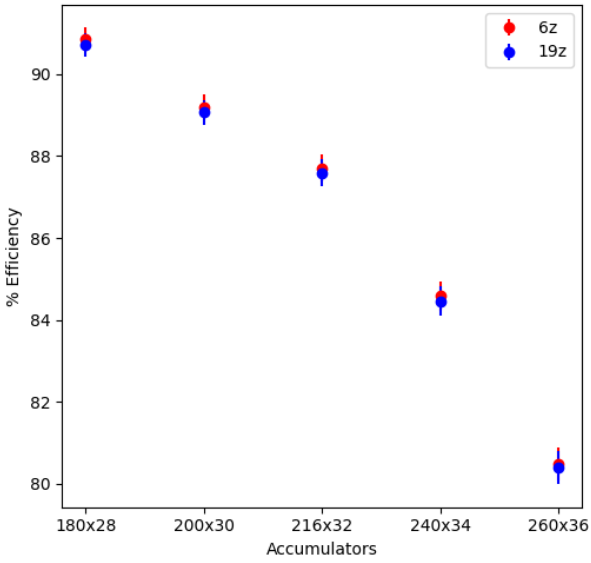

Fig. 5.17: 10k single muon events with pileup 200 efficiency plot, for 5 accumulators with different bin sizes and two different z-slicing (6 and 19).

Regarding the computational time needed to performe all these calculations, in the case of 10k events for the single muon with pileup 200 and 6 z-slices, using a traditional desktop PC with an Intel core I5 2.3GHz and 16 GB of RAM, it takes about 5 hours for the smaller accumulator (180x28) and about 6 hours for the larger one (260x36), while using 19 z-slices the processing time goes from 8 hours and half for the smaller accumulator to about 10 hours and half for the larger one.

Some firmware tests have been performed on the VCU1525 accelerator card with the VU9P FPGA device, using reduced input data sets of single muon and pileup events. The RoI is $0.1 < \eta < 0.3$ and $0.3 < \phi < 0.5$, $|d_0| < 1.8$ mm, $|z_0| < 150$ mm, where d_0 and z_0 are two parameters described in 4.2. 216x32 is the tested accumulator.

Tab. 5.3 summarises the obtained results on software and firmware using the same data sets.

The input data sets F1 and F2 consist of 100 and 1k single muon events without pileup, respectively. The other input data sets (F3, F4, F5, F6, F7, F8) are respectively 65 single muon events with pileup 200 each, related to the 6 z-slices, from z_0 to z_5, where z_0 is the nearest slice to the IP. The input sets implemented on the firmware are all

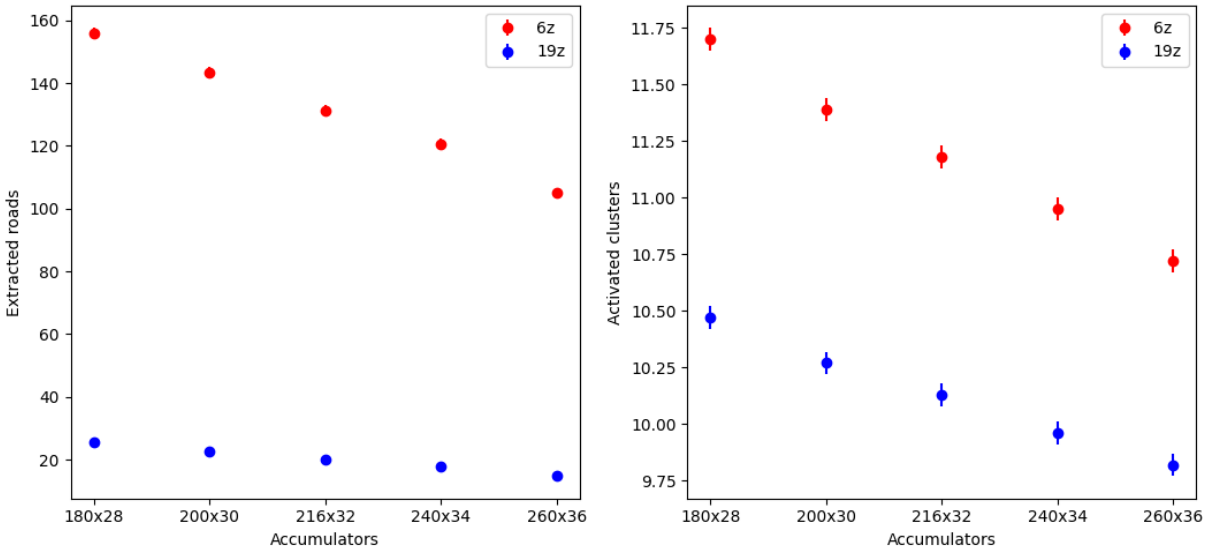

Fig. 5.18: 10k single muon events with pileup 200 extracted roads (left) and activated roads (right) plots, for 5 accumulators with different bin sizes and two different z-slicing (6 and 19).

reduced event sets compared to the number of events used in the software simulations (about 10k events). The table shows the total number of extracted roads and the total number of activated clusters obtained in the software tool and in the accelerator card with the FPGA, given the same data set.

There is a good agreement between the software results and those obtained with the firmware: for the extracted roads, 92.5% in the case of single muon events without pileup and about 99% in the cases of single muon and pileup, while for the number of activated clusters we obtain 92.4 % for input set F1 and about 99% in the other cases. These results lead us to conclude that the software tool is a good emulation of the firmware, at least with these reduced data sets.

The average number of the extracted roads per event in the case of single muon without pileup is 228/100 = 2.2, with an average number of activated clusters per event of 17.13, while in the case of single muon and pileup 200, for example, of the 6th z-slice (last case in the table) the average number of the extracted roads per event is 2835/65 = 43.6 with about 27349-65=420 clusters. This is due to the pileup effect. With 6 z-slices, an event is estimated to be processed in less than 10 μs in terms of

Input Set	Extracted roads (SW)	Extracted roads (FW)	% extr rd SW/FW	Activated clusters (SW)	Activated clusters (FW)	% act clr SW/FW
F1	228	211	92.5	1713	1584	92.4
F2	2169	2156	99.4	16746	16656	99.4
F3	11835	11835	100	134684	134298	99.7
F4	11613	11588	99.7	128123	127862	99.7
F5	6140	6140	100	63688	63170	99.1
F6	7287	7279	99.8	77292	76503	98.9
F7	11545	11539	99.8	127262	126981	99.7
F8	2835	2831	99.8	27349	27331	99.9

Table 5.3: 216x32 accumulator SW and FW results using single muon input data sets. The first two input sets are without pileup, the others are with pileup 200 and one for each of the 6 z-slices, using the VU9P FPGA on a VCU1525 accelerator card.

processing time in the firmware.

Fig. 5.19 shows the Vivado interface with the HT firmware implementation in the case of F1 input data set. The figure shows the total number of clusters activated by the HT algorithm implemented in the firmware (highlighted in the yellow circle), together with the total number of roads extracted (in the Value column).

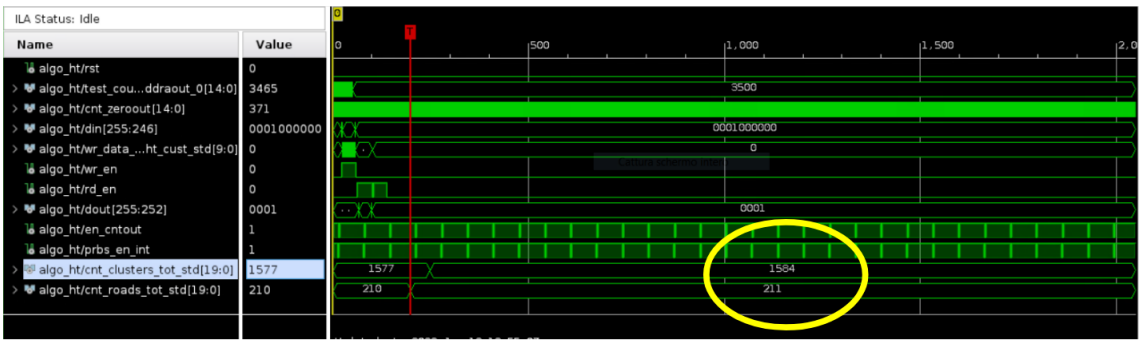

Fig. 5.19: Vivado interface for the VU9P FPGA: roads and clusters extraction.

5.4 Final results on simulated events

In the final version, all the code is ported from Python to the C++ language and runs on Athena, the ATLAS software framework that manages almost all the production workflows such as event generation, simulations, reconstructions.

A working directory is setup on *lxplus.cern.ch*, and batch jobs are run using the HTCondor system. The results are stored in EOS, the disk-based, low-latency storage service of CERN.

Performance evaluations are performed on the HTT simulation framework.

The used samples are:

- Muon with η in $[0.1, 0.3]$: 30.2k
- Muon with η in $[0.7, 0.9]$: 28.3k
- Electron/Pion with η in $[0.1, 0.3]$ and $[0.7, 0.9]$: 10k

The single muon events are with pileup $\langle\mu\rangle = 200$.

