TECHNISCHE UNIVERSITÄT WIEN

HEPHY INSTITUT FÜR HOCHENERGIEPHYSIK

DIPLOMARBEIT

Data Acquisition Systems for Depleted Monolithic Active Pixel Sensors

im Rahmen des Studiums

Technische Physik

eingereicht von

Ing. Bernhard Pilsl BSc

Matrikelnummer: 01622520

ausgeführt am

Atominstitut der Fakultät für Physik der Technischen Universität Wien und
dem Institut für Hochenergiephysik der Österreichischen Akademie der Wissenschaften
(HEPHY)

Betreuung

Assistant Prof. DI Dr.techn. Albert Hirtl (TU Wien / Atominstitut)
DI Dr.techn. Thomas Bergauer (ÖAW / HEPHY)

Wien, December 12, 2022

(Unterschrift Verfasser) (Unterschrift Betreuer)

Abstract

Silicon detectors play a major role in tracking and vertexing in all major particle collider experiments. In recent years new production techniques in the chip industries allow complementary metal-oxide-semiconductor (CMOS) electronics on high-resistivity substrates and high-voltage compatible processes. In high energy physics, these advancements lead to depleted CMOS monolithic active pixel sensors. The possibility of applying high depletion voltages, which is why this detector type is also referred to as HV-CMOS, and using relatively high-resistivity substrates suits this technology well for high radiation environments. Within this approach, the charge-sensitive collecting electrodes, as well as the front-end electronics, are integrated into the same CMOS wafer. This allows for lower noise, higher sensitivity, and also a reduced material budget compared to standard silicon pixel detectors. The use of the commercial standard CMOS process allows for reduced production costs and, therefore, manufacturing of detectors with a large sensing area is possible.

In the context of this master thesis, two HV-CMOS pixel detectors, the RD50-MPW3 developed by CERN-RD50 collaboration and the TJ-Monopix2 developed by the ATLAS collaboration and now studied by the Belle-II collaboration for possible upgrades, have been integrated into data-acquisition frameworks consisting of hardware and software components. Within these frameworks called Caribou, BDAQ53 and EUDAQ, the detectors have been tested in a laboratory environment as well as at the particle-accelerator facilities DESY and CERN.

The recorded data were analyzed and interpreted in order to quantify the performance of these detector prototypes and to compare them with each other. Moreover, simulations were performed and were compared with the measured results.

Contents

1. Introduction

1.1. Detectors in particle physics experiments

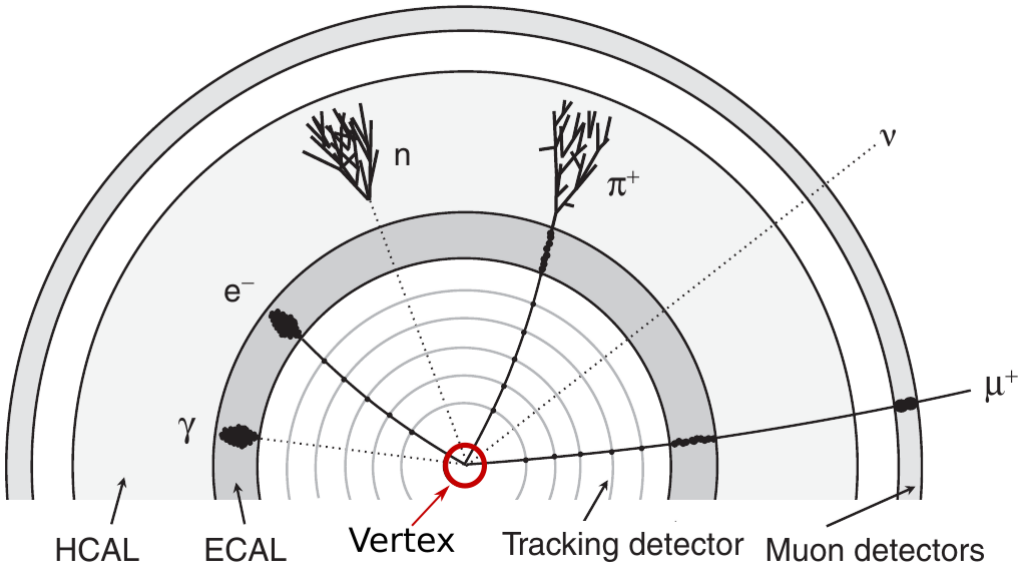


Figure 1.1.1.: Typical layout of High Energy Physics (HEP) detectors consisting of a tracking and vertexing system, an electromagnetic calorimeter (ECAL), a hadron calorimeter (HCAL) and muon detectors. Adapted from [1].

Most large particle physics experiments nowadays utilize a detector system like the one shown in fig. 1.1.1. The inner tracking and vertexing layers are needed to reconstruct the track the collision products traversed on and to fit those tracks to a common vertex (the origin of the daughter particles). The two different calorimeters are used to measure the energy of the particles and thereby to identify them.

The most commonly used detector type at the moment for the tracker is a silicon strip detector, which only offers a one dimensional spatial resolution, which often is sufficient though as the magnetic field, which is used to bend the tracks the particles traverse on and thereby allows to evaluate the mass of the particles from the measured bending radius, can be aligned in a way a one dimensional resolution

is sufficient.

For vertexing though a two dimensional spatial resolution is beneficial as it allows to narrow down the collision vertex better. Pixel detectors offer a 2D resolution and are nowadays the most utilized detector type for vertexing purposes.

Pixel detectors are also being investigated for the whole tracking system for various experiments (for example in the *BELLE-II* experiment [2]) at the moment as the analysis possibilities would nonetheless get enhanced by replacing the strip detectors by pixel detectors.

During this thesis pixel detectors got investigated.

1.2. Future detector development targets

The European Committee for Future Accelerators (ECFA) is a committee picturing (among others) the most important detector Research and Development (R&D) goals for several decades in advance in order to further push the HEP-experiments and the fundamental research performed at particle accelerator facilities.

The most recently published roadmap [3] stresses the importance of HV-CMOS-Depleted Monolithic Active Pixel Sensors (DMAPS) and refers to them as a *key technology*. The characteristics and expectations of / from this technology are quoted as

- Very small pixel pitches resulting in the best position resolution achieved so far.
- The combination of the sensing diode and the readout electronics in the same wafer result in a reduced material budget and it also minimizes multiple scattering.
- The detectors operated in a fully depleted state and with optimized electric field configurations allow to reach timing resolutions in the order of $\mathcal{O}(100\,\mathrm{ps})$ and radiation tolerances of up to $1 - 2 \times 10^{15} \mathrm{n_{eq}cm^{-2}}$.

A comparison of these characteristics with the projected future R&D goals for detectors in fig. 1.2.1 makes it clear why the DMAPS technology is of such high interest and why so much research work is being performed in this area at the moment.

1.3. Usage of DMAPS in current experiments

As the DMAPS technology is rather new, large experiments which are operated in the data production stage at the moment and therefore are already equipped with a working vertexing system, did not yet exchange their tracking detectors to DMAPS. Though a lot of experiments are currently evaluating this technology.

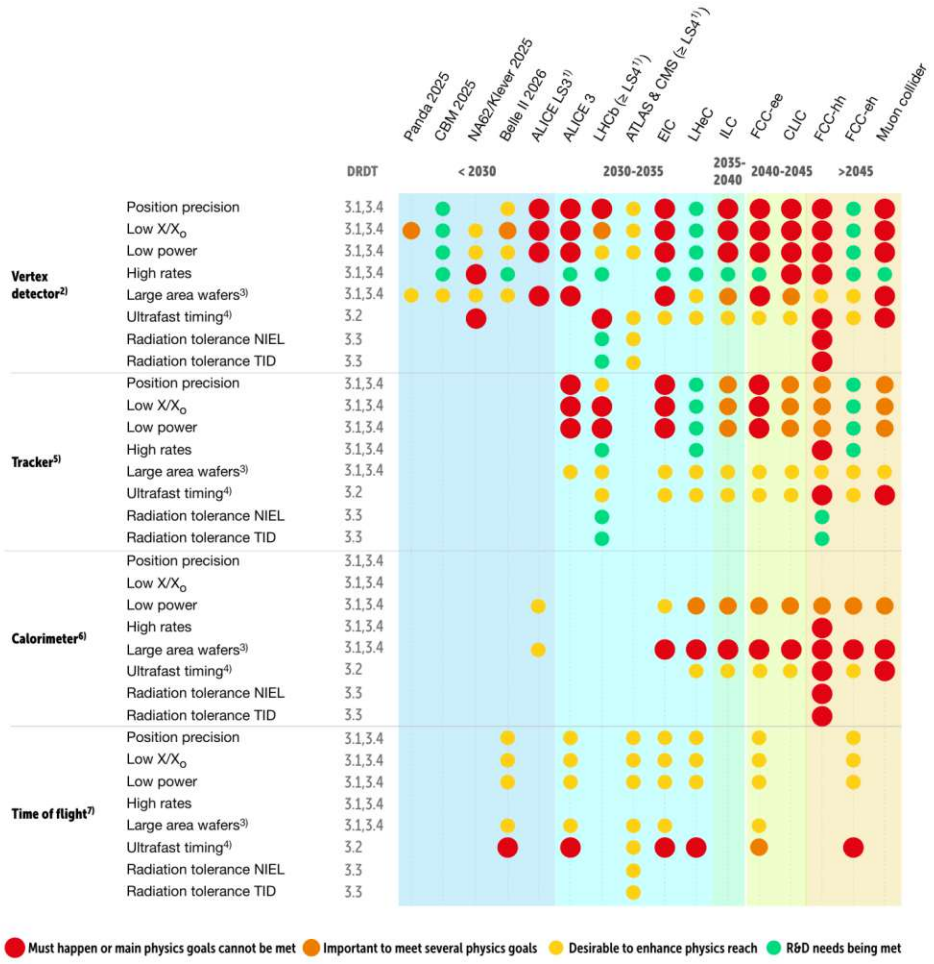


Figure 1.2.1.: Timeline of future detector R&D tasks to meet future physics goals in various experiments. Adapted from [3].

There has also not yet been a opportunity for experiments like the CMS [4] or the ATLAS [5] to upgrade their detector systems.

For the A Large Ion Collider Experiment (ALICE)-collaboration there has been such an opportunity during the Long Shutdown 2 (LS2) of CERN to install DMAPS detectors in their inner tracking system version 2 (ITS2) [6]. For this upgrade a chip named ALICE inner pixel detector (ALPIDE) was designed.

The performance of the ALPIDE after different irradiation levels is shown in fig. 1.3.1. Even after fluences of $10^{15} n_{eq} cm^{-2}$ the chip shows an excellent signal response and also the signal rise time shows only a small increase of 16.7 ns – 19 ns relative to the unirradiated sample [7].

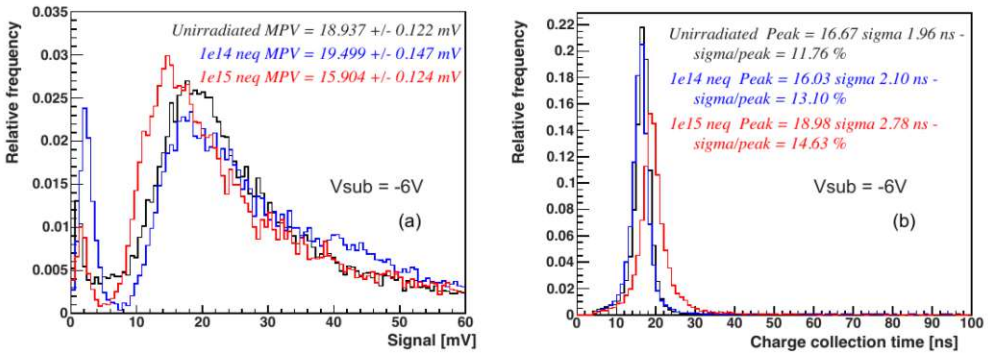


Figure 1.3.1.: Signal response of the ALPIDE chip for different irradiation levels. The black curve corresponds to the performance of the chip before irradiation, the blue one for fluences of $10^{14} n_{eq} cm^{-2}$ and the red one for $10^{15} n_{eq} cm^{-2}$. Figure (a) shows the amplitude distribution for source measurements with a ^{90}Sr source while (b) shows the signal rise time. Adapted from [7].

The material budget of the whole ITS2 as a function of the azimuthal angle is shown in fig. 1.3.2. By far the biggest cause of the material budget arises from passive components used for mechanical support, cooling and shielding. The detector itself (thinned down to 50 μm) only accounts for a material budget of $X/X_0 \approx 0.35\%$ [8].

These two examples illustrate that the DMAPS-technology indeed is capable of meeting the expected demands presented above in section 1.2.

1.4. Structure of this thesis

In this current chapter the physics field in which this thesis was performed was introduced and the motivation for the high interest in DMAPS and the amount of research work performed in this area got justified. In the following two chap-

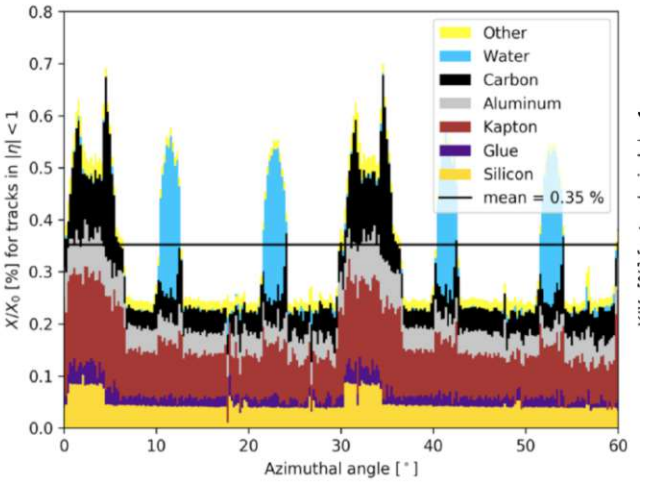


Figure 1.3.2.: The dependence of the material budget on the azimuthal angle of the ITS2 in the ALICE experiment. Adapted from [8].

ters the basics which are needed to understand this thesis are presented. While chapter 2 addresses the physical principles on which particle detectors operate on, chapter 3 lays out the way data acquisition (DAQ)-systems work, their tasks and their necessity. The actual work, the DAQ development, the measurements and the analysis, which was done during the course of this thesis for the two different DMAPS is presented in chapter 4 and chapter 5. The final conclusion and a comparison of those two detectors follows in chapter 6.

2. Physical Background

2.1. Interaction of particles with matter

When particles pass through matter, they, due to various physical effects, eventually will interact with it and thereby loose energy.

2.1.1. Interaction of charged particles with matter

2.1.1.1. Energy loss through ionization / The Bethe-Bloch equation

For a very broad range of particle energies[1] ionization and excitation of the atoms in the passed medium is the dominant energy loss process.

When one particle with a high enough energy passes through matter it interacts with many atoms. An exact calculation for one specific particle therefore is not reasonably manageable, which is why a stochastic approach is taken. The average energy loss per traversed path is given by the following *Bethe-Bloch-equation* [1]

$$\frac{\mathrm{d}E}{\mathrm{d}x} \approx -4\pi\hbar^2c^2\alpha^2\frac{nZ}{m_e v^2}\left(\ln\left[\frac{2\beta^2\gamma^2c^2m_e}{I_e}\right]-\beta^2\right) \tag{2.1.1}$$

The used quantities are listed in table 1.1. In fig. 2.1.1 plots of this formula for various materials are shown. The quantity $\beta\gamma = \frac{p}{mc}$ on the x-axis is chosen as therefore different particle types (differing in mass) can be identified too.

Quantity	Definition
c	speed of light
α	fine-structure constant
n	number density of the atoms in the material
Z	atomic number of the atoms in the material
v	velocity of the incident particle
β	reduced velocity $\beta = \frac{v}{c}$
γ	relativistic factor $\gamma = \frac{1}{\sqrt{1-\beta^2}}$
m_e	mass of an electron
I_e	effective ionization potential of the material

Table 1.1.: The description of the quantities in eq. (2.1.1).

[1] From a few eV up to $\approx 10\,\mathrm{GeV}$

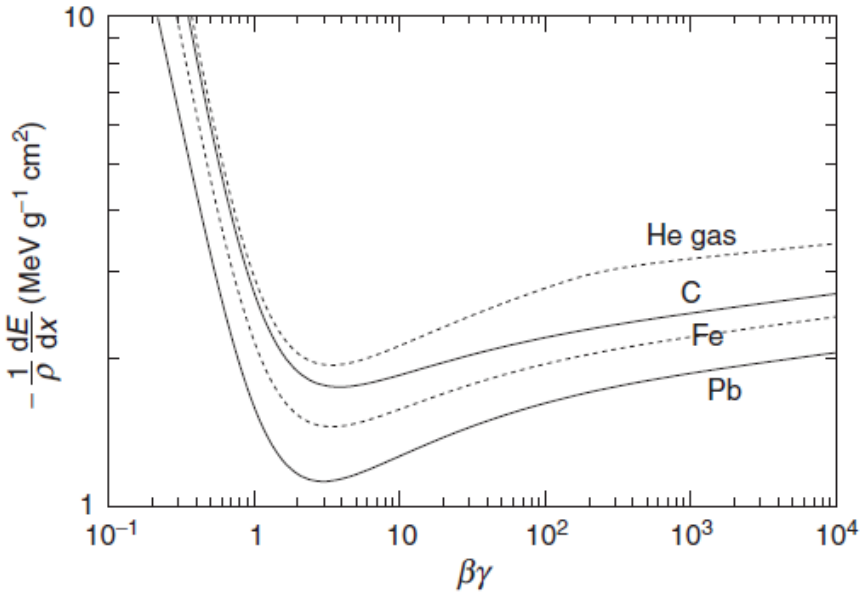


Figure 2.1.1.: *Bethe-Bloch* Energy loss curves according to eq. (2.1.1) of a singly charged particle traversing different materials [1]

It can be seen that there is a minimum for all particles located at $\beta\gamma \approx 3$. A particle with an according energy therefore deposits a minimum of energy in the traversed material, such a particle is referred to as Minimum Ionizing Particle (MIP). A proton for example is a MIP at $E \approx 3\,\mathrm{GeV}$. Detectors are typically designed to work at this minimum [9]. This is because it is the worst case for detecting particles as the least energy for detection is deposited in the material.

Considering the logarithmic scale and looking at higher energies/$\beta\gamma$-values it can be seen that the energy loss remains approximately the same for a broad range of high energies. This is one of the reasons why in HEP-experiments particles with high energies[2] are in use. This way small variations in the beam energy merely influence the measurement.

2.1.1.2. Energy loss through Bremsstrahlung

While at lower energies charged particles mainly interact with the atomic shell of the atoms the material is consisting of, at higher energies particles are able to penetrate the shell and interact directly with the nucleus. This scattering process results in deceleration of the particle. As all accelerated charges emit photons this leads to an energy loss which can be quantified by the following equation[9] :

$$\frac{\mathrm{d}W}{\mathrm{d}t} = \frac{2}{3}\frac{z^2e^2}{4\pi\epsilon_0 c^3}|\ddot{\vec{x}}| \propto \frac{z^4Z^2e^6}{m^2} \tag{2.1.2}$$

[2]The *CERN SPS* beam for example is operating at up to $E \approx 450\,\mathrm{GeV}$.

Here Z corresponds to the charge number of the nucleus, z to the charge number and m to the mass of the incident particle. A calculation taking quantum electrodynamics into account yields in a proportionality factor of $\frac{Z^2 E}{m^2}$. Thus it can be seen that energy loss through Bremsstrahlung is only significant at high energies $\mathcal{O}(100\,\mathrm{GeV})$ for particles like protons or pions, while for electrons or positrons this effect has already to be considered at lower energy levels[3].

The interaction of an electron with a nucleus is illustrated in fig. 2.1.2.

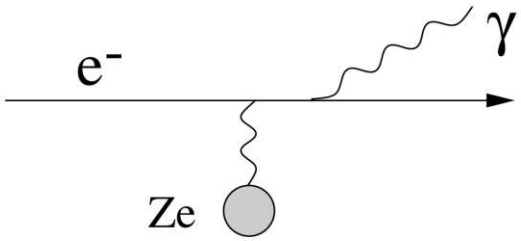


Figure 2.1.2.: Illustration of an electron being scattered by the Coulomb field of a nucleus and emitting Bremsstrahlung in the process. Adapted from [9].

2.1.2. Interaction of photons with matter

The energy of a photon is determined by the following equation

$$E = h\nu \tag{2.1.3}$$

where h corresponds to the Planck-constant and ν to the frequency of the light. Depending on this energy different physical effects act on the photon and on the atoms / molecules of the traversed material.

2.1.2.1. Photo effect

In the so called *photoelectric effect*, as shown in fig. 2.1.3(a), the photon gets entirely absorbed and an electron from an atom within the material is getting emitted. The kinetic energy of the emitted electron is given by $E_e = E_\gamma - E_{bound}$, where E_γ is the energy of the incident photon and E_{bound} is the energy which is necessary to ionize the atom.

[3] For electrons the energy loss through Bremsstrahlung already exceeds the energy loss through ionization at 7 MeV in lead or 100 MeV in air (the difference arises from the Z^2 dependency).

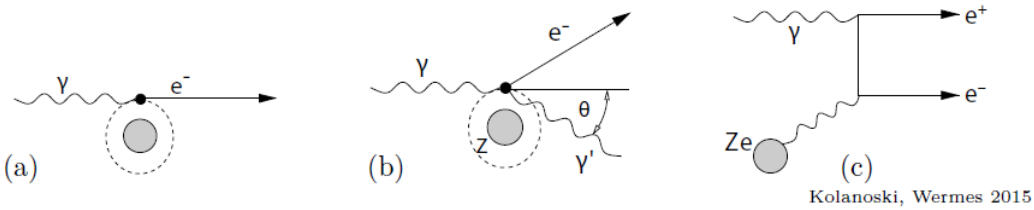


Figure 2.1.3.: Physical effects acting when photons interact with matter. (a) photoelectric effect, (b) Compton effect, (c) pair production. Figure adapted from [9]

2.1.2.2. Compton effect

When an electron is elastically scattered by a shell electron, as shown in fig. 2.1.3(b), it looses energy and is scattered by an angle θ. The kinetic energy of the emitted electron corresponds to $E_e = h(\nu_0 - \nu_1) - E_{bound}$, where ν_0 refers to the frequency of the incident photon, ν_1 to the frequency of the scattered photon and E_{bound} again refers to the energy needed to ionize the atom.

2.1.2.3. Pair production

In fig. 2.1.3(c) the pair production process is sketched. When a photon has an energy of $E_\gamma \geq 2m_e c^2$, where m_e corresponds to the rest mass of an electron[4], a photon possibly creates an electron positron pair. In order to retain conservation of momentum this process may only appear in the field of a nucleus.

2.2. Particle Detectors

2.2.1. Silicon Detectors

2.2.1.1. Semiconductor basics

Silicon in its solid state condenses in a *diamond-lattice*, the primitive cell is shown in fig. 2.2.1. The whole crystal consist of many of these primitive cells next to each other, therefore the potential in which the electrons reside can be considered periodically. When considering $\vec{T}$ as the translation vector of the lattice the periodicity can be formulated mathematically in the following way:

$$V(\vec{r}) = V(\vec{T} + \vec{r}) \tag{2.2.1}$$

When aiming to solve the Schrödinger-equation for such a periodic potential so called *Bloch-functions* arise [10]. These functions show that the available energy states for the electrons in the solid are very close too each other, in fact they do even overlap. This overlap leads to the formation of energy bands.

[4] This corresponds to an photon energy of $E_\gamma \approx 1022\,\mathrm{keV}$

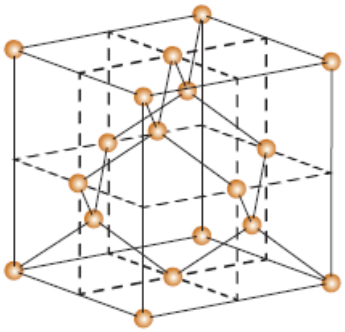


Figure 2.2.1.: The primitive cell of the *diamond-lattice* in which silicon crystalizes. Figure from [9].

Within these bands there is almost a continuum of states which may be populated by electrons, so already small electric fields can increase the (kinetic) energy of electrons and therefore these electrons are considered to be "free" and an electric current can flow. Outside these bands there are *forbidden* states for the electrons.

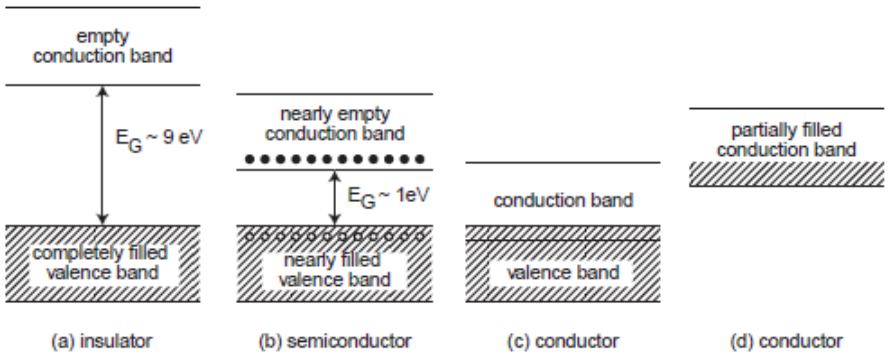


Figure 2.2.2.: Schematic energy bands of insulators (a), semiconductors (b) and conductors (c, d). E_G is the energy of the band gap. Figure from [9].

