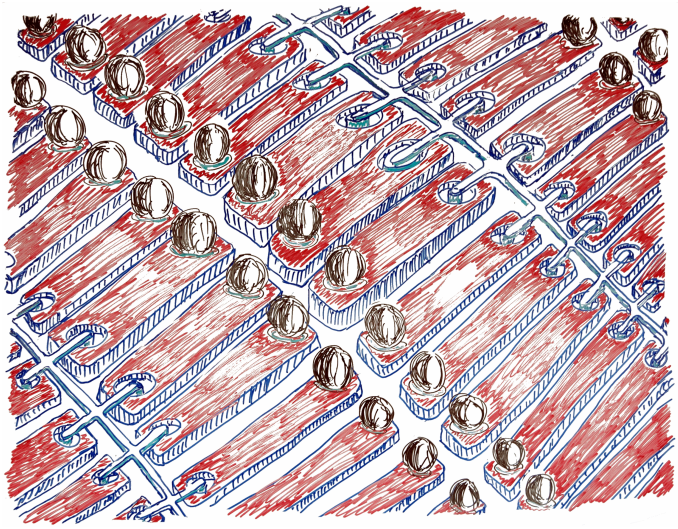
Dissertation
zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften in der Fakultät Physik der Technischen Universität Dortmund

Development and operation of a testbeam setup for qualification studies of ATLAS Pixel Sensors

vorgelegt von
Dipl.-Phys. Georg Troska

Januar 2012



Technische Universität Dortmund - Fakultät Physik
Experimentelle Physik IV

Contents

1. Introduction

The description of particles and their interaction in the Standard Model of particle physics has been proven to be very accurate up to now. Although this model is very successful, some basic phenomena cannot be explained or theoretical estimations are predicted but not yet observed. Famous candidates are the Higgs mechanism, introduced to explain the different masses of particles, the Grand Unified Theory, trying to unify all particles and interactions into one common theory and the super symmetry, which is trying to explain the existence of particles and exchange bosons in parallel.

In order to allow further, more accurate measurements of the Standard Model and its possible extensions, complex experiments are required. The recording of nature's reaction induced by the interaction of two colliding high energetic particles is one of the basic instruments to investigate elementary particles. The energy, set free by the collision, creates further particles with independent properties, such as momentum, charge and energy. With large high resolution detection systems these properties can be recorded.

The LHC is the instrument which induces the interactions of two colliding protons. At one of its interaction points the ATLAS detector is located. The innermost detector, called Pixel detector, located very closely to the interaction point, will gradually lose the ability to identify traversing particles during operating time. The interactions cause radiation damages in the silicon material of the detector. Before it will go completely blind, a new detector layer will be installed inside. As the position of this detector will be even closer to the interaction point a new technology is required. The ATLAS pixel sensor, being the part of this detector sensitive for crossing particles, will be the focus of this study. Furthermore, a complete rebuild of the inner detector of ATLAS is scheduled, with the upgrade of the LHC to higher luminosity. A detector in such a scenario needs to withstand even higher fluences, because of the greater radiation damage.

A new effect first discovered at silicon strip sensors causes a sensor to produce a greater charge signal than expected by extrapolating the available data common for the fluences of the current detector design. The effect occurs when operating a highly irradiated silicon sensor in very cold condition at very high voltages. Research at pixel sensors in order to confirm the effect here are useful, as the lifetime of the sensor can be increased by changing its environmental parameters. A better knowledge of these parameters is also important to quantify the limits of the detector services, like cooling and voltage supply. Strong indications that silicon pixel sensors are affected by the mentioned effect as well will be worked out in this study.

Before installing a new technology into the ATLAS detector one needs to acquire experience with it and needs to prove that the technology is sufficiently working even at the end of the lifetime of the new detector design. In order to achieve a quality attribute of the sensor, special properties of the sensor need to be known. These properties can be identified by performing so called testbeam experiments in which the trajectories of high

energetic particles, which cross the investigated sensor, are determined. The trajectories are being identified by the so called EUDET testbeam telescope.

In order to simulate the radiation damage caused by the particle interactions, test sensors have been irradiated in irradiation facilities before their operation in testbeams.

The focus of this study is the design and construction of a testbeam experiment which can be used to perform tests with ATLAS pixel sensors. During this thesis components of the EUDET telescope were combined with the common readout system of the ATLAS pixel detector. An integration at hardware and software level was done. In oder to identify the sufficient working of the investigated sensors, an online monitor was developed. Furthermore, the reconstruction chain for the testbeam data was modified for the purposes of the ATLAS pixel sensor.

Finally, a generic analysis of dedicated testbeam datasets will be presented. For the first time, highly resoluted testbeam data was recorded with ATLAS pixel sensors irradiated to fluences at the end of lifetime of the insertable b-layer and a pixel detector after an upgrade of the LHC to higher luminosity. Quality attributes of these sensors will be shown.

This thesis begins with an overview about the accelerator LHC and the ATLAS detector. In chapter 3 the fundamental background of the physics of silicon particle detectors will be given in order to explain the ATLAS pixel detector in detail in the next step. Afterwards, a more detailed motivation for this thesis - the coming upgrade of the pixel detector and the charge amplification effect - will be presented. The setup of the already existing testbeam telescope is outlined in chapter 6.

In chapter 7 the mechanics which were provided for the testbeam experiment will be described. In the following, the integration of the ATLAS pixel readout system into the testbeam telescope's data acquisition system is presented. Chapters 9 and 10 contain detailed information about the software used in testbeam, especially the online monitor and the offline software, including the full reconstruction chain. In the last chapters the data taking of the testbeams and the analysis of the data taken is explained.

2. The Large Hadron Collider and the ATLAS Detector

In order to do fundamental research in the context of particle physics, the international laboratory CERN[1] operates a large complex of accelerators and particle detectors. The most significant challenges are the search for the Higgs particle and the confirmation of its theory, the search for heavy SUSY[2] particles and investigations of the quarks of the third generation. Additionally, precision measurements of decay-channels, particle-masses and production cross-sections will be obtained by CERN's experiments to confirm the Standard-Model of particle physics.

Four detectors are connected to the LHC[3]. Two of them are multi-purpose: ATLAS[4] and CMS[5]. For dedicated studies in the area of b-quark physics LHCb[6] was built. ALICE[7] is investigating quark-gluon plasma by studying heavy-ion collisions.

Chapter 2.1 will give an overview about the LHC. ATLAS will be described in chapter 2.2.

2.1. LHC

LHC is the last part in a chain of accelerators at CERN. Figure 2.1 is a schematical illustration of the complex. The initial injection of protons into the complex is realised at LINAC2[8]. The nucleus of a hydrogen atom is separated from its electron in the shell by ionising the atom. Together with many other protons the remaining nucleus will be accelerated to an energy of 50 MeV in the LINAC2. After leaving the LINAC2, the protons will be filled into the PSB[9] (1.4 GeV), then into the PS[10] (26 GeV) and into the SPS[11] (450 GeV) [53]. From the SPS the protons can be injected into the LHC clockwise and counter-clockwise using the transfer-tunnels TI2 and TI8 (compare fig. 2.1).

The LHC has a circumference of approximately 27 km. The protons will be accelerated further to a design energy of 7 TeV in both directions. The magnetic fields needed to

[1] **C**onseil **E**uropéen pour la **R**echerche **N**ucléaire
[2] **SU**per **SY**mmetry
[3] **L**arge **H**adron **C**ollider
[4] **A** **T**oroidal **L**HC **A**pparatu**S**
[5] **C**ompact **M**uon **S**olenoid
[6] **LHC**-**b**eauty
[7] **A** **L**arge **I**on **C**ollider **E**xperiment
[8] **LIN**ear **AC**celerator **2** - accelerator at CERN
[9] **P**roton **S**ynchrotron **B**ooster - accelerator at CERN
[10] **P**roton **S**ynchrotron - accelerator at CERN
[11] **S**uper **P**roton **S**ynchrotron

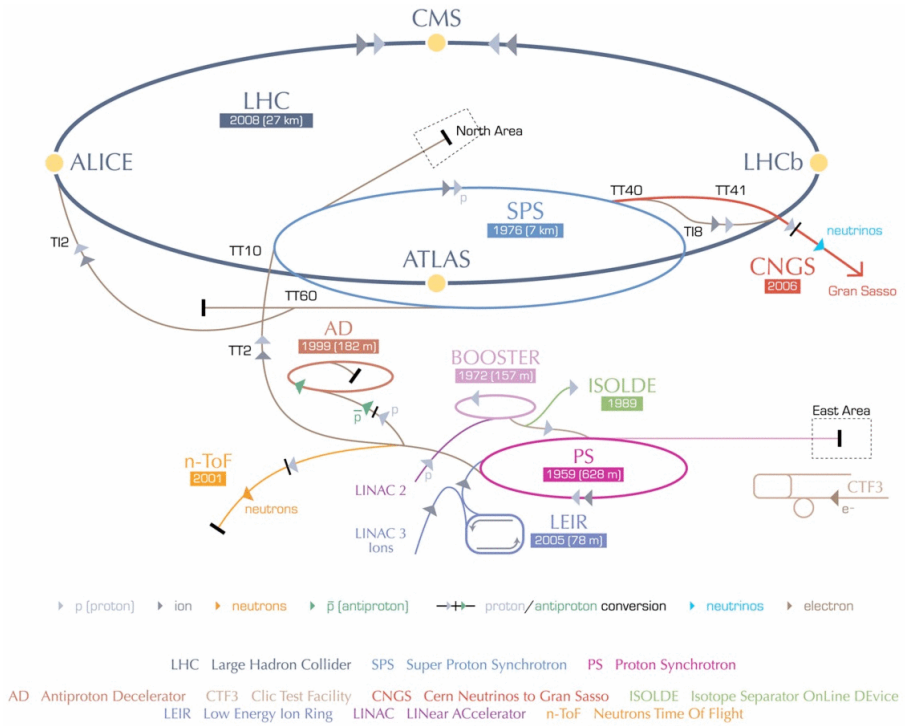


Figure 2.1.: Schematical illustration of CERN's accelerator complex [14]

keep the beam in the circle have a strength of 8.4 T. The current in the superconducting magnets reaches 11700 A in this case.

In figure 2.2 one can see an illustration of the LHC with its connected experiments. The LHC was built into the tunnel that was created for the prior electron-positron-accelerator LEP[1].

At four positions the two beams circulating in opposite directions cross each other. Around these points particle detectors (ATLAS, ALICE, CMS and LHCb) were built to investigate the consequences of the proton interactions. The LHC is designed for a center-of-mass energy of $\sqrt{s} = 14\,\text{TeV}$ and a luminosity of $L = 10^{34}\,\text{cm}^{-2}s^{-1}$ at a bunch crossing rate of 40 MHz with roughly 10^{11} protons per bunch [53].

2.1.1. Current machine settings

At the time being the LHC has not reached its design energy or instantaneous luminosity. The time schedule for the commissioning of the accelerator was heavily influenced by an accident at the LHC in 2009: When the magnetic field was increased to reach the maximum beam energy a missarranged junction between two magnets could not withstand the high current and broke. The resulting heat dissipation led to an enormous pressure in the helium cooling system which caused further damage in several magnets. Almost one whole sector[2] of the accelerator had to be replaced. Therefore the time schedule was delayed for more than one year. As a consequence, CERN's administration decided

[1] **L**arge **E**lectron-**P**ositron Collider
[2] The corresponding sector is called sector 34

to run the LHC at half of its design energy ($\sqrt{s} = 7\,\text{TeV}$) for the next years. The accelerator physicists are still commissioning the machine and are continuously increasing the instantaneous luminosity. At August, 5th 2011 1380 bunches were circulating in the LHC in both directions. At this day the integrated luminosity was roughly $1\,\text{fb}^{-1}$ and the instantaneous luminosity $2,05 \cdot 10^{33}\,\text{s}^{-1}\text{cm}^{-2}$.

2.1.2. LHC-Upgrade

Although the design luminosity of the LHC has not been reached yet, the development of future accelerators is that extensive that first plannings have started already. The LHC upgrade is planned in two phases [73]: For phase 1 a replacement of the LINAC2, which was built in 1978 and is increasingly unreliable, by the LINAC4[1] is planned for 2017. In addition to that, an upgrade of the interaction region magnets and the collimation systems of the experiments will be necessary. With the beginning of phase 2 (scheduled for 2021) the accelerator chain will inject the protons through the PS2[2], as a replacement for the PS and an improved SPS which could be called SPS+[3]. In order to cope with the new PS2 the SPS+ will have new injector- and ejector-magnets. The expected luminosity of the new accelerator system SLHC[4] will be $3000\,\text{fb}^{-1}$ at $\approx 10^{35}\,\text{cm}^{-2}\text{s}^{-1}$ particles after five years. The plan to double the center-of-mass energy of LHC to $\sqrt{s} = 28\,\text{TeV}$ in a so called DLHC[5] accelerator could be a possible project for the future.

[1] **LIN**ear **AC**celerator **4** - accelerator at CERN
[2] **P**roton **S**ynchrotron **2** - accelerator at CERN
[3] improoved **SPS**
[4] **S**uper **L**arge **H**adron **C**ollider
[5] **D**ouble Energy **LHC**

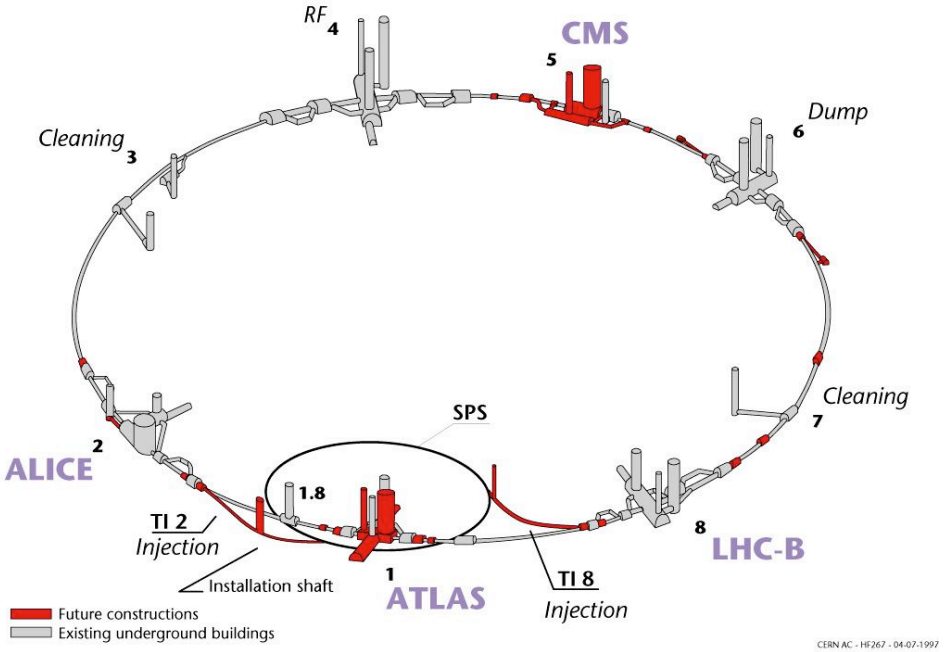


Figure 2.2.: Schematical illustration of the LHC-accelerator with its connected experiments and service caverns [14]

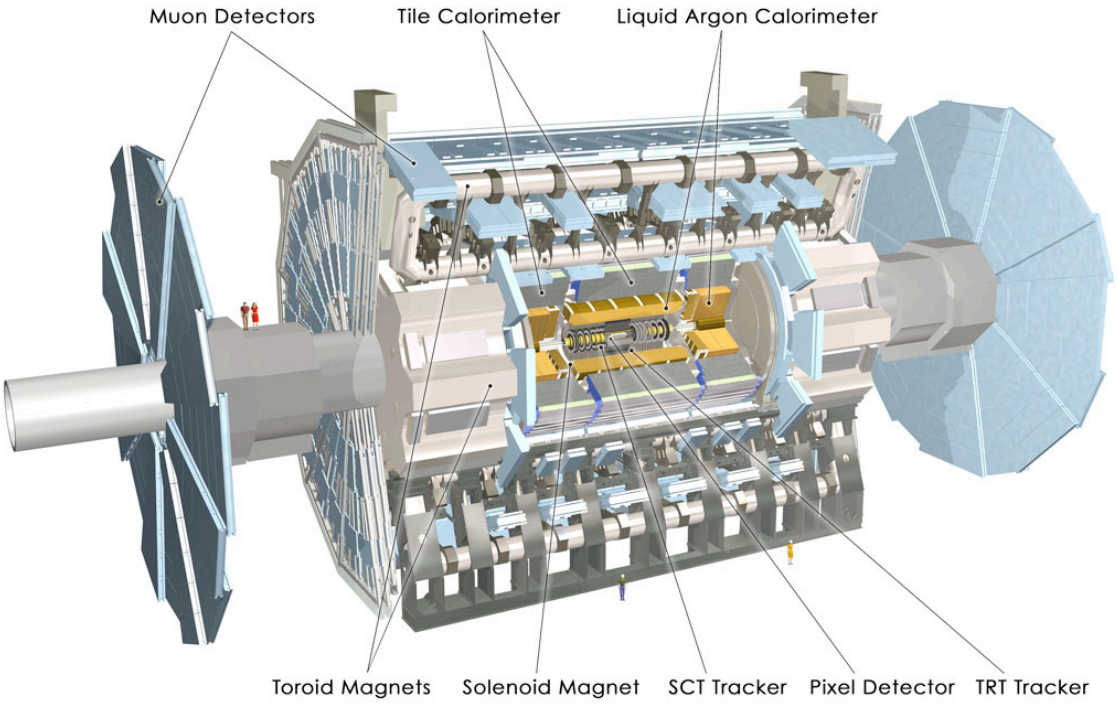


Figure 2.3.: Illustration of the ATLAS-Detector [6]

2.2. ATLAS

ATLAS is one of the multi-purpose detectors at CERN, LHC. It has an overall diameter of 25 m, a length of 46 m and a weight of 7000 t [53]. Figure 2.3 shows a computer aided illustration of the detector. As typical of a particle detector, ATLAS is arranged like an onion with multiple sub-detectors as its layers. The detector systems are divided (from inside to outside) into the tracking system, which is manifested by the inner detector, the calorimeters and the muon-spectrometer. The big toroidal magnet system in the muon-spectrometer is characteristic for ATLAS.

In the next chapters the subdetector-parts of ATLAS will be explained. A special focus is placed on the ATLAS pixel detector as this thesis will describe developments of upgrades at this subdetector later on.

2.2.1. Inner Detector

The inner detector is the part of the detector responsible for tracking. This means that particle trajectories coming from the initial interaction point will be reconstructed by determining the hit positions on the sensors that are located around the beampipe. The particles' momentum can be measured, as the inner detector is surrounded by a solenoid magnetic field. As the traversing particles interact with the matter of the detector itself as described in chapter 3.1, it is important to minimise the amount of matter in the tracking system. Therefore almost all material used in this part of the detector is of low Z (like

aluminum, titanium, beryllium, carbon). The inner detector consists of three different detector-types: The Pixel-Detector, the SCT[1] and the TRT[2].

2.2.2. Transition Radiation Detector

The TRT is the outermost part of the inner detector. It is 2.3 m in diameter and 7 m in length. It consists of tubes with a diameter of 4 mm which are filled with a gas-mixture, consisting mainly of Xenon[73]. A relativistic particle crossing the TRT will create transition radiation which can be detected by a wire in the middle of the tube. A typical particle traversing the TRT will trigger 36 channels in the detector.

2.2.3. Silicon Tracker

Coming closer to the interaction point, the resolution of the attached detectors needs to increase as track density will increase as well due to the smaller radia. As a consequence of the higher track density, the particle fluence will increase as well. This requires radiation-hard components in the tracking system.

The area covered by the SCT is 63 m^2. To minimise the costs of a silicon tracker system this size, the chosen technique for the sensors and their readout is simple and low cost.

The SCT is made of silicon microstrip sensors that are produced in a single sided process. The strip sensors contain 786 strips with 80 μm pitch. Pairs of sensors are connected to each other via wirebonds and form 12.8 cm long strips. A read out electronic is attached to the side of the strip sensors in order to connect to the sensor via wirebonds.

By using a strip detector the position of a traversing particle can be estimated in one dimension only. Therefore, in the SCT, two layers of strip-sensors are rotated 40 mrad relatively to one another. Unfortunately, this system does not work in a high intensity tracking region, as the signals from the two layers are ambiguous when more than one particle traverses the crossed layers. In order to solve these ambiguities a pixelated detector is needed. The SCT has 6.2 million read-out channels, consists of four double layers in the barrel region and nine double layers in the endcaps of both sides to cover the forward region.

2.2.4. Pixel Detector

The subdetector located closest to the interaction point is the pixel detector. It is described in detail in chapter 4.

2.2.5. Calorimeters

There are two different kinds of calorimeters in ATLAS. The one is the EMCAL[3] surrounding the inner detector and inside the HCAL[4]. Both calorimeters are sampling calori-

[1] **S**emi**C**onductor **T**racker
[2] **T**ransition **R**adiation **T**racker
[3] **E**lectro**M**agnetic **CAL**orimeter
[4] **H**adronic **CAL**orimeter

meters. That means that these calorimeters consist of alternating layers of active detector material and passive absorber material.

The EMCAL is a LAr[1] detector with accordion shaped Kapton electrodes and lead absorber parts. [7]. Roughly 190000 channels can be read out. The EMCAL has a total material amount of $> 24\,X_0$ (barrel) and $> 26\,X_0$ (endcaps). The meaning of the radiationlengh X_0 will be described in chapter 3.1.2.

The HCAL, also called TileCAL, is made of alternating iron-plates of 14 mm as absorbers and 3 mm scintiallating material. They are placed radially around the interaction point. The scintillating material is read out by PMT[2]s. Roughly 10000 PMT are used to distinguish between active space-angular and energy.

2.2.6. Muon Spectrometer

The muon spectrometer uses the fact that the momentum of a known particle can be determined by measuring the bending radius in a magnetic field. The muon spectrometer is made of high precision tracking chambers which are mounted in three layers at different radii around the interaction point. The spectrometer is mounted inside the torroidal magnetic field of ATLAS. This magnetic field is created by an air coil torriod magnet sytem. The magnetic field inside the toroid is limited to 3.9 T in the barrel region and 4.1 T in its endcaps [8] [9].

[1] **L**iquid **Ar**gon
[2] **P**hoto **M**ultiplier **T**ube

3. Physics of Planar Silicon Detectors

This chapter is a basic description of the properties and working principles of planar silicon detectors. It starts with the explanations of unirradiated sensors and will go on with the description of irradiated sensors. References for the used equations can be found in [50] and [64]. A good summary is given in [73].

3.1. Physics of unirradiated sensors

The mechanism making silicon particle detectors work is caused by the fact that particles traversing the detector create electron hole pairs that can be measured as charges at electrodes of the sensor. The next chapters will summarise the interactions of particle with matter. The working principle of a planar silicon detector will be described furthermore.

3.1.1. Ionising Energy loss of photons in matter

Interactions of photons with matter can be distinguished in three effects. The Photoelectric Effect, the Compton Effect and the electron pair production.

3.1.1.1. Photoelectric Effect

The Photoelectric Effect is the dominating effect for energies up to 30 keV.