The analysed regions of the detector are defined through:

- ϕ in $[0.3, 0.5]$
- $|d_0| < 2$ mm
- $|z_0| < 150$ mm
- 6 z-slices

taking into account the track definition as defined in [29]. The ITk detector geometry considered is the ITk-22-02-00 [28], consisting of 5 pixel and 4x2 strip layers. In particular, the layers from the 4th to the 12th excluding the 5th have been considered; thanks to the flexibility of the system, different numbers and types of layers can be easily selected and studied.

A lot of tuning is done to reach the highest efficiency and to reduce the number of clusters detected, mostly concerning the number of bins of the accumulator, the number of slices along the z-axis, and the choice of layers to use.

Different accumulator sizes are evaluated, in particular 216×32, 216×64, 108×32, and 108×64.

Track-finding performance tests are performed. The statistical results obtained provide information about:

- events with roads: number of events that have at least one extracted road.
- track-finding efficiency: see Eq. 5.2
- track-finding efficiency in different p_T bins: [1-2 GeV], [2-4 GeV], and [4+ GeV]
- average number of roads per event
- average number of roads per event matched with truth.

A track is matched with a truth particle if more than 50% of its hits are generated by that particle. The roads matched with the truth are good candidates for passing the track fit.

Another important parameter is the road hit combinations average per event; all the combinations of clusters passing the selection have to be fitted by the track fitter, so the average number of road hit combinations per event gives a measurement of how many fits the next stage has to do.

Fig. 5.20 shows the efficiency as a function of p_T and ϕ, for single muon events and pileup $\langle\mu\rangle = 200$ with η in $[0.1, 0.3]$.

The yellow dots represent the offline efficiency, where the offline system provides the best possible tracking information. The overall tracking efficiency should be close to these values in an optimal case. For example, central muons with $p_T > 10GeV$ should be 99% efficient relative to offline. Taking into account this consideration, the 108x64 shows very low efficiency, while 216×32 seems to approximate the offline results better, being closest to them and even better in certain p_T ranges.

The efficiency is shown as a function of ϕ, to identify possible geometrical failures. The efficiency is also shown as a function of p_T; at low p_T the efficiency slowly drops because of known effects, like multiple scattering.

Fig. 5.21 shows the number of roads found per event and track matched to truth per event, for single muon with η in $[0.1, 0.3]$.

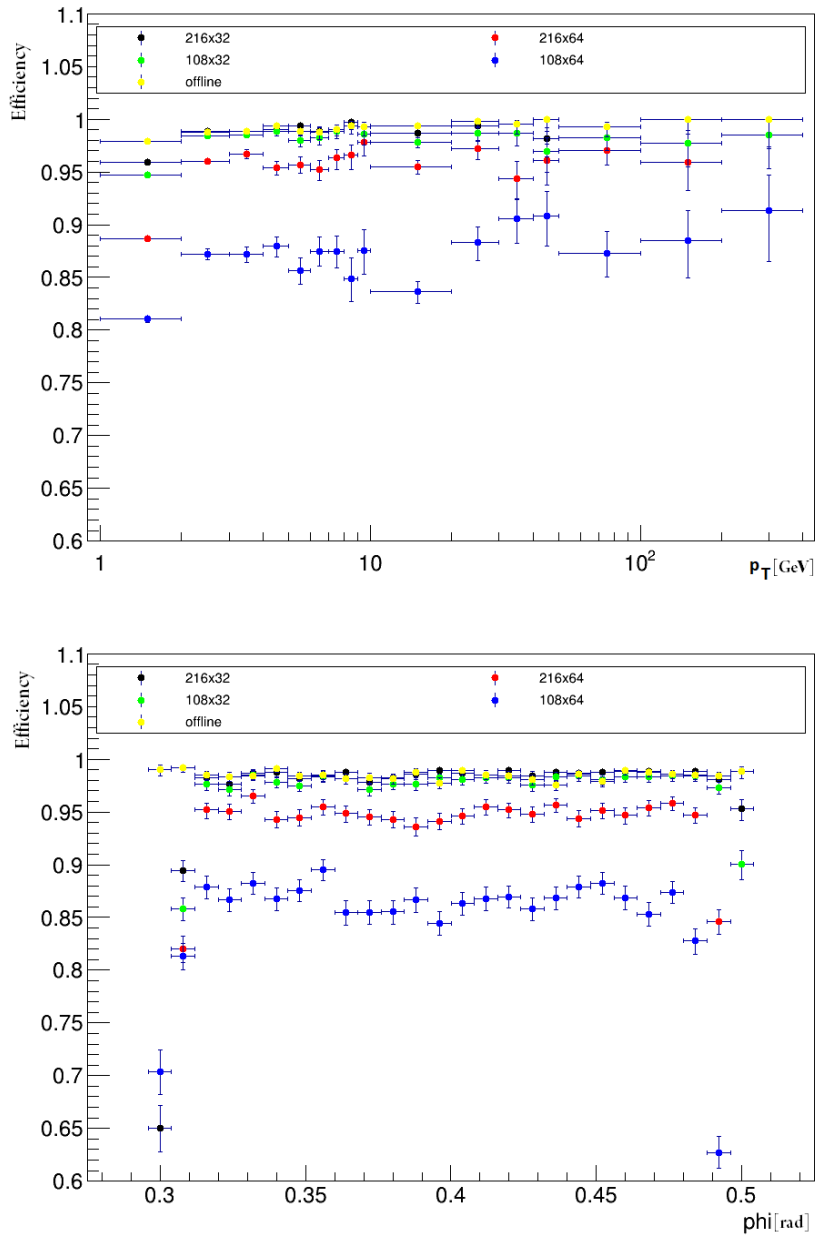

Fig. 5.20: Efficiency plots for *single muon*, $0.1 < \eta < 0.3$.

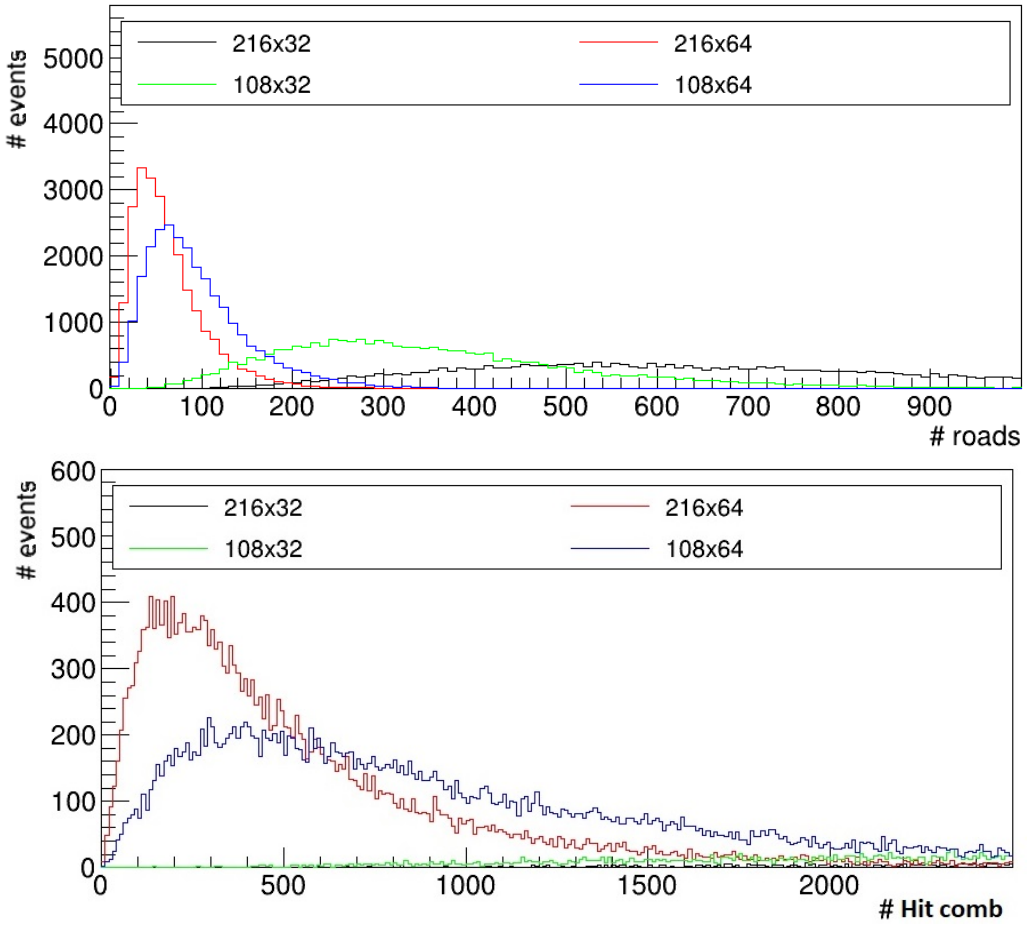

Fig. 5.21: Number of roads found per event (top), number of hit combinations per event (bottom), *single muon*, $0.1 < \eta < 0.3$.

The corresponding numbers of found roads and cluster combinations are quite high (Fig. 5.4), and they will have to be reduced in the forward steps.

	Pseudorapidity interval			
	$0.1 < \eta < 0.3$		$0.7 < \eta < 0.9$	
Accumulator	Mean #roads	Mean #hit comb	Mean #roads	Mean #hit comb
216x32	715.5	25216	1128	54424
216x64	64.46	586.5	108.7	1153
108x32	358	12627	564	27174
108x64	95.9	1226	158.1	2619

Table 5.4: Means for number of roads found per event and for total number of hit combinations per event, 30k single **muon** with pileup 200, four accumulators, two η regions, and 6 z-slices.