As shown in fig. 2.2.2 the energy bands of conductors are either not fully filled, or multiple bands are overlapping. In the case of semiconductors and insulators there is a *band gap* between the fully filled band with the highest energy[5] and the next available (conduction-)band.

The existence of this gap is the cause of the difference between conductors, semiconductors and insulators considering their electrical characteristics.

Intrinsic semiconductors In the case of so called *intrinsic semiconductors* the energy of the band gap E_G is in the order of a few eV's[6] and therefore energy can

[5] This band commonly is referred to as the *valence band* in which the electrons are bound to atoms in the lattice.

[6] For silicon $E_G \approx 1.12\,\text{eV}$ at room temperature

be distributed for example by heat, radiation or incident particles to the electrons which then reside in a higher, not fully populated, energy band. These electrons again are considered "free" in their new band. This makes it possible for electric current to flow. Actually not only the electrons can be considered as charge carriers but also the positively charged *holes* in the valence band take place in transportation of electric current. This behavior is sketched in fig. 2.2.3.

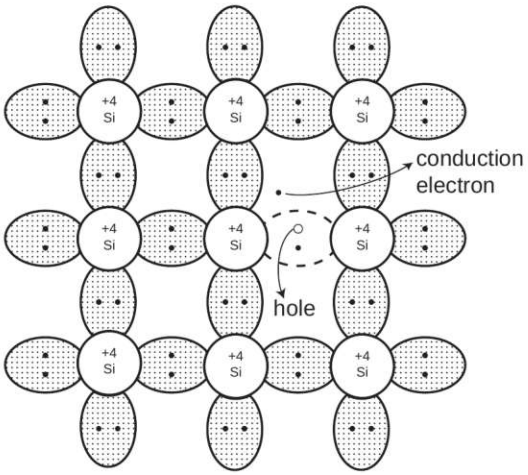
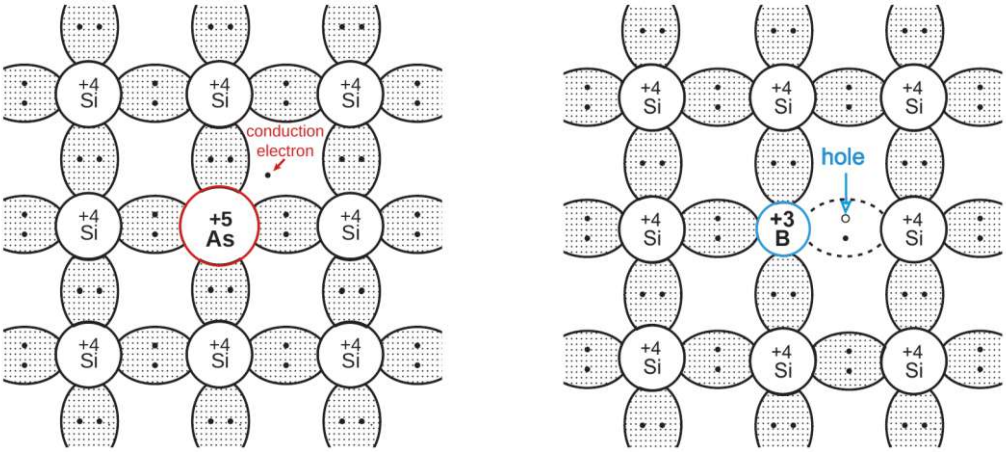


Figure 2.2.3.: An electron and a hole are freed in an intrinsic semiconductor and can therefore move freely. [9]

Extrinsic / Doped semiconductors By adding impurities to the silicon, the electrical characteristics of the semiconductor can be changed. This process is referred to as *doping*.

By adding a small number of atoms with a higher number of valence electrons to the lattice, the excess valence electrons do not fit neatly into the energy bands of the silicon lattice. Therefore they have to occupy states in the conduction band, which leads to free conduction electrons, as shown in fig. 2.2.4a. This is referred to as *n-doping*[7].

On contrary by adding atoms with a lower number of valence electrons this leads to unoccupied states in the energy band of the silicon lattice. This unoccupied states are referred to as *holes* and can be considered to be positively charged charge carriers. The process is called *p-doping*[8] and is shown in fig. 2.2.4b.



(a) n-doped **(b)** p-doped

Figure 2.2.4.: Schematic bonding representation of extrinsic semiconductors. [9]

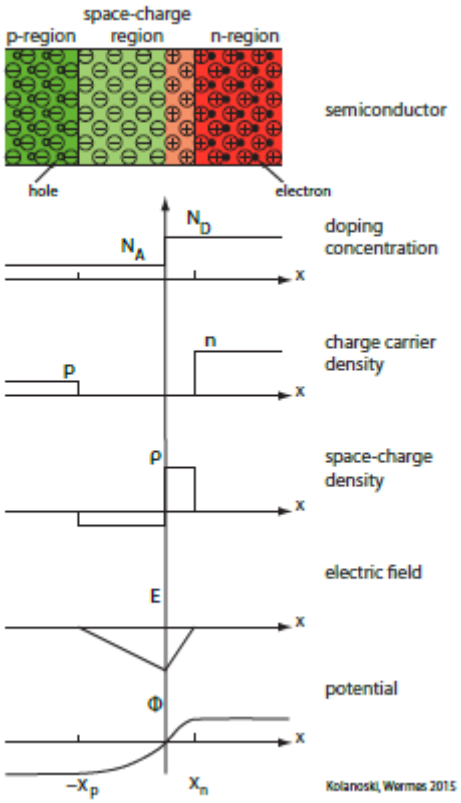


Figure 2.2.5.: Scheme of a P-N-junction also known as a diode. [9]

2.2.1.2. PN junction

By connecting a p-doped semiconductor with a n-doped one, the free conduction electrons from the n-doped part will fill the holes on the p-doped side by diffusion processes. This charge transfer will lead to a depletion region in the junction. As shown in fig. 2.2.5 an electric field builds up which applies a counteracting force to the diffusion of the charge carriers. The current flow stops when the electric field reaches a magnitude at which the energy gain by excess electrons from the n-doped region filling holes in the p-doped region is in equilibrium with the energy need to transfer charges in the potential of the junction.

Such a setup of n- and p-doped semiconductors is widely used in electronics and the component utilizing this behavior is called a *diode*.

When applying a voltage to such a junction:

- With the $\oplus$-pole to the n- and the $\ominus$ to the p-doped side the depleted region expands as more electrons from the n-region will occupy holes in the p-doped region and therefore the electric field in the junction increases, which counteracts the externally applied voltage. This is referred to as operating a diode in *reverse direction*.

- With the $\oplus$-pole to the p- and the $\ominus$ to the n-doped side the externally applied voltage has to exceed the potential in the junction to be able to drive a current, but when it does, a current can flow. This is referred to as operating a diode in *forward direction*.

2.2.1.3. A diode as a particle detector

When operating a diode in reverse direction by applying a reverse bias voltage as stated above the depletion region expands and therefore no current[9] will flow[10]. If energy is deposited in the depleted region, for example by radiation or particles, this causes the generation of electron hole pairs. These pairs are again available for transportation of charge.

This mechanism allows for detection of particles interacting with the diode in the depleted region, by measuring the current flow through the diode. Each time a particle hits the diode a short[11] current pulse or charge, when integrating, can be measured.

2.2.1.4. Charge transport mechanisms in a diode

In a diode mainly two different charge transportation mechanisms occur.

[7] Commonly used atoms for n-doping are for example P, As, Sb, ...
[8] Commonly used atoms for p-doping are for example B, Al, Ga, ...
[9] Besides a small amount of so called *dark current*.
[10] At least until a certain *breakdown voltage* is exceeded.
[11] Typically a few ns long.

1. **Charge transport by drift**
   When an electric field $\vec{E}$ is applied to the junction the equation of motion, according to the *Drude model* [10], for any charge is given by:

$$m_{\text{eff}} \left(\dot{\vec{v_D}} + \frac{v_D}{\tau} \right) = q\vec{E} \tag{2.2.2}$$

   Here $q = \pm e$ corresponds to the charge of an electron (- case) or a hole (+ case) and m_{eff} to the effective mass of the charge carriers. The relaxation time τ takes various parameters like scattering in the lattice into account and can be considered as the mean time between collisions.

   In the stationary case the drift velocity can be approximated to be constant $\Rightarrow \dot{\vec{v_D}} = 0$. This way the drift velocity results in the following current:

$$\vec{j}_{\text{drift}} = \sigma\vec{E} \tag{2.2.3}$$

   In this equation $\sigma = nq\mu$ can be identified as the electrical conductivity with the mobility factor $\mu = \frac{q\tau}{m_{\text{eff}}}$ and the charge carrier density n.

2. **Charge transport by diffusion**
   Due to thermal motion the spatial concentration differences between the differently doped regions are driven into equilibrium by a diffusion current. With the density n, p of electrons and holes, respectively, this process results in the current for electrons $\vec{j}_{\text{n,diff}}$ and holes $\vec{j}_{\text{p,diff}}$ [9]:

$$\vec{j}_{\text{n,diff}} = -eD_n \boldsymbol{\nabla} n, \quad \vec{j}_{\text{p,diff}} = -eD_p \boldsymbol{\nabla} p \tag{2.2.4}$$

   where the diffusion coefficients $D_{n/p}$ of electrons and holes, respectively, are specific for a certain semiconductor at a certain temperature.

The total current for electrons and holes considering both mechanisms results in:

$$\vec{j}_n = -e\underbrace{\mu_n n\vec{E}}_{\vec{j}_{\text{n,drift}}} - \underbrace{eD_n \boldsymbol{\nabla} n}_{\vec{j}_{\text{n,diff}}}, \qquad \vec{j}_p = -e\underbrace{\mu_p n\vec{E}}_{\vec{j}_{\text{p,drift}}} - \underbrace{eD_p \boldsymbol{\nabla} p}_{\vec{j}_{\text{p,diff}}} \tag{2.2.5}$$

Which mechanism dominates the overall charge transport therefore directly depends on the applied electric field / bias voltage.

2.2.1.5. Pixel detectors

When using a p-doped silicon substrate and applying n-doped regions at the surface to it, arranged in a grid, this leads to an array of diodes. The spatial distance between neighboring diodes is referred to as the *pixel pitch*. Such a diode array is the base of a pixel detector, but without additional *readout* electronics not suited to work as an actual particle detector.

The requirements which should be fulfilled by the readout electronics are for example:

- convert the diode current pulse into a charge and amplify it.
- digitize the analog signal and assign it with a Timestamp Leading Edge (TS-LE) as well as a Timestamp Tailing Edge (TS-TE).
- Offer a digital interface for configuration and data transmission to a processing unit[12].

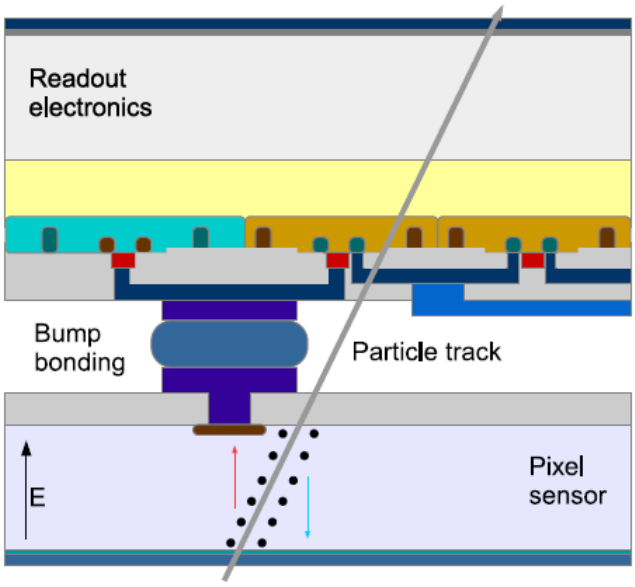


Figure 2.2.6.: Sketch of a hybrid pixel sensor [11].

Hybrid pixel sensors In the case of so called *hybrid pixel sensors* the readout electronics and detector are separate components and are connected to each other by a process called bump-bonding. Such a design is shown in fig. 2.2.6.

On the one hand having two independent components for readout and sensing is of benefit as for example the sensor (often referred to as the Front End (FE)) could be replaced while the readout part is not altered when the composite detector system should be utilized in a different environment, for example in an environment with higher levels of radiation.

On the other hand this technology comes with some major drawbacks:

- Bump-bonding is an expensive and complex assembly process with low output rate.
- The composite system accounts for a substantial increase in material budget and due to the multi layer setup multiple Coulomb scattering is more likely to occur. Both resulting in detectors less suitable for tracking.

[12] In most applications FPGAs perform the direct communication with the readout ASIC.

Depleted monolithic active pixel sensors The technology of DMAPS takes a different approach. The readout electronics and the sensing diodes are integrated in the same wafer. This is shown in fig. 2.2.7. This technology has several advantages to the hybrid approach:

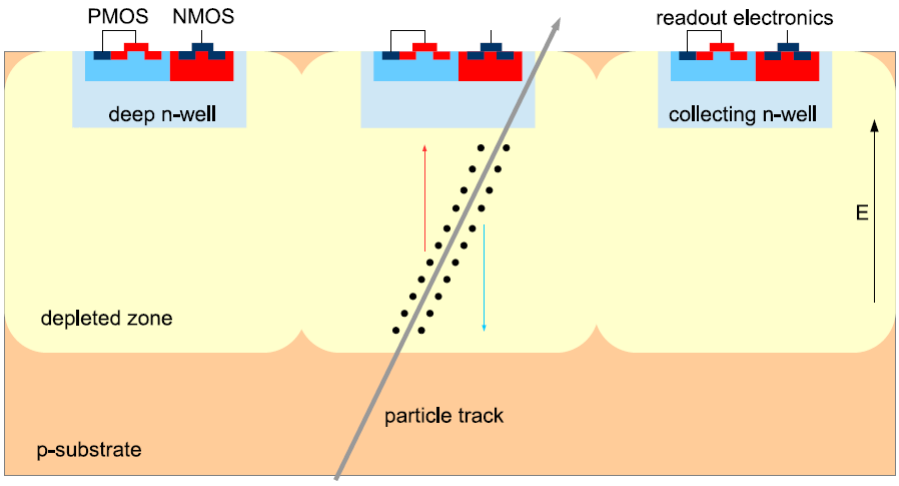


Figure 2.2.7.: Sketch of a DMAPS. Adapted from [11].

- They are fabricated in a commercial standard CMOS process, which leads to reduced costs, a better availability and faster fabrication.
- As there is no two layer setup with bump-bonds in between, the thickness of the sensor can drastically be reduced, which contributes to a lower material budget. Also the likeliness of multiple Coulomb scattering is reduced.

Despite these advantages, the DMAPS-technology still faces some challenges:

- Radiation Hardness
  Each time the detector is hit by a particle this might cause some damage to the detector itself. Thus the detectors will eventually degrade over time until they are no longer usable for tracking in HEP-experiments.

  In particular radiation causes defects in the lattice which can act as traps for charge carriers. When the main charge transport is caused by charge drift instead of diffusion, which, as shown in eq. (2.2.5), can be achieved by applying high bias voltages, this behavior can be counteracted.

  The depletion depth corresponds to the product of bias voltage and substrate resistivity. A bigger depletion depth also allows for a higher radiation tolerance. Therefore not only high bias voltages are aimed for, but also relatively high resistivity substrates [11].

  For the Future Circular Collider (FCC) at CERN radiation levels of $\approx 7 \cdot 10^{17}$ $1\text{MeVn}_{\text{eq}}/\text{cm}^2$ are expected [12].

- Spatial resolution
  In order to be able to perform high quality analyses, the detectors need to achieve a high spatial and timing resolution. The former can be achieved by reducing the pitch size, although charge sharing effects have to be kept in mind and should be aimed to be minimized. The latter depends on multiple factors itself:
  - The integration time in which the charge is collected at the implants
  - Delay in the readout electronics
  - Time walk of the comparator
- Readout rate
  In order to be suitable for high luminosity particle beams the detectors have to be able to be read out with a high frequency.

  For the high luminosity LHC at CERN a particle rate of $\approx 1\,\mathrm{GHz/cm^2}$ is foreseen [13].

All these demands should be met by a clever chip design and is a current subject of research. Both chips evaluated during this thesis are designed as DMAPS and aim to meet the above listed requirements with a bit different design approaches.

Fill factor The two main variants of CMOS sensors are shown in fig. 2.2.8. They are named after the size of the implemented collection electrode.

- In the *large fill factor* structure, in fig. 2.2.8(a), the sensing diode consists of p-substrate located in the deep n-well. The electronics are also located in the charge collecting well. In this design approach a high bias voltage, up to a few 100 V can be applied. As discussed above higher bias voltages are beneficial for radiation hardness. On the other hand the sensor capacitance, dependent on the pixel size, can be in the order of several hundred fF [9]. As a consequence leading to higher noise levels, lower speed and an increased power consumption.
- The second design approach with the *small fill factor* structure, in fig. 2.2.8(b), specifies the sensing diode and the readout electronics being separated by the p-substrate. Within this design only lower bias voltages in the order of a few V can be applied, which leads to a lower radiation tolerance. On the other hand the sensor capacitance is much smaller, in the order of a few fF [9], resulting in lower noise levels, lower power consumption and a faster detection speed relative to the large fill factor structure.

Comparing the two different design approaches shows that both designs come each with their own benefits and drawbacks which are mostly inverted in the other design. Therefore during the design process the application field has to be evaluated and the best suited variant has to be chosen. For example for harsh radiation environments typically the large fill factor structure is preferred.

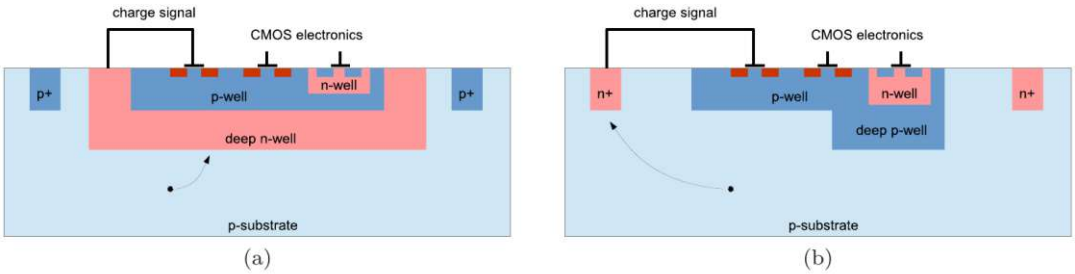


Figure 2.2.8.: The two principally different variants of DMAPS. (a) shows a large and (b) a small *fill factor* structure. Adapted from [11].

The modified Towerjazz process To utilize the benefits of a small collection electrode whilst avoiding the major drawbacks in terms of radiation tolerance a modified process with the *Towerjazz* foundry[13] was implemented. Within this process, as shown in fig. 2.2.9, the full depletion of the sensitive layer gets achieved by implementing a planar junction separated from the collection electrode. When applying a bias voltage depletion starts from the junction and extends over the full pixel area. In [14] a low dose n-type implant was implemented which generates a junction with the p-type epitaxial layer. The implant in this detector is doped low enough to be fully depletable up to the collection electrode with bias voltages of a few V and still obtaining a capacitance of only a few fF.

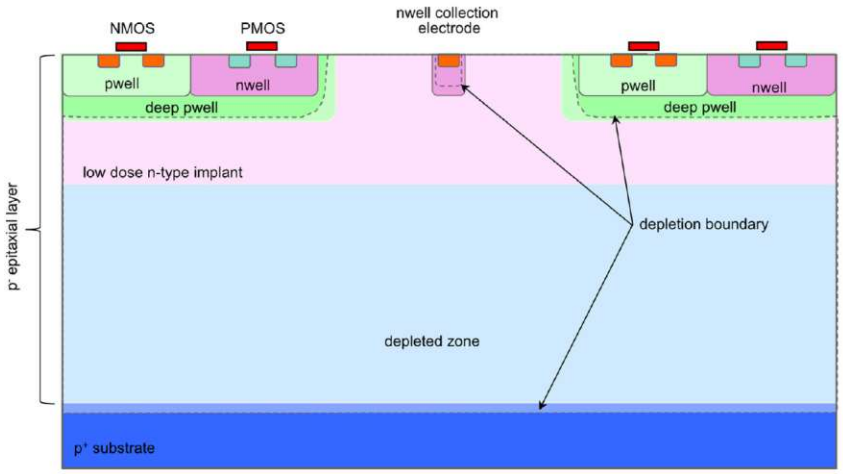


Figure 2.2.9.: The DMAPS geometry in the modified Towerjazz process. Adapted from [14].

[13] Tower Semiconductor Ltd., 20 Shaul Amor Avenue, Migdal Haemek 2310502, Israel, `http://www.towersemi.com`

2.2.2. Scintillation Detectors

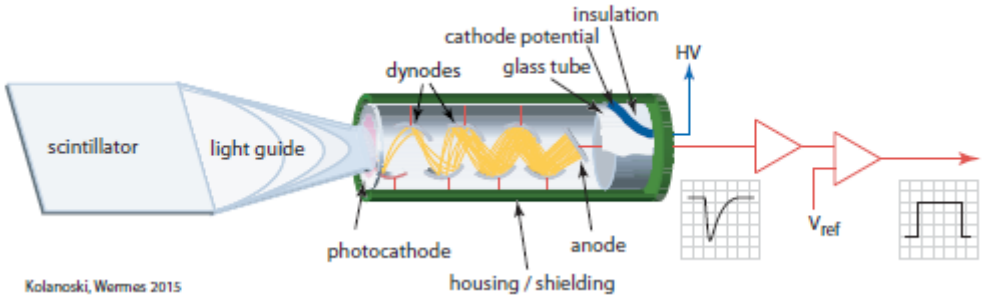


Figure 2.2.10.: Working principle of a scintillation detector. Adapted from [9].

A scintillation detector, as shown in fig. 2.2.10, typically consists of the actual scintillator followed by a photo multiplier and some signal processing electronics.

2.2.2.1. Scintillator

The eponymous scintillator itself consists of dense organic or inorganic crystalline material[14]. An incident particle will eventually deposit (a part) of it's energy in the crystal due to the various effects presented in section 2.1.1. Therefore some atoms in the material get ionized. The so freed electrons recombine with the atoms and emit photons with a wavelength corresponding to the ionization energy.

At this stage there is only a small amount of photons ,which are hard to detect, present. In order to increase the number of photons, typically a Photo Multiplier (PMT) is used.

2.2.2.2. Photo multiplier

The PMT consists of a photocathode, a "multiplier" tube with severval dynodes and the final collection anode.

At the photocathode, due to the photo effect, see section 2.1.2.1, the incident photons from the scintillator emit electrons.

When a voltage is applied the electric field between the several dynodes accelerate the electrons passing the PMT tube. Assuming n dynodes and a linearly divided voltage U the electrons will gain an energy of $\Delta E = e\frac{U}{n}$ at each dynode stage. Therefore at each dynode stage even more electrons, as the incoming electrons gained enough energy to ionize even more atoms, will be emitted. A sketch of this mechanism is shown in fig. 2.2.10.

This multiplying process generates a sizable amount of electrons which lead to an overall current flow between cathode and anode. This current pulse is detected by the signal processing electronics and is interpreted as a particle detection.

[14] Common materials for organic scintillators are for example naphthalene $C_{10}H_8$, anthracene $C_{10}H_{12}$ or stilbene $C_{14}H_{12}$.

2.2.3. Detector characteristics

2.2.3.1. Material Budget

When high energy particles traverse a material they loose energy due to the various mechanisms listed above. The following formula represents the average energy of particles, with an initial energy of E_0, which traversed a distance X in a material:

$$E(x) = E_0 e^{-\frac{X}{X_0}} \tag{2.2.6}$$

X_0 is referred to as radiation length and is dependent on the atomic number of the material. The quantity $m = \frac{X}{X_0}$ is for practical uses often evaluated for a specific detector and is called *material budget*. From the equations it can be seen that a reduced sensor thickness results in a lower material budget, the particles thereby loosing less energy traversing the detector, which is a major goal of the DMAPS approach.

2.2.3.2. Time over Threshold

The detection of a particle in a silicon detector results in a current peak in the detector electronics. In order to digitize the signal usually a discriminator is implemented, for which a threshold is set. Whenever the rising edge of the current pulse exceeds this threshold the discriminator will trigger a Time Stamp (TS) generator and a TS-LE gets assigned. When the descending current pulse falls below the threshold a TS-TE is generated. The Time over Threshold (ToT) is therefore calculated by

$$\text{ToT} = TS_{TE} - TS_{LE} \tag{2.2.7}$$

An example of this process is shown in fig. 2.2.11.