With the Photoelectric Effect an electron in bound state of an atom is knocked out of its position by a photon, when the energy of the photon is higher than the binding energy E_{B}. The resulting energy of the electron will be

$$E_{\mathrm{e}^-,\,\mathrm{kin}} = E_\gamma - E_{\mathrm{B}}. \tag{3.1}$$

3.1.1.2. Compton Scattering

Whereas the Photoelectric Effect describes the destruction of the incoming γ by transfering its energy to an electron the Compton Scattering shows that the conservation of the relativistic four momentum (and with thus also the energy) is granted by an inelastic scattering. The effect describes the conversion of an incoming γ into a lower energetic outgoing γ by hitting an electron. Assuming high energy photons the electron can be treated as approximately free. With

$$E'_\gamma = \frac{E_\gamma}{1 + \frac{E_\gamma}{m_e c^2}(1 - \cos\Theta)} \tag{3.2}$$

in which E_γ and E'_γ is the energy of the γ before and after the scattering the distribution of the scatter angle Θ can be determined. Finally the kinetic energy of the photon will be $E_\gamma - E'_\gamma$. If the sensor is thick enough the whole energy of the γ will be deposited in a compton cascade.

3.1.1.3. Pair Production

When the energy of the incoming γ is high enough pair production becomes possible. This effect explains the destruction of the incoming photon by transfering all its energy into another body. Often only the nucleus is seen as a possible partner, but a reaction with the atomic shell is also possible.

Simplified, the minimum energy, when hitting the nucleus is the energy at rest of the new created electron positron pair $2m_e$. A better calculation takes the recoil of the nucleus into account. Therefore, the minimum energy for pair production is

$$E_\gamma = 2m_e(1 + \frac{m_\mathrm{e}}{m_\mathrm{N}}) \tag{3.3}$$

in which m_N is the mass of the nucleus.

The minimum energy for pair production when hitting an electron in a bound state is higher than for the nucleus. Hitting the electron at rest, the momentum of the incoming photon is shared to the hit electron and the new electron positron pair. A momentum conversion can only be reached when passing the minimum kinetic energy of the pair ($2m_e$) also to the bound electron moving into the opposite direction. Therefore, the minimum energy for pair production in the atomic shell is

$$E_\gamma = 4m_e \tag{3.4}$$

.

3.1.2. Energy loss of charged particles in matter

There a several kinds of interactions for charged particles in matter. The radiationlength, which is one important value in this context, is defined as

$$X_0 = \frac{4Z^2 N_A \rho \alpha r_e^2}{A} \ln \frac{183}{\sqrt[3]{Z}}, \tag{3.5}$$

where Z is the atomic number of the matter[1], N_A is Avogadro's constant, ρ is the density of the matter[2], α is the fine structure constant and r_e the classical radius of an electron.

3.1.2.1. Bremsstrahlung

A charged particle crossing matter will be accelerated by charges in the matter. The acceleration of charged particles causes radiation called Bremsstrahlung. The average

[1] The atomic number Z for silicon is 14
[2] The density ρ for silicon is $2.33\,\mathrm{g\,cm^{-3}}$

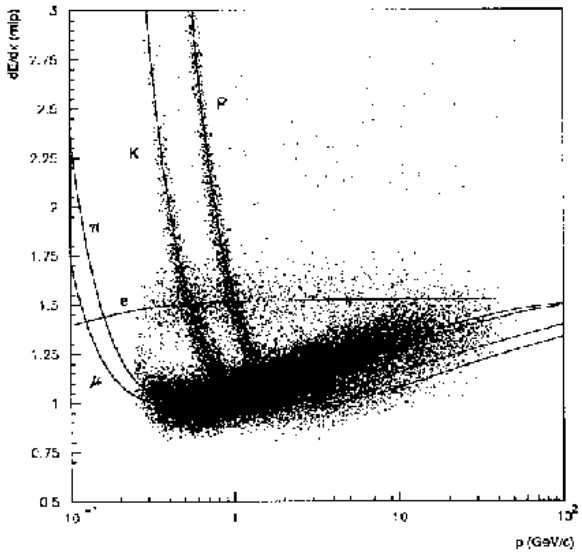


Figure 3.1.: Dependency of $-\frac{1}{\rho}\frac{dE}{dx}$ of the Bethe-Bloch formula for heavy particles [69]

loss of energy per distance $-\frac{dE}{dx}$ caused by Bremsstrahlung can be calculated to

$$-\frac{dE}{dx} = \frac{E_\gamma}{X_0} \propto \frac{1}{m^2}, \tag{3.6}$$

where E_γ is the bremsstrahlungs spectrum.

3.1.2.2. The Bethe-Bloch formula for heavy particles

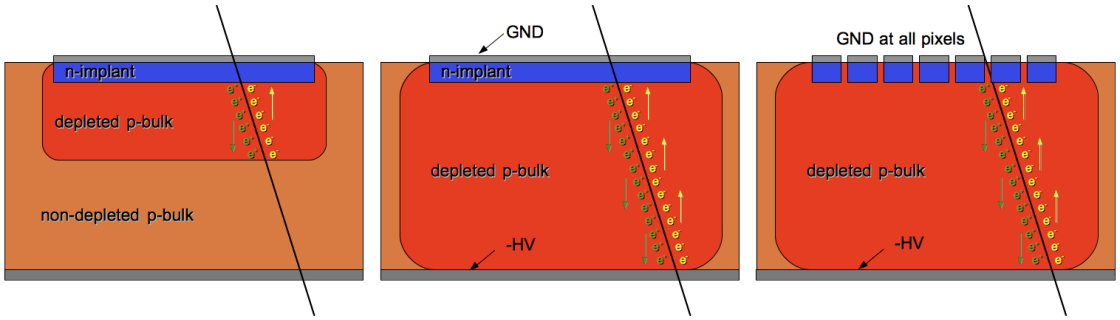
A quantification for multiple kinds of interactions such as elastic and inelastic energy loss of heavier charged particles and Cerenkov radiation but also interaction of low energetic α-particles is realised in the Bethe-Bloch formula [16]:

$$-\frac{dE}{dx} = 4\pi N_A r_e^2 m_e c^2 z^2 \frac{Z}{A}\frac{1}{\beta^2}\left[\ln\left(\frac{2m_e c^2\gamma^2\beta^2}{I}\right) - \beta^2 - \frac{\delta}{2}\right] \tag{3.7}$$

where

c	is the speed of light
z	is the charge of the traversing particle in units of eV
Z	is the atomic number of the medium[a]
A	is the atomic weight of the medium[b]
m_e	is the mass of an electron
r_e	is the classical radius of an electron
N_A	is Avogardro's number ($6.022 \cdot 10^{23}\,\mathrm{Mol}^{-1}$)
β	is the velocity of the traversing particle in units of c
γ	is $(1-\beta^2)^{-\frac{1}{2}}$
I	is the effective ionisation potential[c]
δ	is the density correction.

[a] For silicon the atomic number Z is 14
[b] For silicon the atomic weight A is 28.09
[c] For silicon the effective ionisation potential I is 137 eV



(a) simple pn-junction (b) biased pn-junction (c) pixelated pn-junction

Figure 3.2.: Working principle of (a) a simple non-biased sensor diode, (b) a depleted sensor diode and (c) a pixelated sensor. Particle tracks and the production of electron hole pairs are visualised as well

Figure 3.1 shows the dependency of Bethe-Bloch's normalized energy loss $-\frac{1}{\rho}\frac{dE}{dx}$ with respect to the traversing particle energy. One can see that the dependency has a minimum at

$$-\frac{1}{\rho}\frac{dE}{dx} \approx 1.5\,\frac{\mathrm{MeV}}{\mathrm{g\,cm^{-2}}} \tag{3.8}$$

Traversing particles in that region are called MIPs[1]

3.1.3. Working principles of planar silicon sensors.

The basic working principle of a planar silicon detector is the usage of a pn-junction. Typically silicon is used, as the material can be arranged with common photo lithographical processes. Furthermore, silicon is radiation hard.

The simplest planar silicon sensor is a p-doped substrate and an n-implant induced at one side. With the processes described in the last chapters electron hole pairs will be created along a particle tracks trajectory. This is shown in figure 3.2a. In the area of the pn-junction a region free of charge carriers will appear, which is called depleted zone. The electron hole pairs created in this region will be separated by the surrounding electrical field of this region. Therefore, the charge carriers can be measured as electrical current either at the n- or the p-side of the silicon material.

By applying a reverse bias voltage to the diode made of the pn-junction the depleted zone will grow and will cover the complete sensor substrate finally (compare figure 3.2b). The necessary voltage to fully deplete the sensor is called depletion voltage U_{dep}.

This simple sensor described is sensitive for charged particles and photons and can be used to identify those. One feature, the spatial resolution, is still missing. Instead of using a single n-implant at the sensor surface, the implant can be segmented either in only one direction (strips) or in both directions (pixels). This is visualised in figure 3.2c. In order to apply a reverse bias voltage to the sensor, each strip or pixel implant needs to have the same potential, which means that all implants need to be contacted individually. The same contacts can be used to read out the charge signals for each implant.

[1] **M**inimal **I**onising **P**articles

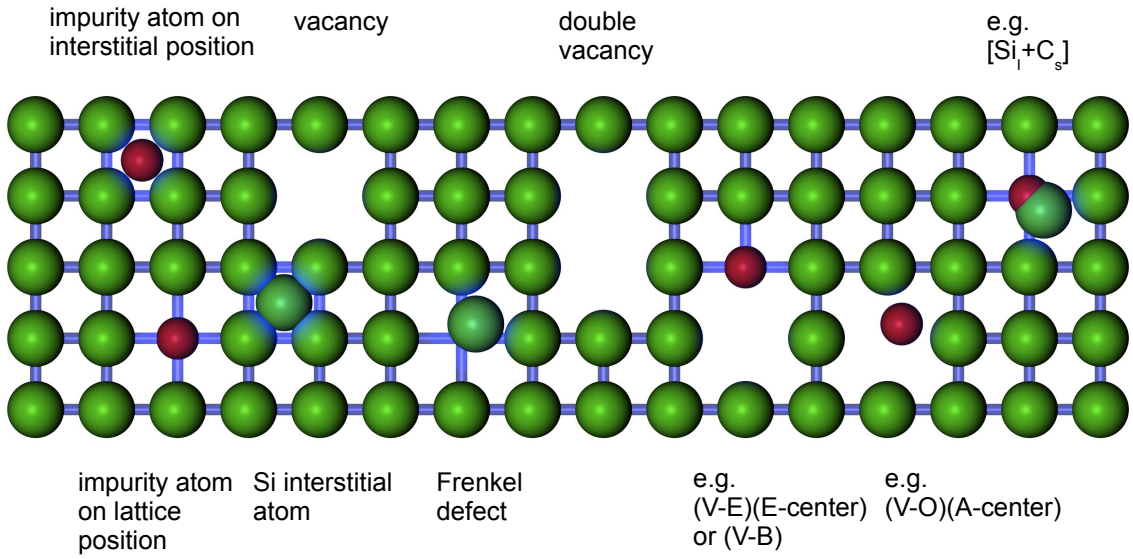


Figure 3.3.: Schematic illustration showing different crystal defects. Original taken from [70] and modified

It is common to use n-implants as pixel contacts, as typically the electron signals are read out. For practical reasons the applied potential of the pixel implants is ground, so that a negative voltage needs to be applied to the sensor's backside.

A more detailed description of the ATLAS pixel sensor is given in chapter 4.2.

3.2. Physics of irradiated sensors

Irradiation influences the working conditions of silicon sensors. The next chapters will give an overview about the effects.

3.2.1. Definitions of Crystal Defect Types

Particles can cross a silicon sensor and deposit energy without effecting the sensor. These effects were described in chapter 3.1. Effects like these will always revert the crystal into its prior state. The so called non-ionising radiation causes effects that will lead to permanent crystal damages by interacting with the atoms in the lattice.

Removing an atom from its position in the lattice requires a minimal energy of 15 keV. This hit atom is called PKA[1]. If the energy transferred to the PKA is high enough it causes further damages in the lattice leading to cluster defects. Finally the traversing particle can convert the nucleus, too.

The NIEL[2] thesis describes that only the non-ionising energy loss has to be taken into account. Furthermore, only the interaction with the PKA depends on the kind of radiation [73]. This means that all further damages are caused by the PKA and not by the

[1] **P**rimary **K**nock-on **A**tom
[2] **N**on **I**onising **E**nergy **L**oss

traversing particles itself. This allows a uniform way to describe the varying effects of different particles such as electrons, pions or hadrons on the PKA. Therefore, one uses the non-ionising energy loss of a fictional neutron with an energy of 1 MeV as the so called standard irradiation. All other kinds of particles are scaled linearly to the standard irradiation which usually has the unit $\mathrm{n_{eq}\,cm^{-2}}$.

Depending on the sensor's temperature the point defects can move inside the crystal lattice. This can lead to a benefical effect called annealing (chapter 3.2.5) or results in more complex structures in the lattice as the defects can combine with each other or other impurities.

Figure 3.3 shows schematically differences of defects in the lattice.

3.2.2. Leakage Current

Irradiation has an influence on the reverse bias current of a silicon sensor. The defects mentioned in section 3.2.1 can lead to the creation of mid-gap states [77]. In good approximation the increase can be described for nearly all materials and fluences by the equation [73]:

$$I_{\mathrm{leak}} = \alpha \cdot \Phi_{\mathrm{eq}} \cdot V, \tag{3.9}$$

where α is the current related damage and Φ_{eq} the standard irradiation mentioned in section 3.2.1. A typical value is e.g. [58]

$$\alpha(80\,\mathrm{min}, 60\,^{\circ}\mathrm{C}) = (3.99 \pm 0.03)\,\mathrm{fA\,m^{-1}} \tag{3.10}$$

after a heat treatment at 60 °C for 80 min.

Beside its dependency on the fluence, the instantaneous leakage current of the sensor depends on the actual sensor temperature. The dependency can be described with this equation [60]:

$$\frac{I(T)}{I(T_0)} = \left(\frac{T}{T_0}\right)^2 \exp\left(-\frac{E_g}{2k_B}\left(\frac{1}{T} - \frac{1}{T_0}\right)\right). \tag{3.11}$$

Here $I(T_0)$ is the leakage current at a known temperature T_0 in units of Kelvin. E_g is the band gap in silicon which is $E_g = 1.2\,\mathrm{eV}$ [15].

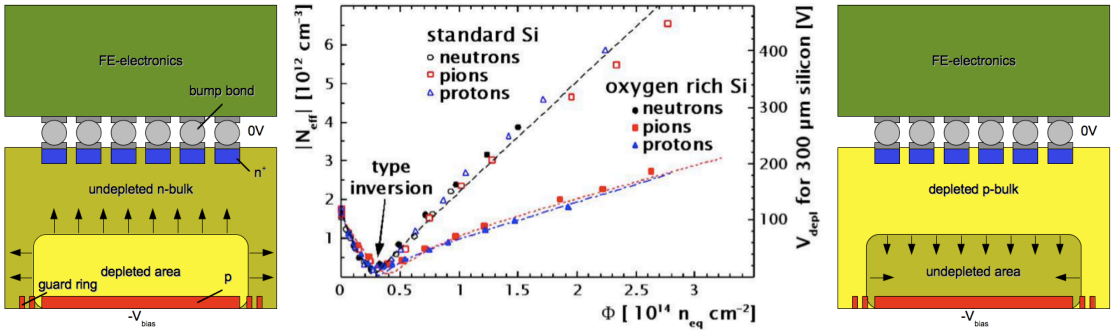
3.2.3. Effective Doping Concentration and Type Inversion

During irradiation the effective doping concentration in n-type material

$$N_{\mathrm{eff}} = N_{\mathrm{D}} - N_{\mathrm{A}}, \tag{3.12}$$

where N_{D} is the number of donators and N_{A} is the number of acceptors, decreases. First it appears to become undoped; when irradiating further on it begins to behave p-type like.

Displacement damages acting as acceptor-like states are the most important reason for this behavior. They are populated by electrons from the bulk, carrying negative space charge. This is the reason for the so called type inversion.



(a) before irradiation (b) effective doping concentration (c) after irradiation

Figure 3.4.: Schematic illustration of the growth of the depletion zone before irradiation (a) and after irradiation and type inversion (c) for a n^+-in-n sensor. The required guard ring is implanted on the p-side of the sensor. Original taken from [87] and modified. (b) shows the effective doping concentration of such a sensor [26]

With the changing of the effective doping concentration comes the change in the depletion voltage

$$V_{\text{dep}} = \frac{e_0 |N_{\text{eff}}|}{\varepsilon \varepsilon_0} \frac{d^2}{2}. \tag{3.13}$$

The type inversion has many consequences for n^+-in-n pixel sensors such as the ATLAS pixel sensor: Before type inversion the sensor cannot be operated in an underdepleted configuration, as the n^+ pixel implants are connected to each other through the non-depleted regions in the n-bulk. In this configuration the depletion zone grows from the p-side of the sensor to the pixelated n-side (compare figure 3.4a). After type inversion the depletion zone grows from the n-side of the sensor (compare figure 3.4c). Even when running underdepleted the depletion zone will cover the pixels n^+-implants which enables a spatial charge separation. The effective doping concentration N_{eff} in dependency of the fluence Φ_{eq} is shown in figure 3.4b.

3.2.4. Trapping

In an irradiated silicon sensor the charge carriers of the pixel signal are captured by damages in the lattice. They are released too late to contribute to the charge signal . The so called inverse trapping propability $\tau^{-1}_{\text{eff}_{\text{e,h}}}$ can be described by

$$\frac{1}{\tau_{\text{eff}_{\text{e,h}}}} = \beta_{\text{e,h}} \cdot \Phi_{\text{eq}}, \quad \text{with} \tag{3.14}$$

$$\beta_{\text{e}} = (4.2 \pm 0.3) 10^{-16}\,\text{cm}^2/\text{ns}, \quad \text{and} \tag{3.15}$$

$$\beta_{\text{h}} = (6.1 \pm 0.3) 10^{-16}\,\text{cm}^2/\text{ns} \tag{3.16}$$

The amount of drifting charge decreases with time due to trapping as

$$N_{\text{e,h}}(t) = N_{\text{e,h}}(0) \exp\left(\frac{-t}{\tau_{\text{eff}}}\right), \tag{3.17}$$

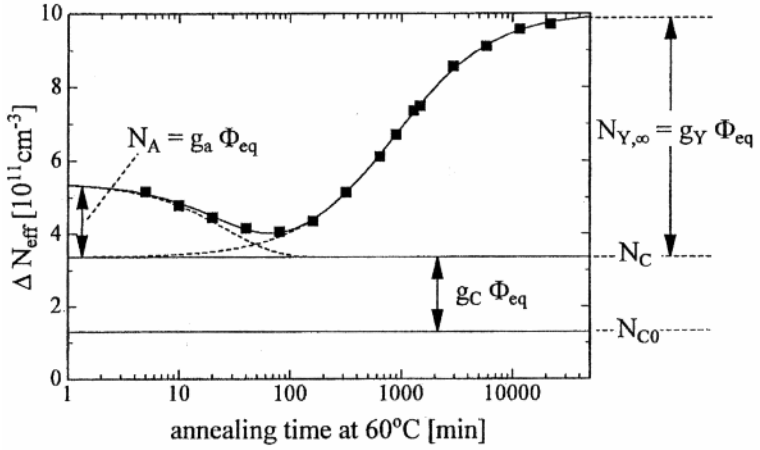


Figure 3.5.: Diagram illustrating the time dependency of the annealing behaviour by showing the effective doping concentration vs. the time at a fixed temperature of 60 °C. The measured points are taken with a diode structure of 25 kΩ cm after an irradiation of $1.4 \cdot 10^{13} \mathrm{n_{eq}\,cm^{-2}}$ [58]

in which $N_{\mathrm{e,h}}(0)$ is the initial number of generated electron hole pairs. [42]

3.2.5. Annealing

The effective doping concentration ΔN_{eff} of a silicon sensor after a fixed irradiation is a function of time and temperature.

It can be described as

$$\Delta N_{\mathrm{eff}}(\Phi_{\mathrm{eq}}, t) = N_{\mathrm{C}}(\Phi_{\mathrm{eq}}) + N_{\mathrm{A}}(\Phi_{\mathrm{eq}}, t) + N_{\mathrm{Y}}(\Phi_{\mathrm{eq}}, t). \tag{3.18}$$

This definition includes three independent effects causing the behaviour illustrated in figure 3.5. N_{C} decribes the constant and time independent part in the definition. It stands for the stable damage in the crystal. The so called 'Short Term' or 'Beneficial Annealing' is parametrised by the function of N_{A}. The function depends on the temperature and explains the decrease of the depletion voltage after type inversion. The 'Long Term' or 'Reverse Annealing' is represented by N_{Y}, which is as temperature dependent as N_{A} and is responsible for the increase of the depletion voltage after passing the minimum of the effective doping concentration.

4. The ATLAS Pixel Detector

The pixel detector covers only a small area of roughly $2\,\mathrm{m}^2$. It consists of three layers in the barrel region and three endcaps on both sides. The barrel consists of 112 identical staves arranged circularly around the beam pipe (fig. 4.1). 13 so called pixel modules are fixed on one stave. The total number of modules in the detector is 1744.

4.1. ATLAS Pixel Module

The ATLAS pixel module is the smallest unity of particle detectors of the ATLAS pixel detector. Fig. 4.2 shows a schematic illustration of one module of the pixel detector. One can see that the module consists of three layers: Located in the middle is the ATLAS pixel sensor, which will be described in chapter 4.2. The analog signals obtained by the sensor need to be transmitted into the readout electronics. Therefore, the so called FE[1] chips are located directly underneath the pixel sensor. 16 FE chips are needed to read out the whole active area of the sensor. The hybrid of pixel sensor and connected FE chips is called bare module.

The bare module becomes a pixel module when attaching it to the module flex. This is a PCB[2] made of Kapton. Apart from the conducting paths, it carries several passive electronic elements, such as decoupling capacitors, resistors and a temperature sensor. The most important part located on the module's flex is the MCC[3]. The digital information of the FEs are passed to the flex and then to the MCC via wirebonds. It functions as a multiplexer and distributor in the ATLAS pixel datastream system. The so called pigtail, carrying a replugable Robinson connector, is attached to the flex. The pigtail connector is the interface to the ATLAS pixel module. It contains supply voltage and signal lines.

4.2. ATLAS Pixel Sensor

The sensitive part of the pixel detector is the pixel sensor. The sensor is made of silicon material which was produced in a double-sided process. Instead of reading out the charge signal at the side of the sensor, as it is done for the SCT, the pixels on the sensor are read out one by one via so called bump bonds. In this technique, the pixel sensor is pressed flatly to the readout electronics to enable 2880 individual connections between the FE and the sensor. Figure 4.3 shows a computer aided illustration of some sensor processing masks.

