Utrecht University

Institute for Subatomair Physics

Bachelor Thesis

ALPIDE Testing Software for the upgraded Inner Tracking System of the ALICE experiment

Author:
Lukas Arts

Supervisor:
Dr. Paul Kuijer
Dr. Andre Mischke

January 11, 2016

Abstract

ALICE, a Large Hadron Collider experiment, will be upgraded during the long shutdown in 2018/2019. The Inner Tracking System (ITS) is one of the sub-detectors of ALICE and forms the heart of the detector. Its main job is to track low momenta particles by the use of silicon detectors. During the upgrade, these silicon detectors will be replaced with modern silicon pixel chips. The demands for these chips are very high because of their harsh operating environment and therefore a higher error rate in the production than normal is expected.

In this bachelor project, a graphical interface for testing software has been designed and created to perform tests and scans on a beta-version of the ALICE Pixel Detector (ALPIDE) chip. The basic communication between the chip and the computer is provided by a library which has been written by INFN (Italy). The graphical interface uses C++ 11 and Qt 5.5.

A register, threshold and noise scan have been implemented. They test the core and individual pixels of the chip. The scans provide data on which a first prediction can be made whether the chip meets its high demands or not. During a scan, the obtained data is plotted and can be examined. The testing software and its structure provide a solid foundation for further development. The software is built on a flexible basis which allows the addition of more scans and tests. Full documentation of the code is attached in the appendix.

Contents

Chapter 1

Introduction

During the long shutdown of the Large Hadron Collider at CERN in 2018/2019, the ALICE experiment will be upgraded to enhance its physical capabilities [1]. In this thesis we will study the upgrade of the Inner Tracking System. These are the detectors closest to the beam pipe and consists of silicon tracking detectors [4]. During the upgrade, these silicon detectors will be replaced with high resolution silicon pixel chips. These so called ALPIDE chips implement the new CMOS MAPS technology [1]. This technology is used in many chips but the very high demands that are imposed on the silicon wafers result in a poor success rate. About 1/3 of the chips produced on the silicon waffle will not work. The high demands that are imposed on the chip in combination with the shipping makes it fragile and vulnerable for cracks.

ALICE cannot afford the case when a newly implemented chip does not work. Therefore, the chips have to be tested before they are placed in the detector. This calls for a method to test the ALPIDE chips. This bachelor thesis describes the development of this method, which includes testing soft- and hardware. The software runs on a desktop computer, which is connected to an external testing module. This module is connected to an ALPIDE chip. The computer sends commands to the testing module where the commands are being translated to meaningful commands for the ALPIDE.

The testing software tests a chip by the use of several scans. These scans test the internal core, pixels and registers of the chip. When an error has been detected, the software gives the possibility to examine the test results to determine the nature and detailed information of the error.

This thesis will focus on the development of the tests and graphical interface of the software. At first we will discuss the ins and outs of detector ALICE. A small theoretical overview will give an introduction to several terms used further on. After this, we will discuss the upgrade of ALICE. The ALPIDE chip is explained in chapter 4 and the testing software is discussed in chapter 5. This chapter will shed more light onto subjects like the hardware, performed scans, graphical interface (GUI) and architecture of the software. Finally, we will discuss the idea's and possibilities for further developments.

Chapter 2

The strong interaction and the ALICE experiment

ALICE is an experiment at the Large Hadron Collider (LHC) [4]. ALICE is an acronym for 'A Large Ion Collider Experiment'. Together with ATLAS, CMS and LHCb it is one of the four main experiments at LHC [8]. Its general purpose is to study heavy-ion collisions and the resulting quark-gluon plasma at extreme temperature and pressure [4]. It will study a wide variety of hadrons, electrons, muons and photons produced in a Pb-Pb collision. Besides the collision of two heavy ions, it is also designed for lighter ion collisions like proton-proton collisions.

The ALICE detector has been built by a group of physicists and engineers from more than 130 institutes in 30 countries. It weights 10.000 tons and measures 16x16x26 m^3 [4]. The detector consists of two main parts. The central barrel holds all main detectors and thereby measures the hadrons, electrons and photons. Besides the central barrel, ALICE also holds a forward muon spectrometer. We'll discuss the central barrel part in more detail in the next sections but first a quick theoretical recap.

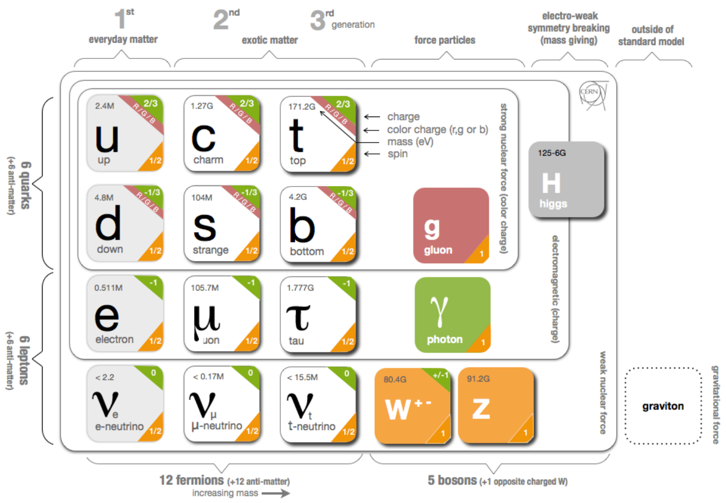
2.1 Theoretical Overview

An atom is made up of two parts. A nucleus and a surrounding electron-cloud. The nucleus is positively charged and the electron-cloud negatively. The interaction between these two is mediated by the electromagnetic force. This force is described by the quantum electrodynamics (QED). The proton's and neutron's which build up the nucleus, consist of quarks. The interaction between those quarks is mediated by gluons and is described by the quantum chromodynamics (QCD) [9]. These quarks and gluons are confined into a proton or neutron in normal circumstances. At extreme values of temperature and pressure the quarks and gluons become disconnected and a quark-gluon plasma (QGP) [11] emerges. This state of matter can be produced in a nucleus collisions.

2.1.1 The standard model

Matter is build up of a finite amount of elements. These elements are represented in the standard model [10]. The standard model contains a total of 12 elementary particles and 4 force interaction particles. The elementary particles are categorized into two groups, the quarks and leptons. Each category consists of 6 particles and 6 anti-particles. These are divided into three generations. The first generation are stable particles. These particles do not decay and they are the building blocks for everyday matter. Higher generation particles are much heavier and decay into lighter particles very fast. There are six different flavour's of quarks. Up, down, charm, strange, top and bottom. Although the lifetime of heavy quarks like top and bottom is short, we can learn a lot of them as we will see soon. Second generation quarks, charm and strange, have a decay time of a couple of picoseconds and the third generation even shorter.

The leptons consist of three generations electrons and neutrino's. An electron and electron neutrino, a muon and muon neutrino and the third generation contains tau and tau neutrino's. The neutrino's are neutrally charged and the electron's are negatively charged. The higher the generation, the heavier the particles. Muon and tau particles decay quickly to the first generation electron and neutrino as this was also the case with quarks. However, there is one difference with respect to quarks. Leptons are fundamental particles, unlike quarks.



1st everyday matter
2nd exotic matter
3rd generation
force particles
electro-weak symmetry breaking (mass giving)
outside of standard model
2.4M 2/3 R/G/B u up 1/2
1.27G 2/3 R/G/B c charm 1/2
171.2G 2/3 R/G/B t top 1/2
charge
color charge (r,g or b)
mass (eV)
spin
4.8M -1/3 R/G/B d down 1/2
104M -1/3 R/G/B s strange 1/2
4.2G -1/3 R/G/B b bottom 1/2
g gluon 1
0.511M -1 e electron 1/2
105.7M -1 μ μon 1/2
1.777G -1 τ tau 1/2
γ photon 1
< 2.2 0 νe e-neutrino 1/2
< 0.17M 0 νμ μ-neutrino 1/2
< 15.5M 0 ντ τ-neutrino 1/2
80.4G +/-1 W+- 1
91.2G Z 1
125-6G H higgs
graviton
strong nuclear force (color charge)
electromagnetic (charge)
weak nuclear force
gravitational force
6 quarks (+6 anti-matter)
6 leptons (+6 anti-matter)
12 fermions (+12 anti-matter)
increasing mass
5 bosons (+1 opposite charged W)
CERN

Figure 2.1: The standard model. [10]

All these particles interact with each other via four elementary forces. The other particles in the standard model are the gauge bosons. They provide a way to transport force between particles. In order of influence on small scales we have strong interaction force, weak interaction force, electromagnetic force and gravity. Gravity is the only elementary force we do not yet understand. Fortunately, gravity is very weak at small length scales and we can therefore neglect it when we study quark-gluon plasma. The strong interaction force interact by gluons, the weak interaction force via Z and W bosons and the electromagnetic force via photons. At the length scales of the QGP, the strong interaction force dominates. This force is described by QCD. To understand the QGP we have to discuss QCD.

2.1.2 Quantum chromodynamics (QCD)

Quantum chromodynamics describes the strong interaction between quarks and gluons [9]. Each quark has a particular flavor or color as they are called. There are three colors: red, green, blue and their anti-colors: anti-red, anti-green, anti-blue. Particles are always observed colorless. This means a combination of three quarks or three anti quarks or a combination of a quark and his anti-quark. Particles made up of three (anti-)quarks are called baryons and particles build up of two quarks are called mesons. Gluons also carry color but carry two colors at once. A color and its anti-color. Therefore gluons also interact with each other.

QCD has two important characteristics:

- **Confinement** The force between quarks does not disappear when two quarks are at large distances. The gluon field will always be able to create another quark anti-quark pair, which in their turn form a meson. This forbids free quarks at normal conditions.

- **Asymptotic freedom** This is a phenomena observed when the energy level is very high. At these high energetic reactions, the quark gluon interaction becomes weak and confinement is no longer demanded. A quark-gluon plasma (QGP) is created.

Asymptotic freedom and the existence of QGP has been predicted as far back as the 1970's by David Politzer and by Frank Wilczek and David Gross. ALICE has proven the QGP to be a real phenomena. This was first done by the RHIC experiment and has been repeated in ALICE.

2.1.3 Quark-gluon plasma (QGP)

The quark gluon plasma (QGP) arises when the temperature and density are very high [11]. These extreme values of temperature and density give rise to asymptotic freedom. In a QGP, quarks and gluons are no longer confined into hadrons. It is believed that the universe was in such a state when it was just a few microseconds old. If we want to learn more about our existence we need to know more about QGP.

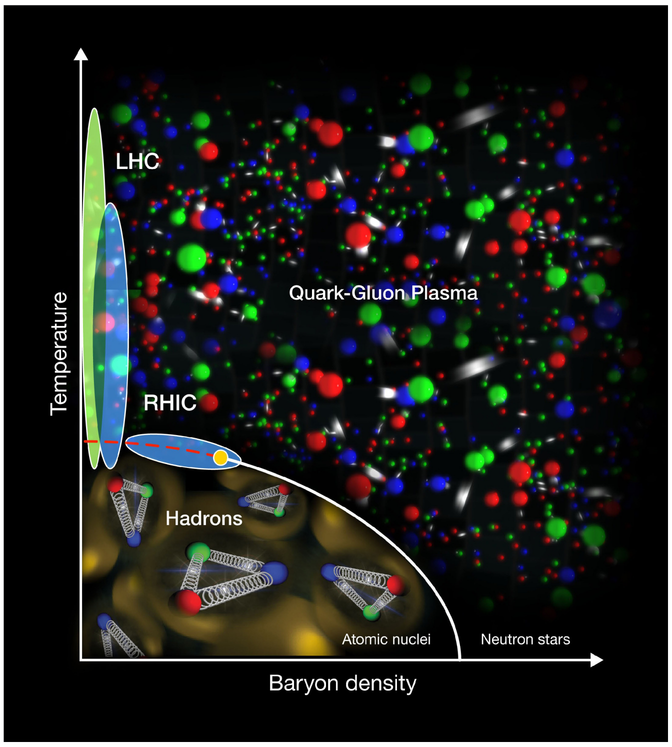


Figure 2.2: Phase diagram of hadronic matter. [5]

The phase diagram in figure 2.2 gives an understanding of the conditions under which a QGP is formed. In ALICE, the collisions of the particles raise the temperature up to an extreme level. The QGP which then emerges, expands very quick and has a short lifetime. The measuring range achieved by ALICE is colored yellow in the phase diagram.