2.2.3.3. Spatial Resolution

The spatial resolution of a detector is defined by the standard deviation of the measurement error along the pixel width. Considering a homogeneous distribution of the incident particles, the probability density of the hit distribution can be approximated by a uniform distribution in the form of $f(x)dx = \frac{1}{a}dx$. Here a accounts for the sensor size (pixel pitch) in one dimension. The standard deviation therefore can be calculated by [9]:

$$\sigma_x^2 = \frac{1}{a}\int_{-a/2}^{a/2} \Delta_x^2 d(\Delta_x) = \frac{a^2}{12} \Rightarrow \sigma_x = \frac{a}{\sqrt{12}} \tag{2.2.8}$$

2.3. Characterization of detectors using a test beam

In order to quantify the performance of a particle detector, it is common practice to install the detector as a Device under Test (DUT) in a so called *test beam* at a particle accelerator facility. The DUT typically is placed into a beam telescope.

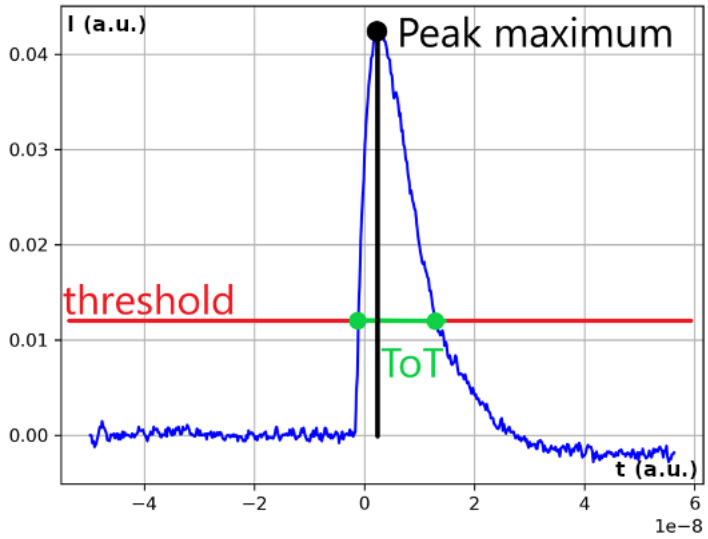


Figure 2.2.11.: Sketch of the measurement of a ToT

2.3.1. Beam Telescope

A beam telescope, like the one sketched in fig. 2.3.1, consists of multiple well known detectors. In between the telescope detector planes the DUT is installed.

When a particle traverses the telescope planes it will cause some so called *hits* at the position the particle interacts with them. When the spatial position of the detector planes, relative to the incident beam, as well as the position of the hit is known, an analysis framework can reconstruct the track the particles traversed.

Depending on the position of the DUT there is a possibility for each track to cause a hit on a certain spot on the DUT.

2.3.2. Measurable Quantities

The following section discusses several physical observables used to characterize particle detectors. It does not aim for completeness, but will only discuss the most important quantities which occurred during the course of this thesis.

2.3.2.1. Cluster size

When a particle hits a detector not only one but multiple adjacent detector pixels can signal an interaction. These multiple hit pixel form a *cluster*.

In order to make sense of the data a cluster analysis has to be performed. Otherwise each pixel which fired might be interpreted as being hit by a separate particle and this would complicate / falsify the data analysis.

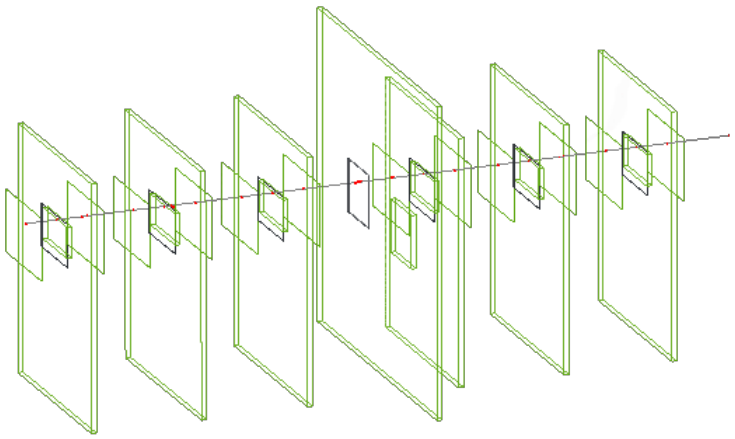


Figure 2.3.1.: A particle beam traversing a beam telescope. Six telescope planes and one DUT are shown. Figure created with the *[VisualizationGeant4]* module of *Allpix*^{2} [15]

2.3.2.2. Correlations between detector planes

When one particle traverses multiple detector planes the hits on these planes should be correlated. If the particle passes the first plane, orthogonally and not getting deflected, and again traverses the second plane orthogonally the position of the hits should be on the same coordinate in the x-y plane (when considering the z-axis as the one the beam is propagating on).

In reality a particle will not be incident exactly orthogonal and gets deflected. Therefore there will be a difference in the coordinates of the hit pixel on both planes. This behavior is sketched in fig. 2.3.2.

The correlation in the x-axis of the two detector planes is calculated by evaluating the quantity $x_{ref} - x_{dut}$. When using multiple particle hits for statistics there should appear a peak in the vicinity of 0 if the telescope is set up / aligned properly, as the deflection as well as the incident angle should be negligible in a first approximation.

2.3.2.3. Spatial residuals

When a particle passes a beam telescope, as sketched in fig. 2.3.3, it causes hits in multiple detector planes. An analysis framework can use the spatial position of the planes and the position of the hit to fit the track of the incident particle.

This track should cause a hit (with a certain possibility) at the various planes in the telescope setup at the position the track intersects them.

The difference between the interpolated track position and the actual hit is referred to as the *spatial residual* and is calculated by the following formula:

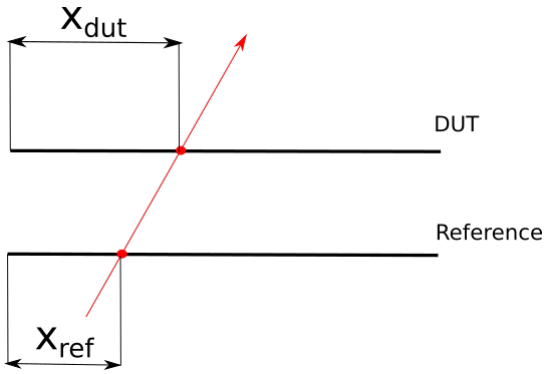


Figure 2.3.2.: Illustration of the correlation of a hit between a reference plane and the DUT. Shown is only the projection of the track on the x-axis, in general the hits and the track will occur in the three dimensional space.

$$\vec{r} = \vec{x}_{\text{Track}} - \vec{x}_{\text{Hit}} \tag{2.3.1}$$

2.3.2.4. Efficiency

When the reconstructed track of a particle intersects the DUT plane it should cause a hit at a certain position. The efficiency ϵ of the DUT is calculated with the following equation:

$$\epsilon = \frac{n_{\text{det}}}{n_{\text{exp}}} \tag{2.3.2}$$

Here n_{det} corresponds to the number of detected hits at the vicinity of the correct / expected position and n_{exp} to the number of expected hits from tracks intersecting the DUT-plane.

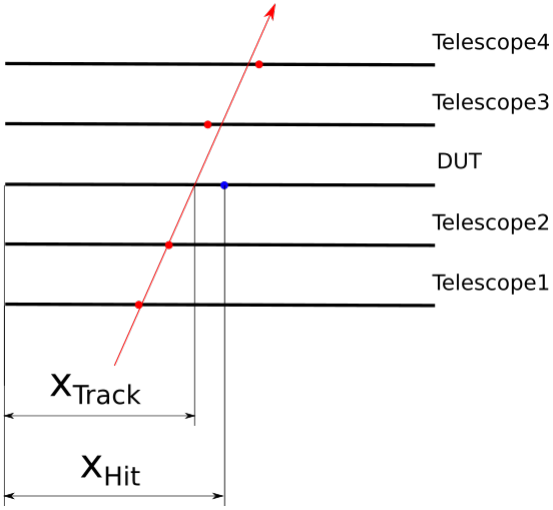


Figure 2.3.3.: Illustration of the spatial residual in a beam telescope. The red line represents the reconstructed track. The red and blue dots correspond to the actually detected position of the hit. Shown is only the projection of the track on the x-axis, in general the hits and the track will occur in the three dimensional space.

3. Data-Acquisition-Systems

3.1. Purpose of a Data-Acquisition-System

The operation of a detector usually requires a system which configures and operates it as well as performing the readout. This is the task of a DAQ-system. A few of the many tasks of a DAQ are:

- Set the proper bias voltages for the detector to work.
- Configure the detector in a way it is intended to be operated.
- Read the data of the detector and store it in a proper format in which the data can be used for analysis to a data storage device.
- Monitor and provide logging in case of errors during operation.
- Provide an interface for the user to operate the detector.

3.2. The Caribou DAQ-system

The Caribou [16] DAQ-system is designed to be a flexible framework for evaluation of pixel detectors. It consists of open source hardware as well as software.

3.2.1. Hardware components

On the hardware side the Caribou system consist of three different boards, which are operated together in order to perform the operation of the detector prototype. Those boards are shown in fig. 3.2.1, the tasks of them are:

- FPGA board
  The *Xilinx ZC706* FPGA evaluation board is the processing unit of the Caribou hardware setup. It hosts a Zynq-System on Chip (SoC) as well as electronics for communication[1] and clock generation. The SoC itself consists of:
  - A FPGA on which the custom firmware (FW) is running. The FW has to be implemented specifically for each chip which should be operated with Caribou.

[1] The communication with the other boards (in case of the Caribou system) is realized via an I^2C bus.

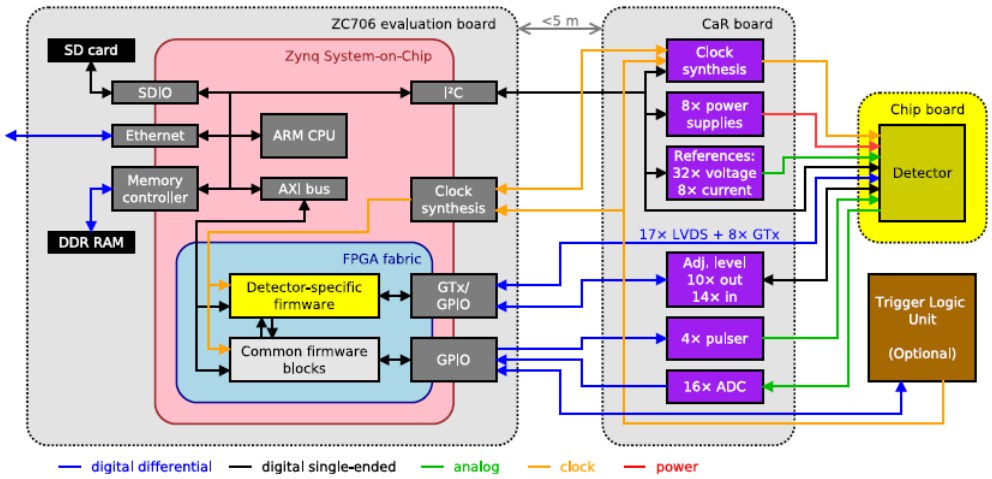


Figure 3.2.1.: The Caribou hardware architecture [16].

  - An ARM CPU on which an embedded Linux operating system and the custom software (SW)-DAQ is running.
  - An AXI bus, which is used for communication between the Operating System (OS) and the FW.

- CaR board
  The Control-and-Readout (CaR)-board hosts several support electronic components for the detectors:
  - Power supplies to set the various bias voltages necessary to operate the chip board.
  - ADCs to measure voltages and currents of the detector.
  - Reference voltages to set for example injection- or threshold voltages for the detector.

- Chip board
  The chip board is a custom made PCB which has to be designed for each detector to be operated with Caribou. It is directly mounted to the CaR-board via a *320-pin SEARAY* connector.

3.2.2. Software framework

On the SW-side there are two layers of abstraction, to simplify the implementation of a detector specific DAQ system, available:

3.2.2.1. Operating system

There is a custom embedded Linux distribution available for Caribou. This distribution is implemented by utilizing the Yocto Project [17]. Specifically the distribution *Poky* [18] was adapted and expanded by a Caribou layer. This OS abstracts the various hardware (HW) components (like the I^2C devices), networking, a file system, ... and therefore simplifies the operation of the Caribou-DAQ.

3.2.2.2. Peary DAQ

Peary [19] is the name of the SW-DAQ, which is an application (written in C++) meant to be operated on the Caribou Linux system. Peary itself is designed to even further abstract the CaR-board-HW[2].

In order for it to work with a custom detector the framework has to be expanded by implementing a new *Peary-device*[3]. The user operates Peary via a Command Line Interface (CLI) and by supplying a custom configuration file.

When Peary got properly extended it can be used to perform various measurements with the detector. Each measurement mode has to be implemented by the user.

3.3. The BDAQ53 DAQ system

The *BDAQ53* DAQ-system is a versatile readout and verification framework specifically designed for the prototype chips of the CERN-*RD53*-collaboration.

3.3.1. Hardware components

As shown in fig. 3.3.1 the BDAQ53 HW-setup consists of the following three boards:

- FPGA board
  The commercially available FPGA board[4] is the processing unit on which the FW is running and performing the direct communication with the detector. This board is mounted to the base board via *triple 168-pin Hirose FX10* connectors.

- BDAQ53 base board
  Besides hosting the FPGA board and providing a connection between the FPGA and the chip board, the base board also supplies some support electronics like a clock generator chip and some signal level translators to be utilized for the detector.

[2]There are for example functions available, which allow the user to set the various supply voltages by specifying the output channel as well as the value of the voltage.
[3]Implementing a Peary-device boils down to implementing a detector specific C++ -class.
[4]The *Mercury+ KX2* evaluation board is in use as the standard FPGA board to be used with BDAQ53.

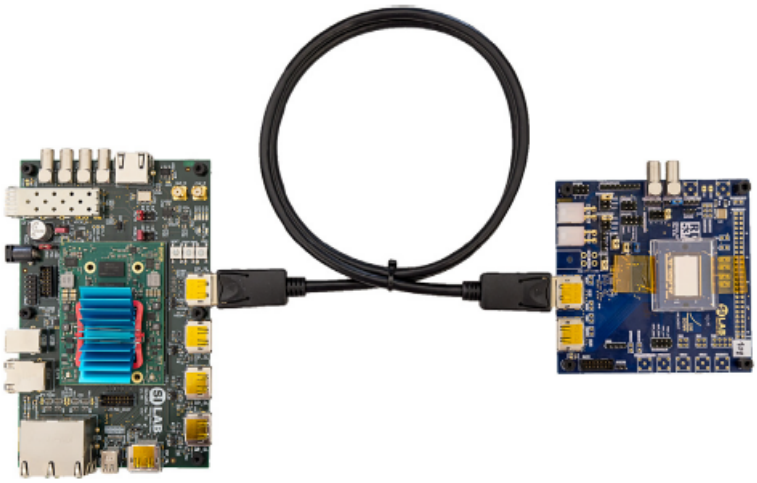


Figure 3.3.1.: The BDAQ53 HW-setup [20]. On the left side the BDAQ53 base board with the mounted FPGA board and on the right side a chip board is shown.

- Chip board
  The chip board, which is connected via a display port cable to the base board, is a custom designed PCB to which the detector ASIC is bonded to. For each detector to be operated with this DAQ a customized board has to be designed.

3.3.2. Software framework

On the SW side, the architecture is shown in fig. 3.3.2, BDAQ53 comes with two abstraction layers.

3.3.2.1. Basil

Basil [21] is the name of the low level modular readout framework utilized by BDAQ53. It consists of:

- Several FPGA FW modules, which can be used to build a custom FW for a detector by connecting those modules together. Basil aims to reduce the effort of developing the FW for a new detector to a minimum. This does apply some constraints on the design of the detector chip itself though. For example the *Basil bus* should be used within the detector.

  Examples for these modules are an interface to the AIDA-TLU, FIFO buffers or a Time to Digital Converter (TDC).

- A Python based control SW which is used as an interface for the next abstraction layer to communicate with the low level FW.

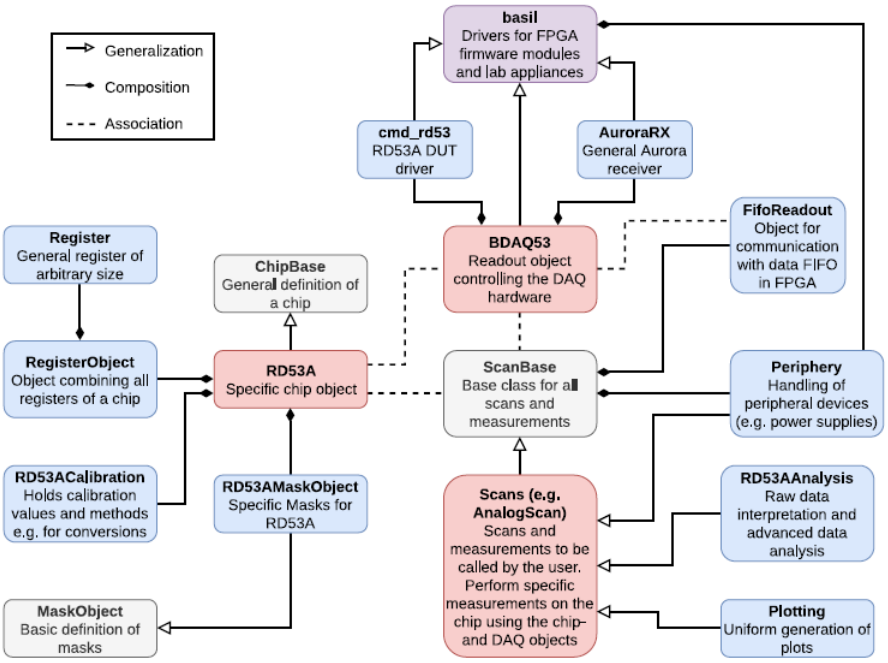


Figure 3.3.2.: The BDAQ53 SW architecture [20].

3.3.2.2. BDAQ53 Python framework

For the top level abstraction layer there is another, python based, framework simply referred to as *BDAQ53* itself. This framework is intended to run on an "ordinary" computer and is connecting to the HW via a TCP connection.

The standard measurements, like injection-, noise- or ToT-measurements, are referred to as *scans* within BDAQ53. These scans are implemented in a very generalized way. In order to operate a new detector type with the framework the user "only" has to supply several configuration files[5] and a python class derived from *ChipBase* and can start performing the most common testing measurement without having to implement them from scratch for every additional detector type. There are also scripts for plotting of the measurements available.

In order for this generalized approach to work, again some constraints are put on how the detector can be operated, or even on the design of the detector itself though.

3.4. Test beam setup

For measurements at a test beam a DAQ-system which supports multiple detector types needs to be set up. This is due to the fact that the data / events of the

[5]The configuration files are in *YAML* format and must contain for example register-names, -addresses, -values, the IP address of the base board, ...

different detectors (DUT, telescope, ...) have to be matched / synchronized in order to be analyzed in the way presented in section 2.3.

3.4.1. Synchronization of the detectors

When a particle traverses the detector setup and accounts for hits in the various planes, this is referred to as an *event*. To be able to perform track reconstruction within analysis the "local events" at each plane have to be merged into "telescope global" events.

3.4.1.1. Synchronization by trigger number

One way of achieving this synchronization is to supply *trigger numbers* to all the detectors.

For this an additional electronics unit, here referred to as Trigger Logic Unit (TLU), needs to be installed.

As shown in fig. 3.4.1 additional / auxiliary detectors need to be installed. In the setup presented here fast scintillators are used as input signal to start event acquisition.

When a particle hits these scintillators within a specified time[6] the TLU increments an internal counter and sends the new trigger number to all detectors in the setup. If the telescope / the DUT detected hits themselve, these hits need to be assigned to this trigger number before further processing happens.

3.4.1.2. Synchronization by timestamp

As the process of distributing the trigger number to all detectors is time consuming, as the trigger number needs to be clocked out by the TLU and therefore at least does take n clock cycles for a n-bit trigger number[7], hence limiting the maximum performance of the system.

This can be avoided by synchronizing the telescope events with a TS. This can be achieved by operating all detectors synchronously, which means operating all of them with the same clock speed.

For this approach to work each detector has to implement an internal counter. When the TLU detects a particle it does not need to send a trigger number to the detectors but produce a timestamped event itself, which can be used as a base for the event building process.

When the detectors measure a particle interaction they have to assign this hit to the current internal TS. As all detectors are operating synchronously, the event building process can be done based on the "telescope global"-TS by matching those TSs with each other.

[6] Typically within a few ns.

[7] In the setup used during this thesis, the trigger number has a length of 15 bit and the TLU operates with a frequency of 40 MHz, the distribution of the trigger number does take $15 \cdot 25\,\text{ns} = 375\,\text{ns}$.

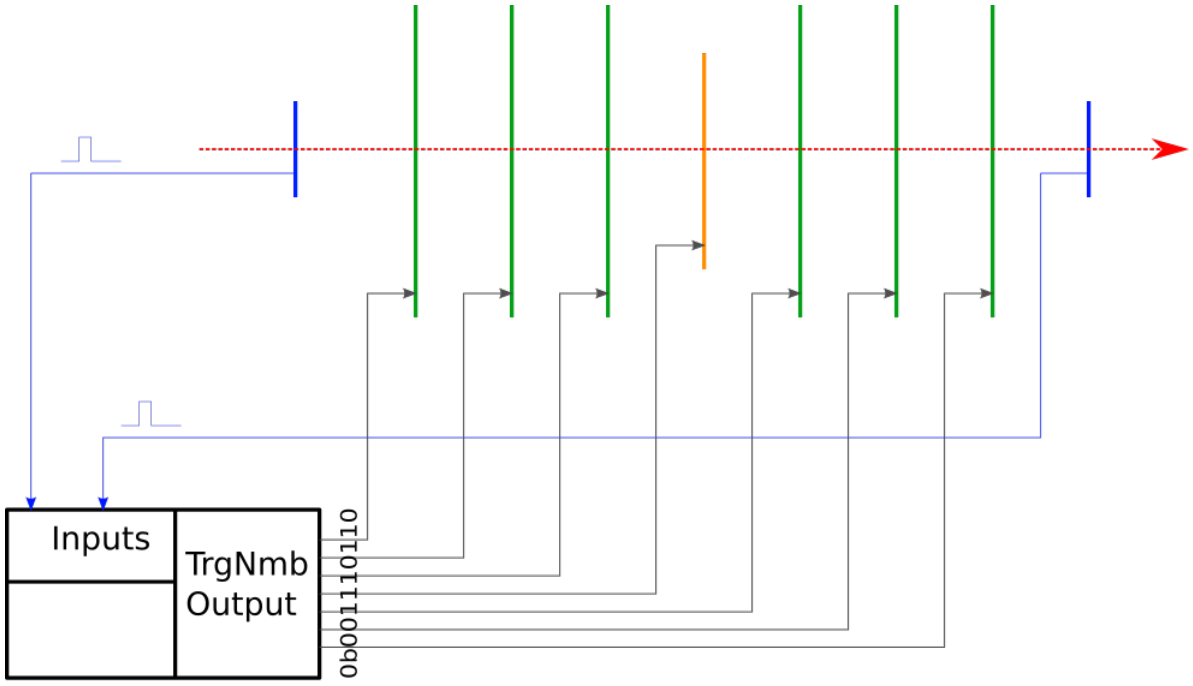


Figure 3.4.1.: Concept of the synchronization by trigger number. The blue planes acount for the auxiliary detectors, the green ones for the telescope planes and the orange one for the DUT. The red line represents the particle beam.

3.4.1.3. The AIDA-TLU

For this thesis the *AIDA2020-TLU* [22], which established itself as the standard TLU for HEP-test beam setups, was utilized for synchronization.

It supports synchronization by trigger number, as presented in section 3.4.1.1, as well as synchronization by TS, as in section 3.4.1.2. The former is referred to as the *EUDET*-mode, as shown in fig. 3.4.2, and the latter as *AIDA*-mode, as in fig. 3.4.3.

In EUDET-mode there is a handshake between the DUT and the TLU. This handshake assures that the DUT is ready to receive the trigger number, so no trigger is lost. In AIDA-mode this handshake is not being performed, as it slows down the setup (no new triggers are being generated until the handshake is done by all detectors in the setup), hence trigger might be lost, which has to be kept in mind as a potential error source.

Multiple electronics lines are utilized for the communication of the TLU and the various DUTs[8], each DUT-connection consists of all of the following lines:

- Trigger
  The trigger line is asserted by the TLU when the scintillators detect a particle. In EUDET-mode after a handshake also the trigger number is clocked out on this line.

- Busy
  The busy line needs to be asserted by the DUT for the handshake sequence. When being held "high" this is interpreted as a detector *veto*, the TLU will pause generating triggers until this line is released by all DUTs.