[1] **F**ront**E**nd Electronic
[2] **P**rinted **C**ircuit **B**oard
[3] **M**odule **C**ontrol **C**hip

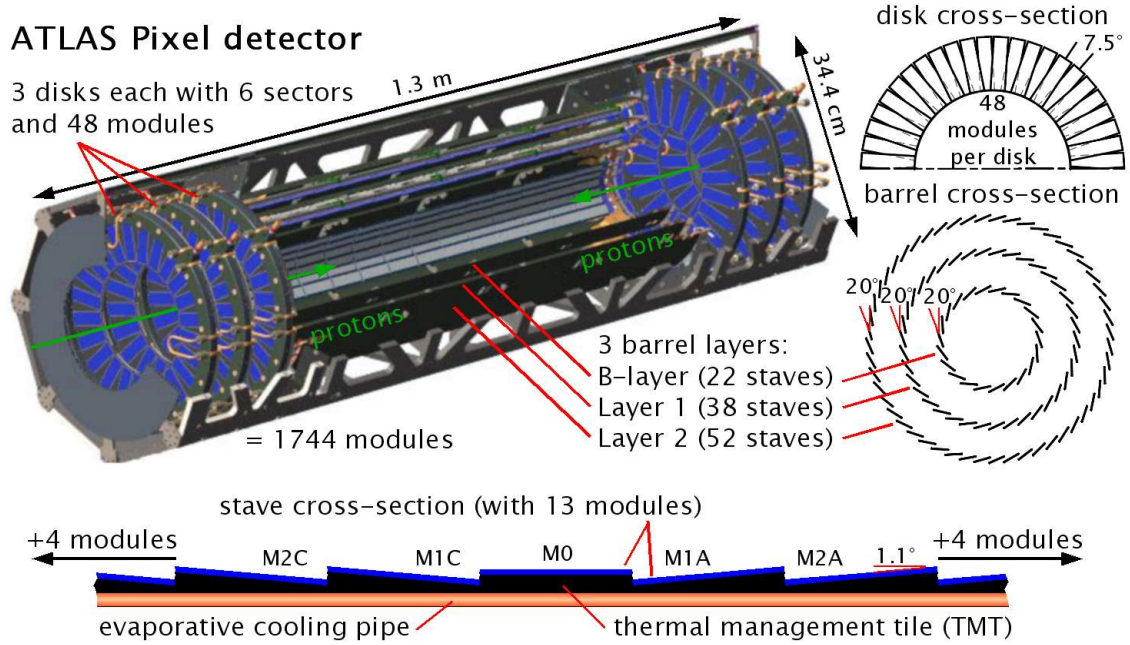


Figure 4.1.: Schematic drawing of the ATLAS pixel detector. One can distinguish the three barrels, the disks, out of which the endcaps are made and the arrangement of the staves in the detector [6, 18, 26]

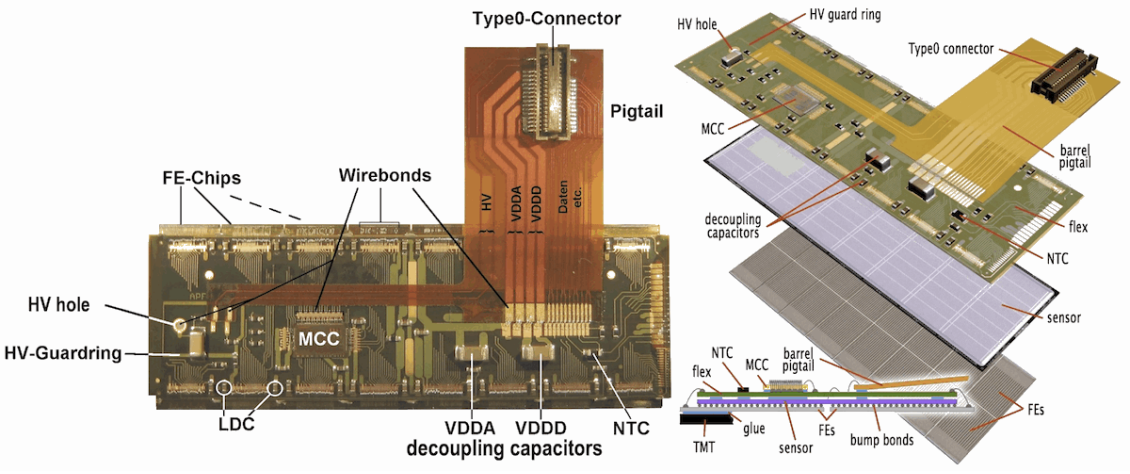
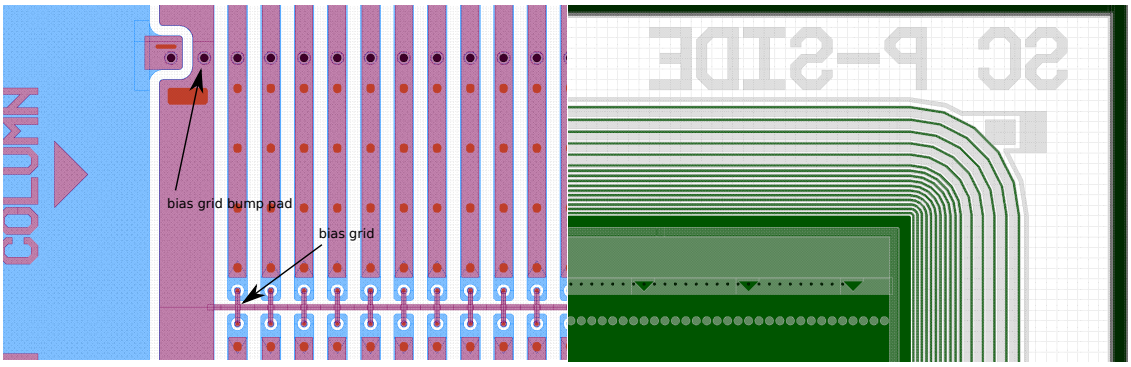


Figure 4.2.: Left: Photograph of an ATLAS pixel module with explanations; Right: Computer aided exploded drawing showing the different layers of FEs, pixel sensor and flex. Originals taken from [26] and modified



(a) pixelated n-side (b) non-pixelated p-side

Figure 4.3.: Computer aided drawing of the sensor design. In (a) one can see the rectangular pixels, the bump bond contact pads and the bias grid. (b) shows the edge of the sensor's backside with the guard rings [73]

The active area[1] of the ATLAS pixel sensor is $60.8 \times 16.4\,\text{mm}$. It is divided into 46080 individual pixel channels that are read out by 16 FE chips in parallel. A thicker sensor generates more charge when a particle is traversing. Whereas a thin sensor minimises multiple scattering in the detector. A value of $250\,\mu\text{m}$ was chosen as it leads to a most probable electron signal of $19600\,\text{e}^-$. Even after irradiation up to the end of the sensor's lifetime, the resulting signal will be higher than the foreseen threshold value of $4000\,\text{e}^-$. In comparison to industrial microchip productions, the high purity of the silicon material is important. The double-sided process of the wafer is not typical either.

The sensor is a n^+-in-n-sensor. This means that the bulk is of n-type with a low doping concentration. One surface is processed with the pattern of the pixels, realised by a high n^+ doping concentration in the pixel implants. The other surface is p doped.

4.2.1. Guard Rings

The cutting edge of the sensor is conductive due to the crystal damages in the lattice induced by the mechanical dicing. To avoid currents over the edges, so called guard rings are used. These guard rings are floating p^+ implantations surrounding the main p^+ implantation to create a potential barrier to the sensor's edge. The guard rings are organised as a multi guard rings structure, to allow a step-by-step potential decrease. In the n^+-in-n sensor technology the n^+-pixel implants are at ground potential, whereas the bias voltage is applied to the p-side. This fact is important as the FE chip can be DC[2] coupled to the sensor. So, there is no need to compensate a high DC bias voltage. Furthermore, the FE-I3[3] can only interpret negative charge signals. Flashovers at the sensor edges cannot happen either, as the complete edge is still on ground potential due to the guard rings.

[1] Measured from the outermost edges of all pixels
[2] **D**irect **C**urrent
[3] **F**ront**E**nd Electronic - Version 3.0

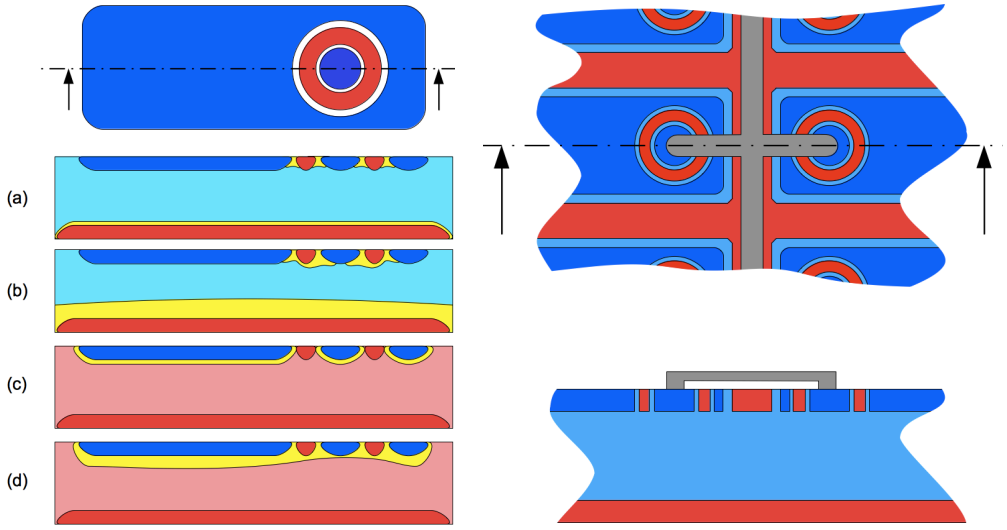


Figure 4.4.: (a) Explanation for the punch through effect at ATLAS pixel sensors. (b) Simplified geometry of the bias grid network (grey) the pixel implants and the bias dots. n-implants are shown in blue, p-implants in red. Not to scale

4.2.2. Bias Grid Network

To allow a testing of the sensor before it is mounted to the FEs a punch through bias grid is implemented. On the other side of the pixel than the bump bond the n^+-pixel implant is structured. A circular n^+ implant is located inside the pixel, but insulated from the n^+ implantation of the pixel itself by a p^+-implant. All of these so called bias dots are connected to each other via an aluminum trace on top of the oxide. An n^+ ring surrounding the whole sensor is connected to these metal traces. Instead of contacting each pixel individually to ground a single contact to the bias grid network can be used to interconnect all pixels simultaneously with ground. A simplified drawing of the geometry of the bias grid network at the bias dots is illustrated in figure 4.4b. In this figure the n-doped material is shown in blue and the p-doped in red. The aluminum trace of the bias grid network connecting all pixels is shown painted in grey.

The underlying effect is the so called punch through effect. In figure 4.4a cross-sections through one pixel as shown in the figure beside are made showing different situations to explain the effect. In (a) and (b) a sensor before type-inversion is shown. Indicated by the yellow area, the pixel implant and the bias dot are insulated from each other, due to a depletion zone caused by the ring shaped p-implant. When the potential difference between the bias dot and the sensor backside increases, the potential of the pixel-implant will be floating when it is not connected through the bump bond. If the potential difference between the floating pixel and the bias dot reaches the punch through voltage called U_{PT} the independent depletion zones will combine, as shown in (b). This is called the punch through effect. It mainly depends on the distance between the bias dot and the pixel implant. It can be roughly estimated to 10 V per 5 μm distance [12, 71].

Through the connection of the depletion zone the potential of the pixel implant will

be fixed to $U_{\text{Pixel}} = U_{\text{Bias Dot}} - U_{\text{PT}}$. The effect also works the other way around: If, for example, a sensor connected through the bump bonds has one missing bump bond connection, the potential of this pixel will be $-U_{\text{PT}}$, as all pixels are connected to each other through the bias grid network. This fact is important as it guarantees a homogenous field geometry in the sensor bulk.

In the cases (c) and (d) the situation for the bias grid is shown at a sensor after the type inversion. As the n-bulk behaves p-like it is pained in light red. Here the depletion voltage grows from the sensor's n-side. As an irradiated sensor is typically attached to a FE, the situation shown is the potential given by the pixel implant. When the punch through voltage U_{PT} is reached, the potential of the bias grid is defined.

The contacting of the bias grid in the production pixel detector was heavily discussed. In fact it is AC[1] coupled to ground. The differences to a DC coupled bias grid are only little. The potential difference of a non-connected pixel to a connected one becomes $2U_{\text{PT}}$ when keeping the bias grid network floating, instead of U_{PT} for a DC coupled bias grid. The decision for the floating bias grid (AC-coupled) was made as a ground connection (DC-coupled) of the bias grid leads to higher noise in the readout electronics.

4.2.3. Moderated p-spray

The pixel implants are isolated from each other by using a so called moderated p-spray [32]. The advantage of this technology is a reduced electric field maximum between the pixels. By using a nitride layer mask, the implant depth of the p-spray can be influenced. It is chosen in such a way that the doping concentration changes between the pixels. The doping profile will reach its maximum in center position between the pixels and will decrease towards the pixels.

4.2.4. Long and ganged pixels

The gap between two neighbouring FEs is 400 μm. Using the nominal pixel pitch in this region would lead to one inactive pixel column, as there would be no way to connect it to a FE. Therefore, the interchip regions have their own pixel geometry. Figure 4.5 illustrates this special geometry. In the z-direction of the pixel sensor, where the nominal pixel pitch is 400 μm, the outermost pixels are extended to 600 μm. These pixels are called long pixels. In the $r\phi$-direction, with a nominal pitch of 50 μm four additional pixels are processed next to the usual. These so called ganged pixels are connected to every second pixel out of the eight outermost pixels connected to the FE. Ambiguities in track reconstruction can be minimised when using this technique, as a traversing particle can hit more than one pixel. The pixels between two ganged pixels are called interganged pixels. In the region between four FEs so called long-and-ganged pixels, a mixture of both techniques is realised.

4.2.5. Radiation hardness

One of the basic design criteria of the pixel sensor was radiation hardness: For the b-layer, being the closest detector layer in ATLAS (55 mm radius with respect to the interaction

[1] **A**lternating **C**urrent

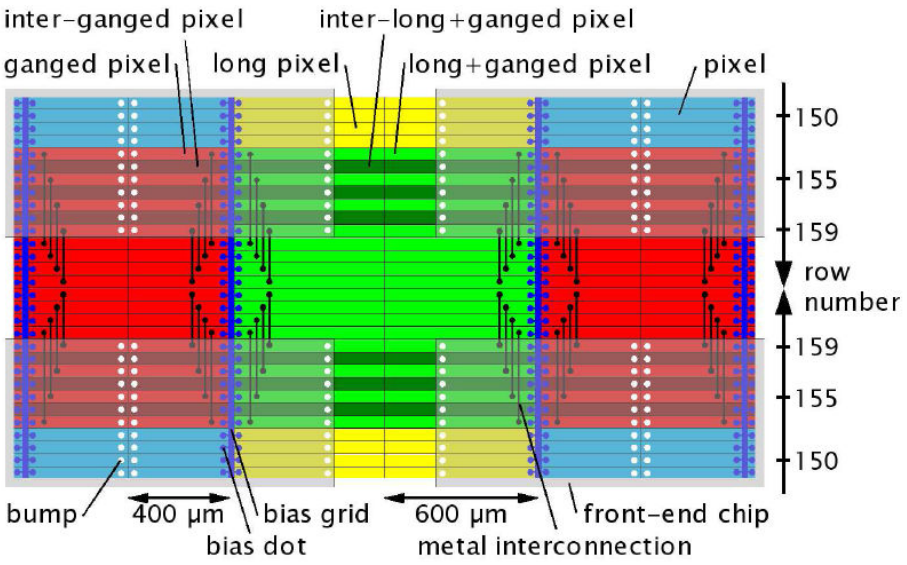


Figure 4.5.: Illustration of the pixel geometry in the interchip region [26]

point), a fluence of $1 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ was simulated. At the designed luminosity time profile of the LHC the depletion voltage is at 600 V after three years of nominal usage. As this is the voltage limit of the ATLAS pixel detector, the depletion zone shrinks from that time on. After five years the depletion zone would be too small to collect enough charge. Therefore, a replacement of the b-layer after five years of nominal usage was planned. As one can read in chapters 2.1.1 and 5.1 the original time schedule was changed and some concepts needed to be updated.

4.3. The ATLAS Pixel Front-End Electronics

The 16 FE chips attached to the ATLAS pixel sensor are the so called FE-I3 chips. One FE-I3 digitises the analogue pixel charges of the dedicated sensor region, does a zero suppression and passes the digital information to the module's MCC. This chapter gives an overview of the features of the ATLAS pixel FE-I3.

The FE-I3 is an ASIC[1] produced with the 6-metal IBM CMOS6SF [26] process. It was realised by using special design rules with the 250 nm DSM[2] technology. Due to the high fluence close to the interaction point, it will have to cope with an ionising dose of 50 MRad [11, 28, 29, 30, 31, 39]. The radiation hardness was verified by the RD49 collaboration [33]. The basic design rules are explained in [38].

Each FE-I3 covers an active sensor region of $7.2 \times 10.8\,\mathrm{mm^2}$ and contains about 2.5 million transistors. It consists of three main parts: The main pixel matrix contains 160×18 individual readout channels. In the middle part - the so called bottom-of-column region - the EOC[3] logics and buffers are located. The third part of the chip contains data

[1] **A**pplication **S**pecific **I**nterated **C**ircuit
[2] **D**eep **S**ub **M**icron
[3] **E**nd **O**f **C**olumn

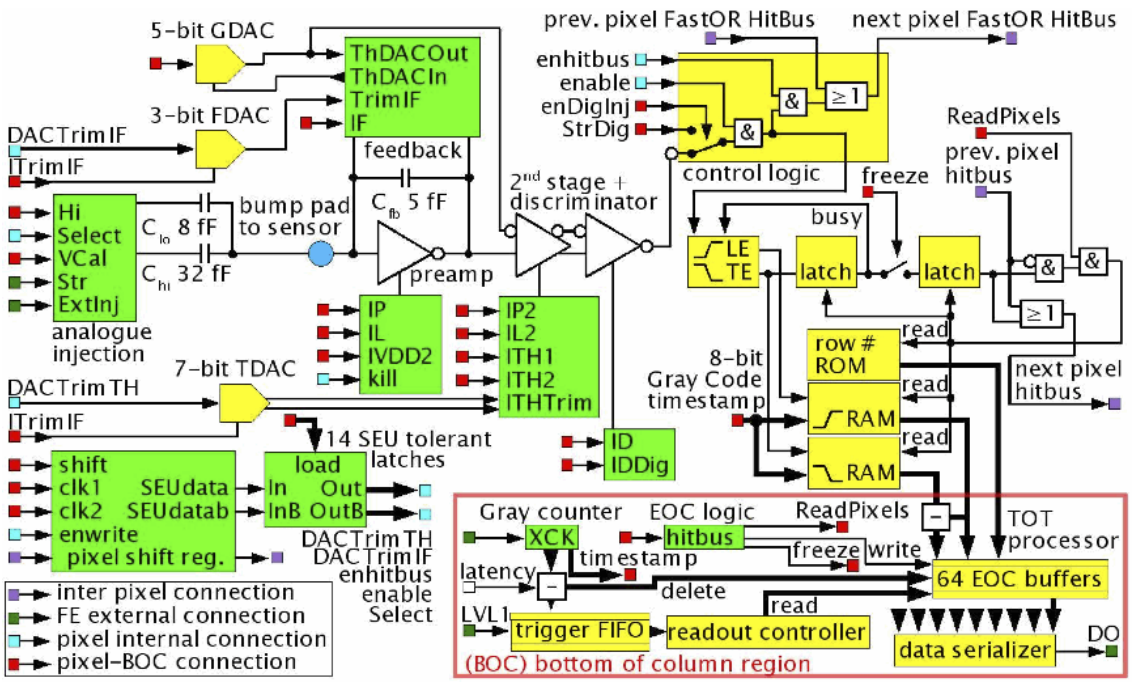


Figure 4.6.: Schematic illustration of a FEs pixel cell, its control blocks and digital readout [26]

serializers and wirebond pads. The FE is overlapping the sensor in order to connect its wirebond pads to the flex on top of the sensor with wirebonds as described in chapter 4.2.

Just like the sensor's pixel pitch, the pitch of the FE-I3's readout channels is $50 \times 400\,\mu m^2$. Each pixel readout channel is made of a high gain, fast amplifier using a feedback capacitor of 5 fC. To compensate the leakage current of the sensor, a DC feedback scheme is implemented in the chip. It can compensate a leakage current of 100 nA per pixel [20, 26]. DC coupled, a second stage fast differential discriminator follows the preamplifier. Several DAC[1]s are used to control the biasing at important nodes in the preamplifier and the discriminator. The threshold can be set from 0 to 1 fC. The design threshold is set to 0.5 fC which corresponds to a $3\,ke^-$ signal.

In figure 4.6 a schematic of a pixel cell and its digital readout is presented. Left from the center the preamplifier and the bump bond connection is shown. The threshold of the discriminator located behind the preamplifier can be set either by using the global GDAC[2] or the local TDAC[3]. The 5-bit GDAC changes the threshold of all pixels of the FE. Its value is distributed to local 5-bit DACs in each pixel. The 6-bit TDAC can be chosen individually for each pixel. The typical step width per DAC count allows a threshold setting with a precision of $75\,e^-$.

In the ATLAS pixel FE-I3 the amount of charge is determined by using a charge sensitive preamplifier, which uses the TOT[4] technique.

[1] **D**igital to **A**nalog **C**onverter
[2] **G**lobal Threshold **D**igital to**A**nalog **C**onverter
[3] **T**hreshold **D**igital to **A**nalog **C**onverter
[4] **T**ime **O**ver **T**hreshold

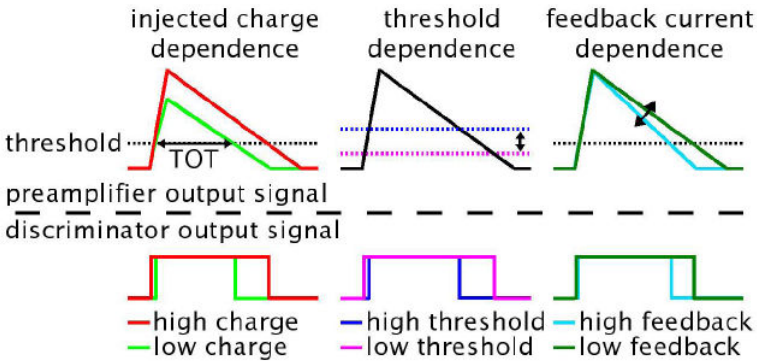


Figure 4.7.: Illustration of the TOT: Charge signal before the discriminator (top) and after (bottom) [26]

Figure 4.7 shows charge signals before and after the discriminator stage to explain the method. When a charge signal reaches the preamplifier and crosses the threshold, it will be decoupled by a constant current supply included in the pixel cell. For the time the signal is above the threshold, the discriminator signal rises to digital high - the length of this signal is the TOT in units of the FEs clock frequency[1]. As one can see from the drawing the TOT is a function of the injected charge.