Although a QGP does not exists very long, we can study it by the use of probes [11]. Probes are particles affected by the QGP, which can be measured outside the QGP. Heavy quarks are one of those probes. They are produced inside the QGP and make their way out of the plasma and turned into mesons. Because of their mass, they interact strongly with the QGP and the properties of the resulting mesons can therefore measured to get 'inside information'. Heavy quarks are produced in two ways:

1. $gg \rightarrow Q\bar{Q}$
2. $q\bar{q} \rightarrow Q\bar{Q}$

Two gluons collide to form a heavy quark anti-quark pair (1) or a quark and anti-quark combine to form a heavier quark anti-quark pair.

2.2 Detector Layout

As we already mentioned, ALICE consists of two main detector parts. In this section, we will zoom in on the central barrel part. The central barrel part holds all the detectors responsible for measuring hadrons, electrons and photons [4]. These particles are measured in several fases to get different kinds of information.

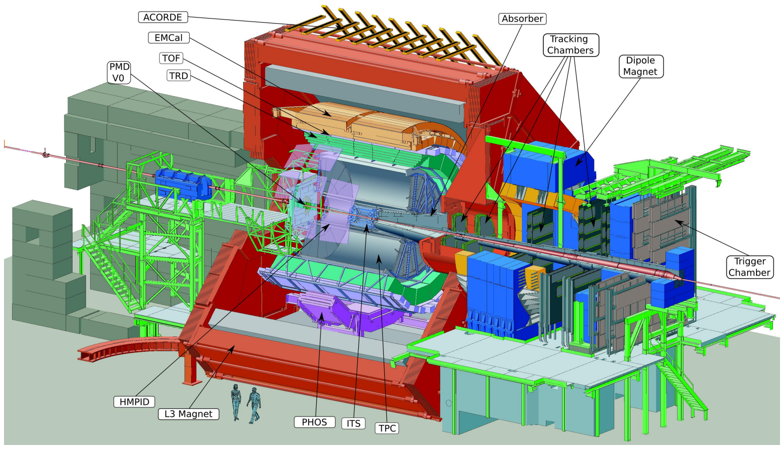


Figure 2.3: A schematic view of the ALICE detector. It consists of different concentric detectors around the beam-pipe. [6]

From inside out, we have the Inner Tracking System (ITS), which contains six layers of high-resolution Silicon Pixel Detectors (SPD), drift (SDD) and strip (SSD) detectors. Furthermore, it consists of a cylindrical Time-Projection Chamber (TPC), three particle identification arrays of Time-Of-Flight (TOF), Ring Imaging Cherenkov (HMPID) and Transistion Radiation (TRD) detectors. At the outer shells of the central barrel are two electromagnetic calorimeters (PHOS and EMCal). All these detectors with exception of the HMPID, PHOS and EMCal cover the full ϕ range of the central barrel. An overview of the placement of these detectors around the beam-pipe is shown in Table 4.1. This table is based on the placement table 1.1 in [4].

Detector	Position (m)	Detection area (m^2)
ITS layer 1,2 (SPD)	0.039, 0.076	0.21
ITS layer 3,4 (SDD)	0.150, 0.239	1.31
ITS layer 5,6 (SPD)	0.380, 0.430	5.0
TPC	0.848, 2.466	32.5
TRD	2.90, 3.68	716
TOF	3.78	141
HMPID	5.0	11
PHOS	4.6	8.6
EMCal	4.36	44
ACORDE	8.5	43

Table 2.1: An overview of all the detectors. The position is the distance from the interaction point to the face of the detector. Since the ITS, TPC and TRD are cylindrical detecters around the beam-pipe, the position corresponds to their inner and outer radius. The dimensions corresponds to the total area of the detector.

This thesis focuses on the silicon pixel detectors in the ITS. Therefore, the ITS will be the subject of the next section.

2.2.1 ITS

The ITS is the detector closest to the beam-pipe. Along with the TPC and the TRD, it belongs to the three tracking detectors [4]. The main tasks of the ITS are to determine the primary vertex and to reconstruct the secondary vertex of heavy flavour and strange particle decays, like hyperons, D and B mesons [3]. Furthermore, the ITS has the specific functions of identifying and tracking low momentum particles, to improve the impact parameter and momentum resolution and to reconstruct the particles that are moving through the dead regions of the TPC.

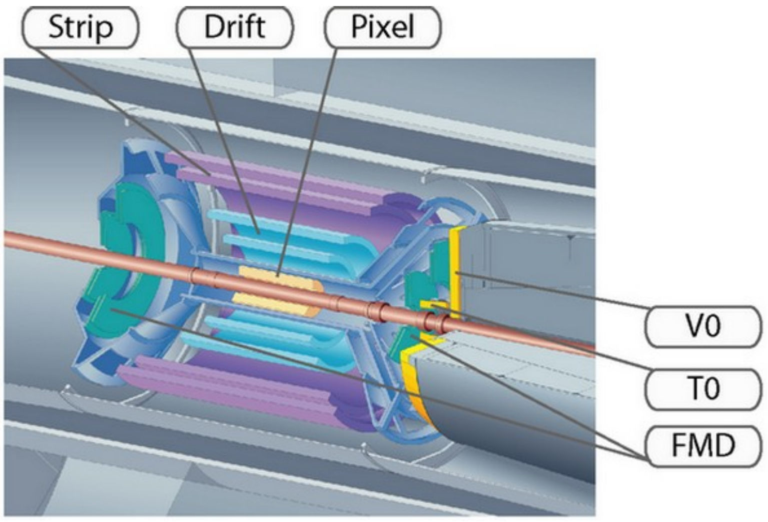


Figure 2.4: A schematic section of the inner tracking system of the ALICE detector. It consists of different concentric detectors around the beam-pipe. [7]

The ITS is build up of six layers of silicon detectors.[4] The four innermost layers have a 2-dimensional resolution because of the very high particle density in that region. The outer layers are just strip detectors (SSD), which cannot determine the vertex of the particle very well but which have a fast readout time. The four outer layers have a analog measurement system, which can determine the energyloss of the particle. By the amount of energy loss over a certain distance, the ITS can determine which kind of particle it deals with. The ITS also provides a particle identifier for low-p_t particles.

Silicon Pixel Detectors The silicon pixel detector (SPD) is part of the two inner shells of the ITS. The SPD's play a big role in determining the primary vertex and the impact parameter resolution, because they are closest to the beam-pipe and thus have the greatest chance to measure decays of the heavy D and B mesons. The SPD consists of silicon pixels which are read out in a binary form. The analog signal coming from the pixels is compared to a certain threshold level. When the output signal exceeds the threshold the readout measures a high signal, otherwise the pixel exerts a low signal. Furthermore, the pixels are arranged in a 2-dimensional matrix. The dimensions of the matrix and the width and length of the pixels changes with respect to the distance to the point of collision. The basic pixel matrix includes 256x160 cells with a width of 50 μm and a height of 425μ [4]. Longer cells are used near the boundary regions of the ITS.

Each SPD is attached to a triangular frame (a half-stave). This frame supplies the cooling system and holds datacables coming from each SPD. A basic half-stave consists of two ladders of 5 chips. The half-staves with the SPD's are placed around the beam-pipe in carbon-fiber sectors. To cover the whole z range of the beam-pipe, two half-staves are connected front-to-tail to form a stave. Each sector supports six staves and there are ten sectors placed around the beam-pipe. A quick calculation gives us a total of 10 (sectors) x 6 (staves) x (20 chips) = 1200 chips. This gives us a total amount of $9.8 * 10^6$ cells.

2.2.2 Other detectors

Time-Projection Chamber (TPC) Besides the ITS there are more detectors responsible for tracking and identifying particles. The closest detector to the point of collision after the ITS is the Time-Projection Chamber (TPC). The TPC is used for robust tracking and identifying relativistic particles with momenta up to 50 GeV/c. The inner radius of the TPC is determined by the maximum acceptable hit density. The outer radius of 2.5m has been chosen so that the length scale for dE/dx for relativistic particles is around the 5-7% [4]. With this resolution, the TPC can be used to identify particles.

Time-Of-Flight (TOF) The main contribution of the other detectors is identifying particles over the whole energy range from non-relativistic to highly relativistic particles. TOF for example is optimized to detect particles with a large range of momenta. It covers an area of 140 m^2 and features 160000 cells at a distance of 4 m from the point of collision [4].

Transition Radiation (TRD) The Transition Radiation Detector (TRD) identifies electrons with a momenta greater than 1 GeV/c with the use of six layers of Xe/CO_2-filled wire chambers. The electromagnetic calorimeter detect photons and neutral mesons created by thermal emission of the quark-gluon plasma.

Chapter 3

ALICE upgrade

Before ALICE was up and running in 2008, there were experimental suggestions for an almost perfect liquid form of a quark-gluon plasma (QGP) [2]. ALICE has confirmed these assumptions and showed that a quark-gluon plasma is created at extreme values of temperature and pressure. To investigate this QGP state of matter, we need high statistics measurements as these will unveil the very rare physics phenomenons needed to understand this phase of QCD.

ALICE had proved herself useful for these probe measurements, but there needed to be an optimization of the hardware to achieve more data with a higher resolution. The ultimate goal is achieving an interaction rate 100 times the current rate. [3]

The ALICE upgrade is divided in two main goals. The first goal is the improving of its physics performance, the second is preparing the detector's for an increase in luminosity. The luminosity will be increased to L = 6 x 10^{27} cm^{-2}s^{-1}. [2] This will lead to a Pb-Pb interaction rate of around 50 kHz. This high interaction rate demands a high readout rate of the detectors. Furthermore, the upgraded detector should have improved vertexing and tracking capabilities at low p_T. The whole upgrade consists of several sub-detector upgrades [2]:

- The inner layer of the ITS is placed closer to the interaction point. This is achieved by a reduction of the beam-pipe from 29.8mm to 19.8mm.
- New silicon pixel detectors with a higher granularity and smaller material budget.
- Wire chambers of the TPC will be replaced by GEM detectors. There will also be a readout upgrade so it is possible to continuously readout the TPC.
- Upgrade of the forward trigger detectors and the Zero Degree Calorimeter.
- Upgrade of the readout process and hardware of the TRD, TOF and PHOS for a higher output rate.
- Upgrade of online and offline systems with respect to the increase of data transmission. With the increase of interaction rate comes an increase of data volume.

3.1 Upgrade objectives

One of the main goals of the new ITS are the heavy-flavour measurements. There are two questions concerning the heavy-flavour interactions with the QGP that need to be answered. The theory behind these two questions is given in references [11] and [9].

- Thermalisation and hadronisation of heavy quarks in the QGP. There are several ways we can measure these characteristics. By measuring the heavy-flavour baryon/meson ratio, by measuring the strange/non-strange ratio for charm particles and studying the elliptic flow for charm and beauty mesons.

- Heavy-quark in-medium energy loss and how the energy loss depends with its mass. This can be studied by measuring the nuclear ratio D and B mesons and by looking at the heavy-flavour production of particles produced by jets.

There is a wide variety of new measurements that will become possible with the ITS upgrade. In general, we can measure the same probes as with the current ITS, but we can measure them with smaller energies.