- Clock
  In EUDET mode the DUTs need to assign their clock to this line, as it will be sampled by the TLU and used for clocking out the trigger number. In AIDA-mode the TLU permanently assigns its internal clock to this line, which should be used as the main clock source by the DUTs.

- Sync / T0
  Each time a data taking *run* is started this line will be raised for one clock cycle. The DUTs need to reset their internal TS counters when this happens. This is crucial for the synchronous AIDA-mode as this procedure ensures that all counters in the telescope setup starting at the same value / time.

3.4.2. EUDAQ

In order to operate the several telescope detectors and the DUT(s) in a test beam, a central control unit for the operator, a decentralized (running on multiple computers) control of the detectors as well as a decentralized data storage is well suited

[8] All primary detectors connected to the TLU are being referred to as DUTs.

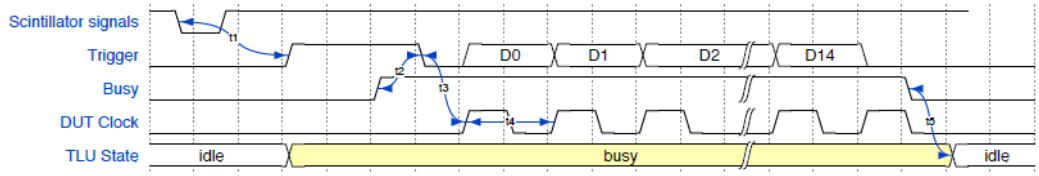


Figure 3.4.2.: Signal lines of the AIDA-TLU operated in asynchronous *EUDET*-mode. Figure adapted from [23]

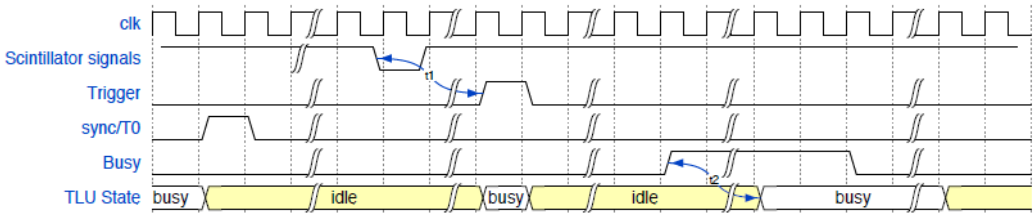


Figure 3.4.3.: Signal lines of the AIDA-TLU operated in synchronous *AIDA*-mode. Figure adapted from [23]

for usability as well as not being limited by the processing power / data storage capability of a single computer.

This is exactly the way the DAQ-system *EUDAQ* [24] is designed, which is the reason this framework was used and expanded during the course this thesis. The various components which are communicating by network connections[9] are shown in fig. 3.4.4. There are several different components in the framework, with their respective purposes being:

- Producer
  A producer is the component which is directly operating a detector. It is performing the following tasks:
  - Applying the Configuration
  - Starting / Stopping the readout
  - Performing the actual readout
  - Build *raw events* from the readout data
  - Sending these raw events to a data collector
- Data Collector
  A Data Collector (DC) is receiving events from one or multiple producer(s). In the latter case it might also perform some synchronization and merging of the received events. Finally it stores the data in a binary format on a data storage device.

[9] If running on a single computer the network connections can be operated on "localhost".

- Monitor
  A monitor is used to display real time information (possibly already some form of pre-analysis) to the operator whilst the measurement is being performed. It receives events from one or multiple data collectors and is a useful tool for debugging purposes and also allows for monitoring of the current data taking run. A monitor typically is not receiving all events from a data collector but only a configurable fraction of them, as otherwise it might be too verbose for the operator to follow/too greedy for processing power.

- Log Collector
  Whenever one of the various EUDAQ-components is raising some kind of log message[10] it is being sent to the log collector which than is displaying the information to the operator and storing it to disk for analysis later on.

- Run Control
  The run control is the central control unit which is used to initialize/configure the various components and starts/stops a run. It is also the (graphical) user interface the operator is working with.

Raw-event A EUDAQ-raw-event is being generated by a producer and being sent to the data collector. It consist of plain/not interpreted data, as processing/interpretation at the producer / data collector level causes a big demand of processing power and would therefore limit the performance.

It typically is stored to a EUDAQ ".raw"-file.

Standard-event In order to make sense of the raw-data for analysis an event-converter needs to be implemented for every detector EUDAQ should be utilized with. This converter is processing the binary data from the raw-event and building an *EUDAQ-standard-event*.

A standard-event consists of the following information:

- Trigger number
- Timestamp
- Standard plane
  For each detector for which data is available in the raw-event. This plane corresponds to the physical detector plane hitmap and should be filled with the corresponding ToT values for each hit pixel.

[10] Available log levels are: DEBUG, INFO, WARNING and ERROR

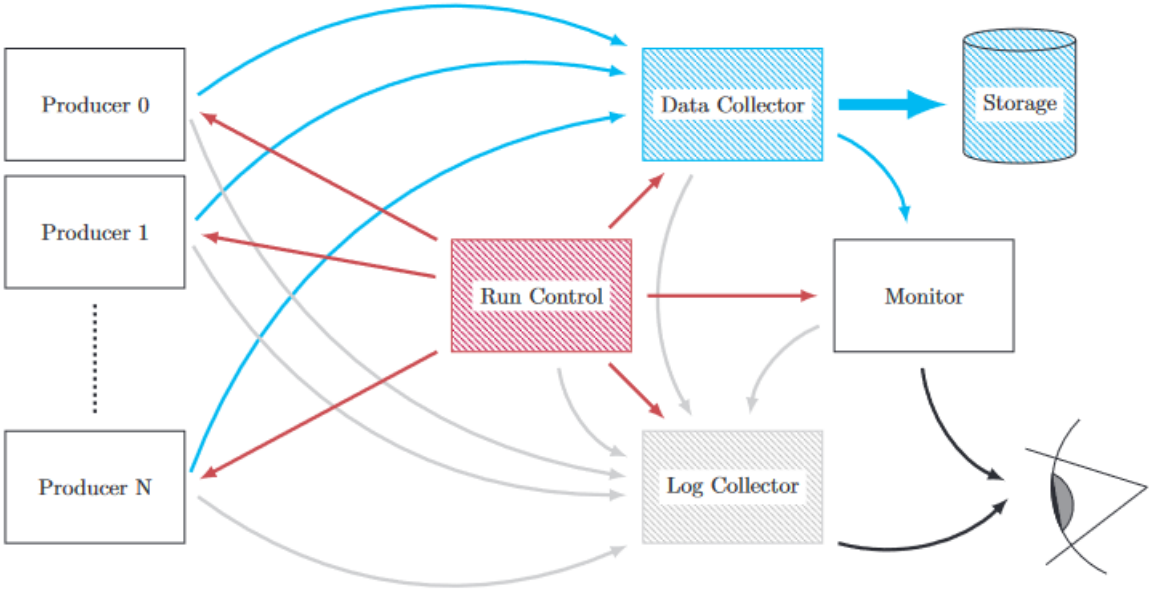


Figure 3.4.4.: The concept of the EUDAQ architecture. The red arrows indicate the control channel, the blue ones the data channels and the grey arrows the log connection. Figure from [24]

4. TJ-Monopix2

4.1. Characteristics of the TJ-Monopix2

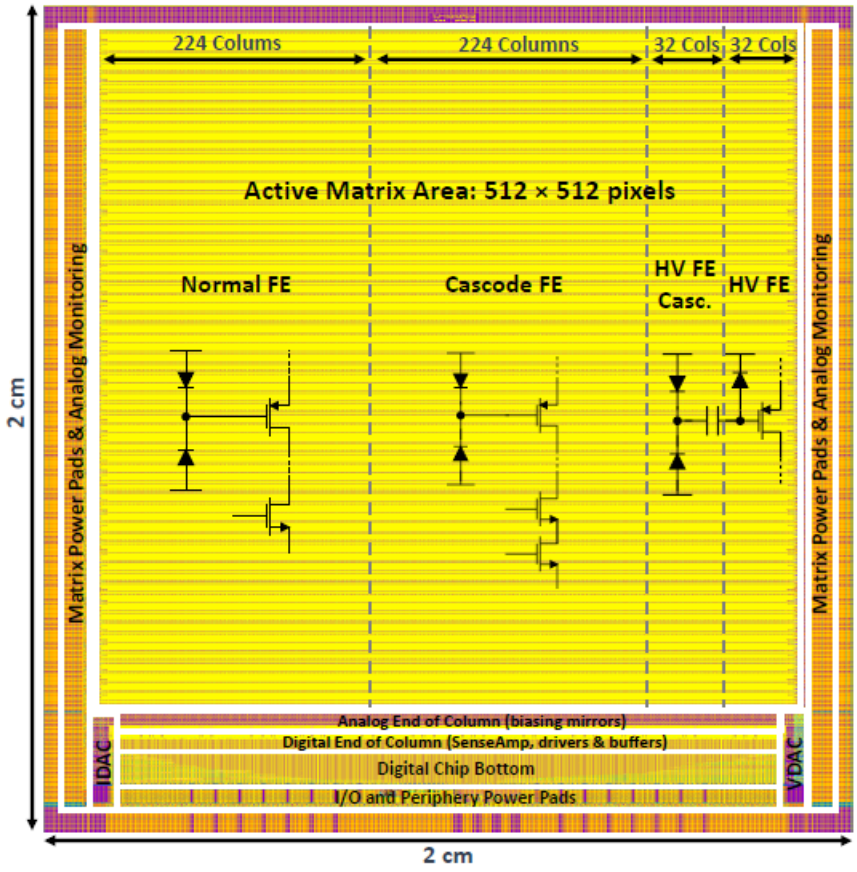


Figure 4.1.1.: The architecture of the TJ-Monopix2 chip [25].

.

The *TJ-Monopix2* [25] was designed by the ATLAS collaboration and is now studied by the Belle-II collaboration for possible upgrades. Its design aims to mitigate some weaknesses of its predecessor the *TJ-Monopix1* and to retain full efficiency even after irradiation. A few measures which have been taken are:

- Reduced detection threshold to $\leq 100e^-$ to detect events with low charge.

- Reduced pixel- / collection-electrode-size to 33.04 μm in order to achieve fast charge collection.

- Adjusted pixel layout to increase charge collection efficiency, especially in the corners.

The whole detector has a size of $2 \times 2\text{cm}^2$ with the active matrix consisting of 512×512 pixels. The chip architecture is shown in fig. 4.1.1. These pixels are separated into two "main" sections, with two "sub" sections each, which consist of different electronic detection circuits, also known as pixel flavors. The different flavors are:

1. The *normal-FE* is a DC-coupled design which supports bias voltages of up to $\approx 10\,\text{V}$.

2. The *cascode-FE* is a sub-flavor of the normal-FE with an additional transistor to increase the preamplifier gain.

3. The *HV-FE* consist of AC-coupled pixels with front-side HV biasing.

4. As for the normal-FE there is also a pixel flavor with an additional preamplifier transistor for the HV design referred to as *HV-FE-Cascode*.

The electronic circuit of the pixels is shown in fig. 4.1.2.

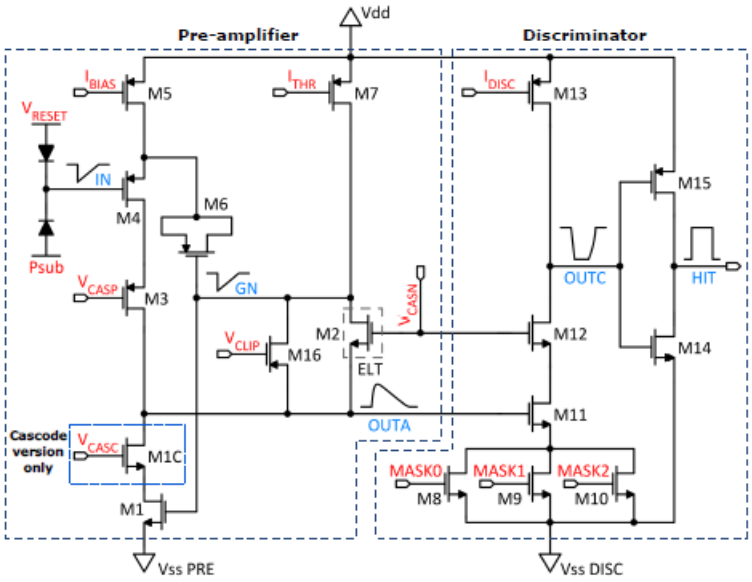


Figure 4.1.2.: The electric FE-circuit of the TJ-Monopix2. In red configurable voltages / currents are shown. Adapted from [25]

The technology used for the TJ-Monopix2 is, at the time of this thesis, one of the top candidates to be used for the foreseen BELLE-II VTX upgrade [2] and therefore its performance had to be evaluated in a test beam.

4.2. DAQ integration

When I joined the BELLE-II VTX-upgrade-collaboration, this chip was already available and (partially) integrated into *BDAQ53*. Scans to record s-curves with injections, scans to verify the digital part or scans to test the chip with a radioactive source were already implemented and working.

The concept of the readout works in the following way: In the FPGA there is a *transmission FIFO* implemented. Whenever a pixel detected a particle interaction data containing the row and column of the pixel as well as the TS-LE and the TS-TE is buffered in the FIFO. Within the BDAQ53-scans this FIFO is readout periodically within an asynchronously running thread.

As the detector should be tested in a test beam at the german electron synchrotron / "Deutsches-Elektronen-Synchrotron" (DESY), my task was to get the DAQ ready for these measurements. In order to do so a connection between EUDAQ and BDAQ53 had to be established, this was done by implementing a specific EUDAQ producer. As stated above in section 3.4 triggering is a very important task for these kind of measurements. Therefore the proper handling of the triggers from the AIDA-TLU also had to be implemented.

4.2.1. AIDA-TLU interface

Parameter	Value
HDMI1_clk	0
DUTMaskMode	lower 2 bits 0b00
DUTMaskModeModifier	lower 2 bits 0b00

Table 2.1.: Configuration of the EUDAQ-producer for the AIDA-TLU for the DUT to be operated in *EUDET*-mode. Here it is assumed the DUT is connected to HDMI port "1". These configuration items have to be included in the EUDAQ configuration file.

For the interface to the AIDA-TLU a Basil module specifically designed for this TLU is utilized within the FW. This modules expects the TLU to operate in *EUDET*-mode, see section 3.4.1.3. In order for the AIDA-TLU to be operated in this mode the configuration as listed in table 2.1 has to be applied to the EUDAQ producer of the TLU.

Whenever the TLU raises a trigger the FPGA will perform a handshake, sample the trigger number and immediately push it to the internal transmission FIFO for the readout by BDAQ53.

4.2.2. EUDAQ integration

4.2.2.1. The Producer

In order to integrate the TJ-Monopix2 into EUDAQ a dedicated EUDAQ producer was developed. Despite EUDAQ being a C++ framework the producer was written in Python[1] as it was also concipated as a BDAQ53-scan (which is implemented in Python) to utilize all the functionality BDAQ53 supports without the need to re-implement them.

The tasks of this producer are:

- Readout hit- and trigger data from the BDAQ53-base-board.
- Assign the hits to the correct trigger number.
- Construct and send EUDAQ events.

Type	Details	
D		
D		
D	EOF	
T		
D	*SOF*	
D		
D		*Event #1*
D		
D		
D	*EOF*	
T		
D	*SOF*	
D	*EOF*	*Event #2*
D	*SOF*	
T		
D	*EOF*	
D	*SOF*	*Event #3*
D		
D		
D	*EOF*	
T		

Table 2.2.: Example of the event construction routine for the data in the FIFO. Type *D* denotes a data word and *T* a trigger word.

Event building Every time hits / triggers were detected and the producer reads out the FIFO multiple 32 bit data words need to be processed. The matching of

[1] For this to work the EUDAQ Python library was used. This library uses *pybind11* [26] to provide Python bindings to existing C++ code.

events to trigger numbers is done in the way laid out in table 2.2. This means the FIFO is split into slices delimited by trigger words. All data words / hits between two trigger words are getting assigned to the trigger number contained in the first trigger word of the current slice.

4.2.2.2. The Event Converter

As mentioned in section 3.4.2 the raw-events from the producer are not suited for analysis, but require interpretation beforehand. This interpretation is done by a *EUDAQ-standard-event-converter*.

The task of this converter is to extract the information of the hit pixels within the data from the producer.

Type	Identifier
Trigger	0b10000
Data	0b01000
Time**S**tamp	0b01001

Table 2.3.: Word type identifiers of the TJ-Monopix2 data.

Data format All 32 bit words from the FPGA FIFO contain a header in the upper 5 bits. The matching of the header to the type of the word is listed in table 2.3.

The remaining lower 27 bits contain data depending on the type:

- **T**-word
  The lower 15 bits of a **T**rigger-word contain the trigger number from the TLU-event.

- **TS**-word
  A **T**ime**S**tamp word contains 27 bits of a TJ-Monopix2 internal TS. This information was not utilized by the DAQ system, as the rest of the test beam setup does not support a TS-mechanism.

- **D**-word
  The remaining 27 bit of a **D**ata word are split into three 9 bit *slices*. At least two D-words are needed to contain the information of one hit pixel. Each time the chip detects one or more hits during the same readout cycle, internally a *frame* is created. This frame is denoted by a Start of Frame (SOF) and an End of Frame (EOF) and contains at least one hit. If there is no data left in a frame, but not all slices of a data word were needed, the remaining slices are filled with an *idle* pattern. The value of these "special" slices are listed in table 2.4.

  All the other slices contain information regarding a hit pixel. For one hit exactly 4 slices are used. In which position which data is stored is shown in

table 2.5. At this stage the timestamps are still gray encoded and have to get decoded for further processing.

Type	Identifier
SOF	0x1BC
EOF	0x17C
Idle	0x13C

Table 2.4.: Identifiers of the "special" 9-bit-slices a data word might contain.

Slice #	Bits								
	8	7	6	5	4	3	2	1	0
1	-	Col: 8 MSB							
2	-	TS-LE: 7 LSB							TS-TE: 1 MSB
3	-	TS-TE: 6 LSB						Col: 1 LSB	Row: 1 MSB
4	-	Row: 8 LSB							

Table 2.5.: The various information about one hit stored in four 9-bit-slices. n LSB/MSB means that the shown n bits are used for n LSB/MSB bits in the corresponding interpreted piece of data.

By interpreting the data in the above presented way the event converter builds a *EUDAQ-standard-event* which can then be used by the analysis framework *Corryvreckan* [27].

4.3. Testbeam measurements at DESY

The measurements leading to the results in the following section have been performed at the Test Beam Facility at DESY Hamburg (Germany), a member of the Helmholtz Association (HGF), in July-2022.

The TJ-Monopix2 was placed in the middle of a *EUDET-type* telescope consisting of six *MIMOSA26* planes [28]. A schematic of the full measurement setup is shown in fig. 4.3.1.

The "rough" detector geometry is shown in table 3.6. This geometry was measured by hand and therefore is not too accurate, as it was not possible to measure any more precise. For a proper analysis a well aligned setup is needed.

4.3.0.1. Alignment

For the alignment procedure with *Corryvreckan* the following steps have been performed:

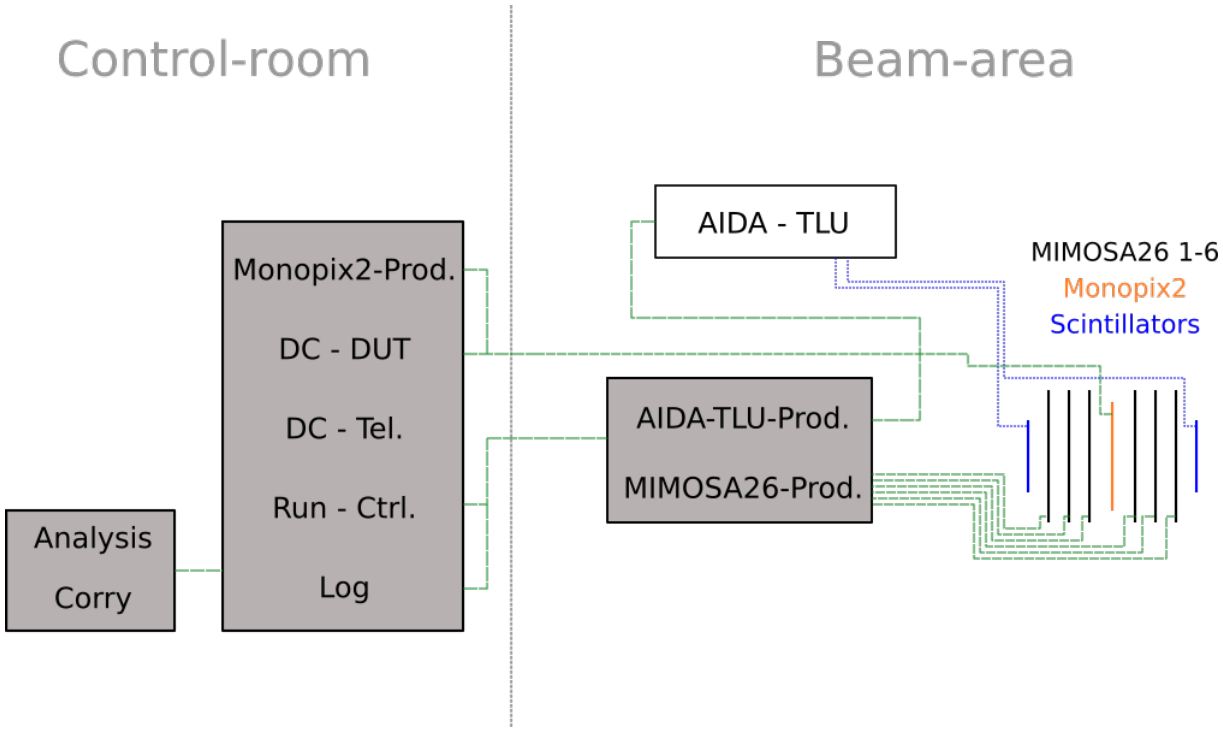


Figure 4.3.1.: The detector and DAQ setup used for the testbeam at DESY. The grey boxes represent computers, the green dashed lines data/network connections and the blue doted lines electric signals. A more detailed view of the detector planes alone is shown in fig. 2.3.1.

Detector	Position z [mm]	Orientation z [°]
MIMOSA26_0	0	0
MIMOSA26_1	35	0
MIMOSA26_2	70	0
Monopix2	98.5	180
MIMOSA26_3	119	0
MIMOSA26_4	153	0
MIMOSA26_5	188	0

Table 3.6.: The "rough" detector geometry used for the measurements presented in section 4.3. The z-axis is considered as the axis on which the beam traverses. The orientation is given in terms of an Euler angle.

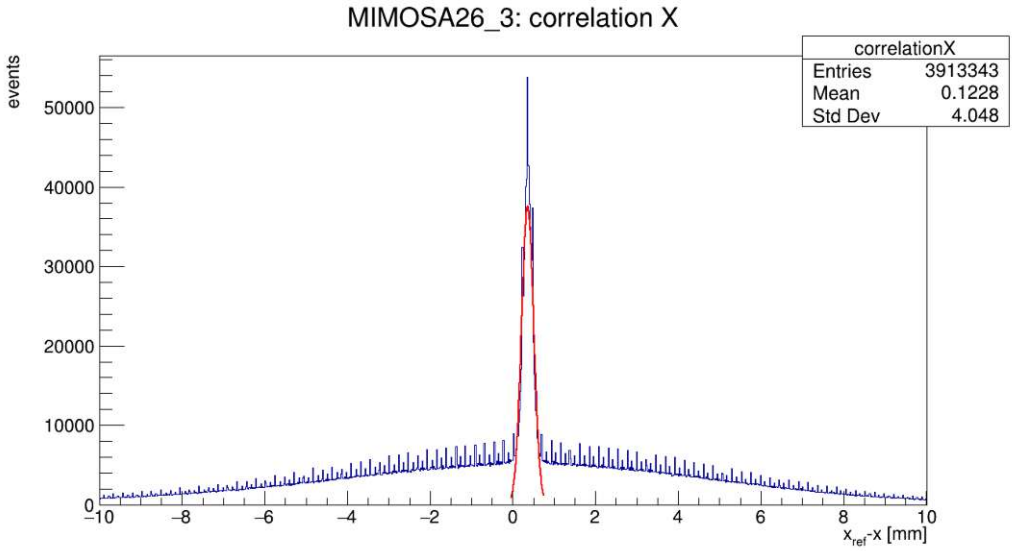


Figure 4.3.2.: Example of the prealignment procedure with the detector "MIMOSA26_3". The red line indicates the fit to the Gaussian function.

1. Prealignment of the telescope
   In a first alignment step the telescope gets pre-aligned with the module *[Prealignment]*. This module processes the correlations of a detector plane corresponding to the reference plane[2]. In an unaligned state the peak of the correlation might be shifted by $\mathcal{O}(mm)$ from 0. As shown in fig. 4.3.2 a Gaussian function is fitted to this peak. In this alignment step the planes get shifted by the difference of the center of the Gauss to 0 in the x-y-plane. After the pre-alignment routine the correlations are centered at 0.