Each signal crossing the threshold with its leading edge is time stamped by using an 8-bit so called Grey Code [45] in the FEs SRAM[2]. This Grey Code is increased with every bunch crossing. The time stamp of the falling edge is stored in a second SRAM. Because of the TOT technique the charge can only be known after the signal's falling edge fell below the threshold. As this point in time is variable one defines a fixed time after the first timestamp to enable a readout. When a LVL1[3] trigger is sent to the FE after this fixed time - called latency - with respect to the first time stamp, the TOT is calculated from the time stamps in the SRAM and written to the EOC buffers.

Because of timewalk, small signals reach the threshold with their leading edge later (compare figure. 4.7). The TOT processor can additionally apply a settable minimum TOT threshold. Charge signals that undergo this threshold will be copied in the SRAM to another time stamp so that the signal will be handled in the context of the correct bunch crossing.

Instead of copying the signal to another time stamp, a sequence of up to 16 external LVL1 triggers can be sent to the FE. Therefore, a 16-word FIFO[4] is used on the FE. The hit information belonging to one of the external LVL1 triggers is taken out of the EOC buffers by the readout controller and send through the data serializer to the MCC.

As the Grey Code has a width of 8-bit, a maximum latency of $2^8 \cdot 25\,\text{ns} = 6.4\,\mu\text{s}$ is possible. The target latency of the ATLAS LVL1 trigger is $2,0\,\mu\text{s}$. Of course, the maximum TOT value is limited by the latency.

[1] The FEs clock frequency is 40 MHz as it is obtained from the LHC, which sends bunch crossing triggers continuously
[2] **S**tatic **R**andom **A**ccess **M**emory
[3] **L**e**V**e**L-1** trigger signal
[4] **F**irst **I**n - **F**irst **O**ut

Beside the setting of the threshold, the feedback current of each pixel cell can be adjusted as well. As one can see in figure 4.7 a change of the feedback current changes the TOT as a consequence. Therefore, an adjustment at the 4-bit FDAC[1] can be used to obtain a homogenous TOT signal from each pixel over the FE by injecting a well known charge.

To do such scans on the pixel DACs, each cell comprises a charge injection circuit (shown in the left of figure 4.6). Two injection capacitors with nominal value of $C_{\text{low}} = 8\,\text{fF}$ and $C_{\text{high}} = 40\,\text{fF}$ are used in order to inject charges into the preamplifier. The charges can be injected either by using the ExtInj[2] line or by generating a voltage step. In this case the capacitors are switched from VDDREF[3] to VCAL[4], which is a voltage provided by a DAC in the EOC region.

Two external voltages are necessary to operate the FE-I3: The digital supply voltage VDD[5], referenced to DGND[6] with a nominal value of 2.0 V and the analogue supply voltage VDDA[7] with a nominal value of 1.6 V, referenced to AGND[8]. For digital voltage an operation range of 1.4 to 2.5 V is common. AGND and DGND are connected on the module. To provide power to the preamplifiers only VDDREF can be used. This line is the reference input for the preamplifiers as well. It is connected to VDDA outside the FE. A typical current consumption is roughly 75 mA on VDDA and 35 mA on VDD [26].

Like the ionisation happening in the sensor, which is used to generate a charge signal for a pixel cell, ionisation can happen in the FE electronics as well. As charges are created by ionisation, the state of a sensitive node in the FE can be changed by this. These so called SEU[9]s do not cause permanent damage to the electronics but can cause, for example, an interim bit flip. All registers are set up in a redundant way in order to create a SEU tolerant FE electronic. Especially the latches in the electronics, which interimly store bits of the registers are made of so called DICE[10] cells [79]. Each of the 14 pixel registers in each pixel consists of these cells. To write a bit into these cells a simultaneous write operation to both nodes in these cells is necessary. The nodes are separated from each other with a distance of roughly 5 μm in order not to be affected by the same SEU. Additionally, redundant reset signals are implemented in the Global and the Command Register. These registers use a tripple redundancy scheme and are therefore even more SEU tolerant.

4.4. Most important scans and tuning steps

Due to process uncertainties in the ASIC production of the FE, each pixel cell in each FE needs to be set to the correct parameters. The most important DACs to be set are the TDAC, the FDAC and their global partners the GDAC and IFDAC[11]. In order to

[1] **F**eedback-current **D**igital to **A**nalog **C**onverter
[2] **EXT**ernal **INJ**ection line
[3] **REF**erence for the digital supply voltage **VDD**
[4] **CAL**ibrated **V**oltage input line
[5] **D**igital supply **V**oltage
[6] **D**igital **G**rou**ND**
[7] **A**nalog supply **V**oltage
[8] **A**nalog **G**rou**ND**
[9] **S**ingle **E**vent **U**psets
[10] **D**ual **I**nterlock storage **CE**ll
[11] Global **F**eedback **DAC**

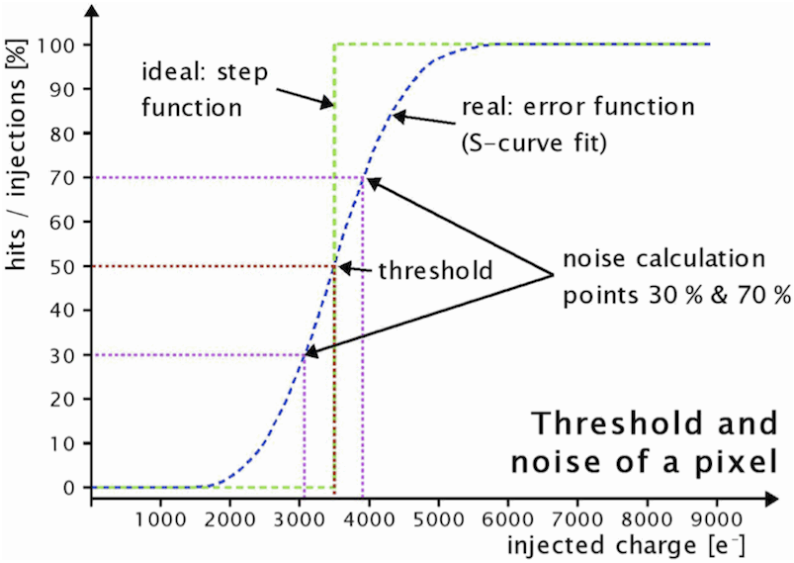


Figure 4.8.: Schematic result of a threshold scan for a single pixel, the ideal step function and the error function are illustrated [26]

guarantee a proper working of all pixels in the sensor under similar conditions scans and tunings need to be made before. These are:

Digital Injection Scan This scan is the most simple scan for testing the condition of the digital part of the FE. Typically 100 digital events are induced into the digital part of each pixel cell. If the cell recognises exactly this number of events it is working properly.

Threshold Scan The threshold scan is the way to determine the position of the pixel's threshold in units of charge. Using the analogue injection circuit pulses with varying charge values are induced into the analogue part of each pixel cell. 100 events per charge step are common. The typical response of each pixel cell is illustrated in figure 4.8. The threshold can be identified as it marks the position where 50 % of the analogue hits are recognised. The function can be parameterised by an error function:

$$f_{\text{error}} = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-t^2} \mathrm{d}t. \tag{4.1}$$

The noise caused by the electrical readout is inversely proportional to the steepness of the transition [80] and can therefore be additionally calculated by this scan.

TOT Calibration Scan In the same way as for the threshold scan analogue reference charges are induced into each pixel cell. This time the TOT response in dependency of the induced charge is analysed. The TOT as a function of the charge can be parameterised by this empirical function for each pixel

$$\mathrm{TOT}(q) = a + \frac{b}{c+q}, \tag{4.2}$$

in which a, b and c are free parameters and q the induced charge. As the raw data stream of the pixel sensor offers only TOT, these three parameters need to be known in order to allow a correct charge conversion into electrons.

Threshold Tuning This tuning step is basically a successive call of threshold scans by using different settings of the TDAC of each pixel. A binary sort algorithm is used to limit the necessary steps for this tuning. A tuning can be applied by using a variable threshold. Therefore, the user can decide which number of electrons the threshold should be set to afterwards. Common setting in the ATLAS pixel detector is $4000\,\mathrm{e}^-$ as threshold.

FDAC Tuning In a similar way as for the threshold tuning the FDAC tuning runs TOT scans with varying FDAC settings. An algorithm sets the FDAC for each cell to a value so that the TOT response for the same injected charge is similar. A typical setting for the FDAC tuning in the ATLAS pixel detector is $20000\,\mathrm{e}^-$ at 30 TOT.

Source Scan In this mode the detector is tested under mostly realistic conditions. A radioactive source can be placed on top of the sensor to induce ionising radiation into the sensor. Typically the scan runs by using a so called self trigger, which is an OR-gate of all enabled pixels. The TOT response of each pixel is histogrammed automatically, so that the charge spectra of each pixel can be analysed.

As the correct setting of the FDAC depends highly on the TDAC, the FDAC tuning needs to be done after the correct threshold tuning. But unfortunately the changed slope of the charge decoupling influences the threshold again. So, a further threshold tuning needs to be applied after the FDAC tuning.

4.5. The ATLAS Pixel MCC

The Master or Module Control Chip of the ATLAS pixel module is the MCC I2.1 [10, 68, 66, 21]. It is produced in the 250 nm DSM technology. In its dimensions of $6.84 \times 5.14\,\mathrm{mm}^2$ it contains about 880000 transistors. In figure 4.9 a schematic overview of the ASIC is shown.

In general, the function of the MCC is to distribute the command signals to the attached FEs and to collect the data signals from these in order to merge them into one common module data stream. This electrical data stream will be converted into optical signals on the Opto Boards and interpreted by the ROD[1]s in the off detector site.

The MCC has two different kinds of interfaces. One of them, called module port, is used to allow a communication with the MCC, respectively the whole module from outside. The other, called FE port, connects the chip to one of the attached FEs. The module port is made of one input data line, called DTI[2] and two output data lines DTO[3] and DTO2[4]. A 40 MHz clock input (CK[5] in figure 4.9) is necessary for the MCC in order to

[1] **R**ead **O**ut **D**river
[2] **D**a**T**a **I**nput (module-port)
[3] **D**a**T**a **O**utput (module-port)
[4] **D**a**T**a **O**utput 2 (module-port)
[5] LHC bunch crossing **C**loc**K**

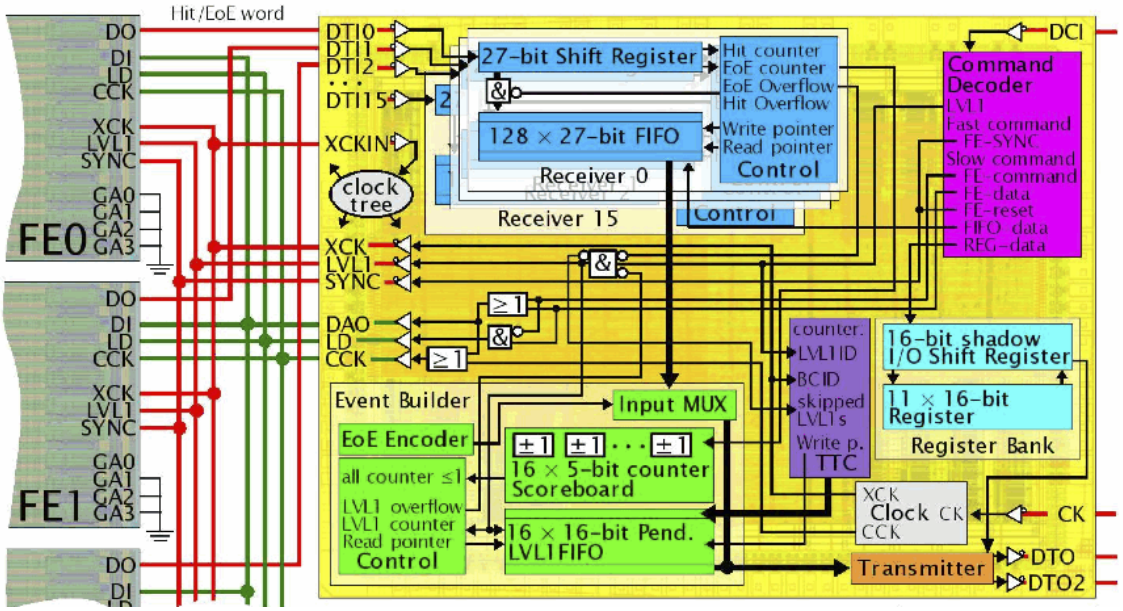


Figure 4.9.: Schematic of the MCC and its routings to the FEs (left). The red lines indicate LVDS signal lines, whereas the green lines symbolise CMOS lines. Original taken from [26] and modified

synchronise the serial signals on all lines. The clock signal, being synchronous with the LHC bunch crossings, is transmitted to the module by a LVDS[1] differential signal.

In the so called single link mode only one of the two output data lines is used. For this case the bandwidths are 40 Mbit/s or 80 Mbit/s. When using the dual link mode, the output bandwidth results in 80 Mbit/s or 160 Mbit/s. For modules close to the interaction point, such as the b-layer modules, the high bandwidth is needed to cope with the higher event rate. The MCC generates the XCK[2] 40 MHz and the CCK[3] 5 MHz from the CK. XCK is used as the main internal clock and also used to clock the FEs.

The FE port uses unidirectional links to interconnect the 16 FEs with the MCC. For each FE an individual LVDS data link (DTI0[4] to DTI15[5]) in phase with XCK is necessary. The resulting bandwidth of each data line is up to 40 Mbit/s.

The connection scheme uses a data push protocol: The FE sends the hit coordinates, ordered by event number and followed by an EOE[6] as soon as they are available in the EOC buffers. The EOE word is important for the event synchronisation. The data can be received by the FE's own receiver block in the MCC [67]. An implemented control state machine checks the number of hits and EOE words in the receiver FIFOs. The FIFO is a 128×27 bit wide dual port SRAM full custom block. The control state machine sets up a Hit or EOE overflow warning in the next EOE word when more than 16 EOE words or $128 - 16 = 112$ hit words are stored, but drops the current word without transmitting

[1] **L**ow **V**oltage **D**ifferential **S**ignal
[2] MCCs main internal **C**loc**K**
[3] **C**ommand **C**loc**K**
[4] **Da**ta **I**nput - channel **0** (FE-port)
[5] **Da**ta **I**nput - channel **15** (FE-port)
[6] **E**nd **O**f **E**vent

it into the FIFO. This is done in order to keep the event synchronisation. As the data coming from the FE is received by the FIFO in a random order, they will be sorted afterwards. To avoid event desynchronisation, which means a mixing of hits belonging to different events, the control state machine can block the LVL1 trigger distribution to the FEs, in case of an EOE overflow.

The event builder is the part of the MCC responsible for merging the hit data. There is a so called scoreboard in the event builder which is made of 16 individual 5-bit up-down counters. Each time an EOE word is detected in one of the receiver blocks, the corresponding receiver channel is incremented in the scoreboard. When all counters do not show zero it indicates a complete event and the event builder control state machine will initiate the shift of the last word from the pending LVL1FIFO[1] to the transmitter. The LVL1FIFO is a 16 bit wide and 16 words deep fully synchronous, single clocked FIFO containing the last four significant bits of the 8-bit LVL1ID[2] counter, the 8-bit value of the BCID[3] counter and the number of skipped LVL1 triggers in the last 4 bits. These counters are implemented in the TTC[4] block. While the BCID counter is incremented in the MCC at each XCK cycle, the LVL1ID counter is incremented each time a LVL1 trigger is received. When the transmission from the LVL1FIFO is done, the receiver FIFOs are read out sequentially through the input multiplexer and send to the transmitter which passes the data out of the module through DTO and/or DTO2.

In contrast to the individual output data lines DTI0 to DTI15 in the FE port there is only one common input connection to all FE realised by DAO[5] LD[6] and a synchronous clock signal CCK. In order to address the FEs individually, the unique 4-bit GA[7] is transferred with each command. Each FE has four CMOS[8] input lines with pull-up resistors, which provide a high state when floating. Setting these pads to ground by wirebonding them to the module's flex, they obtain their individual GA corresponding to their position on the module.

The MCC is relatively SEU tolerant. The command decoder, the event builder control state machine and the register bank are triplicated with a majority logic on all outputs. Warning bits are set in all three blocks if a SEU occurs. This is very important for the Register Bank as it does not contain an automatic correction. Due to their large size the scoreboard and the transmitter are not fault tolerant.

4.6. Staves and cooling

The pixel modules are glued to so called staves. The staves are the mechanical carriers of the modules in the detector made of carbon fibres. Figure 4.10 shows a so called bi-stave shortly before the installation into the detector. Each stave carries 13 modules. As one can see in the picture, the modules are overlapping on the staves. This overlapping -

[1] **F**irst-**I**n **F**irst-**O**ut buffer storing **L**e**V**e**L-1** trigger signals
[2] **L**e**V**e**L-1** trigger signal **ID**entificator
[3] **B**unch **C**rossing **ID**entificator
[4] **T**iming, **T**rigger and **C**ontrol
[5] **D**at**A** **O**utput (FE-port)
[6] **L**oa**D**-signal line (FE-port)
[7] **G**eographical **A**dress
[8] **C**omplementary **M**etal **O**xide **S**emiconductor

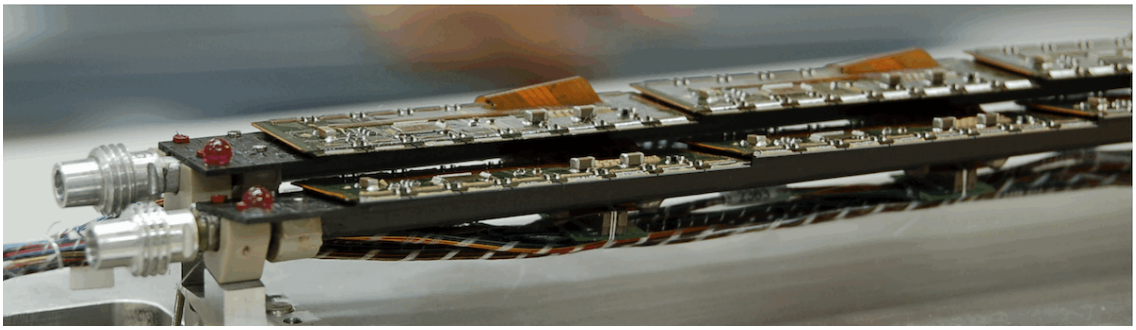


Figure 4.10.: Photograph of a bi-stave during the assembly of the pixel detector. On the left side one can identify the two flanges to connect the staves' cooling pipes to the C_3F_8 cooling system. The modules are shingled over each other on one stave and roofed by arranging the staves to each other

called shingling - is needed as the ATLAS pixel sensor has an inactive region at all edges of more than $1000\,\mu$m. By shingling the modules, the inactive part of the sensor at its short edge will be covered by the other module. The inactive edge at the long sensor side is recovered by overlapping the staves over each other. This is called roofing. In the figures 4.10 and 4.1 the roofing can be seen as well.

A cooling pipe is attached to the stave. The power dissipation of the modules is compensated by using an evaporative cooling system. This means that a cooling liquid - which is C_3F_8 in this case - is pumped into the staves and changes there into its gaseous phase. By changing its phase it is cooled down to temperatures of roughly $-10\,^{\circ}$C. Two staves are attached to each other as they build a cooling loop. So that the inlet and the outlet of a cooling loop is always on the same detector side.

5. Upgrade of the ATLAS Pixel Detector

In this chapter the motivation for this thesis is presented. The necessity of an upgrade of the ATLAS pixel detector and the consequences for the technologies to be used are outlined. Furthermore, charge amplification, a welcome effect to further improve the behaviour of silicon detectors at the end of the lifetime of an upgraded detector, is discussed.

5.1. Upgrade schedule

With upcoming upgrades of the LHC (compare 2.1.2), the ATLAS detector will be upgraded as well. During the time (2013) while the accelerator is prepared to run at the design center of mass energy of 14 TeV, which is called phase 0, a new innermost pixel layer will be installed. As a replacement of the current b-layer with a new one would take more time than scheduled, another option is preferred currently: Instead of removing the whole pixel package to replace the innermost layer, one only removes the beryllium beam pipe, without touching the detector itself. A completely new designed detector layer including a new beampipe with a smaller radius will be installed into the opening. This new layer is called the IBL[1]. A replacement of the Service Quarter Panels, attached next to the detector, at this point in time due to present troubles during the operation is currently under evaluation.

In figure 5.1 one sees a technical drawing of a quarter of the layer in the $r\phi$-direction. The distance of the current innermost detector layer from the interaction point is 55 mm. The inner radius of the current beampipe is 29 mm. In order to clear a space for the new layer the inner radius of the beampipe will be reduced to 25 mm. This new geometry is possible as some tolerances of the positioning accuracy, which had been overestimated before, could be calculated very well with the experiences of the current design.

The new space is needed in order to place 14 staves each with 16 modules of the new FE-I4[2] design between the new beampipe and the existing b-layer. As the place is too limited to shingle the modules on top of each other there is a need for special sensor designs with a little inefficient edge around the active pixelated area. The FE-I4 sensor design uses a pixel pitch of $50 \times 250\,\mu\text{m}^2$, with an active pixel region of $16.8 \times 40.4\,\mu\text{m}^2$. Two FE-I4 ASICs are used to read out one sensor.

A radiation tolerant design is needed for the new IBL detector layer. Simulations have shown that the expected average dose for an IBL detector is predicted to be 1.6 MGy (160 Mrad) at an integrated luminosity of $550\,\text{fb}^{-1}$. The 1 MeV neutron fluence is estimated to $3.3 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$. Due to some uncertainties a safety factor was applied: So at

[1] Insertable b-layer
[2] **F**ront**E**nd Electronic - Version 4.0

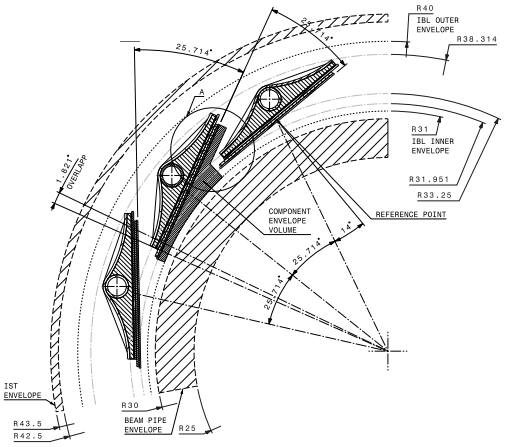


Figure 5.1.: Technical drawing containing the dimensions of the IBL in $r\phi$-direction. Only one quarter of the detector is shown [5]

the end of the lifetime of the IBL the fluence will not be higher than $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$. This corresponds to an ionising dose of 250 MRad. [5]

The tracking resolution benefits from the position of the detector closer to the interaction point. Especially the possibility to differentiate independent tracks with a higher number of pile-up events increases due to the smaller pitch of the pixels.

During the downtime of the LHC upgrade at phase 1 in 2017, a complete rebuild of the pixel detector is currently evaluated.

In phase 2 an upgrade of the whole ATLAS inner detector is being planned. A technical design report for this period has not been prepared yet. Different options of the design of the new inner detector are under discussion. The community seemed to agree on a few points: The SCT will increase its volume and will move to the outer radii and replace the TRT detector. The pixel detector will stay as close as possible to the beam pipe. Estimations on the fluences seem to agree on $2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$ as a maximum including safety factors. The intermediate range of SCT and pixel is the most debatable area. For example, a 5^{th} pixel layer is under discussion.