3.2 Current limitations

As we explained in the previous chapter, the current ITS consists of six concentric layers of silicon detectors. The inner and outer radius of the ITS are respectively 39mm and 430mm. These radii are constraint at one side to the minimum radius of the beam-pipe and at the other side to the optimal distance for the TPC to work. The TPC must be far enough from the beam pipe to obtain a good measurement of the particle, but must be close enough to ensure a good length scale between the inner and outer radius of the TPC. These values were chosen within the boundaries of technological limitations in the days ALICE was built [1].

The ITS was built with the challenge of minimum material budget in mind. Every time a particle passes a chip in the ITS, its energy loss is significantly. To obtain a good measurement of the particle's track and energy, the particle shouldn't be obstructed by materials. This obviously cannot be done, but we can make the material budget as small as possible. The current material budget of the ITS is the lowest value of all the current LHC experiments [1]. With the upgrade we want to reduce this material budget even more.

One of the measurements ALICE can perform today is the determination of the track distance of closest approach. To obtain this determination, ALICE studies the decay of charm mesons in rare decay channels. These measurements can be performed quite good for energies above 1 GeV/c, below that the statistical errors rise above a certain amount that the data becomes meaningless. So we need to obtain better measurements of particles below the 1 GeV/c.

One of the objectives is the study on charm and beauty mesons. As we said before, we can measure charm mesons, but only for high energies. With the current detector, we can't even measure beauty mesons. The decay length of these particles is smaller than the current resolution of the impact parameter. ALICE can therefore not measure these particles in central Pb-Pb collisions because we just don't know what happens at length scales smaller than the impact parameter resolution. We can increase the precision of the impact parameter by reducing the material budget and therefore be more precise about the energy of a particle.

One of the last but crucial limitations of the detector is its read-out rate. It can operate at a maximum rate of 1 kHz. The full interaction rate of Pb-Pb collisions is about 8 kHz. So a large part of the obtainable data is not been used at all. This makes it hard to study decay's of heavy-flavour particles, because their occurrence is very rare.

3.3 ITS Upgrade

We will now discuss the upgrades regarding the ITS. The main goal of the ITS upgrade is the improvement of vertex as well as secondary decay vertices reconstruction. The key features of the ITS upgrade [2] we saw earlier in more detail are:

- **Inner layers closer to interaction point** As we already said, the inner layers of the ITS are placed 17mm closer to the beam-line because the beam-pipe radius is going to be reduced by 9.8mm and the layers are going to be placed closer to the beam-pipe.

- **Reduction of material budget** In the current ITS, the material budget prevents the achievement of higher precision on the impact parameter. Certainly at low p_T it affects the impact parameter resolution strongly, because it is mainly determined by Coulomb scattering in low p_T range. A reduction of material budget will be achieved by a factor of seven by the use of thinner silicon pixel chips ($50\mu m$ instead of the current $350\mu m$).

- **Extra layer and granularity** The ITS will get an extra seventh layer. This will result in a detector containing seven concentric layers in a range of 22mm to 430mm. Besides an extra layer, all the silicon detectors will be replaced by ALPIDE's (silicon pixel detectors). The inner three layers will get SPD's with a cell size of around 30μm x 30μm and the outer four around 50μm x 50μm.

- **Binary readout** The capability of the energy loss detection for the ITS is going to be removed. Studies show that a binary readout of the SPD's and therefore a higher readout speed outweighs a capability of energy identification of particles.

- **Readout speed** Another goal of the ITS upgrade is achieving a higher readout speed. The Pb-Pb interaction rate is increased to 100kHz and to 400kHz in p-p collisions. Indeed, this is twice the amount stated by the objectives.

This thesis will be focusing on the ALPIDE chip. This is the new silicon pixel detector, which replaces the old SPD, SDD and SSD. The ALPIDE chips will be placed into the ITS in a different way the current silicon detectors are placed. For the inner layers, 9 chips together form a half-stave but for the outer layers 14 chips arranged in two rows of 7 form a half-stave [1]. Two half-staves placed next to each other in the z direction form a stave. Finally, a total number of 24.120 chips are used. This results in a 1.26 x 10^9 pixels. This is a factor 100 higher than the current amount of pixels in the ITS.

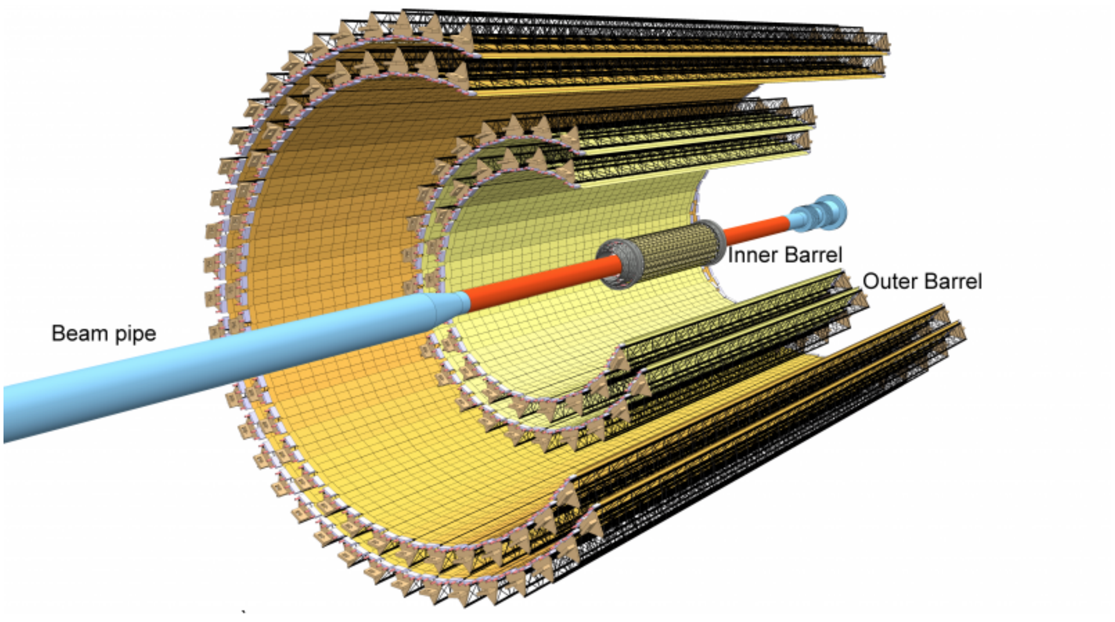


Figure 3.1: The new ITS will feature seven concentric layers of silicon pixel detectors. The granularity will be emphasized with a factor 100. [1]

Developing a new silicon pixel detector was hard because of the very demanding requirements for these detectors. They need a high granularity and read-out speed, a low power consumption and need to be very radiation resistant. Silicon based pixel detectors can be used to match most of the requirements stated. Still there is one disadvantage of this technology and that's the read-out system. A SPD is typically made out of two components. A detection system and a read-out system. The communication between these two creates problems in terms of matching two events together with respect to a certain time frame. When a particle is detected at the detection component, it is read out at the readout component with a time delay. A particle detected at a different pixel is being read-out with a different time delay. So it is very hard to make time-based matches of two particles . To overcome these problems, there was a need for new technology. One which could merge these two systems together.

Such a technology is being developed nowadays and it is called Monolithic Active Pixels Sensors (MAPS). The ALPIDE chip implements this new technology and has a low power consumption and allows fast read-out. The next chapter will describe the ALPIDE chip in detail.

Chapter 4

ALPIDE chip

The ALPIDE (ALICE Pixel Detector) is the pixel chip developed by four institutes. CCNU (China), CERN, INFN (Italy) and Yonsei (South Korea). It implements the MAPS technology and also features and In-Pixel discriminator. This means that a hit is stored binary inside the pixel. With the use of digitization within the pixel there is no need for an analogue driver. This reduces the power consumption significantly.

4.1 Architecture

The chip measures 15.3mm (height) by 30mm (width) [12]. Its pixel matrix consists of 512 x 1024 cells. These cells are arranged in double columns and thereby each double column has 1024 pixels and there are a total of 512 double columns on the chip. Each pixel has a low power discriminator, which compares the particle's charge deposit to a threshold. When it exceeds the threshold, the hit is stored inside the memory. The 1-bit memory only registers a 1 when hit and a 0 otherwise. When a trigger is received, the memory's of each pixel are being readout by a Priority Encoder. This priority encoder logs the pixel's address and sends it to the periphery only if the memory is set to 1. This zero-suppress system ensures very fast readout times because the hit patterns received by the chip are very sparse.

The pixel matrix is divided into four sub matrices, each with a different pixel technology. A layout can be seen in figure 4.1. The four different pixels (S1, S2, S2_DR, S4) differ by the sensitive area size and have a different way of resetting. All the measurements on the chip will be divided in these four matrices to maintain a consistent measurement over the same kind of pixels.

Furthermore, the pixel matrix consists of 32 regions each containing 16 double columns. The regions each have a 16 priority encoders, one for each double column. A schematic overview of a priority encoder can be seen in figure 4.2. When a readout trigger is received, the priority encoders from each region are read out one by one. Each region is read out consecutively, but all 32 regions are being read out in parallel. This means that the matrix reads out columns 1, 16, 32, 48 - 496 at first and than reads out columns 2, 17, 33, 49 - 497 etc. Each region has its own storage for the output data. The data from each region memory is then combined and send off the chip by a top level Readout Chip. The readout time (time to transfer the information from the storage elements inside the pixels to the region storage components) for a central Pb-Pb collision is in the order of 100 ns [1]. This means that a theoretical interaction rate of 10MHz can be achieved. However, this is the read out time when no hits have been received. More hits cause a longer read out time.

A top level control block controls the chip's internal registers and provides a interface for easy accessing these registers. This control block also allows the user to read out the region storage's manually.

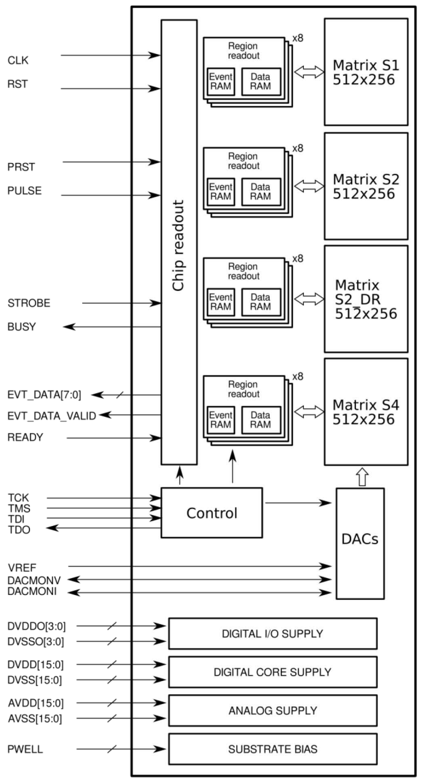


Figure 4.1: A schematic layout of the ALPIDE chip. The pixel matrix has been split up into four different matrices. Each features a different kind of pixel. Every matrix has 8 regions and thereby also 8 readout modules. [12]

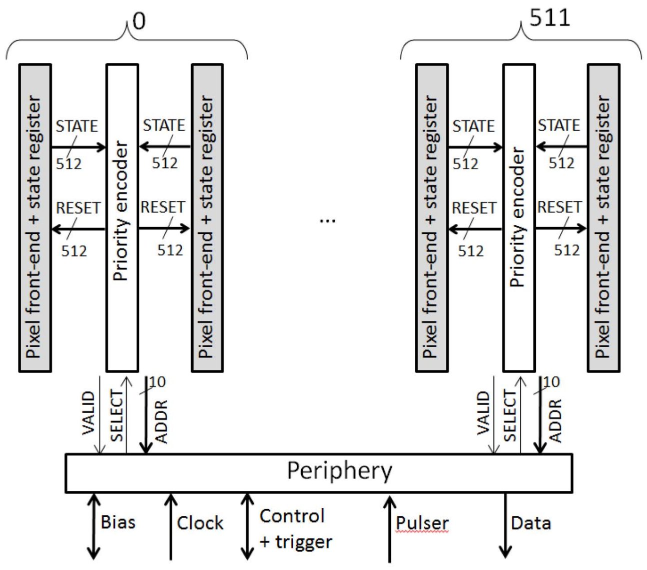


Figure 4.2: The pixel matrix has been build up of 512 double columns. Each double column features one priority encoder which runs between two columns of pixels. When a read out is performed, the priority encoder only stores the address of the pixels where a hit has been received. [12]

4.1.1 The pixel

The selling point of the chip is the front end pixel discriminator. As we already explained, the hit is stored in the pixel's memory unlike other chips where it is stored outside the pixel matrix. The ALPIDE also provides various methods to test this in-pixel discriminator. We can inject an analogue pulse into the pixel to test the working of the discriminator and determine the height of the threshold for example [12]. It is also possible to inject a digital pulse which circumvents the discriminator and directly writes a 0 or 1 into the memory of the pixel. With this method, we can test the memory, priority encoder and region storage chain.