2. Precise alignment of the telescope
   In a second alignment step the telescope is being aligned precisely by using the module *[AlignmentTrackChi2]*. This modules requires track reconstruction to be performed beforehand. It then evaluates the position of the tracks intersecting the telescope planes and compares those with the position of clusters on the planes. A χ^2-test [29] is performed in order to optimize the position of the clusters by minimizing the residuals and shifting the detector planes correspondingl in the x-y-plane.

   As track reconstruction itself depends on the position of the clusters, this alignment is performed iteratively.

   After this stage the telescope is aligned and the so calculated geometry is "frozen" for the DUT alignment steps.

[2]For the reference "MIMOSA26_0" was chosen, as it is the plane in the setup closest to the entry point of the beam.

3. Prealignment of the DUT
   The same procedure as for the pre-alignment of the telescope is again performed for the DUT alone. As the telescope geometry is already aligned precisely fewer degrees of freedom have to be considered, which allows for a more accurate pre-alignment.

4. Precise alignment of the DUT
   For the precise alignment of the DUT the module *[AlignmentDUTResidual]* is utilized. The procedure follows the same scheme as within the *Precise alignment of the telescope* step from above.

At this stage all planes are properly aligned and the actual analysis can be performed.

4.3.1. Analysis

For the following section several runs recorded at the testbeam at DESY have been analyzed with *Corryvreckan* and the results are presented here.

4.3.1.1. Hitmaps

The hitmaps for a run with a "small" collimator[3] are shown in fig. 4.3.3.

The different appearances of the Mimosa26 planes and the Monopix2 plane are a result of the different pixel pitch and the different sizes of the detectors. Due to the size of the collimator the beam spot is clearly visible on all detector planes[4]. For the Monopix2 several columns, as shown in fig. 4.3.3d, have been deactivated, as during the here evaluated run only the "normal" front-end was tested.

4.3.1.2. Cluster Size

The measured cluster size is plotted in 4.3.4. It shows that an average of $\approx$ 1.843 Pixel fired at each particle hit.

4.3.1.3. Residuals

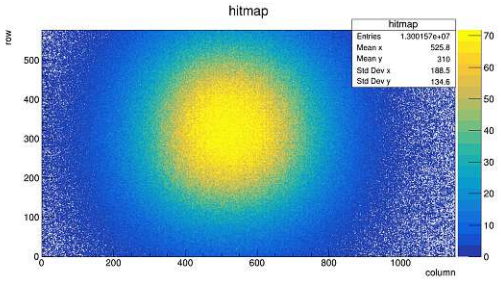
A comparison between the theoretical value of the detector resolution, according to eq. (2.2.8), which evaluates to $\sigma_{x/y} = \frac{33.04\,\mu\text{m}}{\sqrt{12}} \approx 9.538\,\mu\text{m}$, and the measured values of $\approx 9.1\,\mu$m shown in fig. 4.3.5 shows a good agreement of experiment and theory.

4.3.1.4. Efficiency

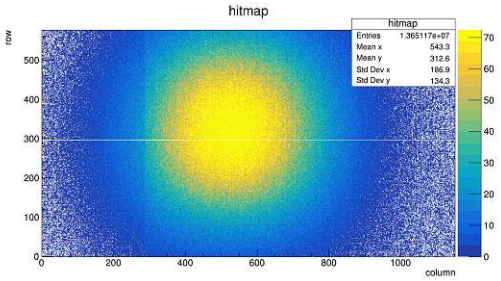
To investigate the detection efficiency of the sensor the *Corryvreckan* module *[AnalysisEfficiency]* was used. The efficiency maps for the "normal" frontend are shown

[3] The "small" collimator has a size of $\varnothing \approx 5\,$mm.

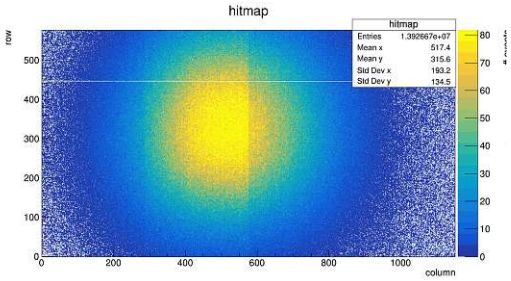
[4] The MIMOSA26 has a size of $\approx 21.2\,\text{mm} \times 10.6\,\text{mm}$ and the TJ-Monopix2 of $\approx 16.9\,\text{mm} \times 16.9\,\text{mm}$.



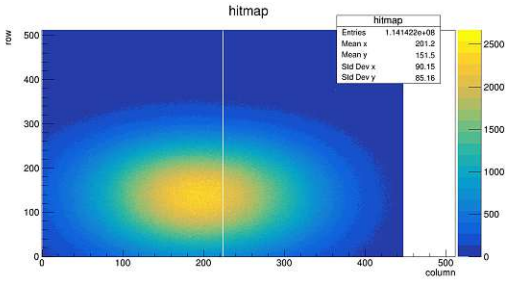
(a) Mimosa26_0



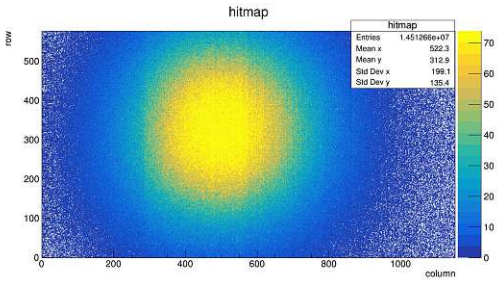
(b) Mimosa26_1



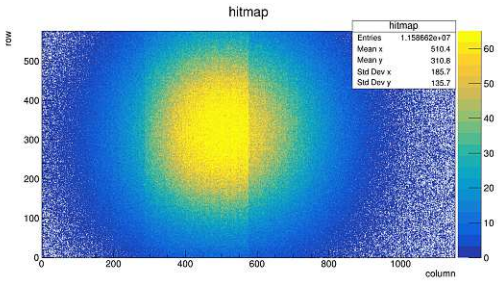
(c) Mimosa26_2



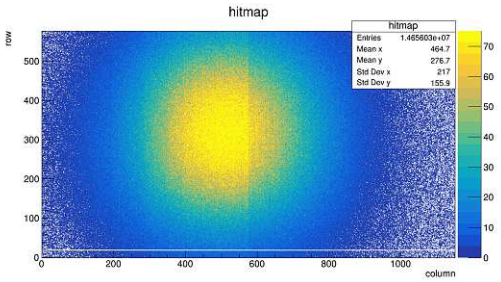
(d) Monopix2. Columns $\in (448, 512)$ have been deactivated.



(e) Mimosa26_3



(f) Mimosa26_4



(g) Mimosa26_5

Figure 4.3.3.: The hitmaps of all detector planes, taken at the test beam at DESY.

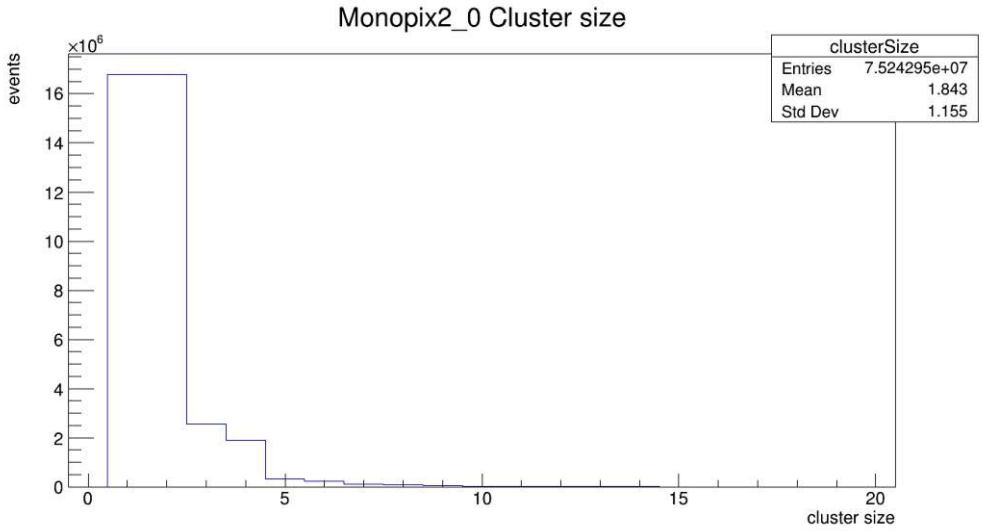


Figure 4.3.4.: Cluster size of the TJ-Monopix2.

DUT	masked columns	total tracks	matched tracks	ϵ[%]
W14R12	-	964649	957539	99.263($\pm$0.009)
W14R12	$\{224\} \cup (448, 512)$	948405	947613	99.917 ($\pm$0.003)
W5R9	-	739812	728472	98.467 ($\pm$0.014)
W5R9	$\{224\} \cup (448, 512)$	727812	721484	99.131($\pm$0.011)

Table 3.7.: Chip efficiency values for different DUTs and applied masks of the "normal" frontend.

in fig. 4.3.6. The upper part of the maps are empty because the telescope planes, due to the different geometry of the detectors, were placed in a way there were no particles causing tracks at these positions.

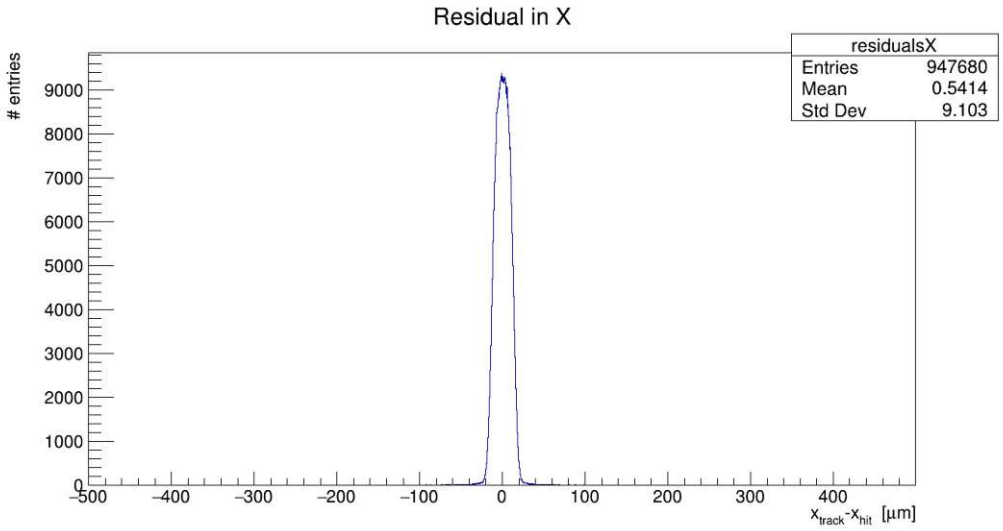
In fig. 4.3.6a it can be seen that there is one column (column #224) which is a lot less efficient than the rest of the detector. Also at the border to the HV-frontend (column #448) the efficiency decreases.

In order to evaluate the efficiency without these columns, they got deactivated/masked for the analysis, as shown in fig. 4.3.6b. This measure resulted in a efficiency close to 100%. The exact chip efficiency values are listed in table 3.7.

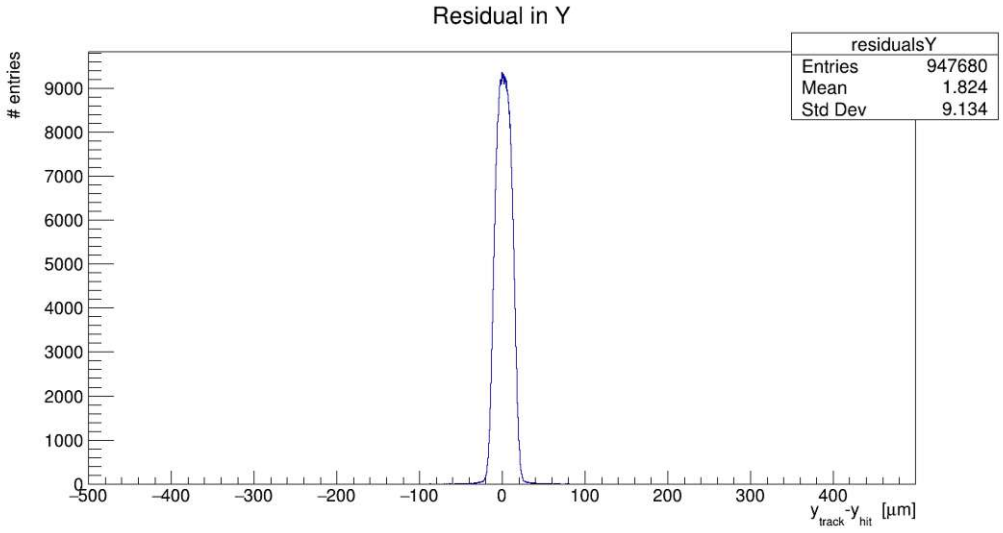
The in pixel efficiency[5] in fig. 4.3.6c shows a slightly decreased efficiency in the edges of the pixel.

The efficiency was also evaluated for the HV-frontend. The efficiency maps are shown in fig. 4.3.7. These plots and the values in table 3.8 show a similarly good efficiency as for the normal-frontend although the HV frontend requires a bias voltage of $\approx -10\,\mathrm{V}$ to be able to compete with the normal-frontend, which got

[5]The in-pixel efficiency is the efficiency for different positions in a single pixel.

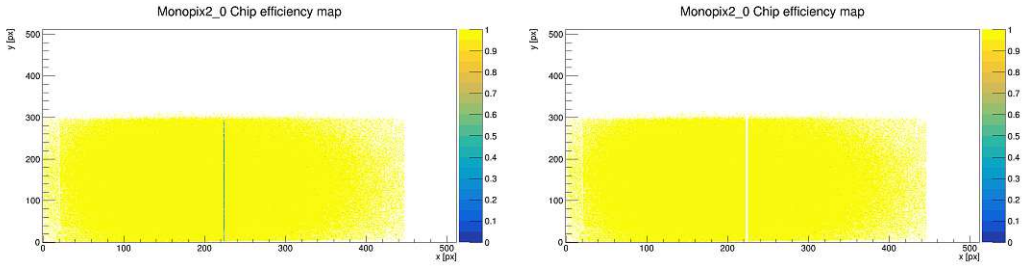


(a) Residual in x-direction.

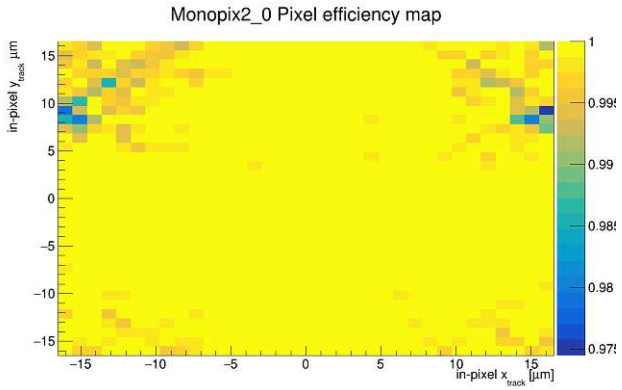


(b) Residual in y-direction.

Figure 4.3.5.: Residuals of the TJ-Monopix2.



(a) No pixels masked **(b)** Columns 224 and $(448, 512)$ masked



(c) In-pixel efficiency

Figure 4.3.6.: Chip efficiency maps of the normal-FE.

bias voltage [V]	total tracks	matched tracks	ϵ[%]
-5	14157	12828	90.612($\pm$0.248)
-6	14601	13894	95.158($\pm$0.182)
-7	14288	13991	97.921($\pm$0.123)
-8	13861	13696	98.810($\pm$0.099)
-9	14497	14334	98.876($\pm$0.089)
-10	14442	14317	99.135($\pm$0.081)

Table 3.8.: Efficiency values for different supply voltages for the "HV" frontend.

biased with only $\approx -3\,\mathrm{V}$.

4.4. Simulation

In order to better understand the TJ-Monopix2 simulations with the *Allpix*^{2} framework [15] have been performed. To be able to validate the simulation results, the simulation was setup to resemble the circumstances of the test beam at DESY. Therefore not only the Monopix2 but the whole telescope with the 6 MIMOSA26 planes was simulated. The detector setup is sketched in fig. 2.3.1. The beam itself was simulated as listed in table 4.9.

Parameter	Value
Energy	4 GeV
Particle type	e^-
Beam size	3 mm
Beam shape	Gaussian

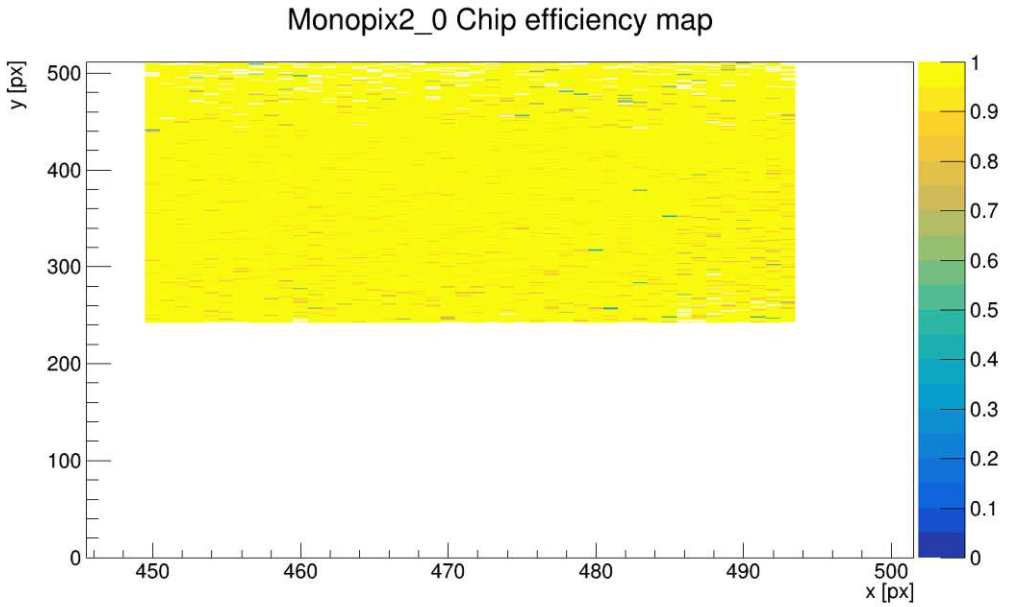
Table 4.9.: Beam parameters of the simulation for the TJ-Monopix2. As the beam shape was set to Gaussian the beam size corresponds to the FWHM of the Gaussian function.

For the simulation results to reproduce the actual detector performance as best as possible and therefore to resemble the experimentally recorded results, the electric field, due to the bias voltage, inside the detector is crucial.

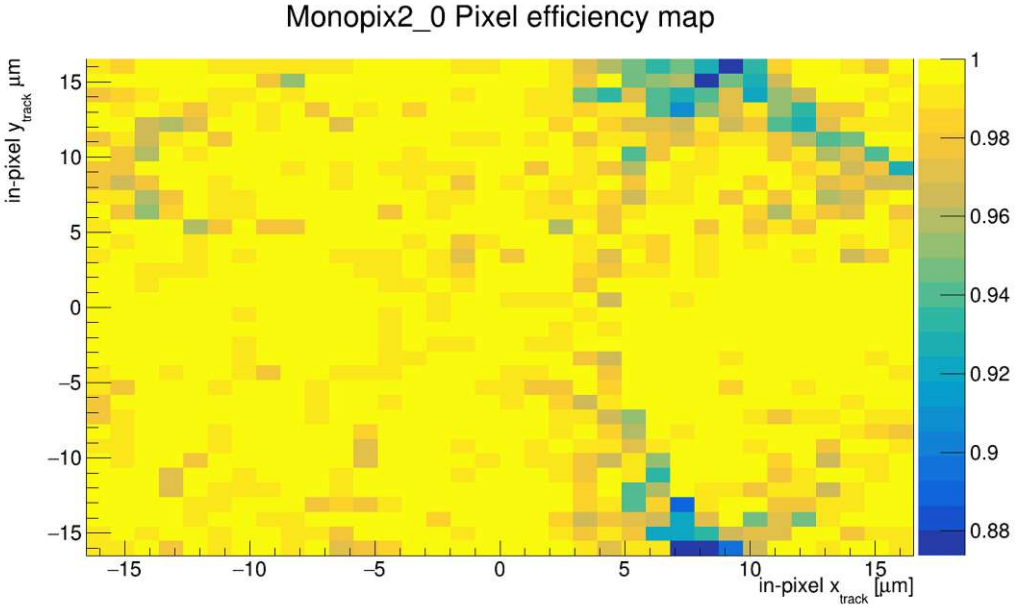
The simplest approximation of the E-field is a linear field, oriented purely in z-direction[6]. This linear model can be justified by approximating the P-N-junction as a plate capacitor and the depleted region in between as the dielectric. The maximum magnitude of the E-field in between the "plates" therefore is calculated by the following equation [9].

$$E_z = \frac{2U_{\mathrm{bias}}}{d} \tag{4.4.1}$$

[6]The z-axes is considered as the one oriented along the smallest spatial dimension (the thickness) of the detector.



(a) Chip efficiency of the HV-frontend. Plot range adjusted to show only active part of the matrix.



(b) In-pixel efficiency of the HV-frontend.

Figure 4.3.7.: Efficiency maps of the HV-frontend with $-10\,\mathrm{V}$ bias voltage.

in which d accounts for the depletion depth and therefore, in our approximation, for the distance between the "capacitor plates".

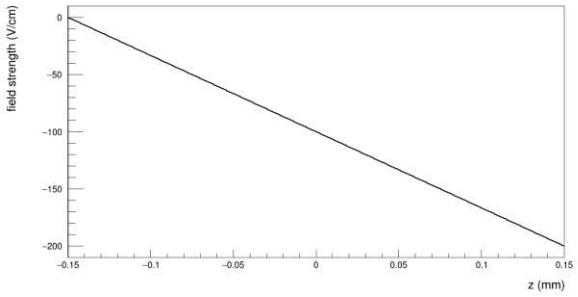
Allpix^2 would also support using precise E-fields from mesh simulation. Such E-fields require TCAD studies [30] of the detector though and would have exceeded the scope of this thesis. The following simulations have been limited to the linear field approximation.

4.4.1. Studies of a fully depleted detector

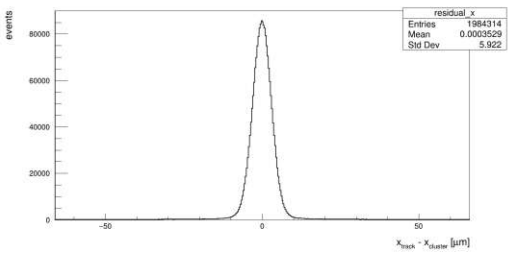
As for most of the runs at DESY the bias voltage was set to $-3\,\mathrm{V}$ and a fully depleted sensor was assumed (for a first study) the maximum magnitude for E_z would be $2 \times \frac{-3\,\mathrm{V}}{300\,\mu\mathrm{m}} = -200\,\mathrm{V/cm}$, which is in agreement with the simulation shown in the 4.4.1a.

For the simulation results presented here the cluster size (in fig. 4.4.1c) and the residuals in x-direction (in fig. 4.4.1b) have been chosen as they can be compared best to the results from the measurements at DESY.

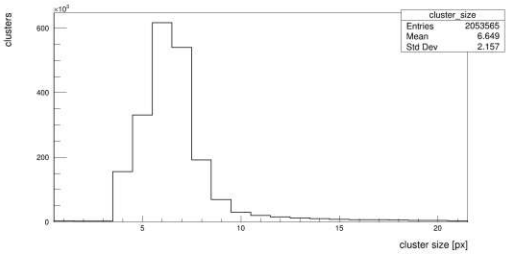
Comparing the simulation results with the experimental ones in section 4.3.1 shows a big difference.



(a) E-field in z-direction



(b) Residual in x-direction



(c) Cluster size

Figure 4.4.1.: Results of the simulation of a fully depleted TJ-Monopix2. A bias voltage of $-3\,\mathrm{V}$ and a depletion depth of the full $300\,\mu\mathrm{m}$ was simulated.

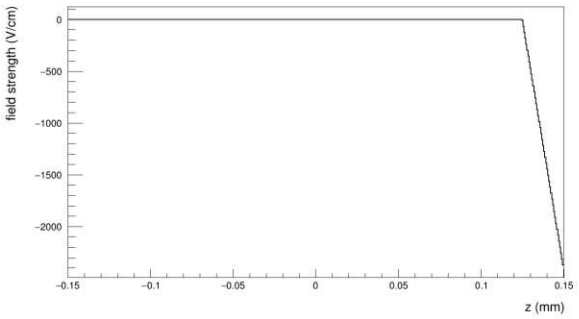
4.4.2. Studies of a partially depleted detector

To reduce the difference between simulation and experimental data the assumption of a fully depleted sensor was dropped. A more realistic scenario, one in which the TJ-Monopix2 is only partially depleted, got investigated.