The equipment to be used either in the new IBL pixel layer or the rebuilt inner detector needs to be qualified for its new environment. The object of this thesis is the construction of a testing facility and the qualification of the ATLAS pixel sensor in an IBL and SLHC detector. Special investigations are necessary in order to cope with new requirements. These are basically the quality control of the sensors after an irradiation similar to that after the lifetime of the corresponding phases. Secondly, the quality control of a changed sensor geometry[1], which was necessary to avoid the shingling of the detectors in the IBL, needs to be done. Further on, the sensor needed to be adapted to the new generation of read-out electronics FE-I4. An influence of the detector performance needs to be analysed.

One subject of this thesis is the construction of a test facility and a connected analysis tool in order to allow qualification studies of ATLAS pixel sensors. Detailed results for

[1] Different sensor geometries with slimmed sensor edges have been produced in order to reduce the inactive area when placing the sensors side-by-side

the qualification of the changed sensor design due to slim edges and the FE-I4 geometry are in preparation and can be found soon in [3, 84].

5.2. Charge Amplification

Important for the qualification of a sensor in the ATLAS detector is the value of the collected charge of the sensor. During the construction of the current detector the behaviour of the sensor when running in the irradiated environment was estimated by using models:

Using equation (3.17), one can approximate [1]

$$q_{\text{eff}}(t) \approx q(t=0) \exp\left(\frac{-t}{\tau_{\text{eff}_{\text{e,h}}}}\right). \tag{5.1}$$

This means that the effectively collected charge q_{eff} is reduced due to radiation damage. In practice, the charge trapping becomes the dominant effect on the signal. Instead of using the inverse trapping times, one can calculate the collection distance at saturation velocity λ_{av} [1]. For irradiations from 1 to $20 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ these values are presented in table 5.1.

The origin of these estimations were sensor irradiations up to lower doses, which had been common for the original ATLAS detector. When comparing these estimations with charge measurements (compare figure 5.2) for silicon strip n-in-p sensors irradiated to similar fluences one sees that the collected charge is much higher than the expected estimations. In figure 5.2a the maximum collected charge of an unirradiated sensor of the same kind is marked as well. One sees that the collected charge increases linearly with the applied bias voltage without saturation. In fact, the collected charge of an irradiated sensor is even higher than the collected charge of an unirradiated sensor of the same kind. This effect is called the charge amplification effect.

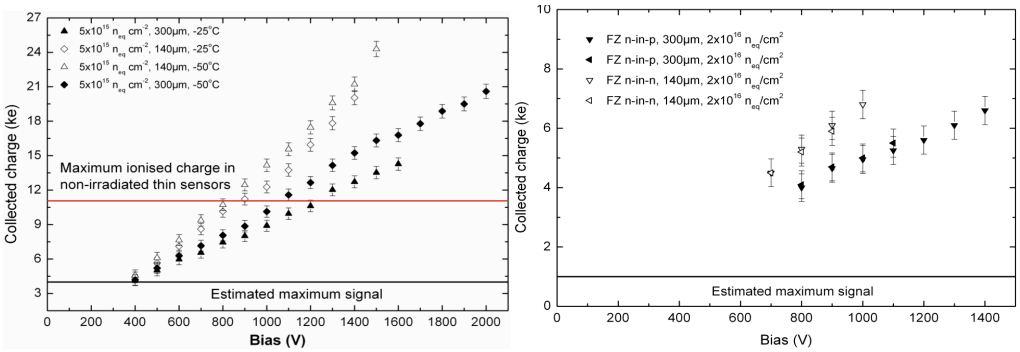
The effect depends on the sensor's thickness as well as on its temperature. At lower temperatures the signal is increased.

In order to apply sensor bias voltages of more than 1 kV a sufficient cooling is compulsory to reduce the leakage current of the sensor.

The origin of this effect is still not understood at the time of this study. Investigations are going on to empirically understand the charge amplification mechanism. There are some indications that the very high electrical field gradients close to the pn-junction lead to something like an avalanche effect in the sensor.

Φ_{eq}	$1 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$	$5 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$	$1 \cdot 10^{16}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$	$2 \cdot 10^{16}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$
λ_{av}	$240\,\mu\text{m}$	$50\,\mu\text{m}$	$25\,\mu\text{m}$	$12\,\mu\text{m}$
q_{eff}	$19\,\text{ke}^-$	$< 4\,\text{ke}^-$	$< 1.3\,\text{ke}^-$	$< 1\,\text{ke}^-$

Table 5.1.: The collection distance at saturation velocity λ_{av} and the estimated effective charge, at different fluences [1]



(a) $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ (b) $2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$

Figure 5.2.: Collected charge of silicon strip n-in-p sensors irradiated to different fluences of 5 and $20 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ vs. the bias voltage and the sensor temperature. The estimated maximum and the common signal of the unirradated reference is illustrated as well [62]

The discovery of the charge amplification effect in silicon strip sensors highly influences the ATLAS upgrade projects. Especially the possibility to use silicon material as sensors irradiated up to SLHC fluences without the need for a more sensitive read-out electronic is very interesting. Although studies to reduce the threshold of the read-out electronic which is currently installed in the detector were made [76], a threshold lower than $1800\,\mathrm{e^-}$ could not be achieved. The charge amplification effect offers the possibility for the planar pixel sensor technology to become the sensor for the IBL and a pixel detector in phase 2.

As part of this study the possibility to operate a planar pixel sensor with the same read-out electronics as common for the current ATLAS pixel detector is investigated. Furthermore, the charge amplification effect which has only been discovered for silicon strip n-in-p sensors yet, is a highly recommended tool to improve the current sensor technology.

6. Overview about Testbeam

Testbeams can be used for multiple purposes. Their origin were fixed target experiments, in which the resulting processes or high energetic particles interacting with a target were studied. Furthermore testbeams can be used to test properties of all kinds of detectors like trackers and calorimeters. In the following chapter the requirements in a testbeam for the investigated studies are being presented.

6.1. General requirements

For studies like those investigated here, high energetic particles are passing through multiple sensors. The energy loss of the crossing particle can be neglected with respect to the total energy of the particle. In the ideal case the consequence is that the particle's trajectory will not be affected when crossing the sensors. That means that effects like beam scattering or bendings do not play an important role. The number of tracks within the readout time of the detector system may only be slightly greater than one. By this definition a reconstruction can be done to obtain particle trajectories from hit pixels in the sensors. Particles in the beam are typically spread over an area in the order of magnitude of the sensor's size. The resulting envelope is gaussian in its origin, but can be collimated with beam absorbers so that the typical shape is limited to its sides.

6.2. EUDET Telescope

EUDET[1] is a project with the aim to support the detector research and development in Europe for a detector at the next large particle accelerator project called ILC[2] [36]. One focus is the permanent operation of a small detector system that could be used in a bigger scale as a tracking system in an ILC detector. On the other hand, a general purpose tracking detector in a particle testbeam is necessary. EUDET combines both requirements in the EUDET testbeam telescope.

6.2.1. Sensor

The heart of the beam telescope is the sensor. The EUDET telescope consists of six MIMOSA[3]-26 sensors. These sensors are so called MAPS[4]. The processing of sensors of that kind is cheap compared to the ATLAS pixel sensors, as it can be realised in a

[1] **DET**ector-project supported by the **E**uropean **U**nion
[2] **I**nternational **L**inear **C**ollider
[3] **M**inimum **I**onising Particle **M**etal **O**xide **S**ensor **A**ctive Pixel Sensor
[4] **M**onolythic **A**ctive **P**ixel **S**ensors

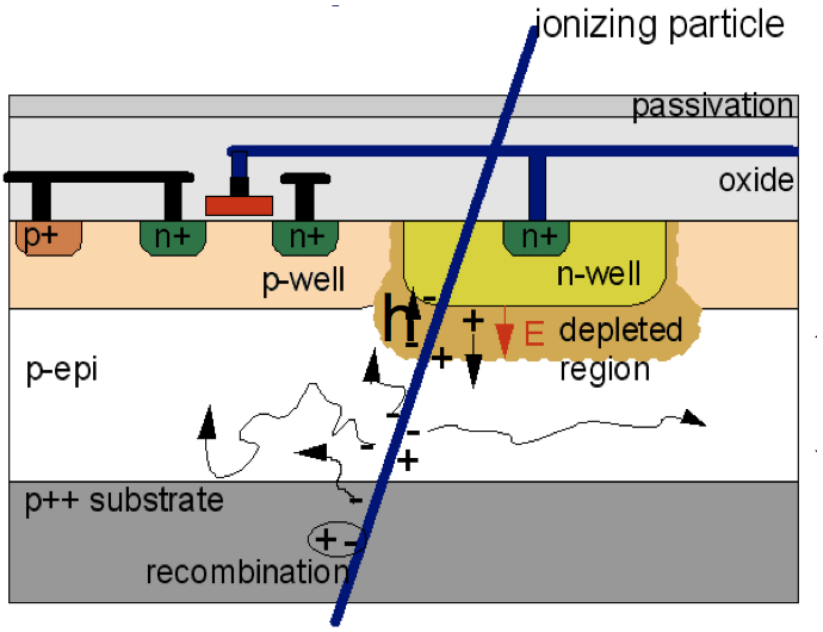


Figure 6.1.: Schematic illustration of a cross section through the MIMOSA-26 sensor [88]

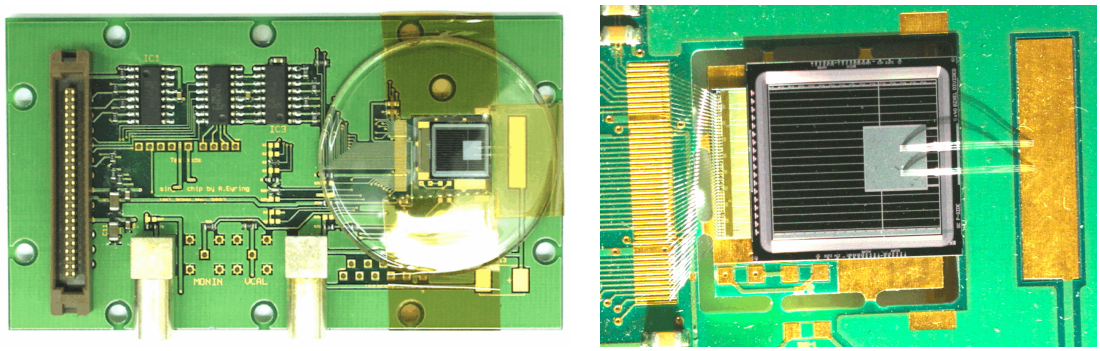
standard single sided CMOS process. Figure 6.1 shows a schematic crossection through the MIMOSA-26 sensor. The epitactical layer (p-epi) has a width of less than 20 μm and is the only active region in the sensor. This width corresponds to the natural depletion depth of the sensor without applying a bias voltage. The charges reach the electrodes through diffusion and through the electric field caused by the mismatch of donators and acceptors. One aim of the detector research in this field is the minimisation of radiationlength. To reach this goal the single-sided sensors can be slimmed from the backside down to 50 μm.

The MIMOSA-26 covers an active region of $21.2 \times 10.6\,\mathrm{mm}^2$. The squarish pixels are arranged in 1152 columns and 576 rows in a pitch of 18.4 μm. Therefore, one sensor is covered by 663552 pixels. For their purpose in an ILC detector the MIMOSA-26 sensors were tested for radiation hardness up to 1 MRad and $2 \cdot 10^{13} \mathrm{n_{eq}\,cm^{-2}}$. For MIPs the sensors show a signal-to-noise ratio of 20:1 to 40:1. Each sensor is read out successively in a loop over all pixels by a data reduction board [19] which uses a so called rolling shutter method. By using an external trigger the data can be received from this board. The data is zero suppressed and digital. The loop over all pixels takes 110 μs [54] and corresponds to the sensitive integration time of the sensor.

6.3. Devices under Test

The sensors to be investigated in a testbeam are called DUT[1]s. Although the testbeam studies in this thesis are mainly made for understanding properties of the detector's sensor, an assembly made of sensor and read-out electronics is necessary. A detailed list of investigated DUTs can be found in chapter 11.

[1] **D**evice **U**nder **T**est



(a) Bonn adapter board (b) Sensor assembly

Figure 6.2.: Photograph of a DUT. (a) shows a so called Bonn adapter board with an FE-I3 single chip assembly attached. (b) shows the sensor assembly attached to the PCB with wirebonds. Original taken from [81] and modified

In any case the sensor assembly is attached to a PCB. The assembly is glued to the board with the sensor on top. The connections to the FE electronics and the sensor's bias pad are realised by wirebonds. The circuits on the board contain mainly LVDS drivers which are used to amplify the signals from and to the FE. Furthermore, decoupling capacitors are attached. Figure 6.2 shows a photograph of the common Bonn single chip adapter board and a closer view of the sensor assembly with the interconnecting wirebonds.

In order to do scans with the DUT, three connectors need to interface the PCB. A 50-pin KEL connector [48] is used to connect the FE readout to the DAQ[1] system. The bias voltage connection and the supply power for the LVDS drivers are made by lemo connectors [52]. When measuring the temperature of the DUT, a PT-1000 temperature sensor is attached to the DUT. This device can be interfaced in addition to record the DUT's temperature.

In comparison to a complete ATLAS pixel module all DUTs are so called SCA[2]s. The special sensor design shares all sensor properties of an ATLAS pixel module except its size. Only one FE chip can be attached to a single chip sensor. To analyse effects in the sensor caused by a varied sensor design, SCAs with changed designs were built as well.

6.4. Coordinate System

It is useful to define a global coordinate system to describe the geometry of the telescope. In this case a right handed coordinate system is used. The origin of this system is the beam axis that is parallel to the z-axis of the system. The x- and y-axis are pointing perpendicular to the beam horizontally and vertically. In the setup at CERN SPS H6B x points towards the Jura mountain whereas y points upwards.

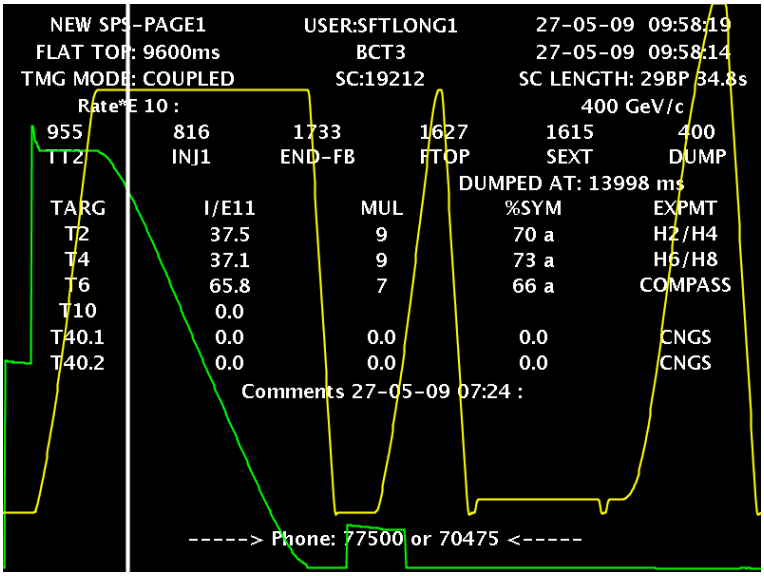


Figure 6.3.: Screenshot of SPS Page-1 illustrating an SPS Super Cycle [78]

6.5. Testbeam at CERN SPS

Plenty properties such as beam profiles, energy, momentum, focus can be set individually. The used particles at CERN SPS are $\pi^{\pm}$. The protons from SPS are accelerated to 400 GeV, ejected out[1] of the circulating beam and converted into $\pi^{\pm}$ by hitting them into a target. A spectrometer magnet filters out the particles of one specific energy. Further on, the beamline contains plenty of magnets for bending and focussing the beam. Beam absorbers and collimaters can be used to filter out particles of other kinds that had been created at the same time. Wire chambers are installed along the beamline to determine the position of the beam.

The beam in the SPS beamlines is not continuously present. The reason for this is that the SPS is connected to multiple experiments and the ejected beam from the SPS is distributed among them. Because of this the SPS fulfills the so called Super Cycle.

6.5.1. SPS Super Cycle

Figure 6.3 illustrates the Super Cycle. One can see two colored graphs in green and yellow. The big vertically white bar moves from left to right in the animated version and indicates the actual position of time in the Super Cycle. The green graph shows the development of the beam intensity, the yellow one the current of the SPS' magnets. In the first part of the Super Cycle the SPS is filled from the PS, which is indicated by the two big steps in the green line. The next step is the increasing of the beam energy inside the SPS. As the SPS is a synchrotron, the beam energy increases proportionally to the magnet current. So one can see the rising beam energy by the rising of the yellow line.

[1] **D**ata **AQ**uisition
[2] **S**ingle **C**hip **A**ssembly
[1] The exact ejection mechanism is described in the chapter 6.5.2.

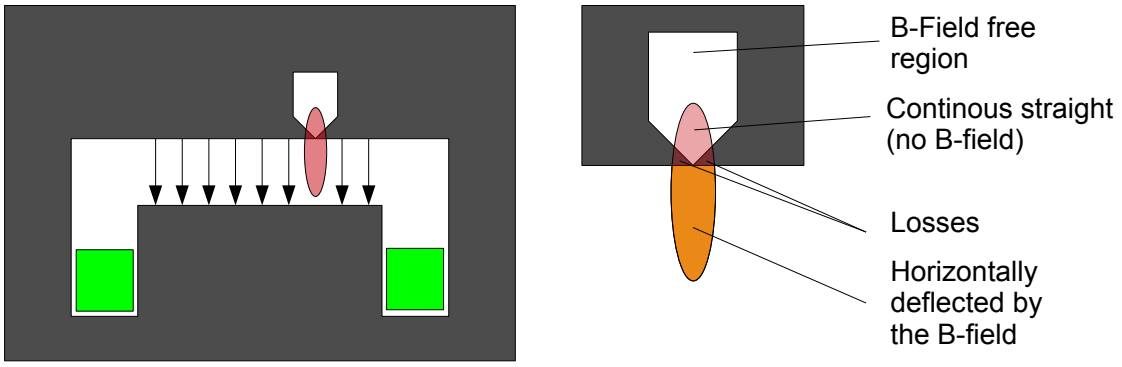


Figure 6.4.: Schematic illustration of a cross section through a septum magnet. The arrows indicate the direction of the B Field. The picture in the right allows a closer view. The beam envelope is illustrated in orange [27]

When the yellow line reaches the plateau at its maximum, the beam is extracted into the north hall experiments such as the H6 beamline. The text in the background (flat top) shows that the duration of the extraction takes 9600 ms.

Beside the beam distribution to the testbeam experiments in the so called north area beam lines further experiments, such as CNGS[1] or COMPASS[2] are delivered with beam in later parts of the Super Cycle. Typically, the very last extraction phase is used to fill the LHC. The overall time of the SPS Super Cycle can of cause vary, depending on the running program, but is typically 42 s. The moment the beam is extracted into the beamline is called the spill. For calculations of the trigger rate one always has to take into account that the beam at SPS is not continuous. The maximum trigger rate is reached during the spill, whereas the average trigger rate is typically a factor of four smaller, due to the ratio of the spill time to the overall Super Cycle time.

6.5.2. Ejection and Beam Wobbling

The typical method to eject a beam from one accelerator into another accelerator or a beamline is to use a kicker magnet. This kind of magnet only fulfills a very low magnetical hysteresis and therefore can change its magnetic field very fast and very precisely. This is necessary in order to kick out only a few bunches from the circulating beam. The injection mechanism into a circular accelerator works in the same way: For the moment the new bunch arrives the kicker magnet brings its onto the beam orbit. Before another particle bunch already being in the orbit arrives the magnetic field needs to be vanished in order not to affect the particles trajectory.

Although this method is known well and works very efficient its not used to extract the SPS beam into the north area beam lines. The reason for this is that the extraction would happen suddenly with the described method, whereas typical experiments at the beam lines need a continous particle flow and constant rate.

[1] **C**ERN **N**eutrino Beam to **G**ran **S**asso
[2] **CO**mmon **M**uon and **P**roton **A**pparatus for **S**tructure and **S**pectroscopy

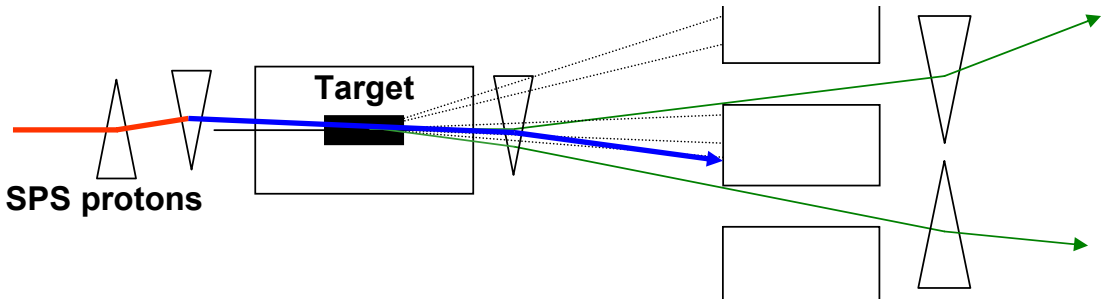


Figure 6.5.: Schematic drawing of the SPS target area. The primary beam, coming from left passes to dipole magnets to change incident angle into the target (wobbling), the secondary particles are generated in the target, passing a spectrometer magnet, are collimated and guided to two beamlines. Original taken from [43] and modified

Figure 6.4 shows a schematic cross section through a so called septum magnet. A magnet like this is used at the CERN SPS to eject the circulating beam into the north area beamlines. The septum magnet works like a dipole magnet with a homogenous magnetic field to bend the beam in the horizontal direction. A part in the magnet's upper pole shoe is hollow. This opening is very close to the gap. When the beta function[1] of the beam is horizontally focussed, but vertically wide enough to graze the opening in the pole shoe, the part of the beam traversing the opening will not be affected by the magnet and will move on straight ahead. The other part will continue its way in the accelerator's circle.

This kinds of ejection can be control by defocussing the beam vertically. The part moving on straight will be ejected. The ejection time depends on the focussing of the beam. The ejected particles will be guided into the beamline's target. The impacting particles are doing hadronic interactions and are producing other particles. Typical particles coming out of the target are [43]

- Remnant protons from the primary beam that are attenuated to $\approx 40\,\%$ of their initial energy.
- Pions, that have been produced in hardronic interactions
- Electrons, produced in electromagnetic processes
- Muons, produced in the decay of the pions

The secondary particles are produced in a large variety of angles from the target. The most energetic particles will have an angle in forward direction with respect to the primary beam. At CERN SPS the so called wobbling mechanism (figure 6.5) is used to give the beamline users a high flexibility of the beam parameters. For example the target station T4 delivers secondary particles to two independent beam lines. The secondary particles are collimated to deliver mono-energetic particles to the beamlines. A spectrometer magnet in front of the collimators allows the user to choose a momentum for both beamlines. In order to choose individual momenta for both beamlines the incident angle into the target of the primary beam needs to be changeable. This is called wobbling.