The number of cluster combinations can be reduced by choosing an accumulator with a lower efficiency, such as a 216x64 accumulator.

The HT we have implemented in the firmware is able to provide a cluster selection efficiency of over 99% for single muon tracks choosing a 216x32 accumulator and a $p_T > 4GeV$, but for this accumulator the average of the number of roads per event and the average number of fits per event are quite high if compared to the same values for the other accumulators, as shown in Tab. 5.4. There is a factor of 2 with the 216x32 accumulator and more than a factor of 10 with the 216x64 accumulator.

Compared to the 216x32, the 108x32 accumulator reduces the number of clusters to be sent to the track fit by about half, and it is slightly less efficient compared to the 216x32. So this could be another possible candidate accumulator for the firmware tests.

If the goal is to minimize the number of roads extracted per event and the number of activated clusters to be sent to the track fit, the 216x64 accumulator seems to be the best accumulator between those analized, but with an efficiency between 95% and 97% in the region [2-4] GeV and between 96% and 98% in [4-10] GeV.

Fig. 5.22 shows the efficiency plots as a function of p_T and ϕ for *pion* events with η in $[0.1, 0.3]$.

Comparing the muon and pion efficiency results, a generalised reduction is shown

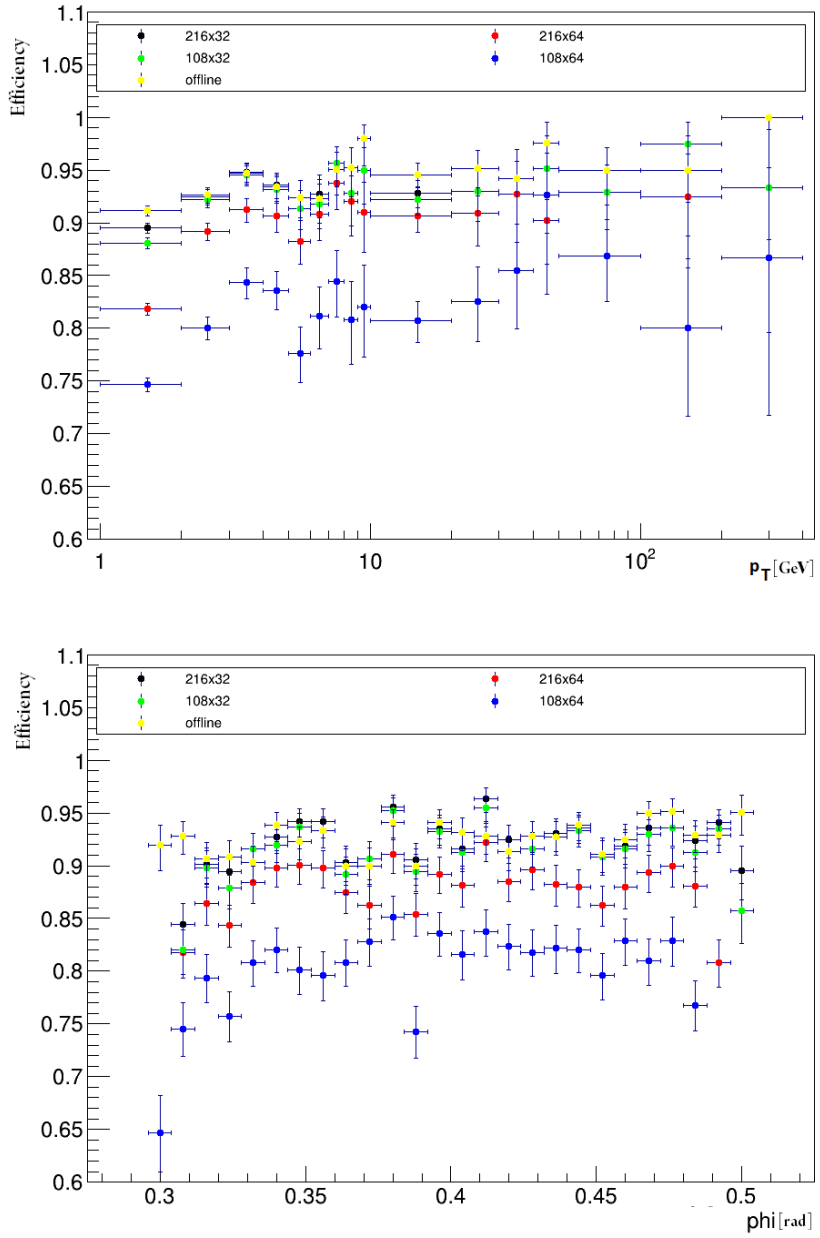

Fig. 5.22: Efficiency plots for *pion*, $0.1 < \eta < 0.3$.

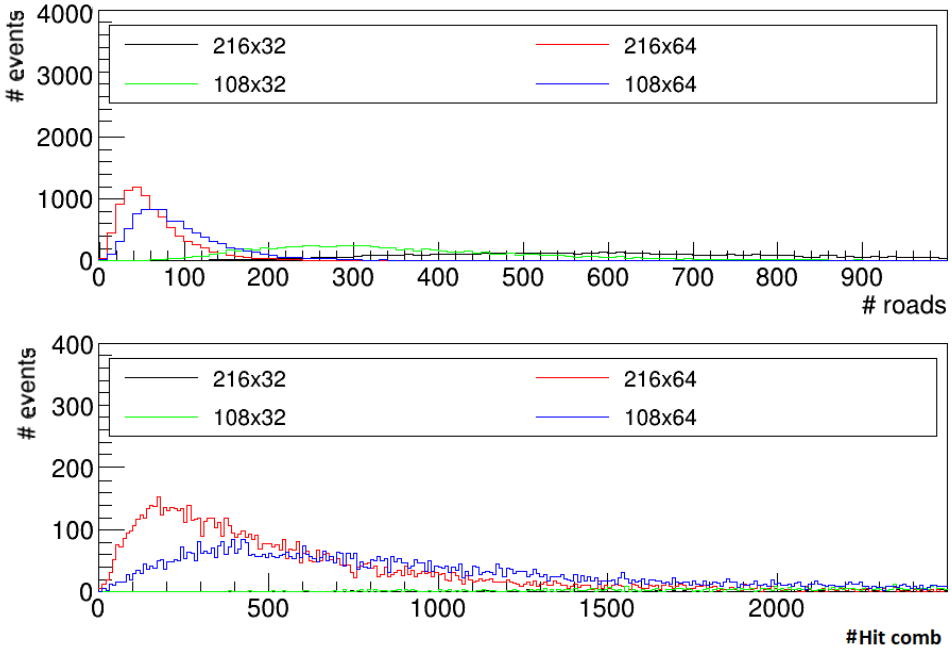

Fig. 5.23: Number of roads per event (top), number of hit combinations per event (bottom), *pion*, $0.1 < \eta < 0.3$.

in all p_T regions, especially for low p_T values. This behaviour is expected, due to the particles and the way they interact with matter. For the 108x64 accumulator, a large statistical fluctuation is seen, so it might not be a good choice to be implemented on an FPGA.

Fig. 5.23 shows the number of roads found per event and the total number of hit combinations per event, for pion with η in $[0.1, 0.3]$.

The choice of which accumulator has to be implemented in firmware must take into account the average number of roads per event and the average number of hit combinations per event parameters, shown in the Tab. 5.5

	Pseudorapidity interval			
	$0.1 < \eta < 0.3$		$0.7 < \eta < 0.9$	
Accumulator type	Avg. number of roads	Avr. number of fits	Avg. number of roads	Avr. number of fits
216x32	720	27000	1100	54000
216x64	66	620	110	1200
108x32	360	14000	550	27000
108x64	97	1300	160	2600

Table 5.5: Average number of roads per event and average number of hit combinations per event for 4 accumulators, 10k **pion**, for different η regions and 6 z-slices.

Looking at another type of particle, the electron, the efficiency of its track reconstruction is lower than that of the muon (Fig. 5.24). Again, these results are as expected due to their different interactions with matter. For $1 < p_T < 2$ GeV, the efficiency of the 216x32 accumulator is slightly lower than the offline efficiency value, while it performs better for p_T values greater than 2 GeV.

As for the muon, for the pion and the electron, in addition to the efficiency, the number of roads extracted and the number of hit combinations (Fig. 5.25) must be taken into account, to evaluate which accumulator is the best to be implemented on hardware.

Tab. 5.6 shown the results for the 10k electron samples.

	Pseudorapidity interval			
	$0.1 < \eta < 0.3$		$0.7 < \eta < 0.9$	
Accumulator type	Avg. number of roads	Avr. number of fits	Avg. number of roads	Avr. number of fits
216x32	720	27000	1100	54000
216x64	65	610	110	1100
108x32	360	14000	550	27000
108x64	97	1300	150	2600

Table 5.6: Average number of roads per event and average number of hit combinations per event for 4 accumulators, 10k **electron**, for different η regions and 6 z-slices.