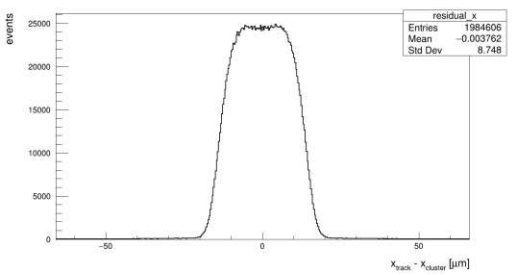
For this study a depletion depth of 25 μm was assumed. The maximum magnitude of E_z according to eq. (4.4.1) evaluates to $E_z = -2.4\,\mathrm{kV/cm}$. In the linear approximation the E-field outside the depleted region should be 0. This behaviour is shown by the simulation in fig. 4.4.2a.

The simulation results for the residual and the cluster size are plotted in 4.4.2b and 4.4.2c respectively.

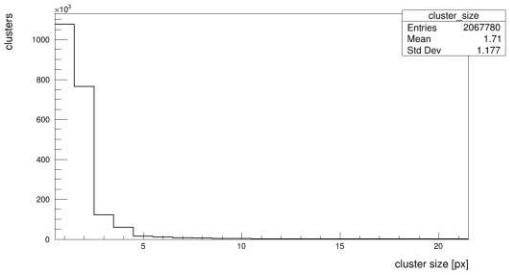
These results resemble the recorded data in section 4.3.1 already much better.



(a) E-field in z-direction



(b) Residual in x-direction



(c) Cluster size

Figure 4.4.2.: Results of the simulation of a partially depleted TJ-Monopix2. A bias voltage of $-3\,\mathrm{V}$ and a depletion depth of 25 μm was simulated.

4.4.3. Possible improvements

In order to further improve the results of the simulation first of all a TCAD simulation should be set up to evaluate the actual electric field and replace the linear approximation by it. By investigating the difference the E-field has on the detector performance, as shown above, this seems to be the most crucial point for an in depth simulation of the sensor.

There are a variety of other simulation parameters which one could try to improve though. For example, as the design of the TJ-Monopix2 aimed for a low detection threshold and low electronics noise [25] this behavior should also be replicated as good as possible. This can be done with parameters of the *[Default-Digitizer]* module.

To get values, which emulate the TJ-Monopix2 best, for this parameters a deeper understanding of the chip is necessary than was available at the time of this thesis. For the simulations presented here the default values provided by the developers of Allpix^2 have been used.

5. RD50-MPW3

5.1. Characteristics of the RD50-MPW3 Chip

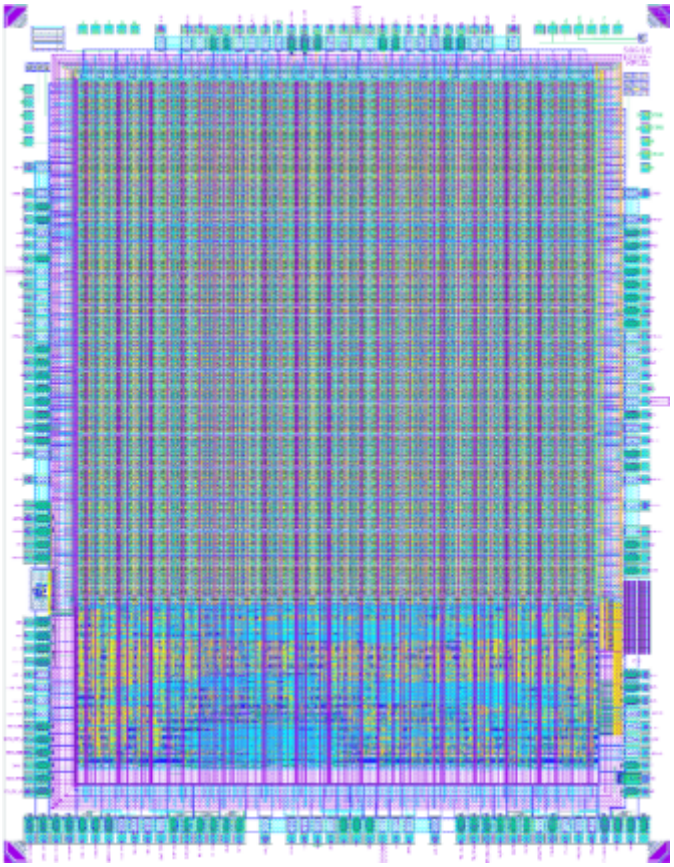


Figure 5.1.1.: Layout of the RD50-MPW3. [31]

The *RD50-MPW3* is the third DMAPS detector of the *RD50-MPW* series designed by the CERN RD50 collaboration. All sensors got fabricated by LFoundry S.r.l.[1] in a 150 nm HV-CMOS process and are designed for harsh radiation environments with high spatial granularity.

The *RD50-MPW1* suffered from a high leakage current in the sensing diodes,

[1] LFoundly S.r.l., Via Antonio Pacinotti 7, Avezzano AQ 67051, Italy, `http://www.lfoundry.com`.

voltage drops across the pixel matrix and crosstalk noise between the lines that carry digital signals [32].

The goal of the *RD50-MPW2* was to mitigate these problems and therefore the design focused entirely on the analog part. With several measures like a guard ring frame at the edge of the chip the MPW2 succeeded in improving the design problems of the MPW1. Though the MPW2 was highly successful some limitations due to design restricted the evaluation of this detector. The active matrix consists only of 8×8 pixel and due to the minimalistic digital periphery only 1 pixel at a time can be readout.

For the third generation, the *RD50-MPW3*, the analog part of the MPW2 was adopted and the chip was expanded by a digital periphery to mitigate the limitations of its predecessor [32]. The active matrix consist of 64×64 pixels with a pitch of $62\,\mu\text{m} \times 62\,\mu\text{m}$ organized in 32 double columns. A schematic view of the chip is shown in 5.1.1.

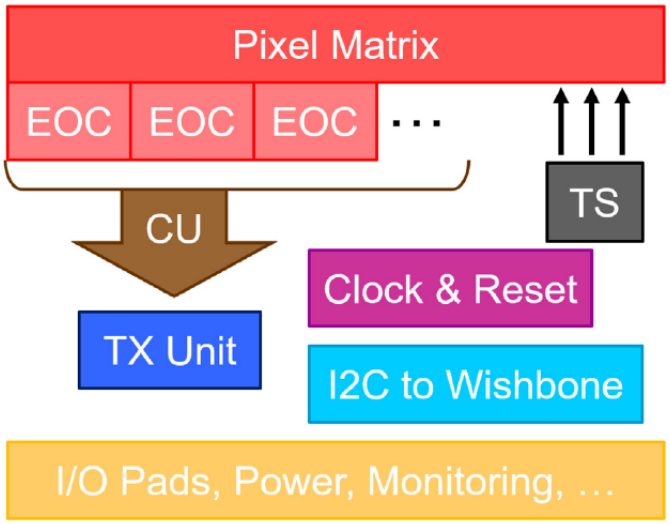


Figure 5.1.2.: The digital periphery of the RD50-MPW3. [31]

The digital periphery, as shown in fig. 5.1.2, consist of an End Of Column (EOC) circuit for each double column which reads out the matrix and stores the data in a buffer. The Control Unit (CU) processes these buffers and pushes the data into a transmission FIFO. This FIFO gets processed by the transmission unit (TX Unit) which adds SOF and EOF words, serializes the data and sends it via a serial line to the off-chip electronics. The timestamp generator (TS) provides an 8 bit timestamp to all pixels, with which the pixel hits are assigned to a TS-LE and a TS-TE. Moreover there is an I^2C to *Wishbone*[2] module converting the external I^2C signals[3] to the internal Wishbone signals.

[2]The Wishbone bus is a open source hardware bus system designed for SoC systems. It is designed as an IP core for semiconductors. The detailed specifications can be found at [33].
[3]I^2C signals are used to configure the detector.

The MPW3 is designed to be operated with an external 640 MHz clock. This frequency defines the maximum possible readout rate. Several tests of the chip showed that due to a too slow discharge of the TS-signal-line inside the chip the values for TS-LE and TS-TE did not behave as expected[4]. This problem was solved by slowing the frequency down to 320 MHz. Thereby the corresponding signal line gained enough time to discharge itself and the TS-LE and TS-TE appeared as proper / changing values in the data readout.

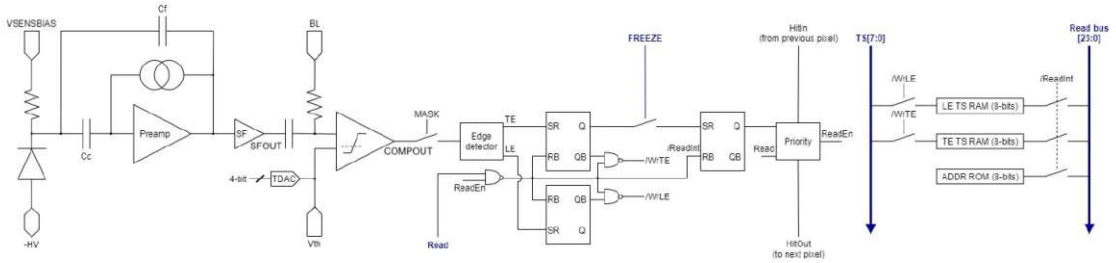


Figure 5.1.3.: Block diagram of in-pixel electronics.

According to the schematic of the in-pixel electronics in fig. 5.1.3 a pixel detects a hit when the AC coupled voltage after the preamplifier output V_{preamp} satisfies the following condition:

$$V_{\text{preamp}} + V_{\text{bl}} \geq V_{\text{th}} + V_{\text{tdac}} \tag{5.1.1}$$

The baseline voltage V_{bl} as well as the threshold voltage V_{th} have to be applied to the chip via external Digital to Analog Converter (DAC)s and thereby act as global parameters for the whole pixel matrix. The Trim-DAC voltage V_{tdac} is generated internally and can be set for each pixel as a 4 bit value[5], which allows to fine adjust the threshold for each pixel.

Furthermore the digitized signal of each pixel can be disabled by masking it. This is also a configuration option for each pixel. A full list of all configuration options per pixel can be found in table 1.1.

5.2. DAQ integration

To evaluate the performance of the *RD50-MPW3* laboratory as well as test beam measurements needed to be performed. In order to implement these measurements the detector had to be integrated into several DAQ-systems.

[4] Those timestamps appeared "frozen" at a fixed value in the digital data readout.

[5] A TrimDAC value < 7 accounts for a negative and one > 7 accounts for a positive shift of the threshold setting. 7 has the least effect and therefore is a "neutral" setting, while the values 0 / 15 account for the biggest shift in negative / positive direction.

Register	Type	Function
EnInj	Boolean	Enable injection
EnSFout	Boolean	Enable source-follower output
TrimDAC	4 bit	fine adjust threshold voltage
Mask	Boolean	Disable data output
EnHB	Boolean	Enable hit-bus output

Table 1.1.: Configuration options for each single pixel. Function description of boolean options correspond to value being set to "true".

5.2.1. Integration into Caribou

For the integration into Caribou a dedicated *Caribou-device*, simply named *RD50_MPW3*, was developed. This device parses a MPW3 specific configuration file and takes care of :

- Setting the bias- and reference voltages on the CaR-board to the configured values.
- Configuring the pixel matrix of the MPW3 chip itself via an I^2C interface.
- Perform laboratory measurements and store them locally on the SD-card of the FPGA-board. The following measurement modes got implemented:
  - Open-shutter
    During an *open-shutter* measurement all the data from the detector is readout for a specified time and a specified configuration. The accumulated data is stored as a hitmap in *CSV*-format. This modus is designated for measurements with a radioactive source.
  - S-Curve
    When performing a *s-curve* measurement an injection voltage will be applied to a pixel[6] for a specified time and the number of hits is recorded. The injection voltage is increased in configurable steps from a minimum to a maximum value. When plotting the number of hits versus the injection voltage a characteristic "S-like" shape should be observed.
  - Noise-Trim
    The goal of the *noise-trim* mode is to calibrate the pixels so they don't detect hits when there are no incident particles, but only electronics noise. This can be done by adjusting the threshold of the comparator by setting the trim-DAC (TDAC)-values for the pixels. The point at which the detector is most sensitive to incident particles, is the point at which a little bit of noise already accounts for hits. This mode aims to

[6] The design of the RD50-MPW3 limits injections to only be applied to one pixel at a time. An s-curve measurements therefore requires a lot of time when the whole matrix should be scanned.

find this set point by scanning the TDAC-values for each pixel. At the ideal value one TDAC-step beneath would account for noise hits while the set point does not produce noise.

5.2.2. User interface

In order to simplify the configuration and operation of the MPW3 a Graphical User Interface (GUI) was developed. A screenshot of it is shown in fig. 5.2.1. This GUI supports:

- The generation of a configuration file which can be parsed by Peary. This file specifies the various bias voltages, the configuration of each pixel and some miscellaneous items like the I^2C-address of the detector, the configuration of the clock, ...
- The execution the various functions implemented in the Peary-device without having to use the CLI of Peary.
- To monitor EUDAQ events. The GUI contains a EUDAQ-monitor which can be connected to the run-control and therefore serve as an online monitor during a data taking session with EUDAQ.

Figure 5.2.1.: The GUI developed to configure and operate the RD50-MPW3.

5.2.3. Integration into EUDAQ

In order to test the RD50-MPW3 at a testbeam at CERN, the detector was fully integrated into the EUDAQ framework. As the MPW3 can be read out with a rate of 320 MHz, a "slow" and a "fast" readout path have been implemented.

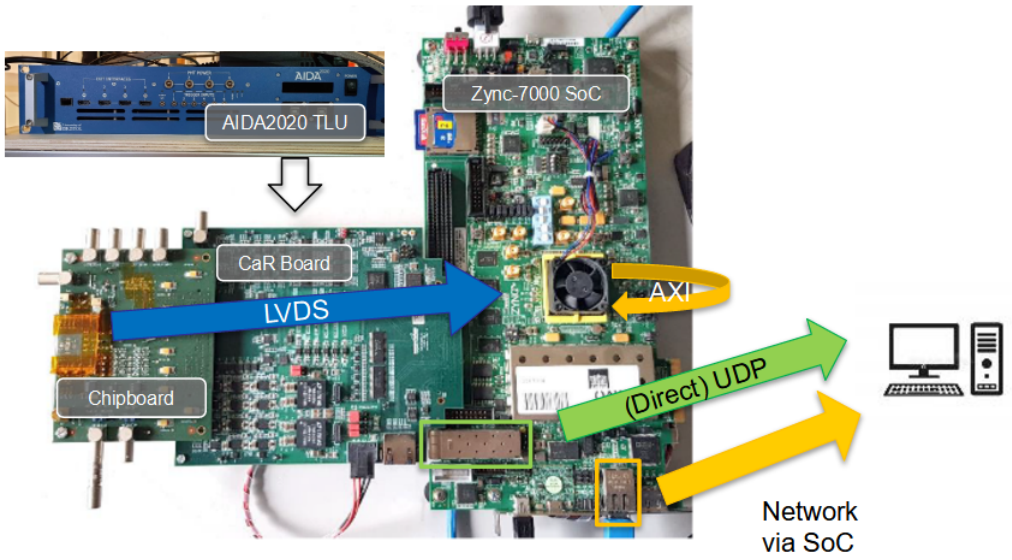


Figure 5.2.2.: Setup of the RD50-MPW3 for test beam measurements. Picture shows *RD50-MPW2* chipboard. Figure adapted from [31]

The "slow" path uses the standard EUDAQ architecture with a producer (directly running on the Caribou system) which is sending events to a standard EUDAQ data-collector[7]. This data channel is limited to an event rate of $\mathcal{O}(10\,\text{kHz})$ and is further referred to as *spy data*.

As the full possible readout rate should be utilized, a second "fast" path was implemented. For this approach a dedicated "fast"-DC was implemented.

Following the various EUDAQ components used for the MPW3 are explained. A schematic of the whole setup can be seen in fig. 5.2.2.

5.2.3.1. The Producer

EUDAQ "out of the box" contains a *Caribou-producer*. This producer is an interface between EUDAQ and the custom implemented Peary-device[8], see above in section 5.2.1.

In both readout modes this producer is initializing and configuring the MPW3-chip as well as the Carboard. It is also constantly reading out a FIFO of the FPGA via the AXI-bus and builds / sends EUDAQ events from the so gathered data during a run. This data corresponds to the above mentioned spy data.

[7] For the "slow" DC the *DirectSaveDataCollector* supplied by EUDAQ was used.

[8] For this to work a few dedicated *virtual* functions have to be implemented in the C++ -class which are executed when EUDAQ wants the producer to perform certain actions, like configuring, starting / stopping a run, ...

5.2.3.2. The "fast" Data-Collector

For the "fast"-data-path the processing power of the Caribou system is not sufficient to build and send events with the full possible readout rate. Therefore the AXI-bus and Caribou Linux layer are entirely skipped for data taking. The FW is reading out the chip with full speed, packing multiple frames into UDP packages and sending these packages via a 1 GBit/s direct ethernet connection to a desktop computer.

On this computer the "fast"-DC is running. For the DC to be fast enough to receive all UDP packages, process and store them it was necessary to implement as a multi threaded application. It consists of the following three threads:

1. **Receiver**
   The *Receiver* thread opens an UDP-port (to which the FPGA is sending data to), receives data as fast as possible and stores it to a first Ring Buffer (RB)-stage *RB1*.

2. **Unpacker**
   The FPGA sends a UDP-packet as soon as an internal FIFO is full[9]. As this means there might be not closed frames in a UDP packet, the task of the *Unpacker* is to scan *RB1* for frames enclosed by a SOF and an EOF word and store the complete frames in a second RB-stage *RB2*.

3. **Writer**
   The *Writer* thread permanently looks if *RB2* buffered full frames, packs them into an EUDAQ event and stores them to the Hard Drive (HD) in a *.mpw3raw*-file[10].

The file stored by the *Writer* is not yet sufficiently processed to be analyzed as it stores MPW3-frames as EUDAQ events. Before the recorded data can be used for analysis multiple frames have to be preprocessed and merged into "real" events[11].

5.2.3.3. The Event Preprocessor

As stated in section 3.4.1 synchronization of the detector events is one of the most important things for a test beam setup. With the MPW3 the AIDA-TLU was utilized for this purpose. With the high readout rate the synchronization by trigger number was no feasible option as the required handshake and the clocking out of the trigger number is too slow.

Therefore a synchronization by TS was implemented. To do so each hit of the MPW3 had to be assigned to a global TS. The chip itself supplies a 8 bit TS for

[9] The FIFO has a depth of 4092 32Bit-words.

[10] For this file type a dedicated *EUDAQ-FileWriter*, which basically is a copy of the "NativeFileWriter" but storing to a file with a different suffix in order to distinguish them from already preprocessed events stored in ".raw"-files, got implemented.

[11] A "real" event should contain all hits belonging to the same trigger or TS.

each hit, which means there will be a TS overflow every $2^8 \times 50\,\text{ns} = 12.8\,\mu\text{s}$, which is not sufficient for a telescope global TS. It got extended by a 8 bit TS overflow counter in the FPGA which is sent within the unused bits of the SOF and EOF words and increments at every 8 bit clock overflow. This corresponds to a time resolution of $2^8 \times 12.8\,\mu\text{s} \approx 3.3\,\text{ms}$, which still is not sufficient. To acquire a bigger and sufficient timestamp which could actually meet the demands for a global timestamp a 64 bit counter based on 50 ns was implemented, thereby allowing for a time of $2^64 \times 50\,\text{ns} \approx 700 \times 10^3\text{y}$, which is more than large enough and which is added to the end of each UDP package. As an UDP package can contain more than just a single hit this leads to timestamping ambiguities within one package. To generate a global timestamp for each single hit the following procedure was set up:

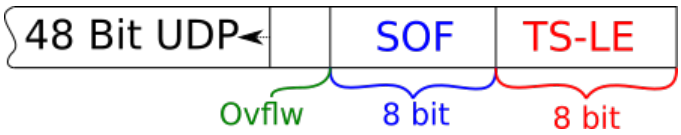


Figure 5.2.3.: Sketch of the global timestamping.

For a starting point the UDP timestamp is used for each single hit inside an UDP package. The lower 8 bit are being replaced by the TS-LE and the bits (8, 15) get replaced by the overflow counter inside the SOF. Inside the UDP package itself overflows of the 8 bit SOF-overflow counter can occur. To also take this possibility into account the UDP package is iterated backwards[12] and such overflows get identified. If an overflow occurs it is applied to the global timestamp at bit(s) ≥ 16 depending on the number of overflows which occurded in the UDP package beforehand. The routine is illustrated in fig. 5.2.3.

In order to implement this algorithm and to fit it into the EUDAQ framework a *Event Preprocessor* in the form of an *EUDAQ-FileReader*[13] was developed. This preprocessor reads data from a ".mpw3raw"-file created by the "fast"-DC and builds time stamped events by executing the following procedure:

1. Load full UDP packages into a buffer.
2. Apply the above illustrated algorithm to generate a global TS.
3. Merge data words with the same TS to one event.
4. Store these events to a native EUDAQ ".raw"-file.

[12] Backwards because the actual timestamp in the frame is closest to the end of the UDP package and thereby closest to the 64 bit UDP timestamp. Thus earlier frames are more likely to have experienced overflows of the SOF overflow counter.

[13] A *FileReader* was chosen as it allows the use of the EUDAQ executable "euCliConverter" to perform the preprocessing.

5.2.3.4. The Event Converter

In both the ".mpw3raw"- as well as in the already preprocessed ".raw'-file there are still uninterpreted data words in a binary format stored.

The task of the *Event Converter* is to extract the information of a hit from these data words, the data format is shown in fig. 5.2.4, and convert them to an *EUDAQ standard-event* for further analysis.

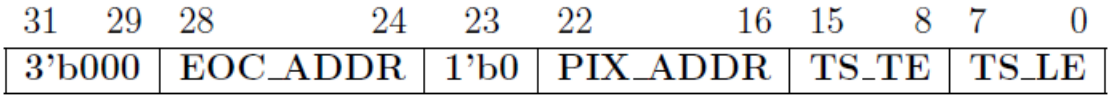


Figure 5.2.4.: The data format of one 32 bit word.

In fact not only one but two event converters had to be implemented:

1. One event converter to convert the preprocessed events. This event converter is the one which should be used for analysis of test beam data.

2. Another converter to interpret the *spy data* from the "slow" data path. As in this data path no preprocessing happens and therefore these events are not asserted to a TS, this converter can only be used for online monitoring purposes. It is not suited for actual data analysis.

5.2.3.5. The Monitor

As mentioned in section 5.2.2 a *Monitor* was implemented in the GUI. This monitor receives *spy data* and plots a hitmap using *Gnuplot* [34]. This monitor is a useful debugging tool during a test beam.

5.3. Laboratory measurements

5.3.1. S-Curve Measurements

In order to test and evaluate the in-pixel electronics of the MPW3 an injection mechanism was implemented in the chip. This mechanism allows to inject a voltage pulse directly to the comparator. This way hits can be simulated without the need of any actual incident particles. In the following measurements injections were used to investigate the threshold behavior of the pixels. According to the condition in eq. (5.1.1) (with $V_{\text{preamp}} \rightarrow V_{\text{inj}}$) no hits will be measured for small injection voltages V_{inj} and all injections should be registered for high enough values of V_{inj}. The transition from no to all hits results in a S-like shape, therefore these measurements are referred to as *S-Curves*.

With the parameters used for the S-curve in fig. 5.3.1 a transition between $0 \rightarrow 100$ hits is expected at an injection voltage of $V_{\text{inj}} \approx V_{\text{th}} - V_{\text{bl}} \approx 1250\,\text{mV} -$

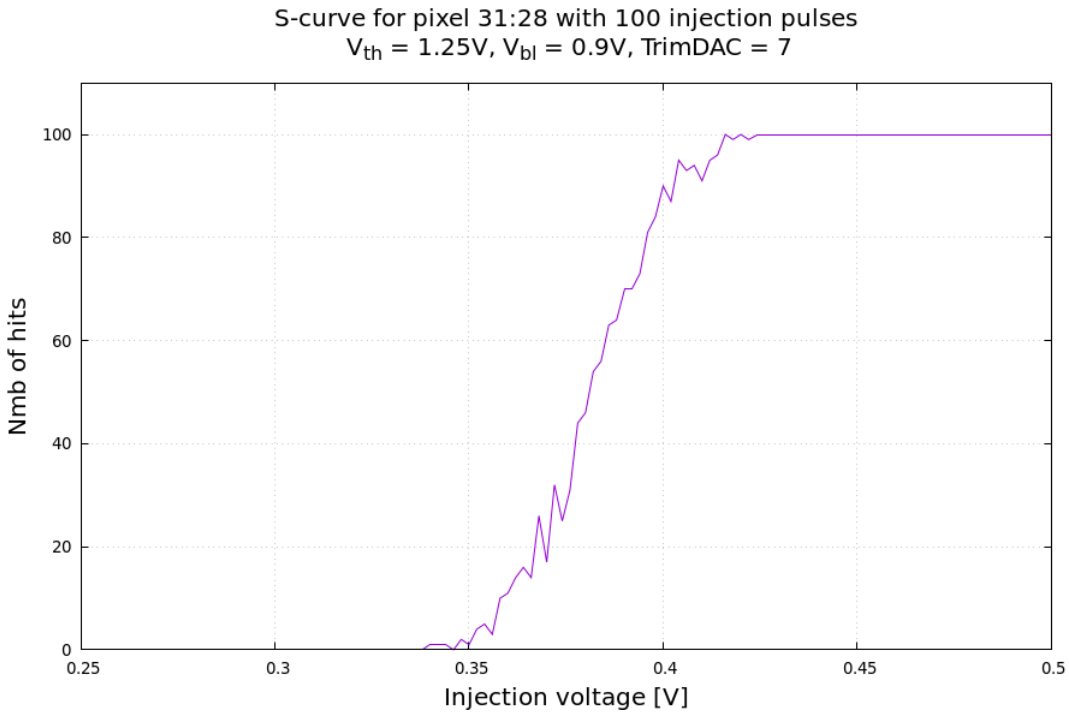


Figure 5.3.1.: S-curve scan for a single pixel. The measurement shows a response of the pixel (half of the sent injection pulses get measured) at $\approx 381\,\mathrm{mV}$. The transition from 0 to 100 hits occurs in a range of $\approx 86\,\mathrm{mV}$.