[1] The envelope of the gaussian beam profile

6.5.3. Interlock System

In order to guarantee the safety of the users' lifes an interlock system is used. It allows an easy access into their beam area without affecting other users much. The H6 beamline is connected to three experimental beam areas H6A, H6B and H6C. These beam areas are built behind each other, that means that one particle traverses all three experiments. Each beam area can be locked through so called interlock doors and is not accessible from outside. A cross passing from one beam area into another is not possible either. Each beam area has access to a beam shutter. This shutter is a big iron block, multiple meters long, that can be moved inside the beam. When the beam shutter stops the beam for one beam area no beam will be delivered for the followed beam area of course. The interlock safety system does not allow to move the beam shutter in order to activate the beam immediately. The user has to stick to some rules represented by interlock modes. Only by using a computer terminal connected to the interlock system the modes can be changed.

With the help of this terminal the interlock system can be set into three different modes:

Free mode The interlock system does not allow beam, the interlock doors are open, everybody can enter.

Access mode The interlock system does not allow beam, the interlock doors are locked, but can be opened with an interlock key.

Beam mode The interlock doors are locked, the beam shutter is open.

The changing from mode 'free' to 'beam' is not allowed immediately. All other changes are allowed with different preconditions.

'Free' is the typical interlock state when a setup is built up or removed from a beam area. Everyone can enter the area. To change from this mode into 'access' a special so called search key is necessary. Only people from the SPS control or users with a special training have the privillege to enter the area and lock the door. It is obligatory that the area is searched by this person, in order to check that no one else is hidden somewhere in there. At different places in the area some buttons need to be set into 'searched' state with the help of the search key. After the area was searched the interlock system is in 'access' mode.

In 'Access' mode the users can get an interlock key out of the interlock door. Eight keys are available. The door can be opened from inside and outside with one of these keys. As this key is a kind of a life insurance it is forbidden to enter the area in this state without a key. To make sure everybody sticks to this the interlock systems goes into 'free' mode when the door is open for longer than 60 s [43]. With the help of the computer terminal it is very easy to set the system into the 'beam' state. In this case the system checks that all interlock keys are at their place, in order to make sure nobody is in the area. If this precondition is fulfilled the system will move the beam shutter out of the beam.

The changing from 'beam' state into 'access' goes in the same way. The system will move in the beam shutter and checks the radioactivity level in the area. If everything is checked and there is no risk of radiation the system will be in 'access' mode. The changing between 'access' and 'beam' mode takes some minutes as the system checking the preconditions is very slow. These interlock modes are the typical modes that are used during a testbeam as changing from the 'Free' mode takes long time and needs help from the SPS Control.

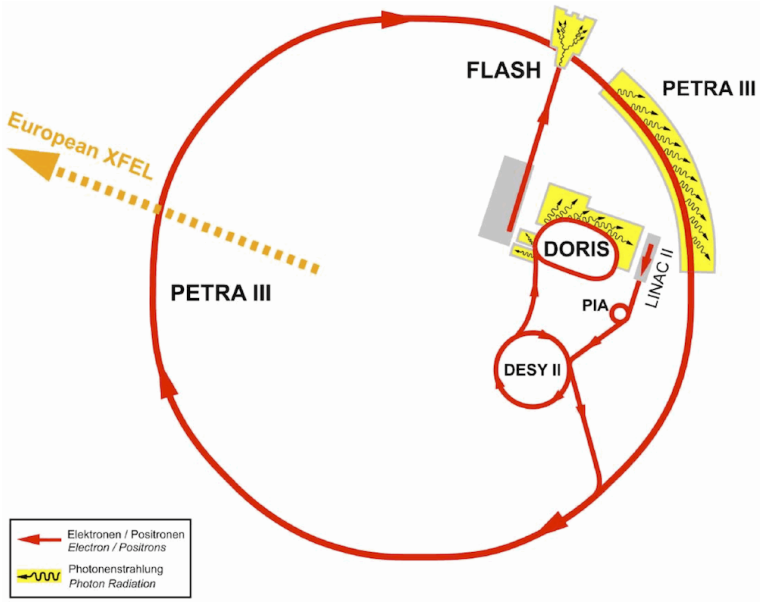


Figure 6.6.: Accelerator complex at DESY [23]

An emergency button at each door allows to the user to enter or leave the area immediately. To guarantee that the area is safe instantanously the SPS beam will be dumped suddenly. The system will be set to 'Free' mode automatically.

6.6. Testbeam at DESY

In comparison to the testbeam setup at CERN the setup at DESY[1] is very compact and easy-to-use. Figure 6.6 shows the accelerator complex at DESY. At the beginning of a chain of accelerators for electrons and positrons are the LINAC II[2] and the storage ring PIA[3]. The maximum energy that can be reached is 450 MeV [24]. At its maximum energy the beam will be transferred into DESY II[4] and accelerated further on to a maximum of 6 GeV [22]. DESY II supplies other experiments namely DORIS[5], PETRA III[6]. The testbeam experimental hall is connected to DESY II as well. Up to the year 2007 the PETRA III accelerators was also a pre-accelerator for HERA[7]. But HERA was turned off and replaced after 15 years of particle collisions at the experiments H1[8], ZEUS[9], HERMES[10] and HERA-B[11] [25].

[1] **D**eutsches **E**lektronen **SY**nchrotron
[2] **LIN**ear **AC**celerator **II** - accelerator at DESY
[3] **P**ositron **I**ntensity **A**ccumulator - accelerator at DESY
[4] **D**eutsches **E**lektronen **SY**nchrotron II - accelerator at DESY
[5] German: **DO**ppel **RI**ng **S**peicher - double-storage-ring - accelerator at DESY
[6] **P**ositron **E**lectron **T**andem **R**ing **A**pparatus **III** - accelerator at DESY
[7] **H**adron **E**lectron **R**ing **A**pparatus - accelerator at DESY
[8] Particle detector at HERA, DESY
[9] Particle detector at HERA, DESY
[10] Particle detector at HERA, DESY
[11] Particle detector at HERA, DESY

6.6.1. Particle properties

The available particles at DESY are electrons and positrons. Their energy is limited to the maximum beam energy of the DESY II accelerator, which is 6 GeV. Due to this the general requirement for the particle energy in a testbeam as described in chapter 6.1 is only realised under special circumstances: With 6 GeV electrons the multiple scattering can not be neglected. Especially when bringing material close to the beam, or even in the beam the particle's trajectory will be affected which will result in a worse resolution. Multiple scattering needs to be included in the track reconstruction in such a case.

6.6.2. Beam Ejection

Instead of ejecting the particles out of the DESY II synchrotron by using a kicker, or a septum magnet, a thin carbon fibre is placed into the accelerator. The circulating particle does electromagnetic interaction and produces photons that are leaving the accelerator ring. In order to do pair production the photons pass a secondary target. A high energetic pair of electron and positron leaves the secondary target. By using a spectrometer magnet with a collimator a monoenergetic particle momentum can be chosen.

6.6.3. Interlock System

At DESY three beamlines that are completely independent from each other are available. All beamlines have an interlock system similar to that one at CERN SPS. The only main difference is that the routine to change from 'Free' to 'Access' state is much more easy as it can be done by the user itself. In addition the time it takes to change from one state into another is much faster (≈ 1 min) than the time at CERN.

7. Testbeam Mechanics

This chapter describes all mechanical building blocks which are necessary to perform testbeams of ATLAS pixel sensors with the EUDET testbeam telescope.

First the mechanical parts of the telescope are described. After that the development of a testbeam box for the DUTs is explained. As there are several requirements for a testbeam box, these are summarised as well.

7.1. Telescope Mechanics

Figure 7.1 shows a photo of the EUDET telescope including all supporting mechanics. The mechanical frame is made of prefabricated aluminum profiles[1]. The whole frame is based on a big mechanical XY stage that can be controlled remotely. By moving the frame on the table perpendicular to the beam in both directions, the user can align the beam spot into the middle of the telescope's sensors. Although the precision of the controller is in units of 100 μm, experience shows that this kind of stages are passing a huge mechanical hysteresis of more than 1 mm.

The mechanical frame is divided into a lower and an upper part. The big xy stage is fixed to the lower part. The two parts are connected in a way that allows to rotate the upper part vertically with respect to the lower part. This rotation can be done by a micrometer screw and measured with a calliper. This mechanical component can be used to rotate the telescope vertically with respect to the beam. The rotation horizontally with respect to the beam can be realized by adjusting the foot screw of the big xy stage.

Two tables are fixed on top of the upper part of the aluminum frame. These tables are the base for the telescope's sensors and are called arms. By using a dovetail guide, the top of the two sensor tables can be moved along the beam axis. This allows an easy access to the devices that are usually attached between the telescope's arms. The position of the sensors on their arms can be set individually. The sensors of the telescope are mounted inside aluminum frames that guarantee an exact alignment of the sensors along the beam axis. A heat exchanger is included under the coverage of the aluminum frames. One can see the attached rubber tubes in figure 7.3 used for pumping temperated water though the frames. On top of these aluminum containments one can see the readout boards.

Typically the DUTs are attached in between the both telescope arms to obtain the maximum tracking resolution. As the mechanical mountings of the DUTs depend on the telescope's users and need to be not directly connected with its support frame a more flexible solution is necessary. The space between the two telescope arms is used to mount a very precise xy stage[2]. On demand a rotary stage can be attached on top to obtain the full flexibility.

[1] So called 'Rose und Krieger' profiles

[2] The accuracy is in the order of 10 μm.

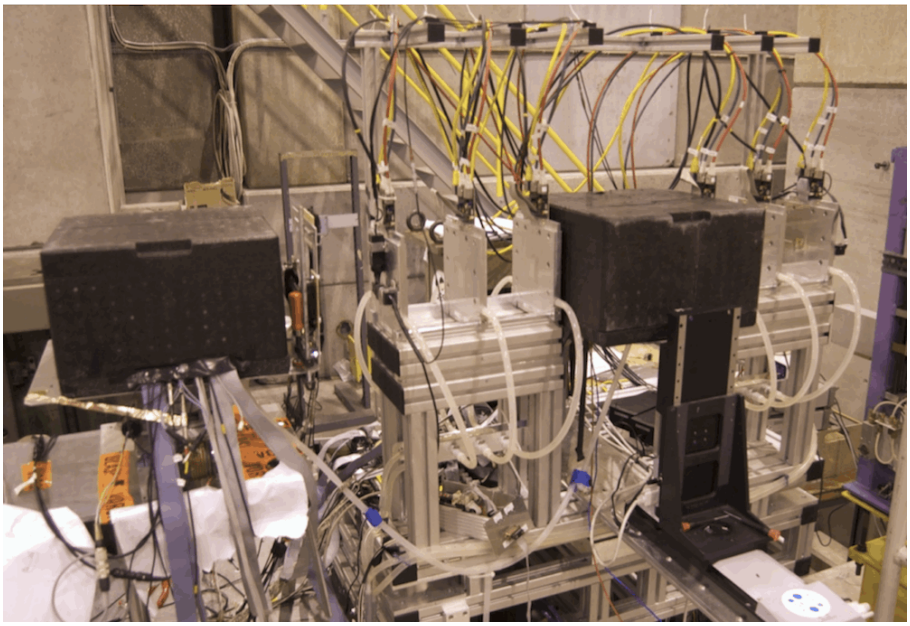


Figure 7.1.: Photograph of the testbeam setup at CERN SPS H6B. One can see the aluminum frame of the EUDET telescope with the six attached sensors. In the middle of both telescope arms the upstream DOBOX is located on the small xy stage. Downstream the telescope the second DOBOX is located.

7.2. DUT requirements

Depending on the sensor's design, its fluence and the intended studies the environmental requirements of the DUTs are changing. In the following sections all requirements and their necessity are discussed:

7.2.1. Tracking resolution

For almost all studies a tracking resolution at least one order of magnitude greater than the pixel pitch is necessary. For the ATLAS pixel DUT the tracking resolution should be roughly $5\,\mu$m in order to allow studies with sub pixel resolution. In this case the tracking resolution depends mostly on the z-position of the telescope's planes and the z-position of the DUT with respect to the telescope. The best resolution ($< 5\,\mu$m) can be obtained by placing the DUT in the middle of the six telescope planes (compare chapter 12.3). As long as the amount of material that is brought into or close to the beam together with the sensor is small enough to neglect effects coming from multiple scattering the tracking resolution will not be affected. Of course only the devices which are placed downstream the material causing scattering will be affected by a worse resolution. Therefore, it is always a good procedure to put DUTs which bring in a lot of matter in the beam at very downstream positions. Even a position close behind the telescope will have a sufficient tracking resolution ($\approx 10\,\mu$m) to do sub pixel studies (compare chapter 12.3), but without affecting the tracking resolution of the whole setup.

7.2.2. Positioning accuracy

In order to extrapolate a track through the DUT, recognised by the telescope, one needs to make sure that the telescope sensors and the DUTs are overlapping in the xy plane during the data taking. This can be controlled very easily by looking at the plane correlations in the online monitor (see chapter 9.4). As the DUTs are typically not well aligned when the system has just been set up, one needs an easy and reproducible way to move the DUTs relatively in the xy plane. A mechanism to move the planes remotely is recommended as this saves a lot of time as the interlock system does not need to be used in that case. In addition, a manual intervention always bears the risk of damaging something that was working before.

When using more than one DUT in the system, for example for doing efficiency measurements, it is important to have a high overlap between these sensors. So, the mechanics should provide the possibility to mount the devices in such a way. A mechanical possibility to move the DUTs relatively to each other in the xy plane would be even better in order to deal with production uncertainties of the accurate position of the sensor on the PCB.

7.2.3. Rotary mounting

Rotary stages are useful because they allow lots of studies: When investigating different incident angles, a possibility to rotate the DUTs at their mounted place is required. Scans of varying incident angles in both directions (φ-rotating around the long pixel side, and η-rotating around the short pixel side) are interesting. If the user needs to change the incident angle several times as this is part of the study, a remote controlled solution is strongly recommended in order to save time at the interlock system.

As long as the change of the incident angle with respect to a perpendicular incident is little, one rotary stage effecting a set of DUTs is sufficient. In this case one has to take into account that the DUTs are shifted out of their best position as they are not completely overlapping anymore. If the shift of the x- or y-position of the DUTs is so great that the sensors are not overlapping at all, a rotary stage per DUT solution is necessary. This solution offers a lot more flexibility as well, as not all sensors need to have the same incident angle. In order to do grazing incident angle studies a way should be provided to mount a DUT 90 ° along the x- or y-axis with respect to the other DUTs.

7.2.4. Cooling

In the case the DUT is an unirradiated sensor the main heat dissipation is produced by the read-out electronics (typical working condition of an FE-I3: $\approx 0.25\,\mathrm{W}$, calculated from values mentioned in chapter 4.3). An irradiated sensor needs a higher bias voltage in order to be fully efficient. With this the leakage current will increase and will bring the heat dissipation of the sensor into the same order of magnitude as the FE's heat dissipation. (for example: $200\,\mu\mathrm{A} \cdot 1000\,\mathrm{V} = 0.2\,\mathrm{W}$). As shown in chapter 3.2.2 the leakage current can be reduced by cooling down the sensor.

The necessity for cooling becomes obvious because the maximum leakage current compensation per pixel in a FE-I3 is limited to 100 nA. Therefore the maximum leakage

current for the sensor is limited to 288 μA. To be on the safe side during testbeam, the limit for the leakage current was 100 μA or under special circumstances 200 μA.

Depending on the sensor's properties a current which is too high leads to so called thermal runaway which means that the sensor is heating up itself with the heat dissipation of the leakage current.

For irradiated sensors a sensor temperature lower than $-15\,^{\circ}$C is sufficient. In order to do studies with highly irradiated sensors ($> 10^{16}\,n_{eq}\,cm^{-2}$), which are fully depleted, even lower temperatures are necessary.

7.2.5. Covering

As the sensors with applied bias voltage are sensitive to light, due to the Photonic Effect, it is important to protect the devices against environmental light sources. This can be realised by covering the device either directly on the PCB or by building an enclosure around one or more DUTs.

Beside the light shielding, covering can also work as isolation and therefore affects cooling. In order to avoid condensation of air, induced by cooling the direct environment of the DUTs, their surrounding volume needs to be flushed either with dry air, or another gaseous medium that does not condensate on the DUTs. In this context the covering acts as an enclosure with exchangeable atmosphere. The condensation of the water on the DUTs needs to be prevented, as shorts on the PCB are possible, which can cause errors in the communication or flashovers in the high voltage part.

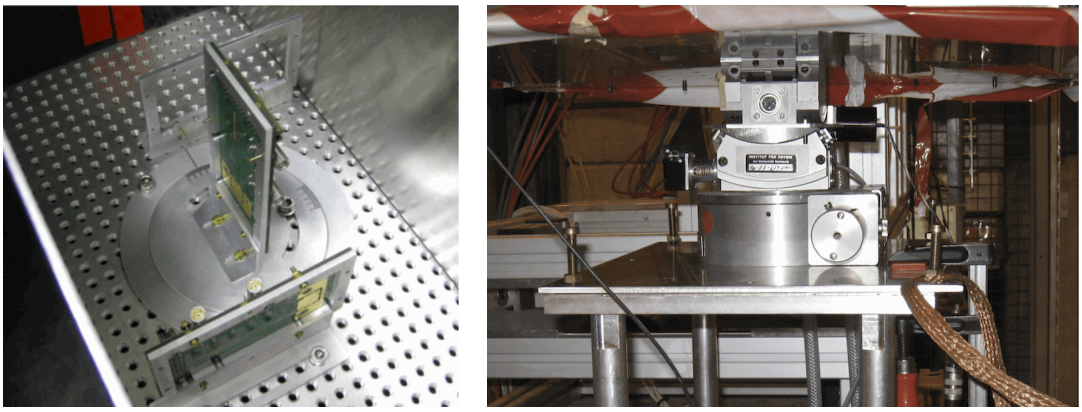
7.2.6. Flexibility

Testbeams demand a high flexibility of the users and their equipment. Therefore, not one exact way to mount the samples should be provided, but a most flexible solution that can be used in the future as well is highly recommended.

7.3. DUT Mechanics

As already mentioned in chapter 7.1 the typical way to adapt the user's DUT to the support frame of the EUDET telescope is to use an xy stage. On top of it a rotation stage can be set up.

In order to do testbeam with ATLAS pixel sensors, which have multiple prerequisites, a more specialised solution was necessary. These were discussed in chapter 7.2. In the context of this thesis three DUT testbeam boxes have been developed. These will be described in chapters 7.3.1 to 7.3.3



(a) Inside of the DOBOX (b) Goniometer under the DOBOX

Figure 7.2.: Photographs of the inside of the first generation of DOBOX and the goniometer, onto which it was attached

7.3.1. DOBOX

The first version of the DOBOX[1] was built for the first testbeam of the PPS[2] community in October 2009 at CERN SPS. The box was created with the aim to do studies with DUTs at SLHC fluences ($> 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$). Grazing angle studies were planned in particular in order to obtain information about the depletion depth under different bias voltages.

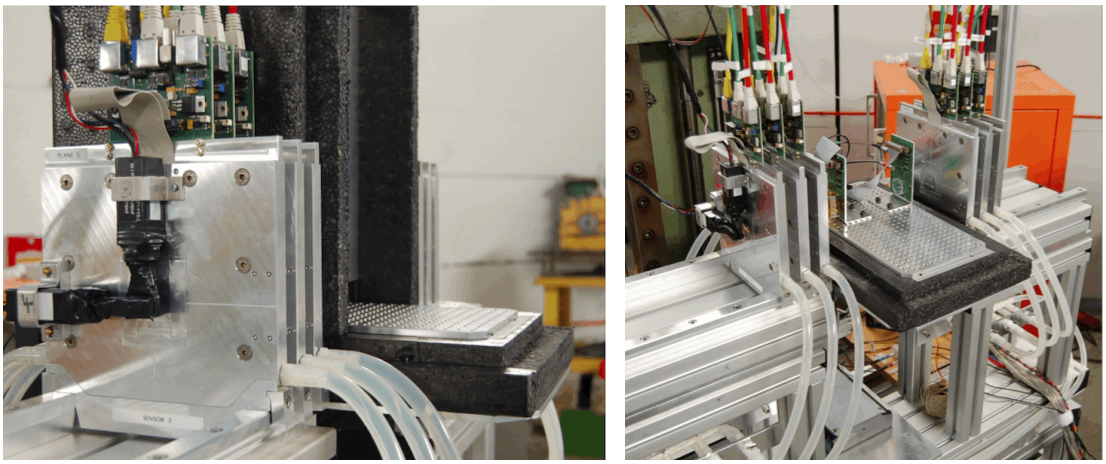
The box is made of PPE[3] foam, which is a material similar to styrofoam but more stable. At the bottom of the box an aluminum board with a grid of M5 threaded holes is provided. These holes have a distance of 1 cm. The box is divided into three parts; the middle part is used for mounting the DUTs. In figure 7.2a the middle part of the DOBOX is shown. One can see possible solutions to mount the devices. In this picture a theoretical beam would pass from the upper left to the lower right or vice versa. Samples are mounted to the thread grid by using L-shaped SCA holders. By using slotted holes the solution offers a lot of flexibility as the devices can be moved horizontally, or rotated around the vertical axis (only the SCA mounted in the middle).

It turned out, that the slotted holes, offering high flexibility, were the main reason for a lot of manual interference, especially as the reproducibility of the accurate positioning of the DUTs cannot be granted.

In order to allow grazing incident angle studies with various angles the box was attached to a goniometer. The goniometer allows a rotation of the box around the x- and y-axis and can be controlled remotely. The software for controlling the goniometer was developed as a part of this thesis. The DUTs for these kinds of studies needed to be attached directly at the center position above the goniometer.

The cooling in this box was realised by using dry ice granulate. The granulate was dumped into the separated parts of the box. The heat transfer was enabled through the aluminum board at the bottom and the L-shaped holder, attached to the DUT's PCBs.

[1] **DO**rtmund testbeam **BOX**
[2] **P**lanar **P**ixel **S**ensor
[3] **P**oly **P**henylene **E**ther



(a) The box half closed (b) Overview of the setup

Figure 7.3.: Photograph of the EUDET telescope at DESY. In between the both telescope arms the DOBOX-2 is attached. In (a) one sees the half closed coverage. In (b) two attached ATLAS FE-I3 samples are visible

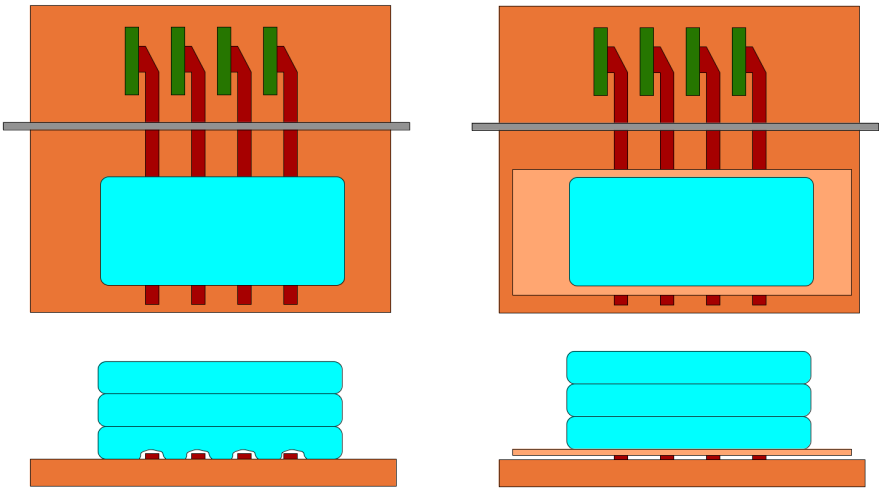
As the temperature of sublimating dry ice is roughly $-78,5\,°C$ a temperature of $-40\,°C$ was expected on the sensors by doing back-of-the-envelope estimations. Temperature sensors attached close to the sensors showed values of $-10\,°C$ to $-20\,°C$. The reason for this was an overestimated thermal conductivity of the PCB itself. Additionally, the thermal contact of the L-shaped holder to the PCB needs improvements.