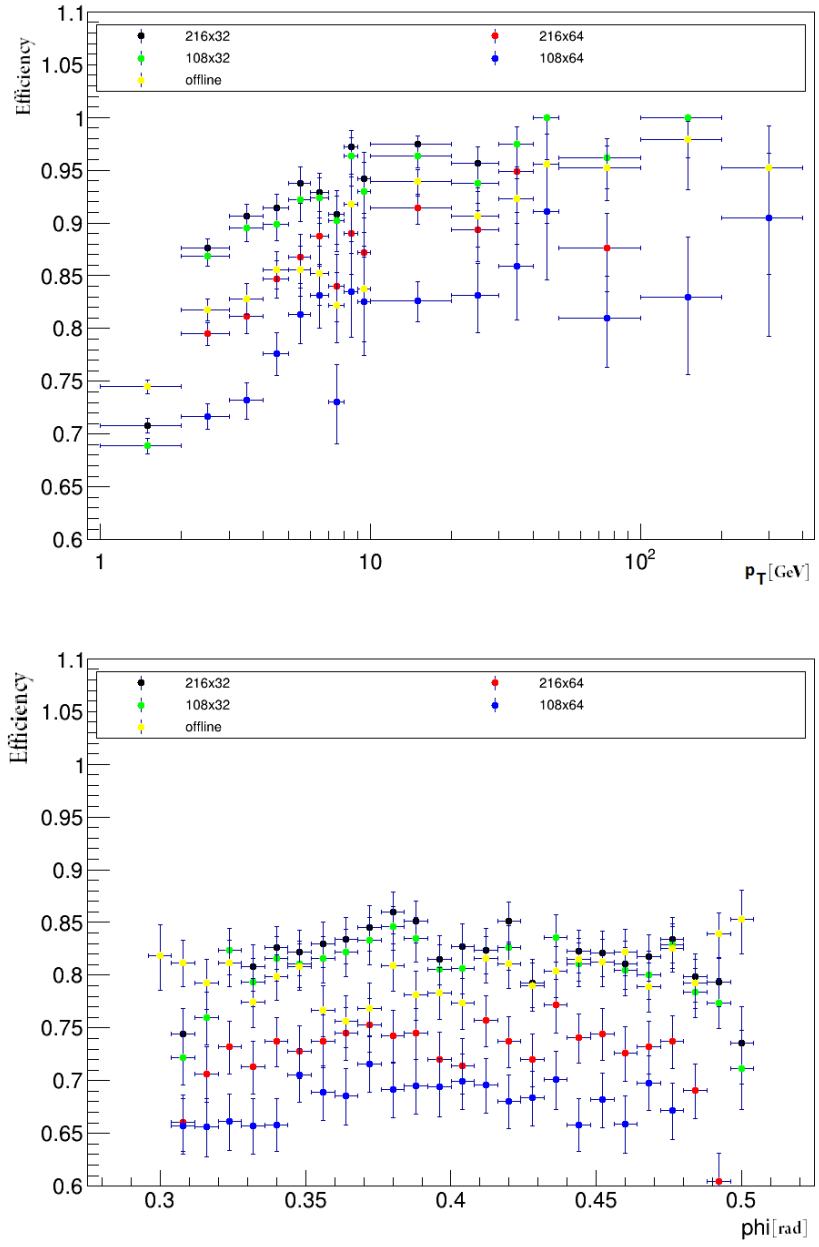

Fig. 5.24: Efficiency plots for *electron*, $0.1 < \eta < 0.3$.

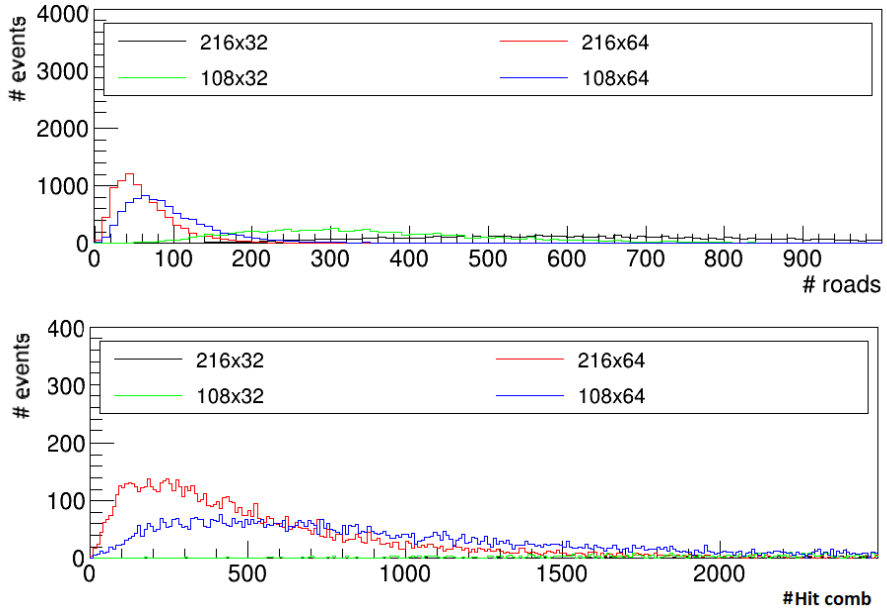

Fig. 5.25: Number of roads found per event (top), number of hit combinations per event (bottom), *electron* with $0.1 < \eta < 0.3$.

The $0.7 < \eta < 0.9$ region is also analysed; the track reconstruction efficiency plots for single muon, pion and electron samples are collected together as a function of p_T in Fig. 5.26.

The behaviour of the 4 accumulators considered in this study is almost the same even in the $0.7 < \eta < 0.9$ region, as shown in Fig. 5.27 and Fig. 5.28.

For the hardware emulation, the target board considered is the Xilinx UltraScale+ FPGA VCU1525; this work is still in progress, but similar results with respect to those previously obtained are expected.

New hardware optimisations and more fine-tuning parameters, especially for the number of bins of the accumulator and the z-slices may lead to better results, the 19 and 21 z-slices have not yet been investigated in depth and will be studied.

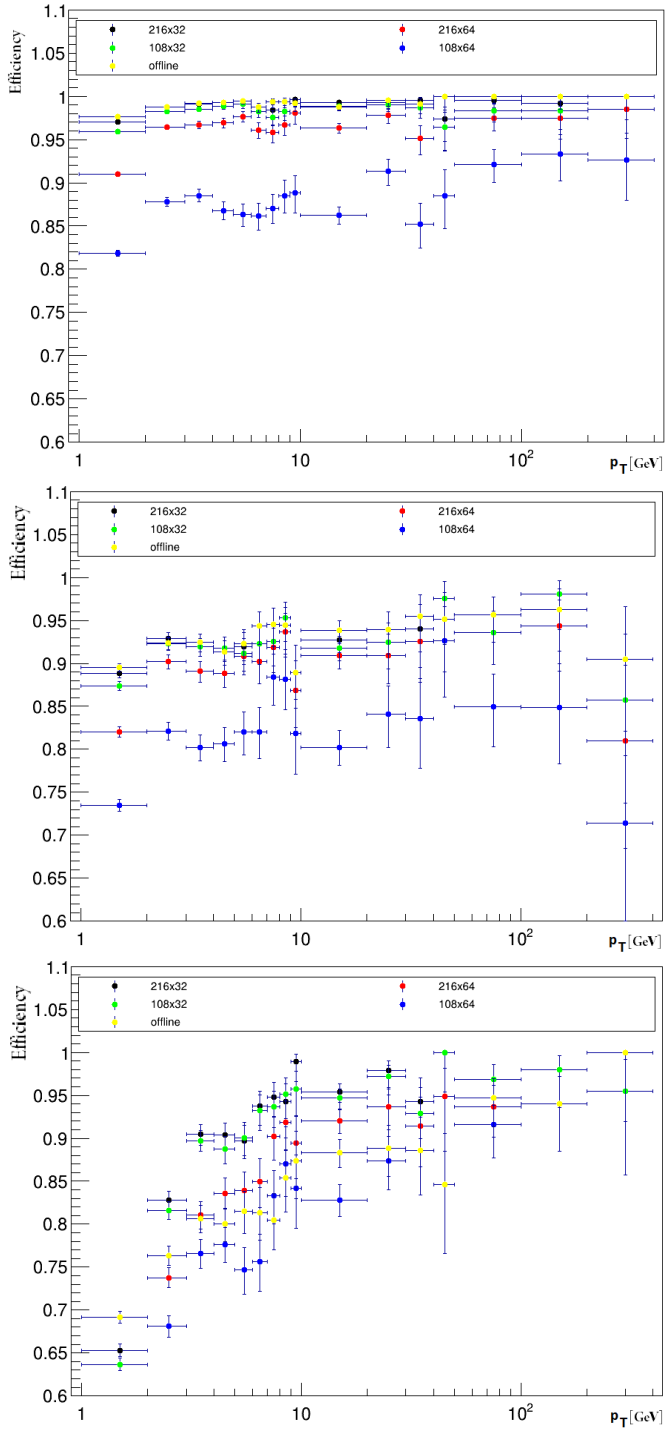

Fig. 5.26: Top to bottom: *muon*, *pion*, and *electron* efficiency as a function of p_T, $0.7 < \eta < 0.9$.