900 mV = 350 mV, which approximately corresponds to the observed behavior of the tested pixel.

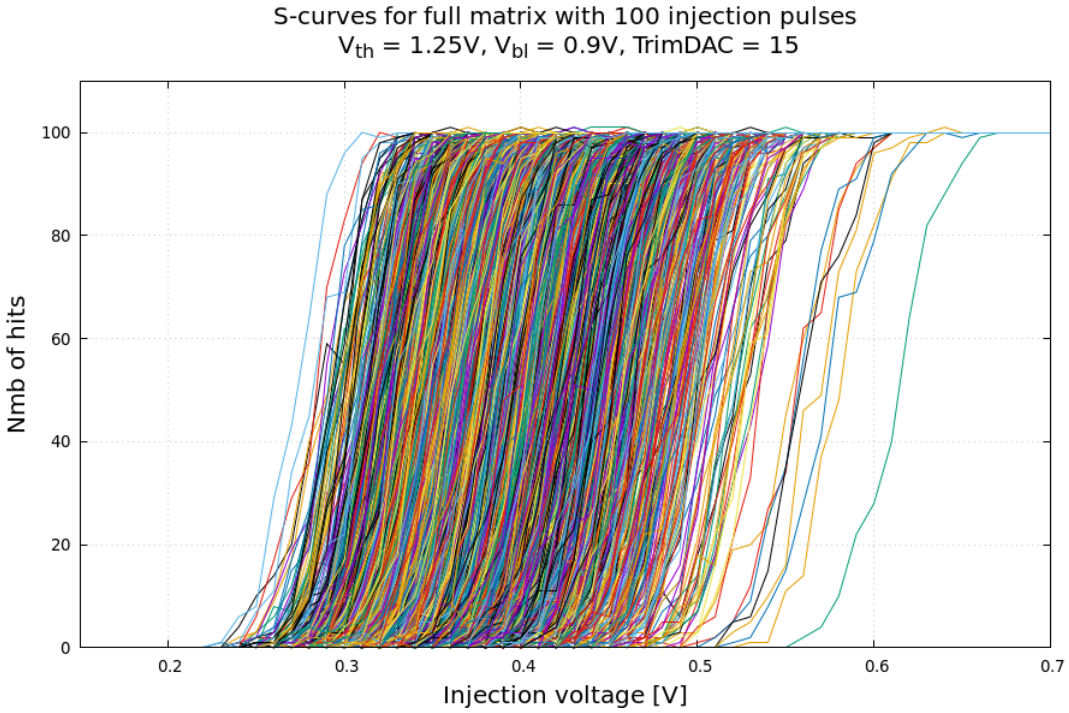


Figure 5.3.2.: S-curve scan of the full matrix. Each line represents one pixel → 4092 lines plotted. A spread of ≈ 340 mV threshold variation is seen for all pixels.

Due to manufacturing tolerances or cross talk from different components of the chip the threshold will be slightly different for each single pixel, even when identical configurations are applied to all pixels. These effects have been investigated with a S-curve scan of the full matrix. The results are shown in fig. 5.3.2. This measurement shows that, even though identical settings for all pixels have been used, the value for V_{inj} at which 50 % of all hits are being detected varies from ≈ 270 mV → 610 mV and therefore the actual threshold setting for the whole matrix differs by up to ≈ 340 mV.

5.3.2. TrimDAC evaluation

To investigate the effect of the TrimDAC setting on the threshold further S-curve measurements with varied TrimDAC values have been performed. An internally generated voltage referred to as VP_{trim} can be configured[14] on which the effect of one TrimDAC LSB depends on. The measurements are shown in fig. 5.3.3.

[14] VP_{trim} is configurable by a 6 bit register and therefore can be adjusted in the range from 0 to 63.

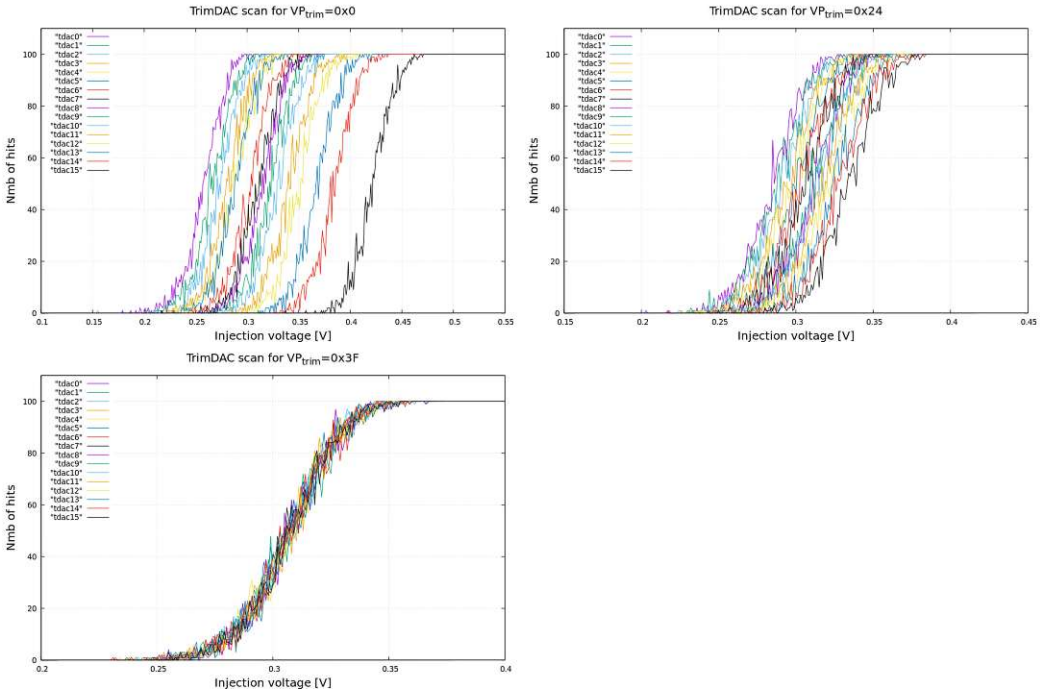


Figure 5.3.3.: TrimDAC scans for different values of VP_{trim}. S-curves with 100 injections for each TRIMDAC setting have been recorded. From these measurements the average effect, of one TRIMDAC LSB for the different VP_{trim} settings on the resulting threshold, can be concluded and is listed in table 3.2.

VP_{trim}	Range [mV]	$\overline{LSB}$ [mV]
0x00	260 → 420	10.0
0x24	280 → 340	3.7
0x3F	300 → 310	0.6

Table 3.2.: Effect of VP_{trim} on the TrimDAC values.

The results of the performed measurements are summarized in table 3.2. These values have to be considered as approximate values which can vary for different pixels and are measured at $\approx 50\,\%$ of the maximum number of hits.

5.3.3. Calibration

In order to operate the detector at a set point at which the threshold is set in a way the pixels are most sensitive to incident particles but do not already produce noise data a calibration procedure was implemented.

By combining the results of the measurements in section 5.3.1 and section 5.3.2 it can already be concluded that acquiring the ideal threshold configuration for the whole matrix will be a difficult task, as the S-curves of the full matrix differ by $\approx 340\,\mathrm{mV}$ and with the maximum available effect of the TrimDAC adjustment only a difference of $\approx 160\,\mathrm{mV}$ can be achieved. The difference between the various pixels can also be seen in the "noise map" shown in fig. 5.3.4. This hitmap was measured by setting $V_{\mathrm{th}} \approx 1200\,\mathrm{mV}$ and performing a readout without an actual particle source. This way just plain noise got recorded.

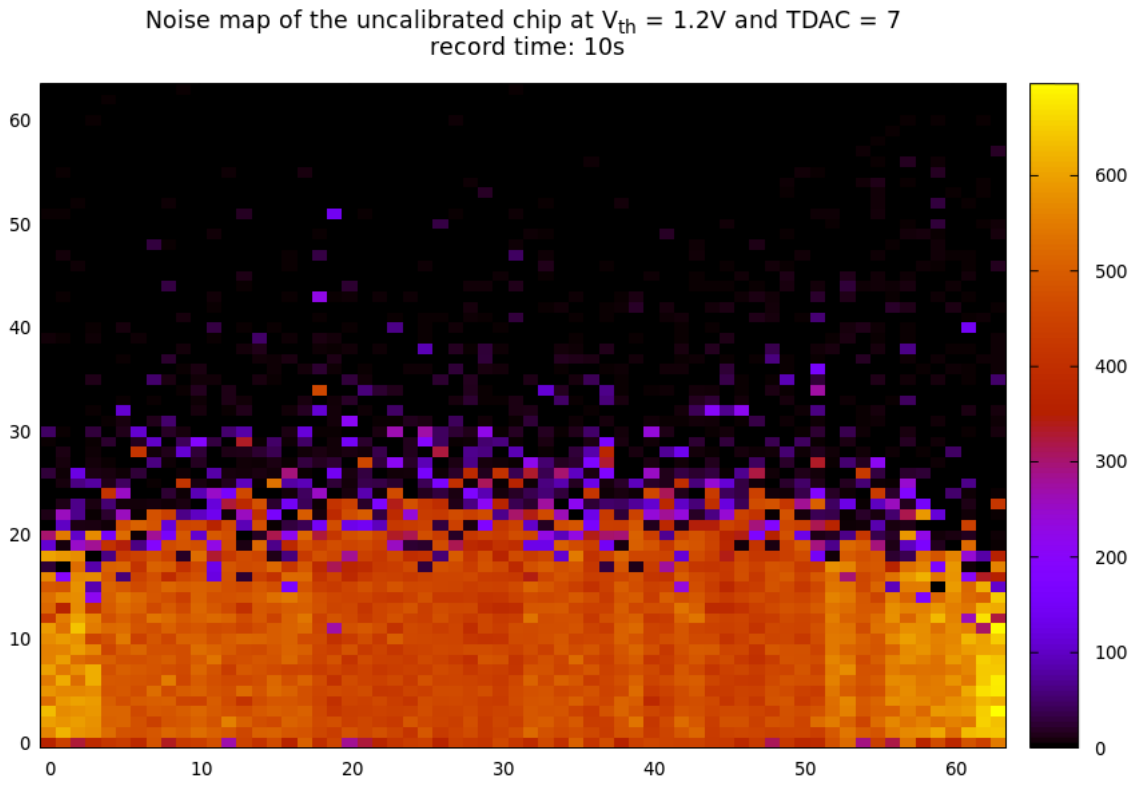


Figure 5.3.4.: Noise hitmap of the uncalibrated chip. This measurement shows a noise level which is about 5 times higher in the bottom half compared to the top half.

The calibration procedure performs the following operations in order to find the ideal threshold setting:

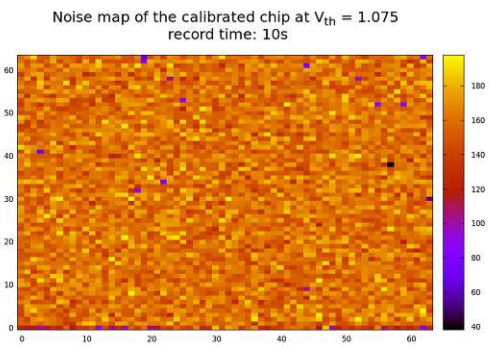
1. Global threshold optimization:
   Several chosen rows (not the full matrix in order to save time) get activated one by one, while the rest of the matrix remains masked, and V_{th} gets scanned from an upper to a lower boundary with configurable decrements per step. With the chosen threshold all data from the current row gets read out. When a pixel starts firing (solely from noise) this pixel will be masked. The scan finishes for the current row when $> 50\,\%$ of the pixels are masked and the threshold value will be considered ideal for this row. The procedure is repeated for all chosen rows and the average value of all ideal thresholds is calculated. This value is considered to be the best suited configuration for the whole matrix.

2. Mask noisy pixels:
   A readout with the above acquired best suited value of V_{th} is performed and pixels which accumulate an amount of hits exceeding a configurable upper boundary get masked. This step is performed to avoid that some noisy pixel totally occupy the readout FIFOs and data would get lost.

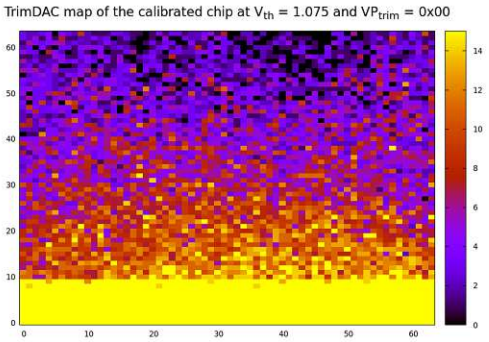
3. TrimDAC optimization:
   The trimDAC value of each pixel is set to the maximum value of 15. Only one row at a time is unmasked. Then a readout is performed. Each pixel knows the following states:

   - *NoHit*
     The pixel did not yet get hit $\Rightarrow$ trimDAC is being decreased by 1 at a time until hits are detected and the state is changed to *Hit*.
   - *Hit*
     The pixel already got hit. Increase the trimDAC value until the pixel stops firing, state changes to *Done* when this is achieved.
   - *Done*
     The trimDAC value is set at a point at which one LSB beneath the pixel is generating noise hits while at the found value the pixel does not account for hits. This is the ideal value for each pixel. It is set at the point at which it is most sensitive to incident particles without producing noise hits.

The results of the calibration routine are plotted in fig. 5.3.5. Though still a sizable amount of noise is measured the hitmap is much more homogeneous than in the uncalibrated case above. The reason noise is measured after the calibration at all (which should counteract exactly this behavior) arises from the different recording time used during the calibration routine and the open shutter measurement which lead to the hitmap. From the map of the trimDAC values can be concluded



(a) Noise hitmap of the calibrated chip.



(b) Map of the TrimDAC values acquired by the calibration routine.

Figure 5.3.5.: Results of the calibration routine. After the calibration the noise hitmap appears much more uniform than in the measurement prior in fig. 5.3.4.

that while the lower part of the matrix requires the maximum setting the values for the upper part are located in the intermediate / lower spectrum of the possible trimDAC range, thus the lower part of the matrix seems to suffer from more noise than the upper part.

5.4. Testbeam measurements at CERN

The measurements in this section have been performed at the CERN-SPS test beam facility.

Due to the above mentioned noise behavior, being different from top to lower part of the matrix, and the calibrated full matrix still producing a lot of noise which accounts for $\mathcal{O}(1\,\mathrm{GB/min})$ on HD usage and therefore would produce a not handlebar amount of data it was decided to mask the lower 32 rows of the matrix and use $V_{\mathrm{th}} \approx 1200\,\mathrm{mV}$ and $V_{\mathrm{bl}} \approx 900\,\mathrm{mV}$ for the measurements performed at the testbeam. A corresponding hitmap is shown in fig. 5.4.1.

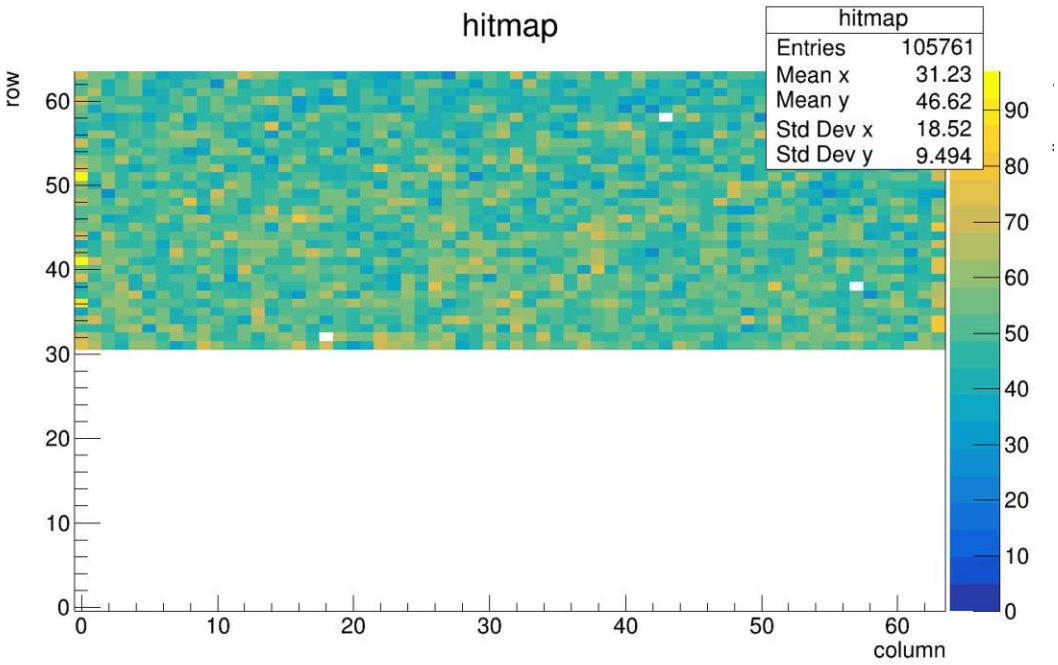
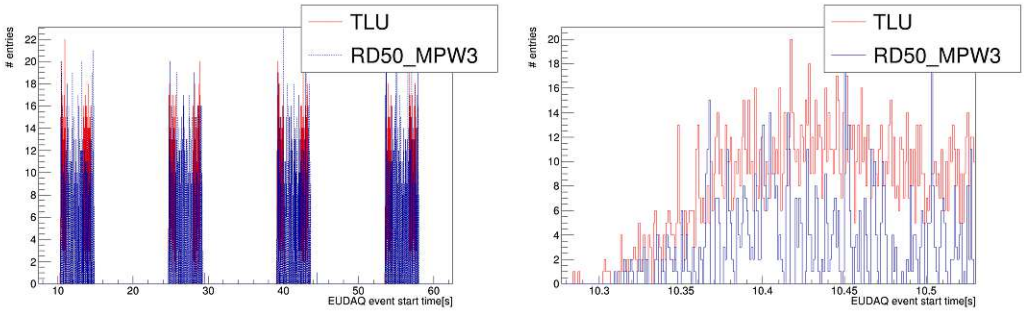


Figure 5.4.1.: Hitmap of the RD50-MPW3.

For this test beam a similar telescope, consisting of 6 MIMOSA26 detector planes, as in the test beam presented in section 4.3 was used. The biggest difference in the setup, besides the DUT of course, was the different synchronization concept. Instead of synchronization by trigger number synchronization by TS got implemented, see section 3.4.1.

To match the events of the trigger number synchronized MIMOSA26, which do not contain any time information, with the MPW3 events, which on contrary only know timestamps, the events of the AIDA-TLU have been used to define the event frame in *Corryvreckan*. The TLU events contain both a trigger number as well as a TS.

A comparison of the MPW3 event times and the TLU event times is shown in fig. 5.4.2. A clear correlation of the event times can be seen from these plots. The



(a) Coarse time scale. **(b)** Fine time scale.

Figure 5.4.2.: Comparison of the event start time of the TLU and the RD50-MPW3. The overlap of the timestamps of the AIDA-TLU and the RD50-MPW3 leads to the conclusion that the preprocessing routine assigns the TS correctly.

pauses in between the various "bunches" can be explained by the CERN-SPS spill structure. The time frame in which the MPW3 generated events but the TLU did not can be explained by an active trigger veto of the telescope detectors.

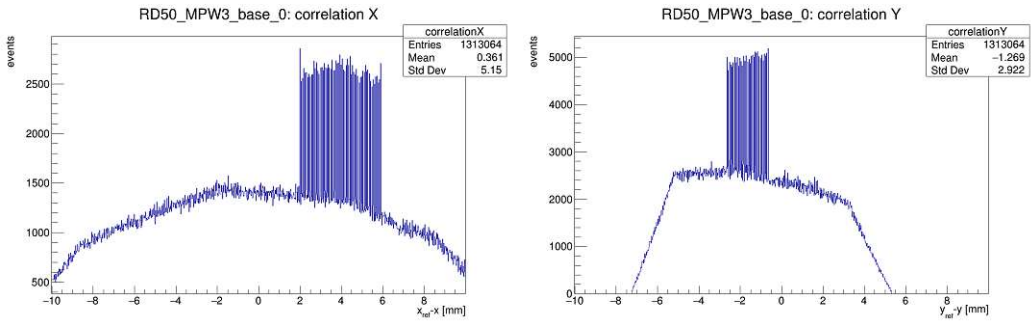


Figure 5.4.3.: Correlations between the MIMOSA26 reference plane and the RD50-MPW3.

Although these event times seem reasonable and no clearly identifiable offset can be seen in fig. 5.4.2b it was not possible, at least during the course of this thesis, to see spatial correlations of the hits between the telescope and the DUT. This can be seen in fig. 5.4.3.

The various "peaks" which can be seen correspond to the position of the chip relative to noisy pixels of the telescope reference plane. As no very distinct peak is identifiable though (compare to fig. 4.3.2) the hits do not appear to be correlated. This most likely indicates a problem with the synchronization / global timestamp generation within the setup which was not solvable neither during the testbeam nor during the course of this thesis.

The lack of correlations does not allow for any alignment procedure which fur-

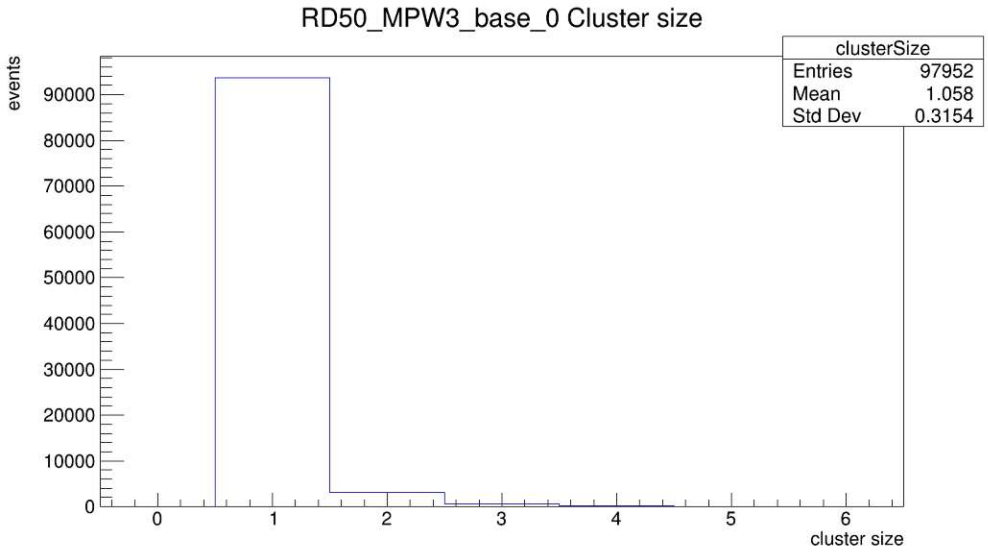


Figure 5.4.4.: Cluster size of the RD50-MPW3.

thermore does not allow for track reconstruction and residuals / efficiency analysis. The analysis of the cluster size can be performed without tracks though and is shown in fig. 5.4.4.

The configuration used for this analysis is listed in appendix A.2.

6. Conclusions

6.1. TJ-Monopix2

The efficiency analysis of the detector done in section 4.3.1.4 shows a good threshold calibration of the chip, as almost all tracks could be assigned to a hit on the chip and also shows that the DAQ is well implemented to readout all the data and assign trigger numbers from the AIDA-TLU to the hits and of course that the chip itself is capable to detect almost all particles. For the HV-frontend a significantly lower efficiency, compared to the normal-frontend, was measured at bias voltages $> -10\,\mathrm{V}$.

The normal-frontend shows a defective column (column 224) in all evaluated DUTs. This column is located exactly at the border between the normal- and the cascode-frontend. This problem needs further investigation on chip design level.