4.1.2 The core

By the term 'core', we mean everything except the pixel matrix. The core has the job to manage the input and output datastream coming in and out the pixel matrix. It has several internal registers [12] which control various things. Registers are memories which can store data for various reasons. The internal registers are in fact memory blocks with a certain begin address and length inside the main memory. If we would look at the main memory, we would see that all registers are placed side by side. A quick overview of the important registers:

Name	Number	Description
Periphery control register	1	Holds all the important information like chip operation mode, pixel configuration data and clustering of pixel data (this will be discussed in section 4.2 'The readout').
Region disable register	1	A region can be disabled to mask its output. This is particularly useful for testing a single region for faulty pixels.
Readout timing register	1	We will discuss the readout sequence in section 4.2 'The readout'. This register stores all time delay data needed for the readout modes.
Region storage register	32	Stores the hits in 8-bit memory blocks.
Column disable register	32	A column can be disabled to supress its data output. Each region has its own column disable register, 32 in total.
Pixel configure register	1	This register holds (temporarily) the configure data for a certain pixel. If you want to configure a pixel or a row/column of pixels, you flush this register.
DAC registers	6	The analog input for the in-pixel discriminators is being provided by the Digital to Analog converters (DAC). The digital values are stored in these registers.
Region status register	32	This stores status data about every region. Status data can contain error codes and information about disabled columns.

Table 4.1: An overview of the important registers of the ALPIDE chip. Every registers has a corresponding name and number. The number is the amount of registers that exist for this kind. For example, the region storage register stores the hit data coming from a region. Every region has such a register and therefore the number 32.

4.2 The readout

The ALPIDE chip has a different readout mechanism [12] than most ordinary pixel chips found in your camera for example. Those pixel chips read out the pixel matrix in a way described as the rolling shutter mechanic [13]. This technique has several advantages and disadvantages. It is best to first study them in detail to explain the difference in readout technique's.

4.2.1 The rolling shutter

The rolling shutter technique is used in most ordinary photo and video camera's [13]. It is a method to capture a scene by rapidly scanning the pixel matrix either vertically or horizontally. The main difference with a global shutter is the fact that not all parts of a scene are captured at the same time. The rolling shutter takes some time to work its way through the matrix and this time delay ensures a delay across different pixels.

One of the advantages of this method is that the pixels can continue gather photons during the readout process. The disadvantage of this method is the distortion of the image in extreme conditions. When the change in the image is faster than the time taken by the rolling shutter to scan the whole matrix, two parts of the matrix contain data taken at different times.

In particle physics you want to correlate two events by time at different positions in the matrix. This is much more difficult with a rolling shutter technique.

4.2.2 The global shutter

The ALPIDE chip makes use of the global shutter technique [3]. This method captures a whole scene at the same time and then sends the data to a processing unit [14]. This is done by storing the hit data on the pixel itself. Between two consecutive readout cycles, the pixel is active and when it registers a photon it writes this hit in his memory. This is different from a rolling shutter in which a pixel is only active when it is been read out. When a read out has been triggered, the pixels shut down and a rolling shutter kind of technique is used to obtain the data of every pixel.

The advantage of this method is the consistency of the data. With a global shutter, all pixels have been active at the same time. This allows the matching of two events at two different positions in the matrix.

The main disadvantage of this technique is its production cost. Because every pixel needs an in-pixel memory, the production costs of these kind of pixels is significantly higher than the of a normal pixel. This is the reason why ordinary consumer hardware does not have this technology. One application where this technology is used is a high speed camera. This camera captures thousands of frames a second, with a rolling shutter this would be impossible to obtain because of the image distortions you will get.

4.2.3 Priority encoder

The priority encoder is one of the most important features of the ALPIDE. It makes it possible to only read out the pixels that have been hit [12]. This greatly improves the read out speed. The priority encoder works in several different stages. When the chip receives a trigger for a read out, the priority encoder starts questioning every pixel if it is been hit or not. If the answer is yes, the priority encoder saves its address in a priority order. The first pixel of the column has the highest priority and the last pixel the lowest. This gives a second advantage for data handling. When an address of a pixel 2/3 in the column has been received we can be sure the first 2/3 of the column did not receive a hit. This makes it reliable and fast.

4.2.4 Readout modes

As we mentioned before, the ALPIDE uses a global shuttering technique. The ALPIDE has two ways of reading out the pixel matrix [12]. They are simply called *Readout A* and *Readout B*.

Readout A captures the discriminator output the whole time until a readout trigger is received. After the trigger, the memory is flushed and stored in the region storages. When the memory is flushed, the pixel becomes active again. This method ensures an almost continues capturing of events.

Readout B only stores the discriminator output when a readout trigger has been received. The duration of this capturing is determined by a pre defined variable. The 'Readout Timing Register' holds the value for this time delay. When the time delay has passed, the readout begins and the pixels become non-active again.

4.3 Testing

Manufacturing a new kind of chip is very difficult and consists of many steps. The ALPIDE chip is based on 180nm technology which is not new to the CPU market but yet is not simple to produce. This is one of the difficulties involved in making the ALPIDE. Others [1] are:

- **Thinning of chip** To reduce material budget of the inner layers of the ITS, the chips are thinned down from $350\mu m$ to $50\mu m$. This process consists of grinding the silicon plate from the bottom down to the desired $50\mu m$. This grinding must be carefully performed and has a high error rate.

- **High demands** Only one out of 10^5 pixels is allowed to be broken or faulty. A maximum of five pixels per chip is allowed to register a false hit. This is a very high demand for a pixel which features in-pixel logic and memory inside an area of $28\mu m$ x $28\mu m$. Certainly not every chip coming from the production line can match this demand.

- **High implementation costs** During the large shutdown in 2018/2019, ALICE is taken apart and rebuild again. The deconstruction of ALICE takes about 3 weeks just as the rebuilding of ALICE. You want to make sure that a large fraction of the chips in the ITS are working at least till the next shutdown.

- **Shipping** The production of the thousands of ALPIDE chips takes place in Israel. This doesn't take a lot of time. The assembly of the modules takes however, is slow and need to be executed with great precision. The modules are therefore developed in parallel at institutes around the world. After their production in Israel, they need to be shipped to these institues. Some modules are created in countries as far as South-Korea. When the chips are eventually shipped to CERN, the possibility for cracks and other disabilities becomes very high.

With these risks in mind, there is a need for a method to test the chip for its capabilities. The ALPIDE chip will be tested at various times throughout the process to precisely determine the nature and cause of each fault. The benchmark-like tests performed will determine if the chip is working or not and will examine the nature of each fault. The tests will be performed by testing software and hardware.

Chapter 5

Testing software

The main job of the ALPIDE testing software is the performing of tests and scans to identify errors in the manufacturing process of the ALPIDE. As explained in the previous chapter, the process of designing and creating the chip involves many carefully performed steps which leads to a high error level. The chips need to be tested for faults at various moments in the process to ensure that the chips which are eventually placed inside ALICE work. In this section, we will discuss various aspects of the software.

First we will briefly go over the measuring set and hardware used in the development. After the hardware, the graphical interface and the scans and tests performed by the software will be discussed. Which tests and the motivation for this tests will be explained. Finally, the inner workings and some parts of the code are explained The structure of the program and its flexibility for expanding is contained in the last section of this chapter.

5.1 Hardware

The hardware used by the software consists of three components. These components are connected via cables and each has a different function in the process of testing. We will briefly describe each one.

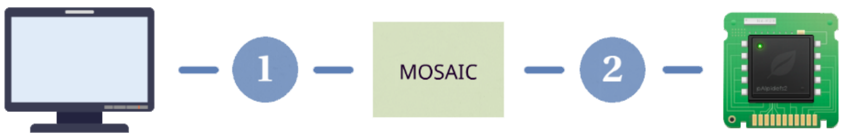


Figure 5.1: A schematic overview of the measuring set used by the software. The computer is connected to the MOSAIC board by an ethernet gigabit cable (1). The MOSAIC board on his turn is connected to the chip with a custom made cable (2).

5.1.1 The computer

First of all, we have the obvious one, the computer. The system is scientific-linux based. Scientific Linux [17] is developed by CERN and Fermilab. It is a version of Red Hat Enterprise Linux which is enhanced to fit laboratory use . The data generated by the measurements is stored onto the RAM during the measurements and is written to the HDD after its completion. Furthermore, the desktop has normal system specifications. The software needs to run on many different systems when it is completed, therefore the software has been written on and for low end desktops.

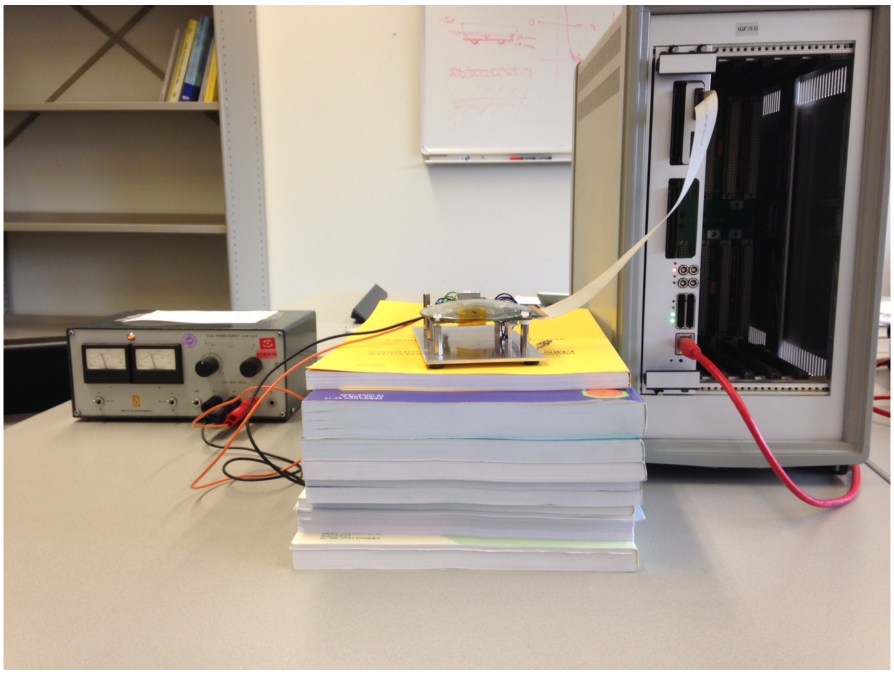


Figure 5.2: A photo of the MOSAIC board and the ALPIDE chip. The two are connected with the white wire which is custom made for this setup. The red Ethernet cable leading to the bottom of the MOSAIC board is the connection to the computer. It allows gigabit transfer from the board to the pc. The ALPIDE chip is powered by the power supply on the left. It supplies a steady voltage of 5.0V to a power regulator which transforms it to the operation voltage of the ALPIDE of 1.8 V.