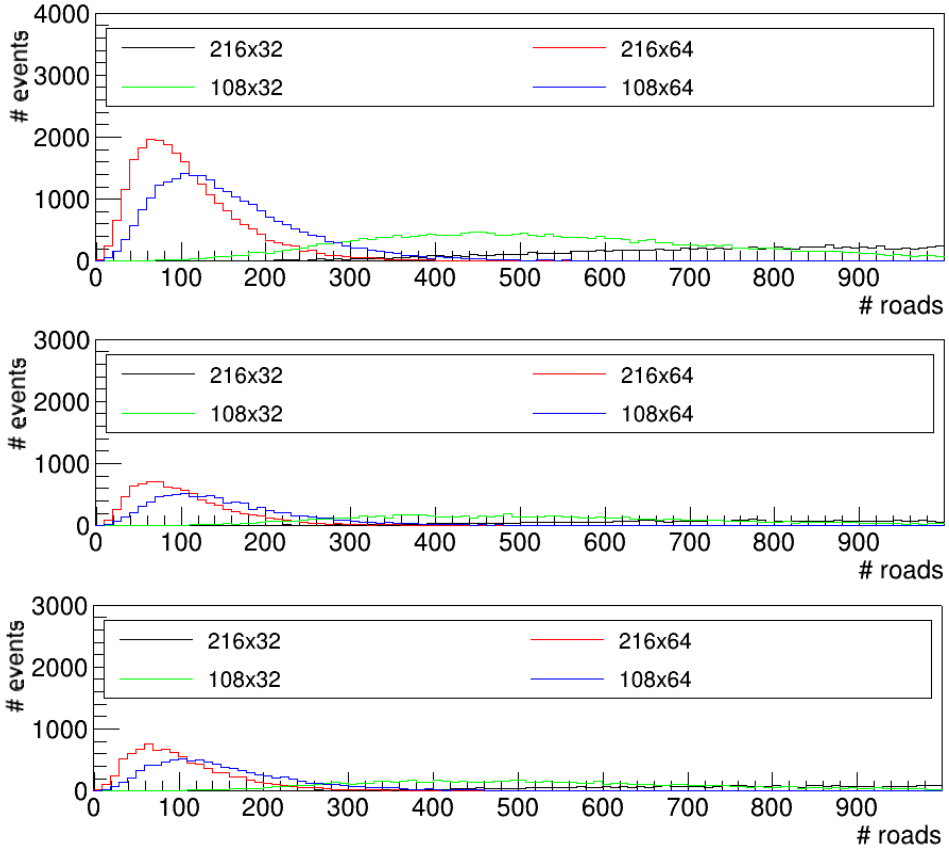

Fig. 5.27: Top to bottom: *muon*, *pion*, and *electron* roads found per event, $0.7 < \eta < 0.9$.

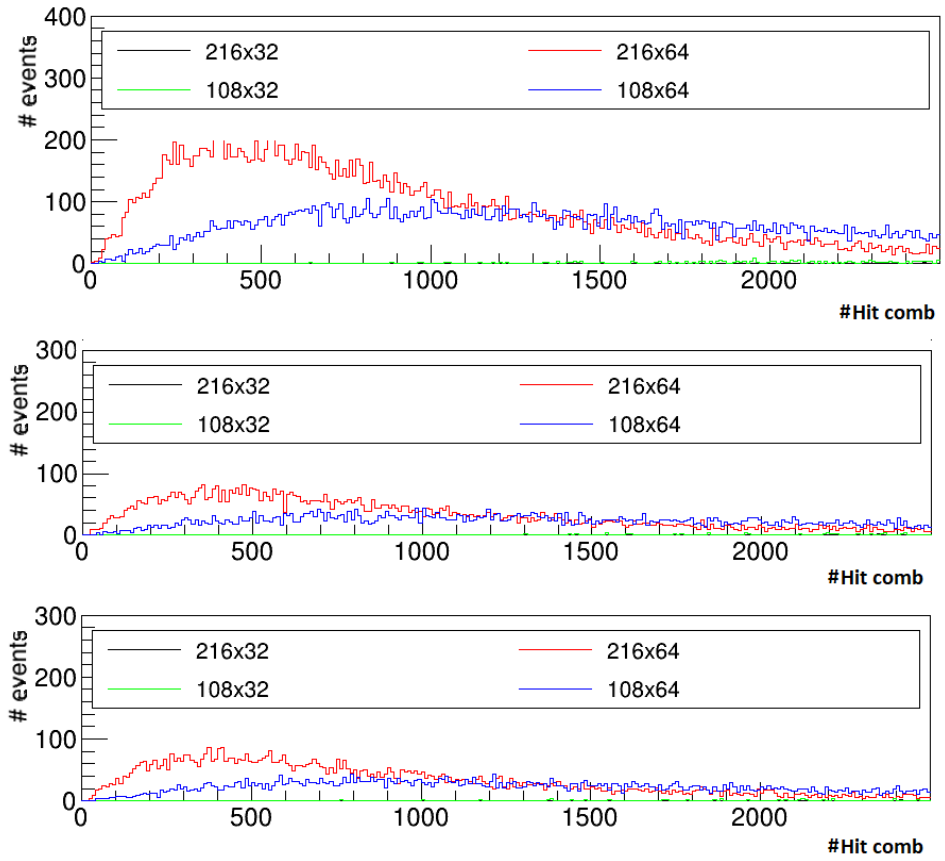

Fig. 5.28: Top to bottom: *muon*, *pion*, and *electron* number of hit combinations per event, $0.7 < \eta < 0.9$.

Conclusions

FPGA devices have proven to be good devices for hosting mathematical algorithms such as the HT. Comparing their behaviour with other commercial devices, FPGA devices are the best compromise for low and fixed system latencies, low power budget, and high data rate; it is perfect for high input-output data transfer rate as the overall latency scales linearly. The simulations are performed on Xilinx Ultrascale+ family accelerator boards.

Currently, the HT is one of the track reconstruction algorithms under evaluation for the Event Filter of the ATLAS Phase-II Trigger and Data Acquisition system upgrade.

The aim of this PhD thesis is to develop a software that emulates the behaviour of an ongoing HT firmware design, in order to study it thoroughly prior to laboratory testing, especially when added with unwanted data simulating a real harsh environment.

Dummy test vectors are generated to feed the software and the hardware with the same input datasets, in order to estimate the performance of the overall architecture in finding candidate tracks and associated clusters.

The system obtained is flexible and able to adapt to any possible changes in the scenario, easily modifiable for the way it has been developed.

The results obtained are in line with expectations, meaning that the integration of the software and firmware designs had been successfully achieved, proving the feasibility of implementing the HT algorithm on hardware and extracting the roads expected from the simulations.

The data corresponding to the ATLAS simulated events are then fed into the software and hardware systems, first using a Python software tool, then the Athena framework, after the code has been ported to the system using the C++ language.

System performance evaluations are carried out to provide a measure of the track reconstruction efficiency and the background rejection capability. A lower track detection capability is obtained compared to other pure software techniques.

Once this work is complete, a process of understanding and tuning the parameters begins, to find the best compromise between efficiency in track reconstruction and firmware implementation feasibility.

These studies of the HT for application to high energy physics experiments are not new, but, in particular for the hardware particle detection tasks in tracking systems, they have never been completed using an FPGA implementation.

New hardware optimisations and more fine-tuning parameters will be the subject of further studies to find the best performance.

Appendix A

Hardware Tracking for the Trigger

For the EF tracking system of the ATLAS TDAQ Phase-II, several options have been considered in the past [27]. Its initial architecture A.1 consisted of a large processor farm with CPUs and GPUs, capable of dealing with the 1 MHz input rate, and a Hardware Tracking for the Trigger (HTT) subsystem based on custom Associative Memories ASICs for pattern recognition and FPGAs for track reconstruction, fitting, and duplicate removal.

My project was born in this earlier project, which was then abandoned.

The HTT is organized as an array of independent tracking units called HTT units. The main hardware blocks of the HTT are ATCA boards called Tracking Processors (TP): the Associative Memories Tracking Processor (AMTPs), equipped with the Pattern Recognition Mezzanine (PRM), and the Second-Stage Tracking Processor (SSTP). They are connected to the EF processor farm via an HTT Interface (HTTIF).