When the dry ice is dumped into the box it needs to be closed and flushed with dry air or nitrogen very shortly after that. Only when the dry ice is completely sublimated the box can be opened again in order to avoid condensation. As the flexible slotted holes need a lot of manual interference, this proved to be a big problem as the cooling cannot be set off immediately.

As the box is heavy (roughly 25 kg, heaviest part is the attached goniometer), it was not possible to mount the box onto the provided xy stage. Therefore, a second big xy stage (like the one onto which the telescope was put) placed behind the EUDET telescope was used to put the DOBOX to its position in the beam.

Because of this the DOBOX is mounted independently from the telescope frame, which is a real advantage. During testbeam it was possible to mount two individual boxes, one in between the telescope planes and the DOBOX downstream the telescope. This solution offers a lot of flexibility, as the DOBOX can be used for specialised grazing angle studies, without affecting the run conditions in the primary box.

The accessibility to this first generation of DOBOX turned out to be very complicated as the whole setup becomes very high. The beam position is at roughly 2 m above the ground. Therefore, one needs to climb on ladders when doing the intervention. One big challange in this context is that the torque that is applied to the edge of the box is limited by the goniometer. This makes it even more difficult to access the box.



(a) Without aluminum board (b) With additional aluminum board

Figure 7.4.: Schematic of the thermal connections via a copper tape (red) between dry ice bricks (blue) and the DUTs (green)

7.3.2. DOBOX II

The DOBOX-2[1] is a redevelopment of the first generation box developed after experiences with the first one. It was first used in a testbeam at DESY in March 2010. In figure 7.3 one sees the EUDET Telescope with the attached DOBOX-2.

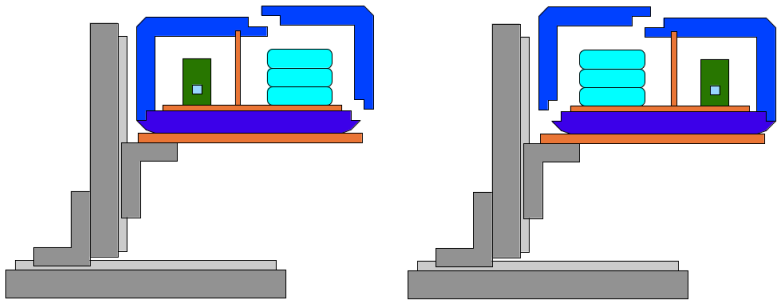
In comparison to the first generation DOBOX the DOBOX-2 is turned upside down. A similar kind of aluminum board is attached to the cover plate of the box, which is now the bottom. By turning the box in such a way the obvious advantage is a much easier access to the devices under test. The former body of the box is now used as coverage.

As the manufacturing of the PPE needed some experience first, the height of the box was not changed from its original size during its production. As a consequence the box turned out to be too big and unhandy. Furthermore, the new coverage was often opened by the force pulling at the cables leading out of the box. For cooling the DUTs dry ice was used again. But instead of using dry ice granulate, which sublimated very fast because of its big surface, dry ice bricks were used. These bricks can be removed easily in comparison to the granulate, which is a further advantage, as the cooling can be turned off much faster. A heating foil is attached under the aluminum plate to reduce the warm-up time additionally.

To avoid condensation, the space in the box is split into two parts. The first part is the volume in which the DUTs are placed. This part is made of half the PPE box to which an aluminum board is attached. This volume can be closed completely above the mounted samples. Additionally, this volume can be flushed with nitrogen.

In order to not effect the isolation of the PPE box too much by cutting it into two halves, the cut was not made straight through the material but with an intermediate step. With

[1] **DO**rtmund testbeam **BOX** second generation



(a) Mounting with a high torque (b) Mounting with a low torque

Figure 7.5.: Schematical cross-section through the DOBOX of the second and third generation attached to the xy stage (not to scale)

these steps in both halves, the parts can be pushed together. These steps are important as well for the mechanical stability.

When the first part is covered, the bottom aluminum board, to which the DUTs are fixed, reaches into the second volume. The DUTs will get cold when dry ice is placed on this aluminum board. The dry ice can be refilled or removed without opening the DUT volume. A copper tape is attached to the SCA[1] and reaches into the dry ice volume to obtain a better temperature gradient, as it can be used for heat transfer. The dry ice is placed on top of the copper tape but is also in contact with the aluminum ground board. Due to the different thermal resistivity of the tape and the board the dry ice sublimates faster where it is in contact with the copper tape (compare fig. 7.4a). Because of this, small caverns surrounding the copper tape were melted into the dry ice which limits a sufficient cooling.

Another obvious difference to the old box is the omission of the goniometer. The goniometer was the heaviest part of the old box and increased the box's weight so much, that the xy stage in between the telescope arms could not be used. The new box loses some flexibility as it cannot be rotated and tilted anymore but the advantage is a greater reproducibility of positioning by using the more accurate xy stage. In addition to that, the tracking resolution between the telescope planes is better than behind the telescope.

Although the weight was reduced a lot it is still very close to the limit of the xy stage. Especially when putting dry ice into the box (which has a load of a few kilograms) the limit is exceeded. During testbeam the box with dry ice could be used, as the table was strong enough to keep the box in place but it could not be moved. The reason for this is that the distance of the dry ice load and mounting suspension in the xy stage was very large (comp. 7.5a). A different mounting of the box on top of the table, as shown in figure 7.5b was not possible, as the xy stage could not be moved out enough to keep the sensor in the beam.

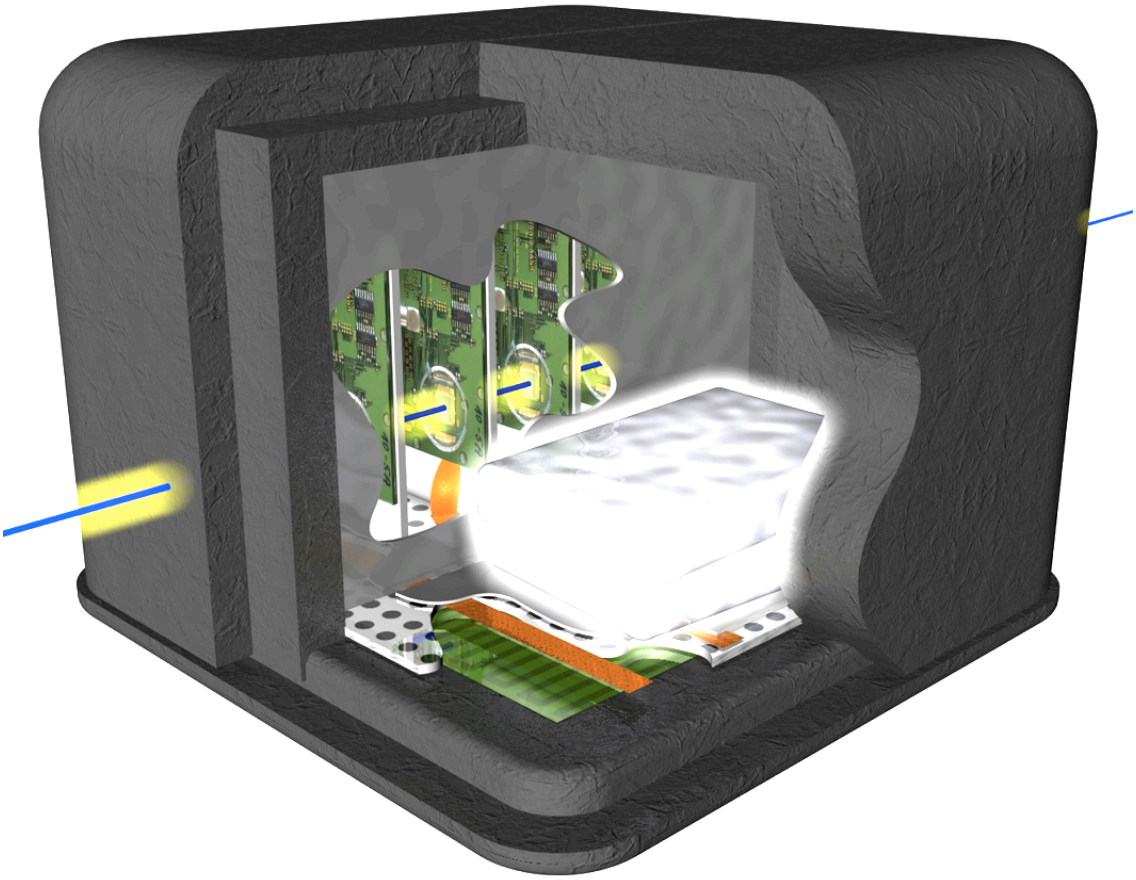


Figure 7.6.: Computer aided drawing of the DOBOX-3. Parts of the box are removed in order to illustrate the inside

7.3.3. DOBOX III

Figure 7.6 shows a computer aided drawing of the DOBOX of the third generation. This kind of box was built twice so that one box could be placed between the telescope arms and the other one behind the telescope. These boxes were used for the first time in July 2010.

The DOBOX-3[1] is basically a reconstruction of the DOBOX-2 with modified external dimensions and a few more features. As the DOBOX-2 was built for an electron testbeam with an energy of roughly 6 GeV, the box was as short as possible in order to keep the telescope arms close together to obtain the highest resolution. The DOBOX-3 was built for the CERN testbeam with 120 GeV pions. Therefore, the z-length of the box is the maximum size allowed by the telescope arms.

In order to avoid problems with the too high torque affecting the mounting suspension of the xy stage, the DOBOX-3 was attached to the xy stage as shown in figure 7.5b. This is

[1] The PCB carrying the sensor comprises an opening right below the sensor. A thin heatsink is glued between PCB and the sensor. So, the surface of the heatsink can be reached from the backside of the PCB.

[1] **DO**rtmund testbeam **BOX** third generation

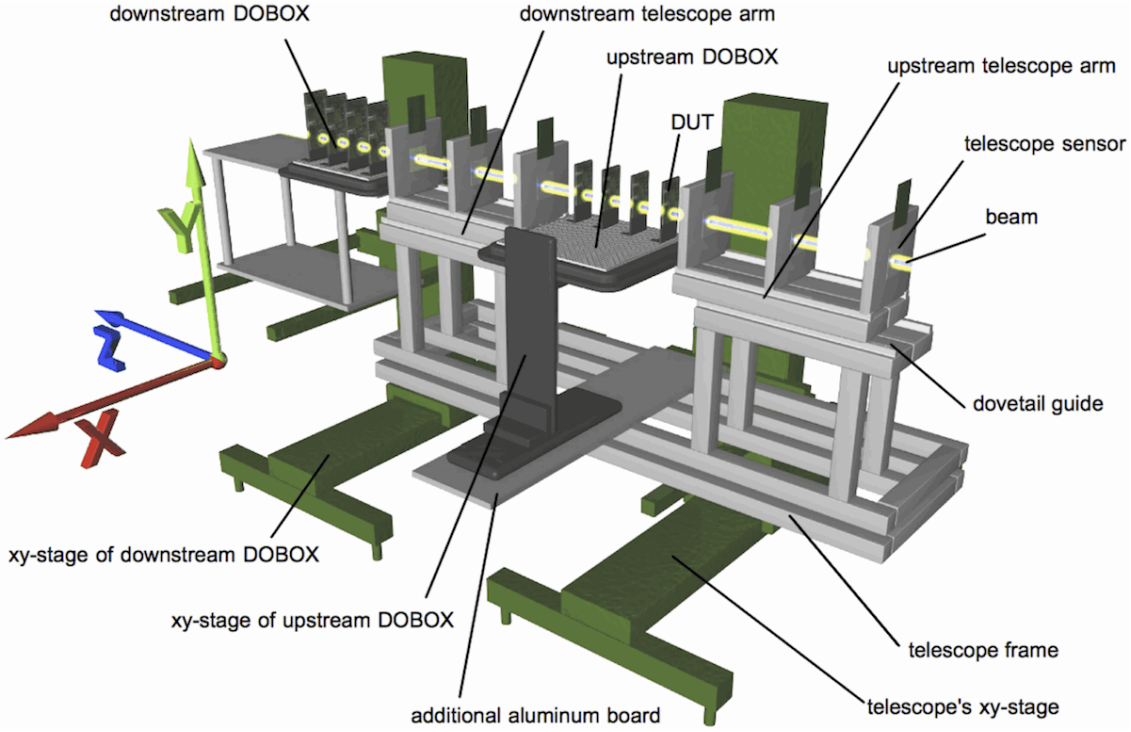


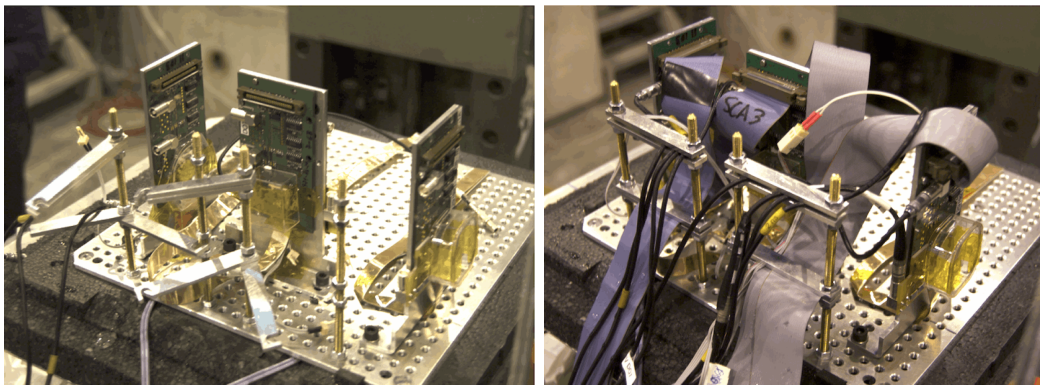
Figure 7.7.: Computer aided drawing of the testbeam setup. In this coordinate system the beam is coming from the right. It passes the first telescope arm, the first DOBOX, attached to the rotated xy stage, the second telescope arm and the second DOBOX attached to a separate xy stage

only possible when the xy stage is turned around 180° along the vertical axis. Therefore, an additional aluminum board which extends the telescope frame as a platform for the xy stage is necessary. In Figure 7.7 a typical mounting situation with two DOBOXes is shown. The second DOBOX is mounted on a separate xy stage in that figure.

As the cooling of the DOBOX-2 worked very well, apart from the bad conductivity between the copper tape and the dry ice bricks, the cooling concept was copied but further improved by attaching a thin aluminum board on top of the copper tape. With this board between the dry ice and the tape, a homogenous heat transfer is granted, and the creation of caverns in the contact area is limited (compare figure 7.4b).

It is possible to reach sensor temperatures below $-40\,°\mathrm{C}$, but a regulation of the temperature is still not possible. By changing the flow rate of the flushing air or nitrogen, the temperature can be changed, but this is tricky. In further box generations which are in preparation, a heater element is attached on the copper tape close to the sensor. Preliminary tests have shown [82] that little heating power ($\approx 5\,\mathrm{W}$) influences the heat dissipation to the dry ice, so that the temperature on the sensor can be changed over a wide range. The system acts like a thermal transistor. A regulator circuit for controlling the power of this element is planned.

The DOBOX-3 offers a lot of possibilities to mount devices. It proved to be the recommended testbeam box for future testbeams. As it will be described in chapter 7.3.5 different SCAs have been adapted to the box and their installation under varying incident angles



(a) Strain relief loose, without cables (b) Strain relief closed, cables attached

Figure 7.8.: Photographs of the DOBOX-3 at the testbeam at CERN in October 2010 to illustrate the strain relief attached to the box

was possible.

A strain relief is attached to the DOBOX-3 to avoid an accidental opening of the coverage. The strain relief is made out of threaded pins that are screwed into the inner aluminum board. Small aluminum shims act as a strain relief as they can be pressed together with nuts screwed onto the threaded pins. The photographs in figure 7.8 illustrate the usage of this system.

7.3.4. OSLOBOX

The OSLO-BOX[1] is a completely independent development of a testbeam box produced by the University of Oslo. Figure 7.9 shows a photograph of the OSLO-BOX.

The box is made of a heavy aluminum ground plate which can be fixed between the telescope arms without using the xy stage. Therefore, the DUTs cannot be moved relatively to the telescope. Each DUT is mounted on top of an individual rotation stage, which allows a full flexibility in this degree of freedom.

The fixed positioning of the DUTs is a big disadvantage, as a manual aligning of the samples is almost impossible. For example, in order to move the box vertically or horizontally it is necessary to unmount the coverage, all samples and the chiller tubes, which takes two hours minimum.

The cooling is realised by attaching the backside of the DUT's PCB onto a board that can be cooled down by using a peltier element. The waste heat of the peltier is dissipated by attaching it to a heat exchanger. The heat exchanger is flushed with cooling liquid by using a chiller. The current in the peltier elements can be controlled by a PID[2] regulator in order to have an almost constant temperature on the sensors.

Unfortunately, the minimum temperature of the frame next to the sensor[3] is limited to

[1] **OSLO** testbeam **BOX**

[2] **P**roportional-**I**ntegral-**D**erivative controller

[3] The temperate of the sensors itself was never recorded directly. Due to thermal conductivity a temperature difference of roughly 5 °C was expected.

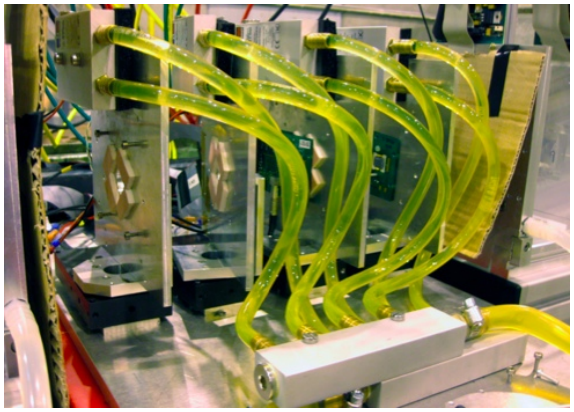


Figure 7.9.: Photograph of the OSLO-BOX without its coverage. Four independent rotary stages and the tubes for the cooling liquid can be seen

$-15\,^{\circ}\mathrm{C}$ as the limit of the chiller is reached then. For a short period of time $-20\,^{\circ}\mathrm{C}$ could be reached, but the regulation was not working properly.

7.3.5. L-mounts

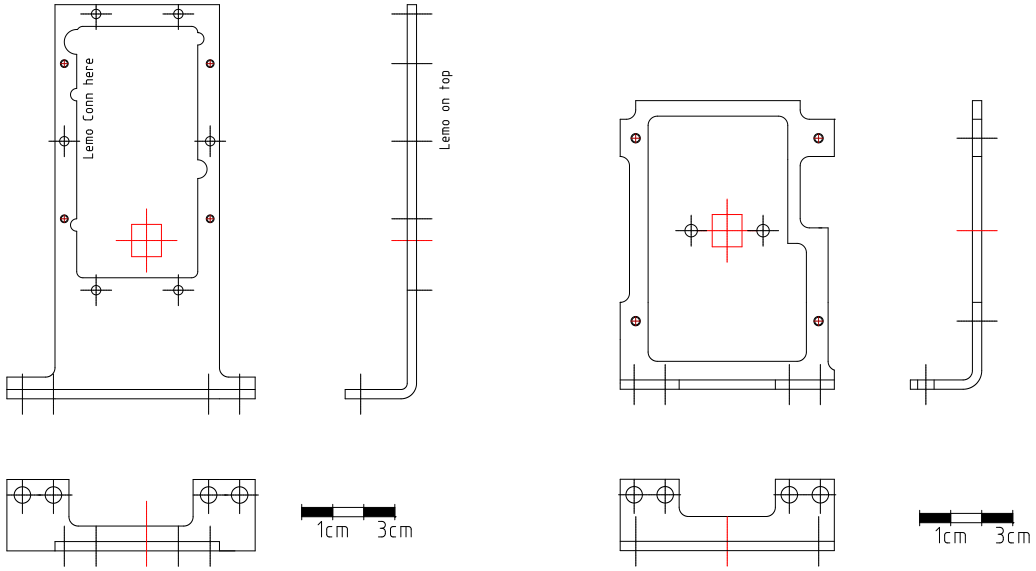
To connect the DUTs, assembled on their PCBs, to the ground plate of the testbeam box, so called L-mounts have been developed. Figure 7.10 shows the corresponding technical drawing. The mounts are made of 3 mm aluminum which is bend 90° at one side. The one side, to which the SCA will be mounted, is milled out, so that only a very thin frame remains in order to avoid multiple scattering of the beam in the aluminum. Some places are milled out further to avoid short circuits with open wires or components on the SCA with the aluminum holder. Threads are tapped in the remaining frame to enable a fixing of the SCA with M2.5 screws.

The other part of the L-mount has drill holes in order to mount the construction to the ground plate of the testbeam box with M5 screws. As one can see from figure 7.10 an individual L-mount needs to be constructed for each type of adapter board. The figure shows the L-mounts for the common FE-I3 Bonn SCA and the Barcelona version of the single chip adapter board. To have the full flexibility to use all versions of the L-mounts in parallel, it was carefully checked that the position of the sensor, mounted to the SCA, (printed in red) is placed in the same xy region with respect to the testbeam ground plate. For testbeams with the new chip generation FE-I4, L-mounts for the corresponding adapter boards were produced as well.

The L-mounts do not contain any slotted holes in order to obtain a completely reproducable positioning of the sensors.