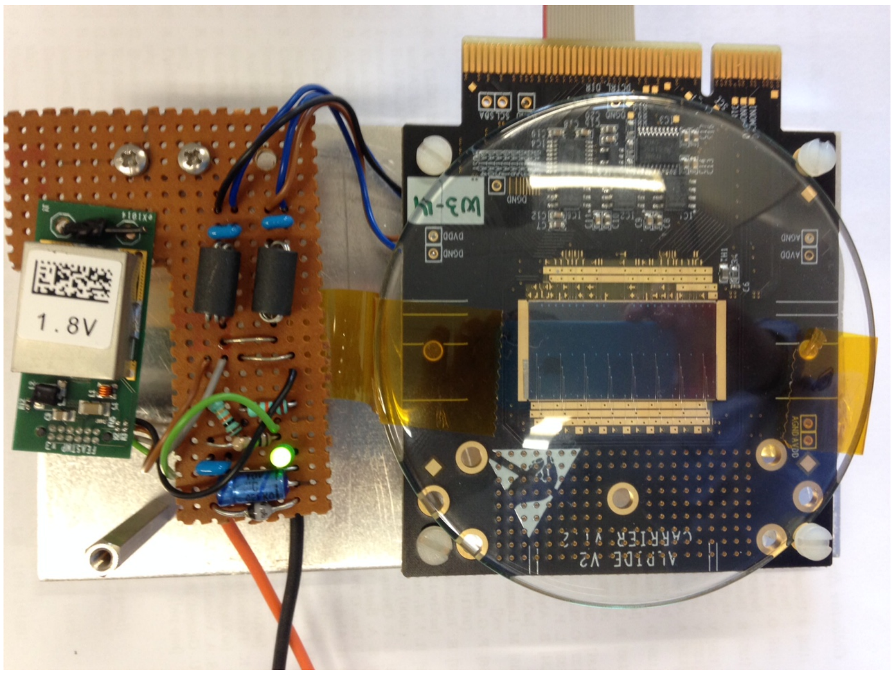


Figure 5.3: A top down view of the ALPIDE chip mounted on its carrier board and on the left its power unit. The glass shield on top of the chip protects the sensitive pixel matrix. The white wire coming from the top is the connection cable to the MOSAIC board. The cable is connected to pins located on the bottom of the carrier board.

The computer is connect to the second component, the MOSAIC board, by a gigabit Ethernet cable. The system communicates with the MOSAIC board by the TCP protocol.

5.1.2 The MOSAIC board

MOSAIC is an acronym for Modular System for Acquisition Infrastructure and Control [15]. It is the interpreter used by the computer to communicate with the ALPIDE. It contains a trigger and pulse system, which delivers basic commands to the ALPIDE. In short, the MOSAIC board conducts the measurements on the chip commanded by the computer. A memory of 1 Gb DDR3 RAM is used to buffer results of measurements on the board itself instead of streaming the data through the IP-bus to the computer. The board sends a 'is-finished' response when it is finished with the queue of commands. The computer in his turn can send another commands for a new test in response. When all tests are done, the results in the board's RAM can be accessed by the computer.

A schematic overview of the components of the board is given below. All the components communicate with each other through the wishbone protocol. This protocol features a communication technique which sends low level commands between components which are synchronized by a clock and consists of a high (1) or low (0) level signal. It is open source and widely used in many integrated circuit systems.

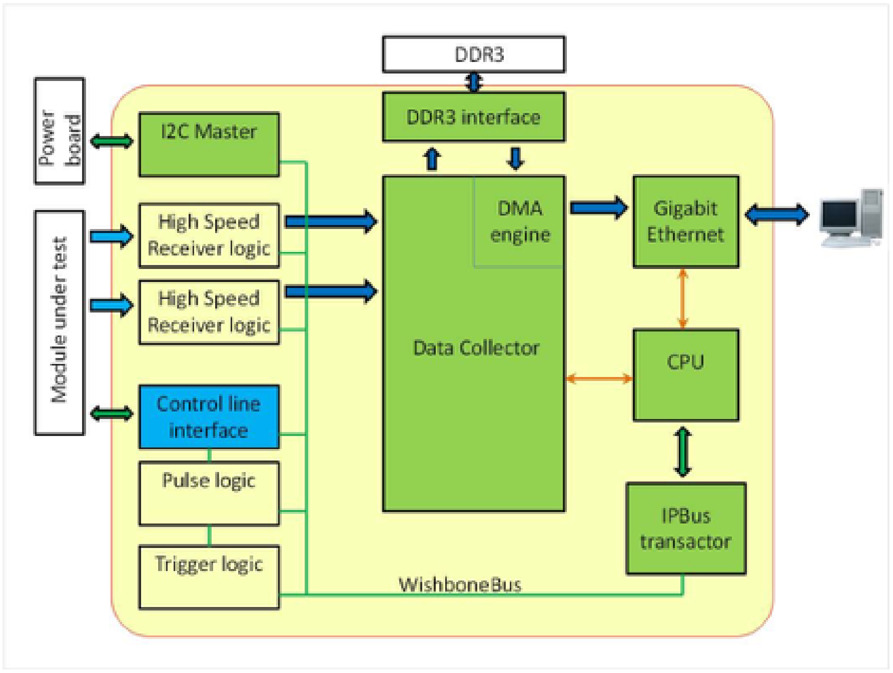


Figure 5.4: A schematic overview of all components of the MOSAIC board. This board is used as an interpreter between the ALPIDE and the computer. All components communicate by the wishbone protocol. The board is connected to the computer through the gigabit Ethernet interface and the ALPIDE chip is connected to the high speed receiver interface. The MOSAIC board generates pulses and triggers based on commands from the computer. [15]

The central component of the board is the data collector. This data collector handles all data coming from the high speed receivers. The data is then stored in the DDR3 RAM where the computer can directly access it through the DMA engine.

Furthermore, the computer is connected to a gigabit ethernet interface which on his turn is connected to the IP-bus transactor through the CPU. With the existence of an IP-bus transactor, it is possible to conduct measurements from an external device as we mentioned in the beginning, in this case the desktop computer. This can be done via the following technique. The pulse and trigger logic can be accessed with the Wishbone protocol [16]. These two components are combined in a single pulser system. This pulser can operate in three different modes. Either in Only Pulse, Only Trigger or External Trigger mode. The commands needed for this pulser to operate are send to the wishbonebus. This technique allows an external device to communicate with the Device Under Test (DUT) by the use of triggers and pulses.

5.1.3 The ALPIDE chip

We did not explained the hardware surrounding the chip which supplies the power and communication interface. The test version of the ALPIDE has been integrated into a carrier board for several reasons. The in and output pins of the ALPIDE have an area which only measures $200\mu m$ x $200\mu m$ [12], and it would be nearly impossible to solder a wire to these pins. Besides that remember the thickness of $50\mu m$. This makes the chip very fragile. In summery, the carrier board features interface logic, makes the chip less fragile and has a wire connector to connect the chip to an interface. The final version of the ALPIDE would not be integrated into a carrier board. Instead, the ALPIDE's will be arranged next to each other onto half-staves and then placed around the beam-pipe, as we said in the chapter about the upgrade of the ITS.

The power is delivered by a power supply. The power supply gives a voltage of about 5V for the ALPIDE to operate. The operating voltage of the ALPIDE is only 1.8V however [12]. For the translation between these two voltages there has been a power converter placed right next to the carrier board. This can be seen in figure 5.3. This power converter converts the 5V to a steady 1.8V.

5.2 Scans performed

The software is designed to perform tests and scans on the ALPIDE, which will determine the correctness of the chip. Three tests have been implemented in total, a register test, threshold and noise test. One can think of many more and we will discuss these idea's and suggestions in the discussion section of this document. In this section, we will go over each test and unveil their use and implementation.

5.2.1 Register test

A chip contains many registers. A register is a small block of memory, which has one particular task. In most cases, registers are used to save variables and data. The ALPIDE has around 100 registers as we discussed in section 4.1.2. A first and simple test is checking the registers if they are storing data the right way. Registers are the core of a chip, if they do not work properly a whole chip won't work properly.

A register test is pretty simple and straightforward. It consists of a writing and reading action on a certain value in a register. If the register is available for write and read actions, we write a couple of bits to this register and wait a while. After a while we read the register and see if the same couple of bits are coming out as we wrote before. If this is the case then we can perform this action for other bit patterns and see if they also give the desired result. If this is not the case then there is something wrong with the register and the register test fails.

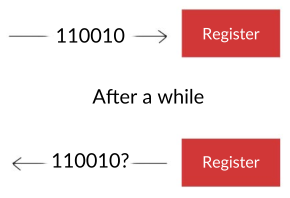


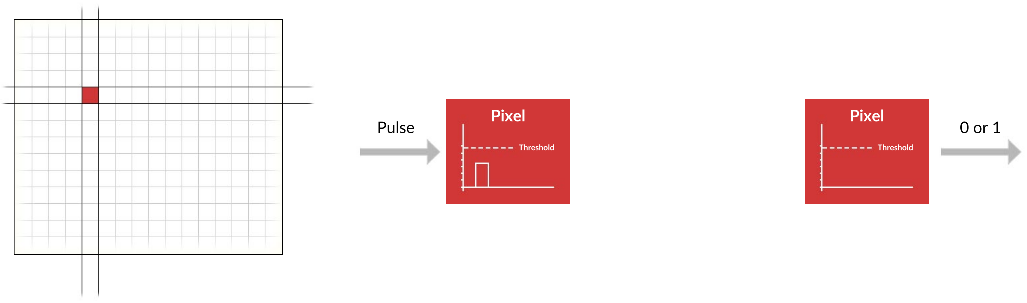
Figure 5.5: A simple explanation of a register test. First we will write a certain bit pattern into the register. After a couple of microseconds we read a value back from the register and match it to the written value.

One of the advantages of this test is its speed. It is a matter of microseconds to perform it. That in combination with the fact that registers strongly determine the working of the chip makes it a good candidate for a first test. It can be performed fast and clean.

5.2.2 Threshold test

With a threshold test we test the in-pixel discriminators and their threshold but first a small recap. We know that the in-pixel discriminators are part of the front-end of a pixel. These discriminators match the incoming analog value from the particle to a threshold. If this analogue value exceeds the threshold, a latch is set (1) in the pixel's front-end, otherwise this latch is left untouched (0).

The height of the threshold is determined by an analog input voltage. A front-end of a pixel is very small and manufacturing uncertainties in wire thickness for example can change the value of this input and thus the threshold a tiny bit. Because we are on the scale of electron charge, these changes can have a large effect. One of the demands for the pixel matrix is a maximum misfiring rate of 10^{-5} [1]. This means only 5 pixels per chip is allowed to record a particle when it should not record anything or record nothing when it should record something. These kind of pixels are respectively called 'stuck-at-one' pixels and 'dead' pixels. This brings the need for a test which can determine the amount of these two kinds of pixels. If the threshold of a pixel is 0 than we can assume a 'stuck-at-one' pixel. In the same way we can detect 'dead' pixels when a threshold is unrealistically high.



(a) Step 1: Select the pixel and mask the rest.

(b) Step 2: Inject the pixel with a pulse.

(c) Step 3: Read the value in the pixel's memory

Figure 5.6: A overview of the steps performed by a 'threshold test'. The outputs of the pixels are summed up to get the total amount of hits for all thousand measurements. The total amount divided by the number of measurements gives the probability a particle with the pulse's height would have been detected.