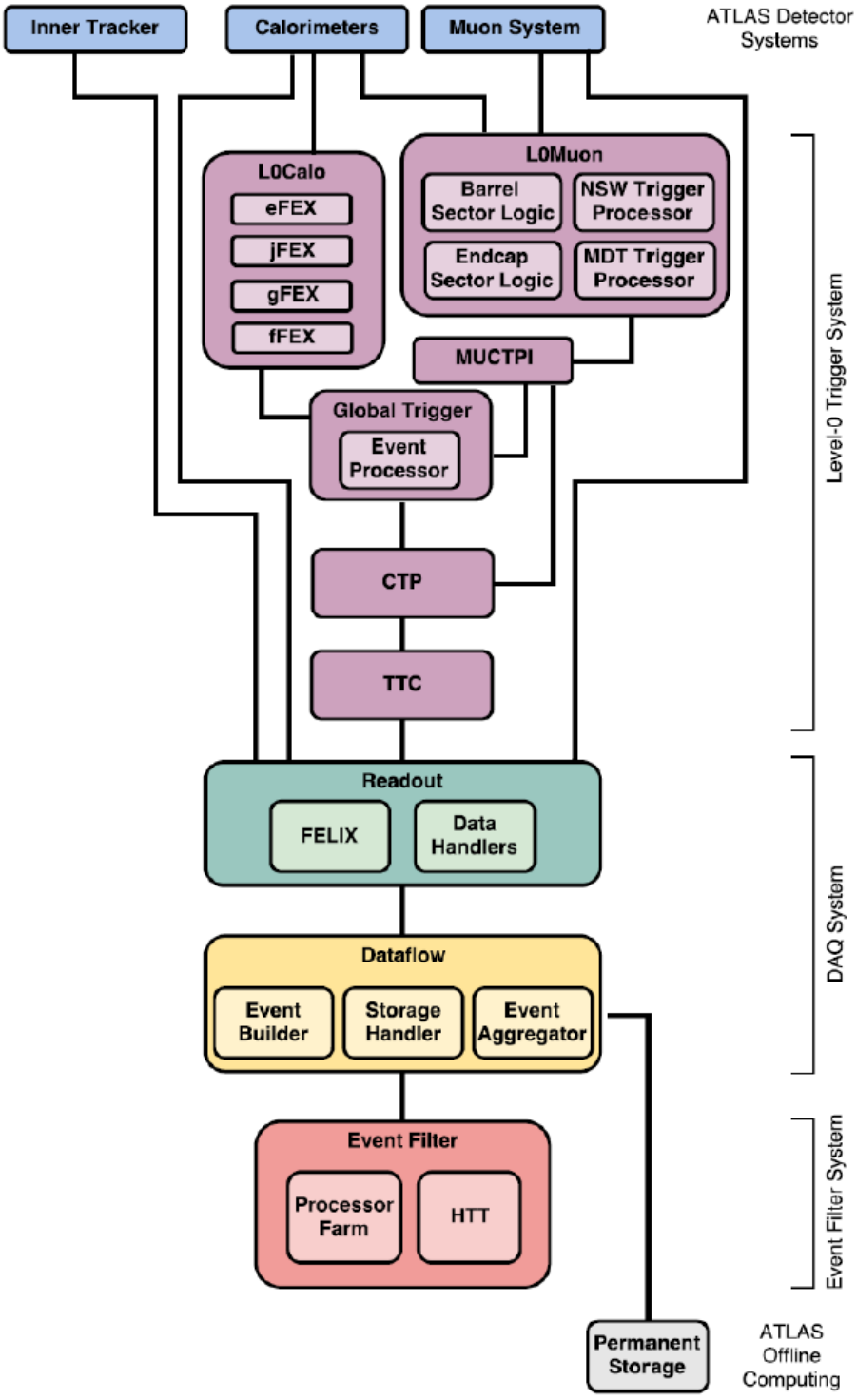

Fig. A.1: ATLAS TDAQ Phase-II - Initial Event Filter System.

The HTT also consists of two subsystems with identical hardware but different functions, the regional (rHTT) and the global (gHTT).

- rHTT reconstructs tracks with $p_T > 2GeV$ at a rate of 1MHz and uses up to 10% of the ITk data, by selecting tracking modules in RoI based on the results of the Level-0 trigger system. It allows a fast initial rejection in the EF of single high-p_T lepton and multi-object triggers from background processes, to reduce the rate to around 400 kHz.
- gHTT reconstructs tracks with $p_T > 1GeV$ in the full detector volume at a nominal rate of 100kHz, closer to offline quality. It achieves further rejection based on a software-based reconstruction, suitable for b-jet tagging, E_T^{miss} soft term calculation, soft jets and pile-up suppression.

Track reconstruction in the HTT proceeds as follows (see Fig. A.2). ITk hits are the input data of the HTT, transmitted via commodity network through the HTTIF.

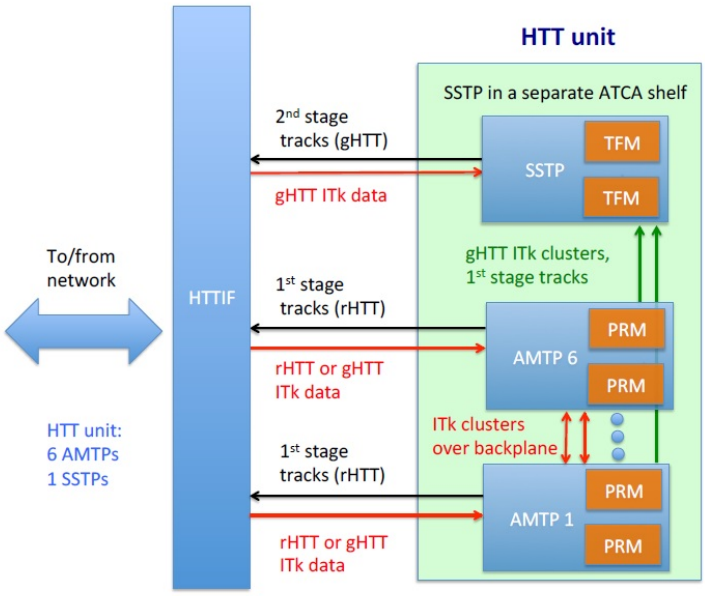

Fig. A.2: Diagram of a HTT unit.

There are two stages of processing. The first stage is characterized by clustering, pattern matching, and removing duplicates performed by AMTPs and used for regional

and global tracking. The hits from eight ITk layers are clustered into consecutive ITk strip or superstrip, then they are compared with a *pattern bank*, a collection of template patterns formed by single-muon tracks in simulated training events. Once all the patterns have been made, similar patterns (see tracks A and B in Fig. A.3) are combined by setting some of the least significant bits, the so-called "Don't Care" (DC) bits, to be ignored in some patterns. Its use offers better efficiency, lower pattern-matching rate and reduced number of patterns to be stored.

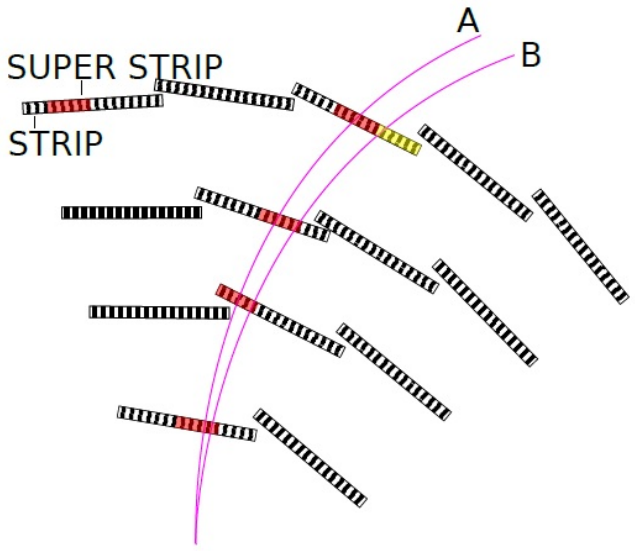

Fig. A.3: Tracks A and B traversing layers divided in superstrips.

At the end, the remaining patterns are sorted in decreasing "popularity" (number of muons that create that pattern) and written in the store pattern bank.

In a second stage, the gHTT system performs a full (13 layers) track fit to achieve the best possible track parameter resolution, using clusters from the 5 layers not included in the 1-st stage fit associated with the extrapolation of the track found in the 1-st stage 8-layer fit. Clusters which are compatible with particle tracks are sent to a track-fitter performed in the Track Fitting Mezzanine (TFM), a board with two FPGA inside the SSTP. The second stage is used only for global tracking.

Finally, the selected candidate tracks and the χ^2 associated to the fit of the track are the output.

rHTT searches for all tracks with $p_T > 2\ GeV$ in limited regions around Level-0 trigger objects, while gHTT searches for all tracks with $p_T > 1\ GeV$ at the nominal rate of 100kHz.

The latency requirement in the HTT depends on how long the data can be buffered and how fast the ITK can be read. If no latency is required on the HTT, data can be buffered in the Event Filter for seconds. If data have to be buffered in the readout electronics, the ITk has to be read out at a much higher rate.

Associative Memories

AM ASICs are based on a technology called content-addressable memory (CAM). This memory compares the SuperStrip Identifiers (SSID) of input ITk clusters with those of the the stored clusters (predefined patterns from tracks) in its memory and returns the address of the data that matches the input (if it finds a match).

This pattern matching method for track reconstruction was developed in the CDF experiment and further optimized in the FTK and HTT projects.

The HTT data input are hits from the eight ITk layers, firstly organized in *clusters* using clustering algorithms and combined into groups of consecutive silicon strip or pixel channel, the so-called *superstrips*. A superstrip has a coarse resolution with respect to single hits. A particle traversing the detector will trace out a *pattern* formed in each layer by one superstrip to which a unique SSIDs is associated. Therefore, a single pattern describes a sequence of eight SSIDs in different layers of the detector.

An example of the pattern matching with AM ASICs in four detector layers is illustrated in Fig. A.4 [38]. In the presented example, each layer is composed of a single module with six strips combined into three superstrips, and the pattern banks contain five patterns. The χ symbol is an X in the table and denotes the *don't care* bits.