The DAQ system used for the readout of the TJ-Monopix2 is limited to rates of $\lesssim 10\,\mathrm{kHz}$ as was investigated by several laboratory measurements. This was no problem for the test beam at DESY as the MIMOSA26 limited the TLU trigger rate to $\approx 1\,\mathrm{kHz}$ by raising the trigger veto line. Should a higher rate be targeted, for which also a faster telescope has to be utilized, the DAQ system needs to be upgraded by a more efficient readout scheme.

Overall the detector performed very well and seems to be a good starting point for the development targeting the BELLE-II-VTX upgrade.

6.2. RD50-MPW3

The biggest difference between the RD50-MPW3 and its predecessor the RD50-MPW2 is the entirely new designed digital periphery. As this part was designed at HEPHY the main objective of the investigations performed during this thesis was to verify the performance of this part. As shown by the measurements in chapter 5 the digital periphery is capable of performing the configuration of the chip as well as the digital data readout, its two main tasks.

6.2.1. Encountered problems

Though the RD50-MPW3 can be considered an improved version of its predecessor and a design success a few problems have been encountered which should be summed up here.

- Noisy lower part of the matrix
  As shown in section 5.3.3 the lower part of the matrix suffers from a higher noise level than the upper partition. This most likely is due to some cross talk from the digital periphery to the active matrix and needs further investigation on chip layout level. A possible next iteration of the chip might also benefit from a broader span of the achievable threshold adjustment by the trimDACs.

- Timestamp generation for synchronization
  The reason why the analysis of the test beam data in section 5.4 did not produce proper results can clearly be identified by the difficult global timestamp generation. On software level there have not been more options with the present chip design and FW as all available time information from the chip has been used. At the moment of writing these lines a more precise 38 bit overflow counter in the SOF and EOF is being implemented and tested which will make synchronization easier in future test beams. The problem could also be accounted for in a next chip iteration by either expanding the actual hit timestamp in the TS-LE and TS-TE or by sending more than one SOF word[1].

- Configuration not working on "first try"
  During the various measurements sometimes (occured "randomly" when performing various measurements) the configuration got not properly applied to the pixels. This was observed within the configuration readback as well as by the chip behavior and might be linked to timing issues within the chip. As this behavior is not really understood at the moment further investigation has to be performed in order to figure out the source of the problem.

6.3. Comparison

Despite the obvious differences between the two investigated DMAPS like the different number of pixels, the different size and the different readout systems this chapter should sum up the not so obvious differences.

The different bias voltage of $\approx -100\,\mathrm{V}$ for the RD50-MPW3 and $\approx -5\,\mathrm{V} \rightarrow -10\,\mathrm{V}$, depending on the frontend flavor, for the TJ-Monopix2 results in a distinctly different cluster size distribution. This can be explained by the dominant, due to the magnitude of the bias voltage, charge carrier transport mechanism in either one of the chips. While for the RD50-MPW3 due to the higher electric fields in the sensitive volume the transport of the charge carriers can be expected to be dominated by charge *drift* the amount of *diffusion* current in the TJ-Monopix2 is substantial resulting in a correspondingly lower / higher cluster size.

[1] The FW could use these SOF words to replace them by a larger timestamp. This cannot be done with the current chip design as with the high readout rate of 320 MHz the FPGA is not able to "pause" the data flow and store additional information into its readout FIFOs.

Due to the lack of proper test beam analysis data for the RD50-MPW3 the efficiency as well as the spatial resolution can not be compared.

The big success of the test beam at DESY and the difficulties encountered at the CERN-SPS one can be directly related to the different implemented synchronization concepts.

Synchronization by trigger number, as done for the TJ-Monopix2 DAQ, with a proper triggering device, like the AIDA-TLU, reduces ambiguities in the telescope global event building process to a minimum, with the draw backs of a reduced readout rate.

Synchronization by TS on the other hand allows for much higher readout rates while the assignment and generation of global TSs[2] is no trivial task. The safest approach would be to already assign for example a 64 bit TS on chip level to each hit. This would lead to an increase of data output[3], resulting yet in a reduced readout rate and an increased amount of data storage use. Thus including a bigger TS already at chip level should not be advertised, but instead a global timestamping of each hit should be implemented as best as possible on FW level.

To compare the theoretical possible hit rates with the current chip and DAQ design of both chips a "back of the envelope"(not accounting for processing overhead) calculation leads with a frequency of 40 MHz of the AIDA-TLU to a trigger rate of $\approx \frac{1}{40\,\mathrm{MHz}} = 25\,\mathrm{ns}$, $25\,\mathrm{ns} \times \underbrace{15}_{\text{length of TLU trigger ID}} = 375\,\mathrm{ns} \Rightarrow f \approx 2.6\,\mathrm{MHz}$ for the trigger number synchronized approach of the TJ-Monopix2.

Within the calculation for the RD50-MPW3 TS synchronized approach the time needed to process the trigger number is 0 as it is not utilized at all. The hit rate thereby is not limited by any triggering device but only limited by the readout speed the FPGA and the detector are capable of. In the here discussed case this leads to a rate of $\frac{1}{640\,\mathrm{MHz}} \approx 1.56\,\mathrm{ns}$, $1.56\,\mathrm{ns} \times \underbrace{40}_{\text{length of 1 hit word}} = 62.4\,\mathrm{ns} \Rightarrow f \approx$ 16 MHz. 40 Bit have been used for the length of one hit word as the data stream is *8 bit - 10 bit* encoded in the design of the RD50-MPW3, thus a payload of 32 bit actually accounting for 40 bit in the data stream.

Thus the synchronization by TS is, just by principle, capable of handling an event rate bigger by the factor of ≈ 6. With all the here discussed drawbacks though which have to be encountered by a clever chip, FW and DAQ design.

[2] Which have to be large enough in terms of bit size to run for several hours at least.

[3] In the case of the current design of the RD50-MPW3 one hit would not account for 32 bit, but in a data usage (in the case of a 64 bit TS) of $\approx 32 - 8 + 64 = 88$ bit. The -8 resigns from the assumption that one TS (TS-LE or TS-TE) could be replaced by the global TS.

Bibliography

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

[2] F. Forti. "Snowmass Whitepaper: The Belle II Detector Upgrade Program". In: *2022 Snowmass Summer Study*. 2022. arXiv: 2203.11349 [hep-ex].

[3] E. D. R. R. P. Group. *The 2021 ECFA detector research and development roadmap*. Tech. rep. Geneva, 2020. DOI: 10.17181/CERN.XDPL.W2EX.

[4] S. Chatrchyan et al. "The CMS Experiment at the CERN LHC". In: *JINST* 3 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.

[5] T. A. Collaboration, G. Aad, E. Abat, et al. "The ATLAS Experiment at the CERN Large Hadron Collider". In: *Journal of Instrumentation* 3 (2008), S08003–S08003. DOI: 10.1088/1748-0221/3/08/s08003.

[6] J. P. Iddon. "Construction, Commissioning and Performance Measurements of the Inner Tracking System Upgrade (ITS2) of ALICE at the LHC". Presented 29-06-2022. 2022.

[7] I. Berdalovic et al. "Monolithic pixel development in TowerJazz 180 nm CMOS for the outer pixel layers in the ATLAS experiment". In: *JINST* 13 (2018), p. C01023. DOI: 10.1088/1748-0221/13/01/C01023.

[8] D. Colella. "Upgrade of the ALICE experiment beyond LHC Run 3". In: *International Journal of Modern Physics E* 31 (2022). DOI: 10.1142/s021830132240002x.

[9] H. Kolanoski and N. Wermes. *Particle Detectors, Fundamentals and Applications*. Oxford University Press, 2020.

[10] S. Bühler-Paschen, H. Michor, and M. Reissner. *Festkörperphysik 1*. TU Verlag. 2018.

[11] R. M. Hernandez. *Overview of CMOS sensors for future tracking detectors*. 2019. DOI: 10.48550/ARXIV.1910.07284.

[12] M. I. Besana, F. Cerutti, A. Ferrari, et al. "Evaluation of the radiation field in the future circular collider detector". In: *Phys. Rev. Accel. Beams* 19 (11 2016), p. 111004. DOI: 10.1103/PhysRevAccelBeams.19.111004.

[13] *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*. Tech. rep. Geneva: CERN, 2017. DOI: 10.17181/CERN.FOZZ.ZP3Q.

[14] W. Snoeys, G. Aglieri Rinella, H. Hillemanns, et al. "A process modification for CMOS monolithic active pixel sensors for enhanced depletion, timing performance and radiation tolerance". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 871 (2017). DOI: 10.1016/j.nima.2017.07.046.

[15] S. Spannagel, K. Wolters, D. Hynds, et al. "Allpix2: A modular simulation framework for silicon detectors". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 901 (2018), pp. 164–172. DOI: https://doi.org/10.1016/j.nima.2018.06.020.

[16] T. Vanat. "Caribou – A versatile data acquisition system". In: *PoS* TWEPP2019 (2020), p. 100. DOI: 10.22323/1.370.0100.

[17] Linux-Foundation and Yocto-Project. *The Yocto Project*. 2022. URL: https://www.yoctoproject.org/ (visited on 07/20/2022).

[18] Linux-Foundation and Yocto-Project. *Poky: A reference distribution of the Yocto Project*. 2022. URL: https://www.yoctoproject.org/software-item/poky/ (visited on 07/20/2022).

[19] *Peary Caribou: DAQ framework for the Caribou DAQ System*. URL: https://peary-caribou.web.cern.ch/peary-caribou/ (visited on 07/21/2022).

[20] M. Daas, Y. Dieter, J. Dingfelder, et al. "BDAQ53, a versatile pixel detector readout and test system for the ATLAS and CMS HL-LHC upgrades". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 986 (2021), p. 164721. DOI: https://doi.org/10.1016/j.nima.2020.164721.

[21] SiLab-Bonn. *basil: A data acquisition framework in Python and Verilog*. URL: https://github.com/SiLab-Bonn/basil (visited on 07/20/2022).

[22] P. Baesso, D. Cussans, and J. Goldstein. "The AIDA-2020 TLU: a flexible trigger logic unit for test beam facilities". In: *Journal of Instrumentation* 14 (2019), P09019–P09019. DOI: 10.1088/1748-0221/14/09/p09019.

[23] P. Baesso. *Manual AIDA Trigger logic unit (TLU v1E)*. University of Bristol, 2019.

[24] P. Ahlburg, S. Arfaoui, J.-H. Arling, et al. "EUDAQ—a data acquisition software framework for common beam telescopes". In: *Journal of Instrumentation* 15 (2020), P01038–P01038. DOI: 10.1088/1748-0221/15/01/p01038.

[25] K. Moustakas. "Design and Development of Depleted Monolithic Active Pixel Sensors with Small Collection Electrode for High-Radiation Applications". PhD thesis. Rheinische Friedrich-Wilhelms-Universität Bonn, 2021.

[26] *pybind11*. URL: https://github.com/pybind/pybind11 (visited on 07/25/2022).

[27] D. Dannheim, K. Dort, L. Huth, et al. "Corryvreckan: a modular 4D track reconstruction and analysis software for test beam data". In: *Journal of Instrumentation* 16 (2021), P03008. DOI: 10.1088/1748-0221/16/03/p03008.

[28] I. Rubinskiy. "An EUDET/AIDA Pixel Beam Telescope for Detector Development". In: *Physics Procedia* 37 (2012). Proceedings of the 2nd International Conference on Technology and Instrumentation in Particle Physics (TIPP 2011), pp. 923–931. DOI: https://doi.org/10.1016/j.phpro.2012.02.434.

[29] R. Frühwirt. *Wahrscheinlichkeitsrechnung und Statistik für Studierende der Physik*. Bookboon, 2015.

[30] D. Dannheim, K. Dort, D. Hynds, et al. "Combining TCAD and Monte Carlo methods to simulate CMOS pixel sensors with a small collection electrode using the Allpix2 framework". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 964 (2020), p. 163784. DOI: 10.1016/j.nima.2020.163784.

[31] P. Sieberer, C. Irmler, H. Steininger, et al. "Design and characterization of depleted monolithic active pixel sensors within the RD50 collaboration". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1039 (2022), p. 167020. DOI: https://doi.org/10.1016/j.nima.2022.167020.

[32] E. Vilella. "Development of high voltage-CMOS sensors within the CERN-RD50 collaboration". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1034 (2022), p. 166826. DOI: https://doi.org/10.1016/j.nima.2022.166826.

[33] OpenCores. *Wishbone B4, WISHBONE System-on-Chip (SoC)Interconnection Architecture for Portable IP Cores*. 2010. URL: https://opencores.org/howto/wishbone.

[34] T. Williams and C. Kelley. *Gnuplot*. 2022. URL: http://www.gnuplot.info (visited on 07/28/2022).

List of Figures

List of Tables

Acronyms

ALICE A Large Ion Collider Experiment.

ALPIDE ALICE inner pixel detector.

CaR Control-and-Readout.

CLI Command Line Interface.

CMOS complementary metal-oxide-semiconductor.

CU Control Unit.

DAC Digital to Analog Converter.

DAQ data acquisition.

DC Data Collector.

DESY german electron synchrotron / ”Deutsches-Elektronen-Synchrotron”.

DMAPS Depleted Monolithic Active Pixel Sensors.

DUT Device under Test.

ECFA European Committee for Future Accelerators.

EOC End Of Column.

EOF End of Frame.

FCC Future Circular Collider.

FE Front End.

FW firmware.

GUI Graphical User Interface.

HD Hard Drive.

HEP High Energy Physics.

HW hardware.

ITS2 inner tracking system version 2.

LS2 Long Shutdown 2.

MIP Minimum Ionizing Particle.

OS Operating System.

PMT Photo Multiplier.

R&D Research and Development.

RB Ring Buffer.

SoC System on Chip.

SOF Start of Frame.

SW software.

TDAC trim-DAC.

TDC Time to Digital Converter.

TLU Trigger Logic Unit.

ToT Time over Threshold.

TS Time Stamp.

TS-LE Timestamp Leading Edge.

TS-TE Timestamp Tailing Edge.

A. Corryvreckan Configuration

In this appendix the configuration files to operate *Corryvreckan* for alignment and analysis of the test-beam data from DESY are listed. The values "< ... >" account for filenames which have to be adjusted to the file-system of the computer Corry is being executed on.

A.1. TJ-Monopix2 Configuration

A.1.1. Alignment

The configuration-files of the alignment process are listed in lst. A.1 to lst. A.4 and should be performed from first to last for a full alignment procedure.

Listing A.1: Prealignment Telescope

```
[Corryvreckan]
log_level                = "INFO"
log_format               = "DEFAULT"

number_of_events         = 50000
detectors_file           = <rough_geo>
detectors_file_updated   = <geo_telPrealign>
histogram_file           = <out_telPrealign.root>

[Metronome]
event_length             = 2us
triggers                 = 1

[EventLoaderEUDAQ2]
type                     = "MIMOSA26"
get_time_residuals       = true
file_name                = <record_tel.raw>

[ClusteringSpatial]

[Correlations]

[Prealignment]
time_cut_abs             = 100
type                     = "MIMOSA26"
method                   = "gauss_fit"
fit_range_rel            = 100
```

Listing A.2: Alignment Telescope

```
[Corryvreckan]
log_level                = "INFO"
log_format               = "DEFAULT"

number_of_events         = 50000
detectors_file           = <geo_telPrealign>
detectors_file_updated   = <geo_telAlign>
histogram_file           = <out_telAlign.root>

[Metronome]
event_length             = 2us
triggers                 = 1

[EventLoaderEUDAQ2]
type                     = "MIMOSA26"
get_time_residuals       = true
file_name                = <record_tel.raw>

[ClusteringSpatial]

[Correlations]

[Tracking4D]
min_hits_on_track        = 6
exclude_dut              = true
track_model              = "gbl"
time_cut_abs             = 10s

[AlignmentTrackChi2]
iterations               = 4
prune_tracks             = true
align_position           = true
align_orientation        = true
max_associated_clusters  = 1
max_track_chi2ndof       = 10
```

Listing A.3: Prealignment DUT

```
[Corryvreckan]
log_level                = "INFO"
log_format               = "DEFAULT"

number_of_events         = 50000
detectors_file           = <geo_telAlign>
detectors_file_updated   = <geo_dutPrealign>
histogram_file           = <out_dutPrealign.root>

[Metronome]
event_length             = 2us
triggers                 = 1

[EventLoaderEUDAQ2]
type                     = "MIMOSA26"
get_time_residuals       = true
file_name                = <record_tel.raw>

[EventLoaderEUDAQ2]
name                     = "Monopix2_0"
type                     = "Monopix2"
get_time_residuals       = true
file_name                = <record_dut.raw>

[ClusteringSpatial]

[Correlations]

[Prealignment]
time_cut_abs             = 100
name                     = "Monopix2_0"
method                   = "gauss_fit"
fit_range_rel            = 20
```

Listing A.4: Alignment DUT

```
[Corryvreckan]
log_level                = "INFO"
log_format               = "DEFAULT"

number_of_events         = 50000
detectors_file           = <geo_dutPrealign>
detectors_file_updated   = <geo_dutAlign>
histogram_file           = <out_dutAlign.root>

[Metronome]
event_length             = 2us
triggers                 = 1

[EventLoaderEUDAQ2]
name                     = "Monopix2_0"
type                     = "Monopix2"
get_time_residuals       = true
file_name                = <record_dut.raw>

[EventLoaderEUDAQ2]
type                     = "MIMOSA26"
get_time_residuals       = true
file_name                = <record_tel.raw>

[ClusteringSpatial]
#use_trigger_timestamp   = true

[Tracking4D]
min_hits_on_track        = 6
exclude_dut              = true
track_model              = "gbl"
time_cut_abs             = 10s

[DUTAssociation]
log_level                = INFO
time_cut_abs             = 100
use_cluster_centre       = false

[AlignmentDUTResidual]
iterations               = 4
align_orientation        = true
align_position           = true
max_associated_clusters  = 1
max_track_chi2ndof       = 10
```

A.1.2. Analysis

The following listing shows the configuration for a full analysis of the test-beam data.

Listing A.5: Analysis

```
[Corryvreckan]
log_level                = "INFO"
log_format               = "DEFAULT"

detectors_file           = <geo_dutAlign>
histogram_file           = <out_analysis.root>

[Metronome]
event_length             = 2us
triggers                 = 1

[EventLoaderEUDAQ2]
name                     = "Monopix2_0"
type                     = "Monopix2"
get_time_residuals       = true
file_name                = <record_dut.raw>
shift_triggers           = 1

[EventLoaderEUDAQ2]
type                     = "MIMOSA26"
get_time_residuals       = true
file_name                = <record_tel.raw>
shift_triggers           = 0

[ClusteringSpatial]

[Correlations]

[Tracking4D]
min_hits_on_track        = 6
exclude_dut              = true
track_model              = "gbl"
time_cut_abs             = 10s

[DUTAssociation]
log_level                = INFO
time_cut_abs             = 100

[AnalysisEfficiency]

[AnalysisTelescope]

[AnalysisDUT]
```

A.2. RD50-MPW3 Configuration

For the testbeam analysis of the RD50-MPW3 the geometry file in lst. A.6 and the Corry config in lst. A.7 have been used.

Listing A.6: Geometry file for MPW3

```
[TLU_0]
orientation             = 0,0,0
orientation_mode        = "xyz"
position                = 0,0,0
role                    = "auxiliary"
type                    = "tlu"

[MIMOSA26_0]
number_of_pixels        = 1152,576
orientation             = 0,0,0
orientation_mode        = "xyz"
pixel_pitch             = 18.4um,18.4um
position                = 0,0,100mm
spatial_resolution      = 5um,5um
time_resolution         = 230us
type                    = "mimosa26"
role                    = "reference"

[MIMOSA26_1]
number_of_pixels        = 1152,576
orientation             = 0,0,0
orientation_mode        = "xyz"
pixel_pitch             = 18.4um,18.4um
position                = 0,0,190mm
spatial_resolution      = 5um,5um
time_resolution         = 230us
type                    = "mimosa26"

[RD50_MPW3_base_0]
mask_file=mpw3_mask.txt
number_of_pixels        = 64,64
orientation             = 0,0,0
orientation_mode        = "xyz"
pixel_pitch             = 62um,62um
position                = 0,0,470mm
spatial_resolution      = 18um,18um
time_resolution         = 50ns
type                    = "mpw3"
role                    = "dut"

[MIMOSA26_2]
number_of_pixels        = 1152,576
orientation             = 0,0,0
orientation_mode        = "xyz"
pixel_pitch             = 18.4um,18.4um
position                = 0,0,660mm
spatial_resolution      = 5um,5um
```

```
time_resolution          = 230us
type                     = "mimosa26"

[MIMOSA26_3]
number_of_pixels         = 1152,576
orientation              = 0,0,0
orientation_mode         = "xyz"
pixel_pitch              = 18.4um,18.4um
position                 = 0,0,760mm
spatial_resolution       = 5um,5um
time_resolution          = 230us
type                     = "mimosa26"

[USBPIXI4B_10]
number_of_pixels         = 336,80
orientation              = 0,0,0
orientation_mode         = "xyz"
pixel_pitch              = 250um,50um
position                 = 0,0,800mm
spatial_resolution       = 14um,70um
time_resolution          = 10ns
type                     = "USBPIXI4B"

[MIMOSA26_4]
number_of_pixels         = 1152,576
orientation              = 0,0,0
orientation_mode         = "xyz"
pixel_pitch              = 18.4um,18.4um
position                 = 0,0,880mm
spatial_resolution       = 5um,5um
time_resolution          = 230us
type                     = "mimosa26"
```

Listing A.7: Analysis MPW3

```
[Corryvreckan]
detectors_file=              <geometry_file.geo>
log_level=                   "INFO"
histogram_file=              <output_file.root>

[EventDefinitionM26]
detector_event_time =        "TLU"
file_timestamp =             <record_tlu.raw>
file_duration =              <record_mimosa26.raw>
time_shift =                 0
shift_triggers =             1

[EventLoaderEUDAQ2]
type=                        "TLU"
shift_triggers =             1
file_name =                  <record_tlu.raw>

[EventLoaderEUDAQ2]
type=                       "MIMOSA26"
```

```
file_name =                     <record_mimosa26.raw>

[EventLoaderEUDAQ2]
name =                          "RD50_MPW3_base_0"
type =                          "RD50_MPW3_base"
file_name=                      <record_mpw3_preprocessed.raw>
shift_triggers =                0
buffer_depth =                  2048
# shift_time_us =               3200

[Clustering4D]
time_cut_abs =                  1ms


[Correlations]
do_time_cut =                   true
time_cut_abs =                  230us
```

B. Allpix^2 Configuration

B.1. Detector models

Listing B.1: Detector model of the TJ-Monopix2

```
type = "monolithic"

number_of_pixels = 512 512
pixel_size = 33.04um 33.04um

sensor_thickness = 300um

#PCB
[support]
thickness = 1.6mm
size = 100mm 100mm
offset = 20mm 0mm
hole_size = 17mm 17mm
hole_offset = 20mm 0mm
material = "g10"
```

B.2. Simulation

Listing B.2: Simulation config of the TJ-Monopix2

```
[AllPix]
log_level            = "WARNING"
log_format           = "DEFAULT"
number_of_events     = 50000
detectors_file       = <detectors_file>
multithreading       = true
workers = 18

[GeometryBuilderGeant4]
world_material       = "air"

[DepositionGeant4]
physics_list         = FTFP_BERT_LIV


# Replicate DESY beam parameters
particle_type        = "e-"
source_energy        = 4GeV
source_position      = 0 0 -10mm
beam_size            = 3mm
beam_direction       = 0 0 1
number_of_particles  = 1
max_step_length      = 5um

# ==== Telescope setup

[ElectricFieldReader]
type                 = "mimosa26"
model                = "linear"
bias_voltage         = -4V
depletion_depth      = 15um
output_plots         = true

[ProjectionPropagation]
type                 = "mimosa26"
temperature          = 293K
charge_per_step      = 10
integration_time     = 15ns
diffuse_deposit      = true
output_plots         = true

[SimpleTransfer]
type                 = "mimosa26"
max_depth_distance   = 5um
output_plots         = true

[DefaultDigitizer]
type                 = "mimosa26"
electronics_noise    = 13e
```

```
threshold              = 60e
threshold_smearing     = 4e
qdc_smearing           = 0e
qdc_resolution         = 1
output_plots           = true

# ===== DUT setup

[ElectricFieldReader]
name = "Monopix2_0"
model = "linear"
depletion_depth = 25um
bias_voltage = -3V
output_plots = true

[GenericPropagation]
name = "Monopix2_0"
temperature = 293K
charge_per_step = 50
integration_time = 100ns
output_plots = true

[SimpleTransfer]
name = "Monopix2_0"
max_depth_distance = 10um
output_plots = true

[DefaultDigitizer]
name = "Monopix2_0"
threshold = 100e
#electronics_noise = 80e
#threshold = 500e
#threshold_smearing = 30e
#qdc_smearing = 350e
#qdc_resolution = 16
#qdc_slope = 1.0376e
#qdc_offset = -500e
output_plots = true

[DetectorHistogrammer]
```