The test consists of several stages. These stages are combined into so-called 'mask stages'. In a single mask stages a certain amount of pixels is tested. When more mask stages are used, more pixels are being tested.

First of all, a selection of pixels is being made based on the current mask stage. A selection generally consists of 2096 pixels. This selection is uniformly distributed over the pixel matrix and each region has the same amount of selected pixels. This ensures a steady read out time for each region. After the whole pixel matrix has been masked, the selected pixels are being unmasked. This results in the fact that only the data of the selected pixels is read out during a read out of the matrix.

Secondly, an analog pulse is injected into the pixels. This is one of the testing methods for the front-end we mentioned in section 4.1.1. This analog pulse resembles a particle. The height of the pulse is analogue to the particle's energy. This way we can simulate a particle hit. In the test we inject pulses of different energy levels ranging from a low energy to a high energy. We expect the threshold somewhere between these two energy's.

Each time we inject a pulse, the pixel is read out a 1000 times and we keep record of the number of times the pixel outputs a 1. This measurement gives us the probability that a particle has been detected. These measurements are performed in a single batch by the MOSAIC board. The computer sends the command to trigger the chip a thousand times and the board stores the results in its RAM. When we get a number of hits for a certain pulse height, we plot this relation and get the following result for a perfectly working pixel:

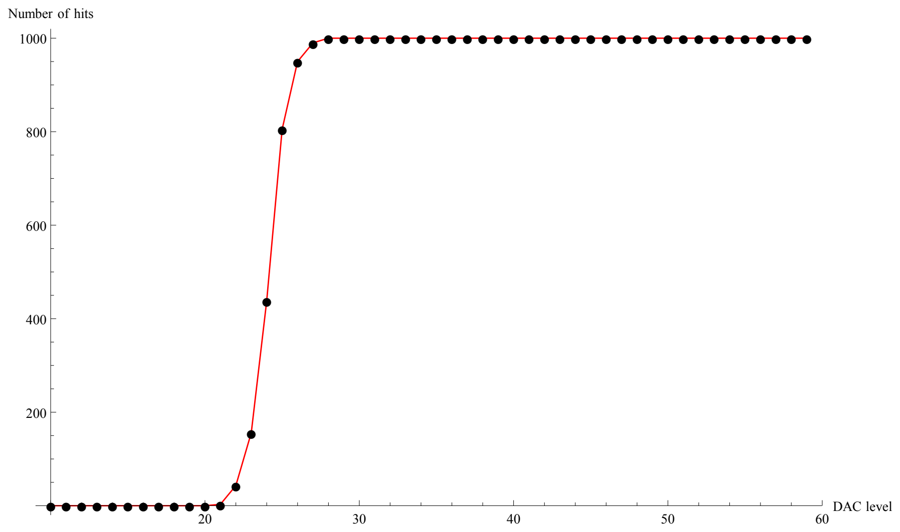


Figure 5.7: This plot resembles a perfect working front-end pixel chip. The input level of the Digital to Analog Converter (DAC) of the injected pulse is plotted against the number of hits the pixel detected out of 1000 measurements. The threshold can be seen as the DAC level where the number of hits is around 500. This means a probability for passing particle of 50%.

This plot has an error-function like shape. When we would fit an error function and differentiate this with respect to x, we would obtain a Gaussian distribution. This Gaussian distribution would give us the threshold which the DAC level of the maximum. This is further and more in depth explained in the bachelor thesis 'Testing the pALPIDE v2 chip for the upgrade of the ALICE Inner Tracking System' of Pim Verschuuren [19]. The testing software aims on giving fast and moderately accurate results. Therefore we don't use the method of fitting and integrating but we simply determine the DAC level when the probability of a hit exceeds 50%. This is the point where the number of positive outputs exceeds 500 in the case of a 1000 measurements. This would give us a high accuracy on smooth plots. However, this technique fails at plots which are very steep around the threshold energy. Nevertheless, comparison between these two technique's only showed a small difference in thresholds and does not affect the average threshold on large scale.

After we obtained the thresholds for each pixel in the selection, we sum them up for each threshold to plot a histogram for the thresholds. This gives us an overview of the threshold distribution. The width of the distribution resembles the amount of different values. Obviously, we want the distribution as small as possible. Such a histogram is shown in figure 5.8.

The histogram has been divided into four plots. Each plot represents a matrix with its own distribution. As we discussed in chapter 4 'The ALPIDE' the pixel matrix consists of four different matrices. A matrix is build up of 8 regions. Each matrix has a different kind of pixel and thus a different threshold when the input voltage is kept the same for all four matrices. Eventually, the best performing matrix will be chosen based on tests like these.

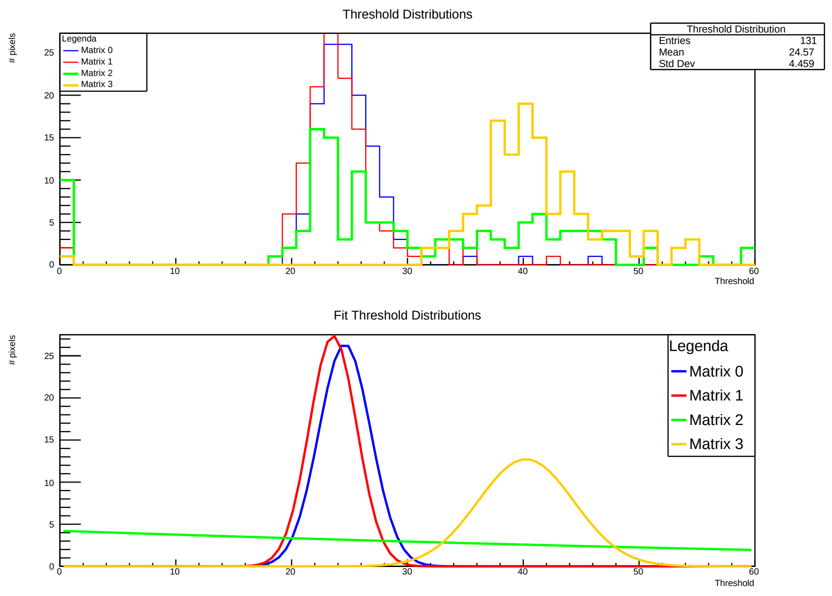


Figure 5.8: A threshold distribution can be made if the threshold per pixel has been calculated. In this histogram, we plotted four matrices because each of them has its own pixel hardware which differs from the others. [19]

Matrix 1 and 2 are the smallest and highest peaks. So those perform best based on these results. Matrix 4 is the worst with the widest and lowest distribution. This means a large difference in threshold among pixels.

5.2.3 Noise test

We can determine 'stuck-at-one' and 'dead' pixels with the threshold test but we cannot measure how accurate the in-pixel discriminator works. In an ideal situation, we want the threshold to be a steady value which is perfectly determined. In the real world however, we see a threshold which has small variations. We can see this for example in the distribution of figure 5.7. For a perfect threshold we expect to see a perfect vertical line. Right before the threshold, the number of hits would be zero and right after the threshold the number of hits would be at its maximum. Clearly we don't see these kind of results.

This is caused by the noise in the threshold and the signal. Just like the difference in thresholds among pixels is caused by the little imperfections in the front-ends of the pixels, the noise is also caused by these imperfections. So the threshold does not only differ from one pixel to another, but it also differs over time. This difference is called the noise. We can measure the noise in different ways. One method was described in the previous paragraph. This matched the amount of noise to the steepness of the threshold plot.

The mathematical technique is a bit trickier. This involves an differentiation of the fitted error function as we described in the previous section. The differentiated function gives us a Gaussian distribution. The mean of this Gaussian distribution resembles the threshold and the variation resembles the noise. If we would want an accurate calculation of the noise we should use the mathematical technique.

As we said however, the software aims on giving fast results. Just like the threshold calculation we are going to perform a quick but sloppy calculation. The noise is defined by the difference in DAC levels where the probability is 10% and 90%. Figure 5.9 illustrates the method in detail. We find the first data point exceeding the 10% mark (in this case 100 hits) and the first data point exceeding the 90% (in this case 900 hits). The difference between the DAC level of these two data points is the noise. The precise calculation of the difference is also shown in the figure.

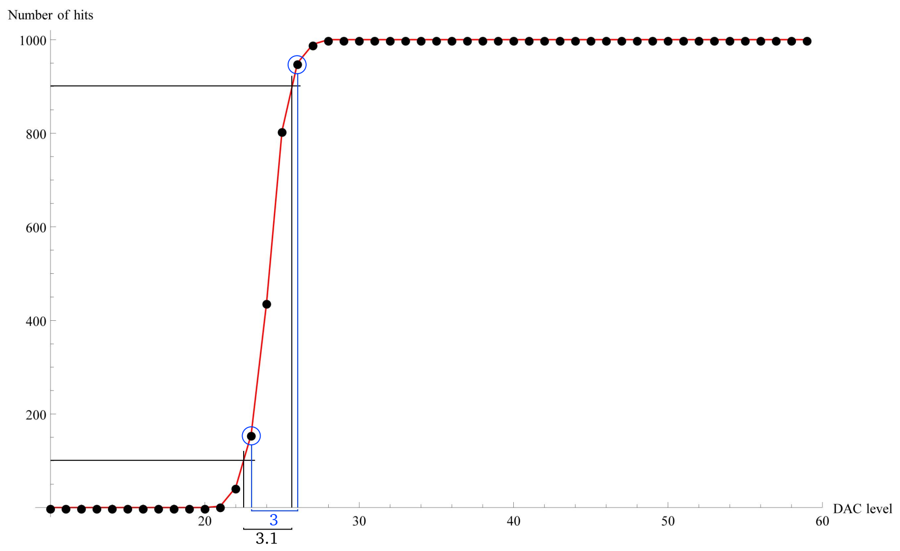


Figure 5.9: This figure illustrates the method used by the software to determine the noise easy and quick. The noise can be estimated by the difference points where the pixel registers a hit in 10% of the cases and 90% of the cases. We loop over all the data points and search which point exceeds the 10% mark and the 90% mark first. We then subtract their x-values and we obtain a measure of the noise. This figure shows the difference in an exact calculation and the quick calculation.

After we obtained the noise for every tested pixel, we can also make a histogram for the noise distribution just like we did for the threshold distribution.

5.3 Software Interface

We already discussed the hardware and the performed tests. This gives us an almost complete view of the testing software. However, we still miss a crucial component of the program, the graphical interface. A program can be as well written and can have as many features as needed but if the graphical interface is not accessible for users, it is worthless. This is also applicable to the ALPIDE testing software. The demands for the graphical interface are high:

- **Fast and targeted** The interface needed to provide fast information, which is easy to understand. The new ITS will contain 24000 chips [1] and with a fail rate of 30% in mind we need to test more than 30000 chips. This brings the need for fast information. The interface features real time test results. When a mask stage during a threshold scan is completed, a histogram is made directly or updated with new test results. This gives real time feedback during a threshold scan which can sometimes take more than 30 minutes.

- **Advanced tests results** Fast results are very useful for users that want simple data to determine if the chip is working or not. Unfortunately, this is not the only target audience. The software is also going to be used by researchers who want to know more in depth information about the chip. Region based threshold information for example.