In hardware implementation, AM chips store a large bank of precomputed and simulated patterns called *roads*. These patterns represent physical regions of the detector defined by the physic events of interest. Each bank is stocked with an address in the memory, and its patterns are generated using large samples of simulated muon tracks with track parameters that cover the RoI. For example, if one hit occurs in a certain region defined by a specific pattern, then all the hits inside the same region must be

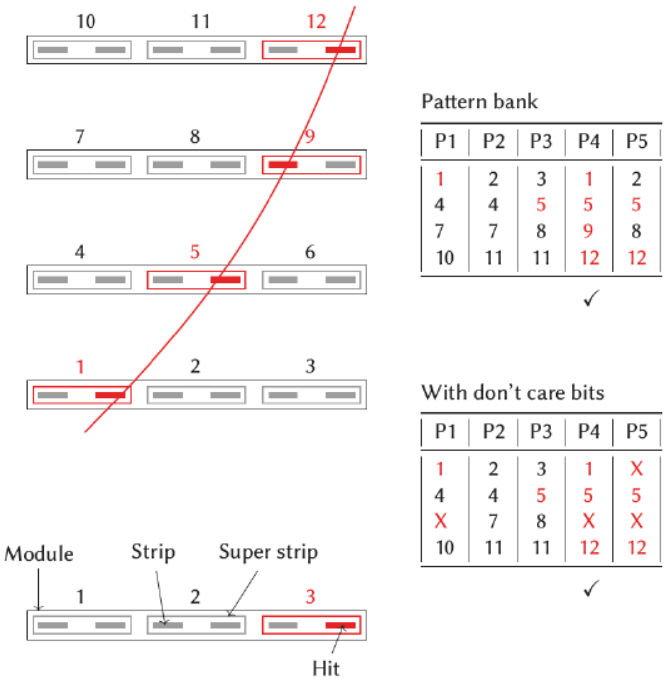

Fig. A.4: Example of the Pattern Matching with AM ASICs in four detector layers.

studied and processed in the track fitting. AM ASICs programmed with a pattern bank will try to match an input pattern and output the address of the matching pattern.

Approximately one million patterns are necessary to cover a RoI of 0.2×0.2 in $\eta \times \phi$ for tracks with $p_T > 2\ GeV$.

The "don't care bits", or ternary bits, are used to combine two or more similar patterns, to free up space and include more patterns increasing track finding efficiency. "Don't care bits" match regardless of the input.

Track fitting

A track-fitter is implemented in the firmware and is performed in an FPGA in the PRM (see Fig. A.2). It takes the full resolution hits of the roads passed by pattern matching and calculates the p_i and the χ^2 track fit parameters. The track parameters p_i are calculated using a linear interpolation:

$$p_i = \sum_{j=1}^{N} C_{ij} x_j + q_i \tag{A.1}$$

where x_j are the coordinates of the full resolution clusters and $(C_{ij}\ ,q_j)$ are unique constants for each sector, where *sector* is defined as a combination of one module from each of the 8 layer used in the first-stage processing. The constants are determined from a large sample of simulated muon tracks with the same parameter ranges and distributions as those used in generating the patterns. The quality of the fit is evaluated through a linearised χ^2 method:

$$\chi^2 = \sum_{i=1}^{N-5} \big(\sum_{j=1}^{N} A_{ij} x_j + k_i \big)^2 \tag{A.2}$$

where A_{ij} and k_i are the additional constants needed per sector.

χ^2 can be computed with FPGAs technology. Both fitting constants and constants used to compute χ^2 have to be stored in internal memory on FPGA, external memories, or both of them. Since they need to be retrieved from memory for each sector in the event, the fitting hardware could run into a bottleneck issue. In order to fit a region of $\eta \times \phi = 0.2 \times 0.2$, several thousand sectors are required. This corresponds to about forty million coefficients that need to be stored in external memories on the PRM or in the internal FPGA memory. With some optimization, this quantity of memory can be reduced.

In the second stage, track fitting is performed on the Track-Fitting Mezzanine (TFM) in a FPGA. For every track the TFM calculates the five helix parameters and the χ^2 of the fit using the same equations as in the first step. The TFM receives all the hits from the detector layers not used by the PRM and the 8-layer tracks from 6 PRM cards. The TFM implements two functions: *Extrapolator*, which finds near a PRM track the hits on the additional silicon layers, and the *Track Fitter*, which fits the hits on the PRM track with each combination of hits on the other layers and applies a χ^2 cut. The duplicated tracks in an event are then removed using the HitWarrior algorithm, before sending the track candidates to the Event Filter.

Appendix B

Software Code

All the Python and C++ code developed during this PhD thesis is accessible by clicking on the QR code below.

To get access, please send an email to `francesca.delcorso@unibo.it`.

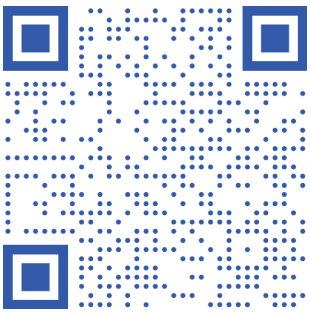

Fig. B.1: QR code for the software.

Acknowledgments

I graduated in Computer Science in 1996; the PhD in Physics, which started in 2019, was a great challenge for me, as you can well imagine, but the beauty of life is also to always set new goals and try to achieve them in the best way.

I would like to thank all those who have contributed to the realisation of this PhD. In particular, I would like to thank my supervisor, Alessandro Gabrielli, for his constant support, his kindness and his openness to everything, from electronics to real life problems. Thank you for your recommendations, hints, opportunities and experiences!

I would also like to thank Francesca Pastore, for supervising and supporting my ATLAS qualification task, and all the people involved in the HTT project.

I cannot forget to thank my colleagues Fabrizio, for working hard with me and sharing his knowledge on firmware, and Kazuki, for his support and invaluable feedback on physics and data analysis.

Thank you all for making the physics a bit clearer for me!

If I have been able to do this PhD, I also owe it to Franco Semeria, my supervisor in the Calcolo e Reti group, and to the director of INFN Bologna, Eugenio Scapparone, for allowing me to do it.

I would also like to thank my family for their support.

Finally, I'd like to say that it's never too late to embark on a new and fascinating journey into fields of knowledge different from those we know, and to leave our comfort zone for a while, to become more complete and better people. Great opportunities and wonderful experiences can arise that will reward us for our efforts. It's up to us to know how to grab them!

Acronyms

AFP ATLAS Forward Proton.
AM Associative Memories.
AMTP Associative Memories Tracking Processor.
ALFA Absolute Luminosity For ATLAS.
ATLAS A Toroidal LHC ApparatuS.
CERN European organization for nuclear research.
CMS Compact Muon Solenoid.
CSC Cathode Strip Chambers.
DSP Digital Signal Processing.
ECAL ELectromagnetic CAlorimeter.
EFT Event Filter Tracking.
EMB ElectroMagnetic Barrel.
EMEC ElectroMagnetic End-Cap.
FCAL Forward Calorimeter.
FPGA Field-Programmable Gate Array.
GeanT GEometry ANd Tracking.
HDL Hardware Definition Language.
HEC Hadronic End-Caps Calorimeters.
HLT High-Level Trigger.
HL-LHC High-Luminosity LHC.
HT Hough Transform.
HTC Hadronic Tile Calorimeters.
HTT Hardware Tracking for the Trigger.

IBL Insertable Barrel Layer.

IDE Integrated Development Environment.

IP Interaction Point.

ITk Inner Tracker.

JSON JavaScript Object Notation.

KF Kalman Filter.

LAr Liquid-Argon.

LHC Large Hadron Collider.

LLPs Long-Lived Particle.

LRT Large-radius tracking.

LS Long Shutdowns.

LUCID Luminosity measurement using Cherenkov Integrating Detector.

MDT Monitored Drift Tubes.

NSW New Small Wheel.

PD Pixel Detector.

QCD Quantum Chromodynamics.

RoI Region of Interest.

RPC Resistive Plate Chamber.

SCT Semi-Conductor Tracker.

SM Standard Model.

SSTP Second-Stage Tracking Processor.

SSID SuperStrip Identifier.

STGC Small strip Thin Gap Chambers.

TDAQ Trigger and Data Acquisition.

TDR Technical Design Report.

TFM Track Fitting Mezzanine.

TLAs Trigger Level Object analysis.

TRT Transition Radiation Tracker.

TP Tracking Processor.

TV Test Vector.

ZDC Zero-Degree Calorimeter.

Bibliography

[1] LHC Study Group, *The Large Hadron Collider Conceptual Design*, CERN-AC-95-05 (LHC), 1995, https://cds.cern.ch/record/291782.

[2] M. Krause, *CERN How We Found the HIGGS BOSON*, World Scientific Publishing Co., Nov 2014. https://cds.cern.ch/record/1748524.

[3] ATLAS Collaboration, *The ATLAS Experiment at CERN Large Hadron Collider*, JINST, vol. 3, S08003, 2008.

[4] CMS Collaboration, *The CMS Experiment at the CERN LHC*, JINST, vol. 3, S08004, 2008.

[5] LHCb Collaboration, The LHCb Detector at the LHC, JINST, vol. 3, S08005, 2008.

[6] ALICE Collaboration, *The ALICE experiment at the CERN LHC*, JINST, vol. 3, no. 08, p. S08002, 2008.

[7] *ATLAS HL-LHC Industry web site*, https://project-hl-lhc-industry.web.cern.ch/

[8] ATLAS Collaboration, *ATLAS Inner Detector: Technical Design Report*, 1, CERN-LHCC-97-016 ATLAS-TDR-4 (1997), https://cds.cern.ch/record/331063.

[9] ATLAS Collaboration, *ATLAS Pixel Detector: Technical Design Report*, CERN-LHCC-98-013 ATLAS-TDR-11, https://cds.cern.ch/record/381263.

[10] Y. Unno, *ATLAS silicon microstrip detector system (SCT)*, Nucl. Instrum. Meth., vol. A511, pp. 58–63, Sept.2003.

[11] E. Abat et al., *The ATLAS Transition Radiation Tracker (TRT) proportional drift tube: Design and performance*, JINST, vol. 3, P02013, 2008.

[12] ATLAS Collaboration, *ATLAS Insertable B-Layer Technical Design Report*, ATLAS-TDR-19 CERN-LHCC-2010-013, 2010, `http://cds.cern.ch/record/1291633`.

[13] D. Caforio, *Luminosity measurement using Cherenkov Integrating Detector (LUCID) in ATLAS, in Astroparticle, particle and space physics, detectors and medical physics applications*, Proceedings, 10th Conference, ICATPP 2007, Como, Italy, October 8-12, 2007, 2008, pp. 413–417.

[14] ATLAS Collaboration, *Zero Degree Calorimeters for ATLAS*, CERN-LHCC-2007-001 LHCC-I-016, Jan 2007, `https://cds.cern.ch/record/1009649`.

[15] L. Adamczyk et al., *Technical Design Report for the ATLAS Forward Proton Detector*, Tech. Rep. ATLAS-TDR-024 CERN-LHCC-2015-009, May 2015, `https://cds.cern.ch/record/2017378`.

[16] S. Jakobsen, P. Fassnacht, P. Hansen, and J. B. Hansen, *Commissioning of the Absolute Luminosity For ATLAS detector at the LHC*, Dec 2013, presented 31 Jan 2014, `https://cds.cern.ch/record/1637195`.

[17] P. Jenni, M. Nessi, M. Nordberg, and K. Smith, *ATLAS high-level trigger, data-acquisition and controls: Technical Design Report*, ser. Technical Design Report ATLAS, Geneva, CERN, 2003, `https://cds.cern.ch/record/616089`.

[18] ATLAS Collaboration, *ApprovedPlotsDAQ*, `https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ApprovedPlotsDAQ`.

[19] ATLAS Collaboration, *Letter of Intent for the Phase-II Upgrade of the ATLAS Experiment*, CERN-LHCC-2012-022 LHCC-I-023, `https://cds.cern.ch/record/1502664?ln=en`.

[20] ATLAS Collaboration, *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*, ATLAS-TDR-030 CERN-LHCC-2017-021, https://cds.cern.ch/record/2285585?ln=en.

[21] ATLAS Collaboration, *Technical Design Report for the ATLAS Inner Tracker Strip Detector*, ATLAS-TDR-025 CERN-LHCC-2017-005, https://cds.cern.ch/record/2257755.

[22] ATLAS Collaboration, *ATLAS Phase-II upgrade Scoping Document*, CERN-LHCC-2015-020 LHCC-G-166, https://cds.cern.ch/record/2055248?ln=en.

[23] ATLAS Collaboration, *Technical Design Report: A High-Granularity Timing Detector for the ATLAS Phase-II upgrade*, ATL-TDR-031 CERN-LHCC-2020-007, https://cds.cern.ch/record/2719855?ln=en.

[24] ATLAS Collaboration, *ATLAS Liquid Argon Calorimeter Phase-II Upgrade Technical Design Report*, ATLAS-TDR-027 CERN-LHCC-2017-018, https://cds.cern.ch/record/2285582?ln=en.

[25] ATLAS Collaboration, *Technical Design Report for the Phase-II Upgrade of the ATLAS Tile Calorimeter*, ATLAS-TDR-028 CERN-LHCC-2017-019, http://cds.cern.ch/record/002285583.

[26] ATLAS Collaboration. *Technical Design Report for the Phase-II Upgrade of the ATLAS Muon Spectrometer*, ATLAS-TDR-026 CERN-LHCC-2017-017, https://cds.cern.ch/record/2285580?ln=en.

[27] ATLAS Collaboration, *Technical Design Report for the Phase-II Upgrade of the ATLAS Trigger and Data Acquisition System*, ATLAS-TDR-029 CERN-LHCC-2017-020, https://cds.cern.ch/record/2285584?ln=en.

[28] ATLAS Collaboration, *Technical Design Report for the Phase-II Upgrade of the ATLAS Trigger and Data Acquisition System - EF Tracking Amendment*, ATLAS-TDR-029-ADD-1 CERN-LHCC-2022-004, https://cds.cern.ch/record/2802799.

[29] *Global Track parameter*, Available online: `https://atlassoftwaredocs.web.cern.ch/trackingTutorial/idoverview/`

[30] P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures*, Conf. Proc. C 590914 (1959) 554-558, `https://inspirehep.net/literature/919922`.

[31] P. V. Hough, *Method and means for recognizing complex patterns*, Dec., 1962. US Patent 3, 069, 654.

[32] R. O. Duda and P. E. Hart, *Use of the Hough Transformation to Detect Lines and Curves in Pictures*, Commun. ACM, Vol. 15, pp. 11–15, Jan. 1972.

[33] D. H. Ballard, *Generalizing the Hough Transform to detect arbitrary shapes*, Pattern Recognition Vol. 11, No.2. pp. 11 1122. 1981.

[34] L. Rinaldi et al., *GPGPU for track finding in High Energy Physics*, Proceedings, GPUHEP2014, Pisa, Italy, September 10-12,2014. 2015, 17. `https://arxiv.org/abs/1507.03074`.

[35] N. Pozzobon, F. Montecassiano, P. Zotto, *A novel approach to Hough Transform for implementation in fast triggers*, Nucl. Instrum. Meth. A,834 (2016), 81.

[36] K. A. Mohammad Kamal Azmi, W. A. T. Wan Abdullah, and Zainol Abidin Ibrahim, *Hough transform method for track finding in center drift chamber*, AIP Conference Proceedings 1704.1 (2016), p. 030014, doi: 10.1063/ 1, 4940083.

[37] R. Gonzalez, R. E. Woods, *Digital Image Processing*, Global Edition, 4th Ed., Pearson, 2018.

[38] M. Mårtensson, *A search for leptoquarks with the ATLAS detector and hardware tracking at the High-Luminosity LHC*, PhD thesis, Uppsala U., 2019.

[39] F. Alfonsi, *Study and Optimization of Particle Track Detection via Hough Transform Hardware Implementation for the ATLAS Phase-II Trigger Upgrade*, PhD thesis, Bologna U., 2021.

[40] A. Gabrielli et Alt., *Hardware Implementation Study of Particle Tracking Algorithm on FPGAs*, Electronics 2021, 18 Oct. 2021, https://www.mdpi.com/2079-9292/10/20/2546.

[41] F. Alfonsi, F. Del Corso, A. Gabrielli, *Simulated Hough Transform Model Optimized for Straight-Line Recognition Using Frontier FPGA Devices*, Electronics 2022, 9 Febr. 2022, http://dx.doi.org/10.3390/electronics11040517.

[42] F. Alfonsi, F. Del Corso, A. Gabrielli, *Hough Transform Proposal and Simulations for Particle Track Recognition for LHC Phase-II Upgrade*, Sensors 2022, 24 Febr. 2022, https://doi.org/10.3390/s22051768.

[43] F. Alfonsi, F. Del Corso, A. Gabrielli, G. Levrini, K. Todome, *Hough transform Software, Firmware, and Hardware investigation for fast tracking in Phase-II LHC upgrade*, Book of Abstract, TWEPP 2022 Topical Workshop on Electronics for Particle Physics, p.37-38, 19-23 Sett. 2022, https://indico.cern.ch/event/1127562/book-of-abstracts.pdf.

[44] F. Alfonsi on behalf of the ATLAS TDAQ Community, *A FPGA Implementation of the Hough Transform tracking algorithm for the Phase-II upgrade of ATLAS*, poster session, ACAT 2022, Bari, Italy, 23-28 Ott. 2022, https://indico.cern.ch/event/1106990/contributions/4991260/attachments/2533440/4359527/poster_acat2022_fabrizio_alfonsi_pre_20221015.pdf.

[45] XILINX, *UltraScale+ FPGAs Product Selection Guide (XMP103)*, Xilinx VU9P, Available online: https://docs.xilinx.com/v/u/en-US/ultrascale-plus-fpga-product-selection-guide.

[46] NI, *FPGA Fundamentals*, Available online: https://www.ni.com/it-it/innovations/white-papers/08/fpga-fundamentals.html.

[47] *Anaconda web site*, Available online: https://www.anaconda.com/

[48] *Python official web site*, Available online: https://www.python.org/

[49] *Microsoft Visual Studio Code web site*, Available online: `https://code.visualstudio.com/`

[50] *JavaScript Object Notation*, Available online: `https://www.json.org/json-it.html`

[51] *Matplotlib: Visualization with Python*, Available online: `https://matplotlib.org/`