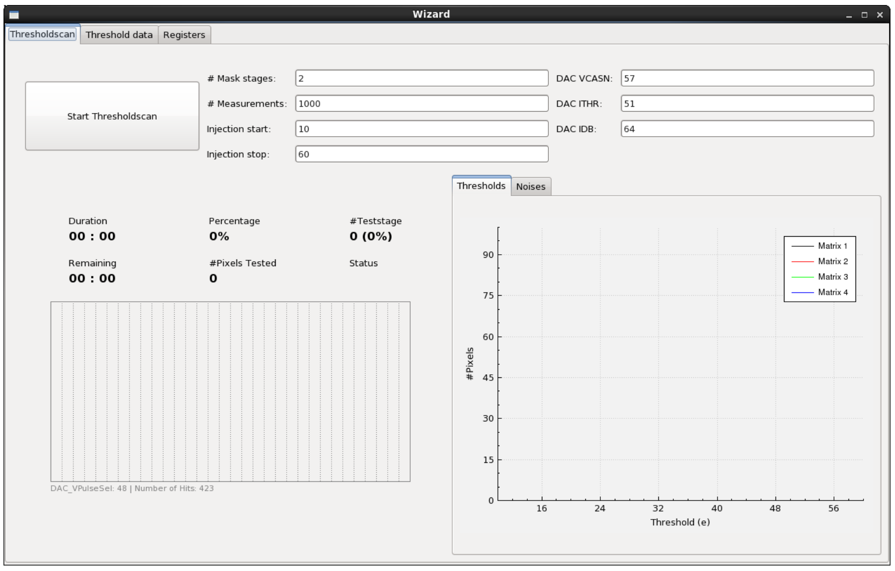
- **Full control** Users need to have full control over tests they perform. This means less hard coded variables and more user defined variables. The software features the possibility to change the testing parameters. A standard setting has been chosen as default. This allows basic testing without parameter adjustment for the inexperienced user.

- **Export** Online plotting is featured by the software but this uses rough and sloppy calculations as we discussed in the previous section. An export option was therefore demanded to allow offline data examining.

The software has been designed as a tab-based application. The application consists of three tabs as you can see in figure 5.10. These tabs can be seen as a stack of different screens. Tabs in the background will remain active. Each tab has its own function in the process of testing. We will discuss each of the tabs in the following subsections.

5.3.1 Tab 1: Thresholdscan

This tab features all parameters and shows progress data of the threshold scan. The view has been divided into three regions. A region to start the test and adjust the testing parameters, a region to get feedback about the progress and a region to quickly get an overview of the real time test results.



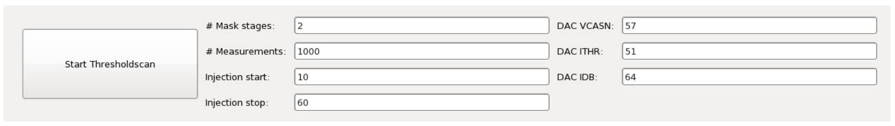
Mask stages: 2
Measurements: 1000
Injection start: 10
Injection stop: 60
DAC VCASN: 57
DAC ITHR: 51
DAC IDB: 64

Duration
00 : 00
Percentage
0%
#Teststage
0 (0%)
Remaining
00 : 00
#Pixels Tested
0
Status

DAC_VPulseSel: 48 | Number of Hits: 423

Figure 5.10: A screenshot of the thresholdscan tab. The application is tab-based and contains three tabs. A Thresholdscan, threshold data and registers tab.

Figure 5.11 gives a detailed overview of the parameter inputs and the start button. The inputs have a default value for which basic scans can be performed. The number of mask stages has a default value of two, this means the scan operates in two stages, each stage tests 2096 pixels. A scan can have a maximum of 250 mask stages. This follows from the total amount of pixels divided by the amount of pixels tested in one mask stage.



Mask stages: 2
Measurements: 1000
Injection start: 10
Injection stop: 60
DAC VCASN: 57
DAC ITHR: 51
DAC IDB: 64

Figure 5.11: A screenshot of the opening screen. The application is tab-based and contains three tabs. A Thresholdscan, threshold data and registers tab. These tabs will be discussed in the next sections.

The number of measurements determines the number of times a pixel is being read out. Default value is 1000. The injection start and injection end set the range of DAC levels to be tested. We normally expect a threshold of about 25-30 and thus the default range ranges from 10 to 60. At last, the VCASN, ITHR and IDB determine the threshold height. These parameters aren't meant to be changed.

When the threshold scan is running, region 2 gives us feedback about its progress. A threshold scan can take quite some time as we already said and therefore regular feedback about its progress is demanded. The figure to the right shows a running threshold scan performing its first mask stage. The software calculates its running time, total percentage, current mask stage with its percentage, remaining time and total amount of pixels tested. It also shows the current status which can be either 'pixel selection' or 'scanning'. The remaining time is being calculated based on its running time and total percentage. The red dots resemble the position of the pixels currently being tested. The pixels are uniformly spread over the matrix for best testing results.

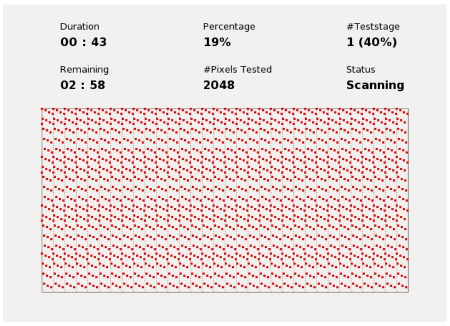

Duration 00 : 43
Percentage 19%
#Teststage 1 (40%)
Remaining 02 : 58
#Pixels Tested 2048
Status Scanning

Figure 5.12: Part of the threshold scan tab which gives feedback about the progress of the current threshold scan.

The last region gives a quick overview of the data resulting from the last tested mask stage and those before. This means the histogram is based on data coming from mask stage 1 when the scan is currently at mask stage 2 and that the histogram is based on mask stage 1 and 2 when the scan is currently measuring mask stage 3. Furthermore, the scale of the x-axis will automatically adopt to the current measuring range and the y-axis will scale with respect to the highest measured value.

The tabs at the top will give access to an overview of the noise histogram. This means that both the noise and threshold for each pixel are being calculated online during the scanning.

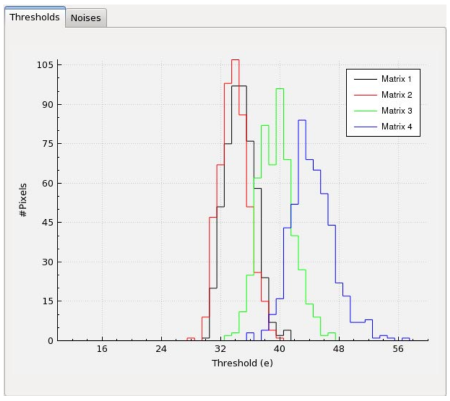
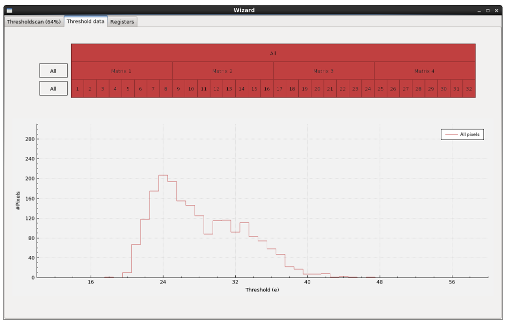


Figure 5.13: Part of the threshold scan tab which gives quick access to real time data acquired by the last performed mask stages.

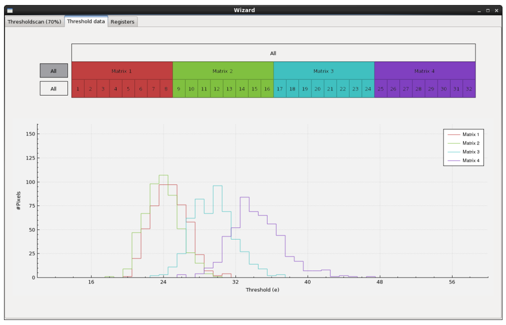
5.3.2 Tab 2: Threshold data

The threshold data tab features a system where the user can view and compare data coming from the threshold scan. Based on the selected regions or matrices in the 'data selector' above, the plot shows different histograms. We will further study the options made available by the 'data selector'.

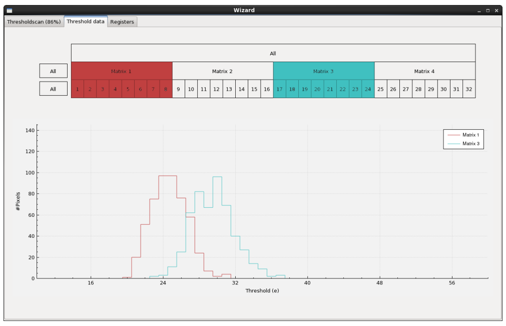
There are a couple of different plotting scales available. You can view a threshold distribution of the whole pixel matrix, per matrix and per region. In case of the last two, it is also possible to show multiple distributions at the same time.



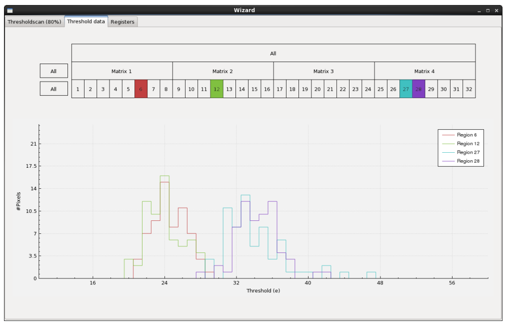
(a) A threshold distribution of the whole pixel matrix.



(b) Four distributions, each for a different matrix.



(c) Two distributions of matrix 1 and 3. Useful for comparing.



(d) Four distributions of four different regions. Useful for comparing on region level.

Figure 5.14: Four viewing possibilities in the threshold data tab. This tab is designed for in depth examining and comparing of threshold data. The feature of comparing two regions or matrices is useful for revealing anomalies in regions or matrices.

The possibility to view a distribution for the whole pixel matrix, which contains four different kind of pixels, does not seem very useful for now. However, the best performing pixel is eventually being chosen and used for the whole matrix in the next version of the ALPIDE. The testing software has also been designed for future versions of the ALPIDE and this is one of those features.

The colors of the different distributions are generated dynamically and are based on selection size. The feature to examine the distributions on different scales gives the possibility to quickly discover anomalies in particular regions.

5.3.3 Tab 3: Registers

The register tab features the results of the register test. As we explained in section 5.2.1, the register test is performed very easily and quick. Therefore this test is always performed automatically during the launch of the program.

The registers are divided into three parts based on function. A part that contains the region based registers, a part for all core registers and a part containing the 11 DAC registers. The DAC registers are actually half the width of a normal register, therefore two DAC registers are placed next to each other to form one full register.

The region based registers are designed to give a quick but superficial overview. There are several registers per region. A region storage register, column disable register and a region status register. The result of these registers are combined into a single block. If one of those registers fails at the register test, the block representing that register will be red. A neat feature that could be implemented is a detailed view of a region register that would be shown after a mouse click on a specific region. The current register view of regionregisters is not very useful when there's something wrong.

The control registers are registers controlling the overall performance of the pixel matrix in contrast to region registers which only control the performance of one region. This test version of the ALPIDE only contains four control registers. A next version has a lot more registers and an expansion of tests is therefore desirable.

The DAC registers are half the width of a normal register as we said before. We can only test full registers and therefore 6 tests are needed for 11 DAC registers.

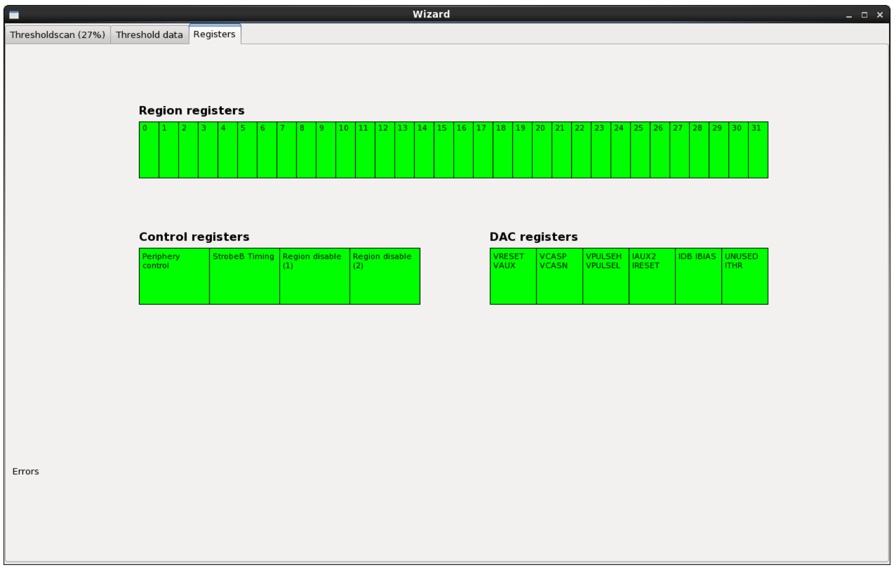


Figure 5.15: A screenshot of the register tab. It shows all tested registers. A register is shown in green if it passed the test, it turns red otherwise.

5.4 Architecture

Till this point we discussed the why and what of the software, the different kind of scans and tests and we also showed the graphical representation of results. But how is the application structured and how is the code build up? In this last section we will discuss the language and coding software choice, use of threading, the underlying class tree and library use.

5.4.1 Language

C++ version 11 in combination with the Qt 5.5 framework has been used to code the application and create the graphical interface. C++ has been used to program the back-end and Qt for the graphical interface. Qt [18] is a widely used cross-platform toolkit to create applications with a graphical user interface (GUI). Qt Creator was chosen as code editor because of the easy Qt environment. Qt Creator makes the compiling very easy without the need of stumbling around with qmake (Qt version of cmake) and cmake. The standard GCC compiler is being used to compile the application.

5.4.2 Qt 5.5

Qt 5.5 is the most recent version of Qt, a toolkit for developing application software and graphical user interfaces [18]. It is a version of standard C++ with several extensions. One of them includes signals and slots. These two take care of the handling of events. Signals can be seen as out going calls of a class and slots as incoming calls. A certain signal can be connected to a particular slot to establish a connection between them. If the class 'emits' a signal, its connected slot function will be executed. This simplifies communication between classes which is a very important part of application development.

Other extensions include 2 dimensional canvas drawing and many non-GUI functions like database implementations. Qt provides a very useful framework for application development.

5.4.3 Classes

The GUI consists of many widgets which are placed inside so called layouts. A layout is also a particular kind of widget. In summary, the GUI can be represented as a tree of widgets. The root of this tree is a mainView widget. This resembles the actual window of the application. This window contains a tabView manager which provide interface for the three tabs discussed in the sections above. Each tab contains a layout widget which in their turn contains the widgets representing data, buttons, graphs or input forms. A total view of the tree is shown below. The widgets and classes are further explained in the code documentation appendix.

```
WizardMain
    TestWindow(tabWidget)
        ThresholdScan (tab)
            CanvasThreshold
            ThresholdPlotRealtime
            QPushButton
            QLineEdit
        ThresholdDetail (tab)
            ThresholdSelector
            ThresholdPlotDetail
        RegistersTest (tab)
            CanvasRegister
```

Besides the widget classes there are several back-end classes which perform the basic functions of the application. The 'chiptest' class for example performs all tests and scans and handles the resulting data. The visual feedback of the test and scans are being handled by other classes. It outsources a lot of simple visualization work and only concentrates on testing. Another back-end class is 'testdata', which stores global variables.

5.4.4 Threading

Before we discuss threading with respect to our application. It is useful to learn a bit about threads. A thread is a sequence of programmed instructions that runs on its own. Multiple threads can run at once if the CPU supports multiple cores. Every core accounts for a thread and therefore multiple actions can be performed at the same time. Threads communicate with each other by the use of queue's. A queue of a particular thread can be seen as some sort of inbox which stores all incoming messages from other threads. When a thread has 'free time' it processes the messages inside the queue.

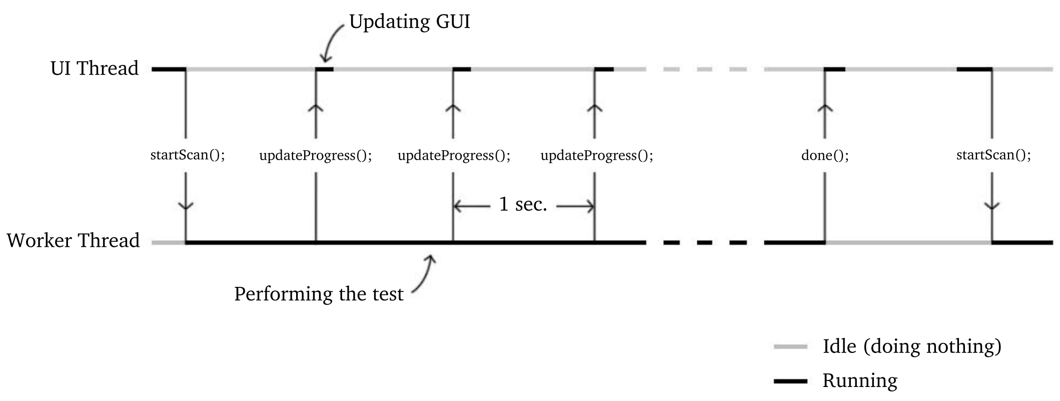


Figure 5.16: The application runs two threads simultaniously all the time. An UI thread and a worker thread. The worker thread performs all the test and the UI thread gives the user graphical feedback. The two threads communicate by the use of signals and slots. The worker thread sends updates every second to keep the UI thread provide with progress data.

The testing software runs two threads at all times. An UI thread and a worker thread. The UI thread provides the GUI and the worker thread performs all tests and scans. The reason for this choice of structure is the following. The tests performed by the application are thread-blocking. This means that during a test, no other processes can run on that thread. If you would run tests on the UI thread, the whole application would seem to freeze because no graphical updates would be processed. This gives a very bad user experience. User experience is the way a user feels about the software. If the software is slow and doesn't respond, it will irritates the user or even make him to quit the program. The software needs to respond quick and give the user the feeling he is in control. Therefore the application runs a specific thread, which is only intended for scanning and testing.

Figure 5.16 illustrates the use of threading in a graphical form. The chiptest class implements a core threading class of Qt. The chiptest thread is basically a while loop which runs forever. If a test is being performed, the while loop stops because the test is thread-blocking. If a test is finished the while loop continues running on forever till another test is started. A few things happen inside this while loop. The class has a status variable which holds the current status of the thread. This status is a simple enum which contains the following values. IDLE, STARTTHRESHOLD and THRESHOLD_SCANNING. When the thread is in IDLE mode it just runs the status check over and over. If a STARTTHRESHOLD status has been set, the threshold scan is being launched and the status is being changed to THRESHOLD_SCANNING. When the test is completed, the status is turned back into IDLE and the story repeats itself.

5.4.5 Library

The low level interface for communicating to the ALPIDE via the MOSAIC board has been written by the INFN institute in Italy. This is a framework containing 26 classes which are distributed over 50 files. INFN is still working on new versions of this framework for newer versions of the ALPIDE. One of the demands for the software was its flexibility. The code should be extended with ease.

Therefore, the framework written by the INFN group has been compiled into a single library (.a) file. This file can easily be replaced with a updated version of the framework.

Chapter 6

Future development

The testing software written during this bachelor project is far from finished. The final version of the application is demanded to have a dozen of tests implemented and should be able to not only test single chips but also test half-staves containing 9 or 14 chips. Another planned feature is the use of an online database to store test results of tested ALPIDE's. In the future this database could be analyzed and quantitative demands for the final ALPIDE could be formed. All these desired features should be implemented before the testing software is put into work.

However, if we concentrate on the current status of the software, we can conclude a few things. First of all, the application has multiple tests implemented: the register, threshold and noise test. These tests combined give a good impression of the workings of the ALPIDE. The threshold and noise tests each give visual feedback and their data can be examined within the application during a running test. The register test does not yet contain all the registers needed for a next version of the ALPIDE but it has been programmed with the ease of extension in mind. Furthermore, the overall structure of the program, the use of threading and the created link between the existing framework and the GUI gives a solid foundation for expanding. The tab based GUI gives the opportunity to easily add different kinds of tests just by adding more tabs.

In the past half year of programming, there has been made a lot of progress. When the new test version of the ALPIDE chip arrives in the first quarter of 2016, the software should provide a solid foundation for further development.

Bibliography

[1] The ALICE collaboration, *Technical Design Report for the Upgrade of the ALICE Inner Tracking System*, CERN, Geneva Switzerland, 2013.

[2] The ALICE collaboration, *Upgrade of the Inner Tracking System Conceptual Design Report*, CERN, Geneva Switzerland, 2012.

[3] Felix Reidt for the ALICE collaboration, *Upgrade of the ALICE Inner Tracking System*, CERN, Geneva Switzerland, 2015.

[4] The ALICE collaboration, D. Anstett, P. Bouvier, B. Cantin, M. Connor, F. Dalla Santa, C. David, C. Decosse, A.Ferrand, D. Gabriele, U. Genoud Prachex, C. Hervet, P. Ijzermans, S. Junod, X. Lagrue, Y. Lesenechal, S. Maridor, D. Meunier-Picard, B. Moris, E. Paulat, S. Philippin, L. Radulescu, J. Van Beelen, M. Van Stenis, D. Williams, *The ALICE experiment at the CERN LHC*, CERN, Geneva Switzerland, 2008.

[5] Phys.org *Closing in on the border between primordial plasma and ordinary matter* http://phys.org/news/2012-08-border-primordial-plasma-ordinary.html, accessed January 2016.

[6] The ALICE collaboration *ALICE EMCal Physics Performance Report* http://arxiv.org/pdf/1008.0413.pdf accessed January 2016.

[7] Vito Manzari *The present Inner Tracking System - Steps forward!* http://alicematters.web.cern.ch/?q=ALICE_currentITS accessed January 2016

[8] CERN *The Large Hadron Collider* http://home.cern/topics/large-hadron-collider, accessed January 2016

[9] The ALICE collaboration, *Alice Physics*, http://aliceinfo.cern.ch/Public/en/Chapter1/results.html, accessed January 2016.

[10] W N Cottingham, D A Greenwood, *An Introduction to the Standard Model of Particle Physics*, Cambridge Univ Press 2nd ed, 2007.

[11] Helmut Satz, *The Quark-Gluon Plasma, A Short Introduction*, Fakultät für Physik, Universität Bielefeld, Germany, 2011.

[12] ALICE ITS ALPIDE development team, *pALPIDEfs datasheet*, CERN, Geneva Switzerland, 2013.

[13] Wikipedia, *Rolling Shutter* https://en.wikipedia.org/wiki/Rolling_shutter, accessed January 2016.

[14] Joe Rubinstein *How does a global shutter work?* http://www.digitalbolex.com/global-shutter/ accessed January 2016.

[15] INFN sez Bari, CAD service, *USER'S MANUAL, ALICE Pixel IT Barrel MOSAIC Test System*, INFN, Italy, version 1.0b, 2015.

[16] OpenCores *WISHBONE System-on-Chip (SoC)Interconnection Architecturefor Portable IP Cores* http://cdn.opencores.org/downloads/wbspec_b4.pdf, 2010.

[17] Wikipedia *Scientific Linux* https://en.wikipedia.org/wiki/Scientific_Linux, accessed January 2016.

[18] The Qt company *Qt - Qt for Application Development* http://www.qt.io/application-development, accessed January 2016.

[19] Pim Verschuuren *Testing the pALPIDE v2 chip for the upgrade of the ALICE Inner Tracking System* Universiteit Utrecht, 2016