7.3.6. Wedges and rotation adapters

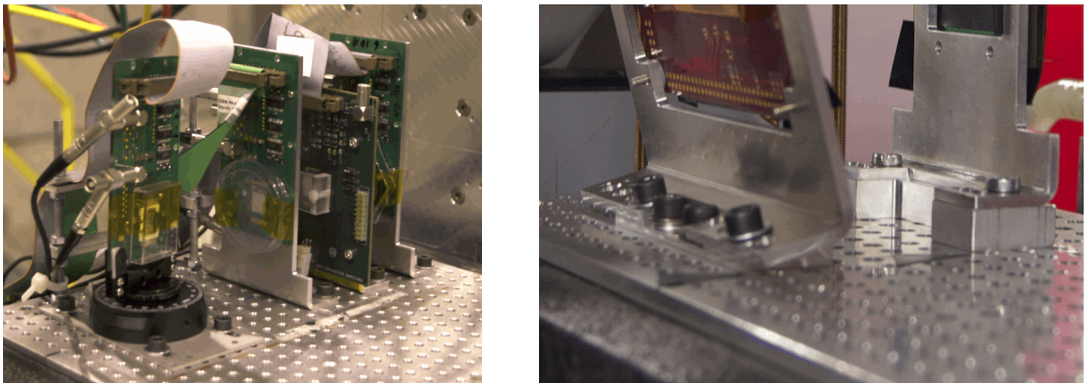
Special wedges have been developed to tilt the L-mounts along the horizontal axis. So far, wedges for 15° and 30° have been produced. These special adapters can be attached to the bottom part of the L-mount. To attach the wedge to the ground plate, a cut-out was



(a) L-mount for the Bonn SCA

(b) L-mount for the Barcelona SCA

Figure 7.10.: Technical drawings of the L-mounts for the Bonn and the Barcelona SCA



(a) L-mounts attached to the ground plate

(b) Wedges to tilt the L-mounts

Figure 7.11.: Photograph of the L-Mounts with SCAs attached to the DOBOX. Wedges are used to tilt the L-mounts.

made in the L-mount to obtain the necessary space. The photograph in figure 7.11b shows a mounted wedge. As the position of the sensor moves up when the sensor is mounted to the wedge, all other L-mounts need to be raised up as well in order to compensate the shift, as shown in that figure as well.

The wedges allow a rotation around the horizontal axis only. Special adapters to rotate the L-mounts vertically are under construction.

8. Testbeam Setup

In this chapter the setup which is used to record data in a high energy particle testbeam is presented. Some already existing parts of the setup and their extensions, which had been necessary for a combined data taking, are described as well.

8.1. Existing Testbeam Hardware

In the following sections all electronic hardware components used for the testbeam experiments in this thesis will be described. The telescope's trigger system, realised by the Trigger Logic Unit, will be explained in chapter 8.1.1. The read-out electronics which enables the telescope's data acquisition is discussed in chapter 8.1.2. The DAQ system acquiring data from the ATLAS FE-I3 SC[1]s is called TurboDAQ system and is presented in chapter 8.1.3. In order to enable a simultanous readout of both systems, as it is required in a testbeam, extensions to the TurboDAQ system had been necessary. These are summarised in detail in chapter 8.2.

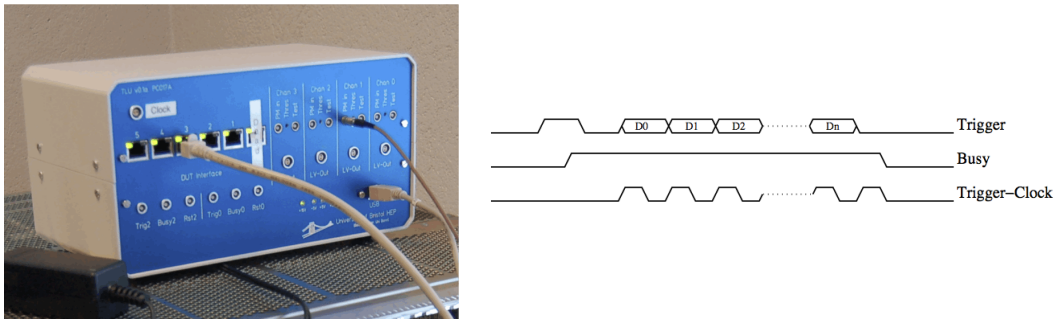
8.1.1. TLU

At the beginning of the read-out chain of the EUDET testbeam telescope and all kinds of DUTs is the Trigger Logic Unit (compare figure 8.1a). The TLU[2] needs a connection to all attached DAQ systems that are going to be read out during data taking. The function of the TLU is to distribute a trigger signal to all of them. The signal of four scintillator detectors are used as input to the TLU. Two of these scintillators are attached crosswise to the front of the telescope, the other ones behind.

Main parts of the TLU are made of a fast commercial FPGA[3] board [61]. The device can be configured via a USB[4] interface to generate a trigger depending on the input signals of the scintillators. Most common used is a coincidence of all four scintillators. A self trigger can be generated alternatively in order to test all DAQ systems without beam.

The distribution of the trigger is possible from the TLU's DUT interface. Up to six independent DAQ systems can be attached to one TLU. The DUT interface is realised by RJ-45[5] connectors, which are the commercial connectors for ethernet networking cables. The interface standard of the DUT interface is LVDS. Therefore, two wires of the eight-wired ethernet cable are used to send a trigger signal.

[1] **S**ingle **C**hip
[2] **T**rigger **L**ogic **U**nit
[3] **F**ield **P**rogrammable **G**ata **Array**
[4] **U**niversal **S**erial **B**us
[5] **R**egistered **J**ack-**45**



(a) Trigger Logic Unit (b) Trigger handshake

Figure 8.1.: (a) Photograph of the TLU, (b) Representation of the TLU trigger handshake [19]

After sending a trigger signal, the TLU expects a BUSY signal from each DAQ system. For this BUSY signal, a separate LVDS return line is necessary, which requires two more wires. An active BUSY signal in one of the connected DAQ systems prohibits the sending of further triggers from the TLU. Depending on the DAQ system, BUSY can mean that either the time the data collection takes for a sensor is not over, the system is in a kind of latency (compare chapter 4.3) or FIFOs on a DAQ system are completely filled, so that the complete read-out of the next event cannot be granted.

Beside the distribution of the trigger itself, the TLU can distribute a TLU-ID[1] to the DAQ systems. This ID[2] is very useful in order to guarantee a correct merging of the different data streams from all DAQ systems, based on this TLU-ID. To allow a sending of the TLU-ID, the connected DAQ system needs to send a clock signal not faster than 10 MHz, through a second LVDS return line back to the TLU. By receiving the clock signal, the TLU sends the bits of the TLU-ID through the trigger line. During this phase the BUSY needs to stay high, in order to distinguish a trigger signal from an ID signal. This sequence of bits is called TLU trigger handshake, which is illustrated in figure 8.1b.

The last two wires of the trigger cable connection can be used as a reset signal to the DUTs, which is only used in special cases.

8.1.2. EUDET Setup

Figure 8.2 shows a schematic overview of all components necessary to enable the read-out of the telescope's MIMOSA-26 sensors. The sensors are described in chapter 6.2.1. A summary of all components of the EUDET telescope can be found in [37].

The data acquired from the sensors is stored in so called EUDRB[3]s. These are fast FPGA DAQ boards with a VME[4] interface. One EUDRB is necessary for each sensor. The data is stored in FIFOs on the EUDRBs and is read out periodically by a DAQ PC. This PC is a VME computer, attached to the same VME backplane. It is often called MVME[5].

[1] **T**rigger **L**ogic **U**nit - **ID**entifier
[2] **ID**entifier
[3] **EU**DET **D**ata **R**eduction **B**oard
[4] **V**ersa **M**odule **E**urocard
[5] **M**imosa's **VME** PC

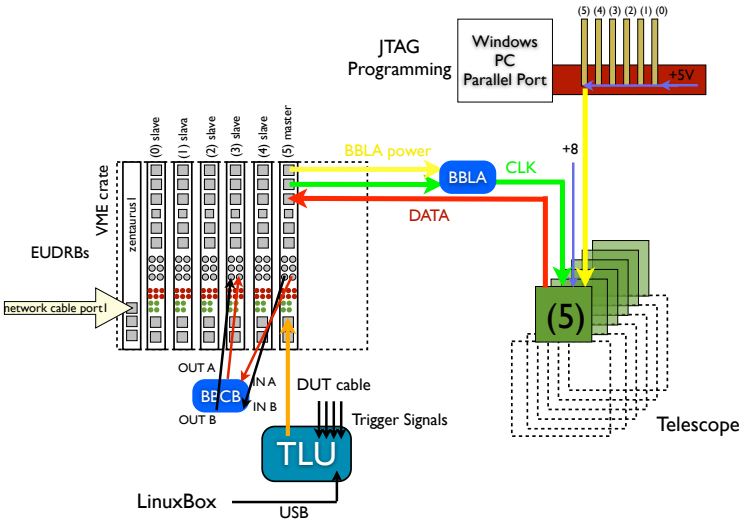


Figure 8.2.: Schematic overview of all cable connections within the EUDET telescope setup [37]

The readout speed of the six EUDRBs can be almost doubled by using two independent VME crates. Two MVMEs are needed in that case; one for three EUDRBs.

The TLU trigger signal reaches the EUDRBs at one designated master board per VME crate. From this point the trigger is distributed to the other EUDRBs in the crate by using so called BBCB[1]s. Beside this the function of the BBCBs is the distribution of a common clock signal to all devices.

The clock and trigger signals are sent to the sensors by passing the BBLA[2], which are used to adjust the LVDS signal levels from the EUDRBs to the sensor. Each sensor needs its own BBLA.

The configuration of the sensors is realised by a separate JTAG[3] interface. Each sensor is connected to a JTAG adapter board, which uses a parallel port connection.

8.1.3. TurboDAQ Setup

The TurboDAQ setup is not the standard read-out chain for ATLAS pixel modules. The standard way to allow data acquisition inside the ATLAS detector is very complicated, as wire-to-optical converters and vice versa and a lot of special and fragile hardware is needed. TurboDAQ was developed to allow tests of ATLAS pixel modules or SCAs in a laboratory.

Figure 8.3 shows a schematic overview of all components in a typical TurboDAQ setup, as used in laboratories. In the lower right part in that figure one can see an ATLAS pixel SCA. Either an SCA or a full module (not in the figure) can be connected. The SCA is connected to the TPCC[4] via a 40-pin ribbon cable with KEL connectors. Only

[1] **B**lue **B**ox for **C**lock **B**uffering
[2] **B**lue **B**ox **L**evel **A**dapter
[3] **J**oint **T**est **A**ction **G**roup
[4] **T**urbo **P**ixel **C**ontrol **C**ard

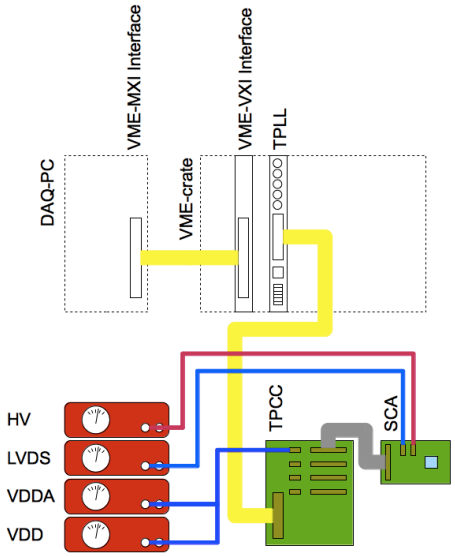


Figure 8.3.: Components and cable routings of a standard TurboDAQ setup for laboratory tests

ATLAS pixel FE electronics up to version I3 can be used with TurboDAQ. The main function of the TPCC is multiplexing between up to four connected SCAs or modules. The TurboDAQ setup can only readout or configure one of the SCAs or modules connected to one of the four ports on the TPCC. The TPCC was originally built for characterising many modules successively. Therefore, the multiplexing was useful. For testbeams, in which all connected SCAs need to be read out simultaneously, it cannot be used in that way. Beside the multiplexing, the TPCC contains LVDS drivers for the differential signals from and to the FE. To achieve the fast communication speed with the FE, an FPGA is used.

The TPCC is the place onto which the low voltages (VDDA and VDD) for the SCA are attached[1]. The SC adapter board itself contains LEMO connectors for the high voltage[2] to deplete the sensor, and a separate LVDS voltage to operate LVDS drivers on the PCB of the SCA.

The TPCC is connected to the TPLL[3] with another ribbon cable. The TPLL is the communication adapter to the FE. It is a complex multi-layer board with an FPGA and a connection to a VME backplane. With the help of the TPLL, the FE can be configured, scanned, tested, or set to acquire data.

The settings in the TPLL can be changed or read-out through the VME backplane and a commercial VXI[4]-to-MXI[5] adapter interface into a DAQ PC[6]. This PC is a standard desktop computer.

[1] In case a module is connected it is recommended to apply the low voltages via so called module adapter cards, in order to reduce the voltage drop
[2] Although the LEMO connectors are not qualified for voltages higher than 800 V, tests have shown that voltages higher than 3 kV can be applied to that connector.
[3] **T**urbo **P**ixel **L**ow **L**evel
[4] **V**ME-bus E**X**tension for **I**nstrumentation
[5] **M**ultisystem E**X**tension **I**nterface
[6] **P**ersonal **C**omputer

8.1.4. TurboDAQ with the BAT telescope

In former times the TurboDAQ setup was already used in testbeams, for example with the BAT[1] telescope [26, 55, 56]. That setup had several constraints:

Triggering The trigger was realised by a coincidence of scintillators with NIM[2] electronics. Furthermore, it was secured that no second trigger appeared directly after the first within the latency. To guarantee this, analogue NIM electronics were used. A trigger timestamp, as it is realised for the EUDET telescope, did not exist.

Parallel readout As only one SCA can be attached to one TPLL in parallel readout mode, up to eight TPLLs were connected to one DAQ PC, running multiple instances of the DAQ software.

Data acquisition The data of the different detector parts had been stored on their local DAQ PCs. A sequential merging was done at a later step.

The most important limitation is the non-existing synchronisation of trigger timestamps. The amount of desynchronisation was so large, that a complete run was stopped and restarted after one spill in order to resynchronise the system again [2].

It was problematic to attach multiple TPLLs in one system to the same DAQ PC, as it led to ambiguities between the TPLLs.

With the separate storage of the data, online monitoring or correlations between the sensors is not possible. Because of this, manual aligning of the sensors is very hard.

8.2. Testbeam Extensions to the TurboDAQ Setup

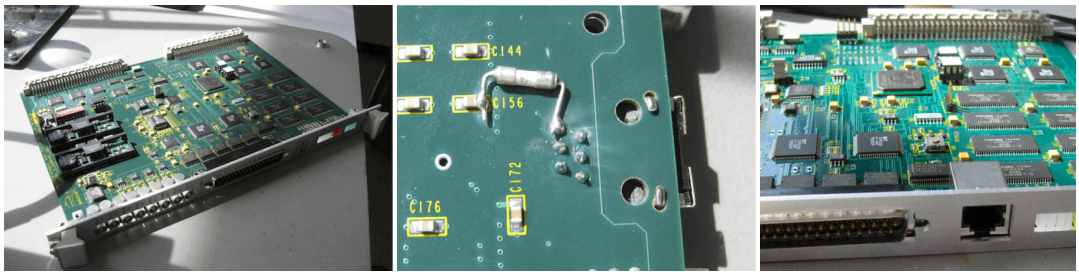
In order to use the TurboDAQ setup in a testbeam with the EUDET telescope, several extensions to this system had been necessary.

The main change of the TurboDAQ setup is a new interface in order to connect it to the TLU. Plenty solutions on how to attach the TLU trigger and its ID to the data taking system had been discussed. The final solution was to modify the TPLL:

An unsoldered connection on the TPLL board was used to interface to the TLU. The connection had the solder footprint for a RJ-12[3] connector[4]. To allow a communication of this port to the rest of the board, a modification of the FPGA was necessary. The improvements to the FPGA code are described in chapter 8.2.2.

As the digital signal standards of the TPLL and the TLU differ, their signals need to be adapted. This adaption is summarised in chapter 8.2.3.

[1] **B**onn **A**tlas **T**elescope
[2] **N**uclear **I**nstrumentation **M**odule
[3] **R**egistered **J**ack-**12**
[4] The RJ-12 is similar to the well known RJ-45, but with six instead of eight pins



(a) before extensions (b) microfuse attached to the trigger interface (c) trigger interface in the front panel

Figure 8.4.: Photographs of the TPLL

8.2.1. Conception of the extensions

The main change to be done is modifying the TPLL. On the TPLL a LEMO connector for trigger-in is already implemented in order to use the TurboDAQ setup with an external trigger, like a scintillator. Connecting the TLU to the TPLL means to reroute the trigger-in signal coming from the TLU. It needs to be handled as if connected to the LEMO connector. This is the first step in setting up the mentioned trigger handshake. This kind of rerouting can be done by changing the FPGA code, which contains the logic settings in the FPGA. Most parts of the extensions of the original FPGA code was done by Markus Mathes. For a complete overview the amount of changes and the added registers these are described here as well.

By changing this code, basically everything can be changed. Because of this, the modifications have been implemented in a transparent way for the users. This means when turning on the TPLL it needs to behave exactly like a usual TPLL without extensions. The extensions need to be activated by switching one register.

When the trigger signal coming from the TLU reaches the TPLL, it needs to send back the BUSY signal. For the time of the latency of the SCA readout, the BUSY signal will stay high. As the TLU cannot send a second trigger when the BUSY signal is high, it is secured that the latency time is not interrupted by such a signal. This is important, as the TPLL cannot handle such a situation.

With the BUSY signal sent out, a 10 MHz clock signal will be sent to the TLU as well. The clock can be generated from the 40 MHz clock, which is sent to the SCA to replace the LHC bunch crossing clock.

The TLU will send its trigger ID on these clock signals to the TPLL. With the changes of the FPGA code in the TPLL a second FIFO, the so called EUDET FIFO, was implemented. The arriving trigger ID will be stored in that FIFO. The TPLL trigger ID[1] will be stored in the first and the second FIFO every time an event occurs to allow a synchronised readout of both FIFOs. Depending on the settings of the registers, a 64-bit timestamp can be stored there as well.

In order to guarantee that the FIFOs will not overflow, the BUSY signal will rise up immediately when one or both of the FIFOs are almost full. Almost in this context

[1] On every trigger the TPLL will generate an internal 16-bit trigger ID

means that at least one more event can be stored.

8.2.2. Changes in the FPGA code of the TPLL

The transparency of the implementation mentioned before means, that the TPLL should appear as a regular TPLL with all its functionality. The additional extensions for the testbeam usage need to be enabled. Table 8.1 summarises the new implemented testbeam registers. These registers can be read out and modified by the DAQ PC via the interfacing VME backplane.

The register 0x21 is the common readable and writable control register for the EUDET extensions. The bits 15-8 represent the run mode. By booting the FPGA, the standard setting will be 0x0 for the whole register so that the TPLL will appear as a regular TPLL without EUDET extensions.

Beside some more test modes, the testbeam mode (bits 15-8 to 0x10) is the regular mode with enabled EUDET extensions. In the first 8 bits of this register the user controls a lot of settings concerning the trigger and the storage of different trigger IDs, e.g. if the bit 2 is set, triggers are accepted from the TLU through the new RJ-12 connector.

A new EUDET FIFO was implemented for storing the common identifiers. While the regular FIFO[1] is used for storing the event-based information, the EUDET FIFO is only used for storing the trigger IDs. Each time a trigger is accepted, the event-based information is stored in the data FIFO. The trigger IDs depending on the settings in bits 5-7 are stored in the EUDET FIFO. One is able to select the TPLL trigger ID which is generated in the TPLL, the TLU trigger ID which is sent from the TLU and a 64 bit timestamp.

The TPLL trigger ID contains the following information: The first 4 bits are always 0x8 and the last are 0x1. Because of this one can easily see that the collected data from the FIFO is a TPLL ID. The bits 25 and 24 contain the number of the following 32-bit words that are stored in the EUDET FIFO and belong to the same event. These words can contain the TLU trigger ID or the timestamp. The bits 14-8 contain the TPLL trigger ID.

The TLU trigger ID is simply stored in the first 15 bits. The first 32 bits of the timestamp correspond to the UNIX standard time. This time needs to be set in register 0x23. The second 32 bits contain the sub-seconds in units of 25 ns.

Register 0x22 shows a status about the EUDET extensions. The FIFO fill level is represented in 2 bits and also an overflow bit can be read out. In regular use the overflow will never happen as the TPLL will send a BUSY signal to the TLU when the FIFOs are almost full, which disables the sending of more triggers.

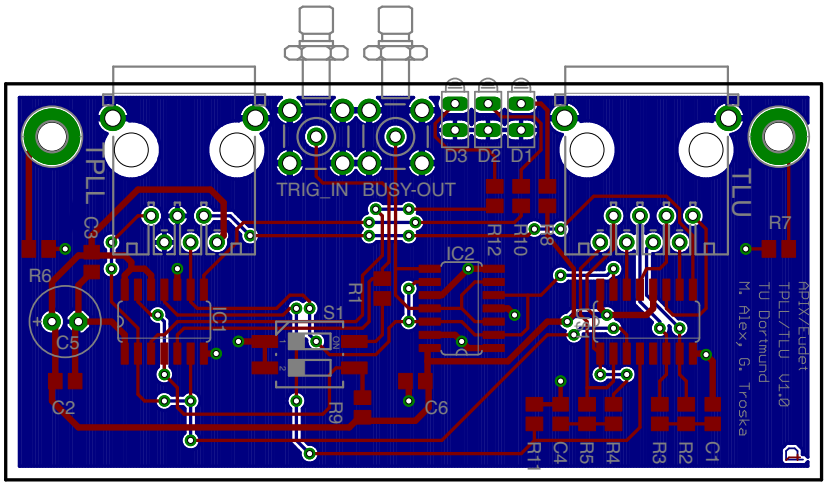
Register 0x22 can be used for writing as well but is only used for debugging.

As the 32-bit POSIX[2] time inside the FPGA is set to 0 when booting the TPLL, one can write (and also read) this time in register 0x23.

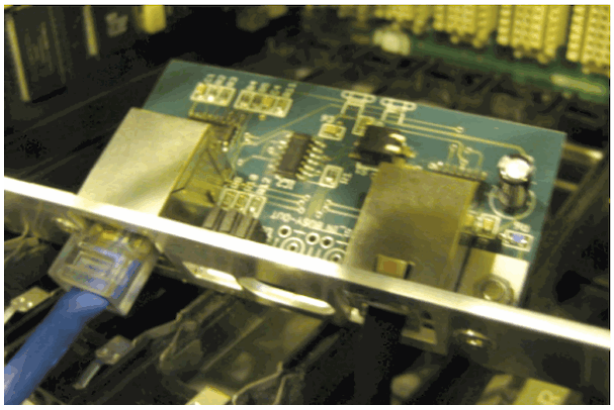
[1] The regular FIFO is now called data FIFO in order to distinguish it from the EUDET FIFO
[2] **P**ortable **O**perating **S**ystem **I**nterface for Uni**X**

0x20 (R)	**Indicator for EUDET interface availablility** (0xeded)
0x21 (RW)	**EUDET Extensions Control Register** Bit 0: Accept LEMO TPLL trigger Bit 1: Accept MONHIT trigger Bit 2: Accept EUDET trigger Bit 3: Accept Software trigger Bit 4: Stay busy (wait for clear of wait flag) Bit 5: Write TPLL Trigger ID to EUDET FIFO (32bit) Bit 6: Write EUDET Trigger ID to EUDET FIFO (32bit) and enable clock out Bit 7: Write timestamp to EUDET FIFO (64bit) Bits 15-8: Run modes: 0x00 Transparent Mode (like without EUDET extensions, but allows to accept also the other inputs beside LEMO), acts also as reset 0x10 Testbeam mode Test modes: 0xf0 On trigger write timestamp - writes 64 bits to EUDET FIFO (sec since 1970-1-1 (32bit), subsec ticks of 25 ns (32bits)) 0xfa EUDET, feeds back TRIG as BUSY on EUDET interface 0xfb EUDET, trig handshake, writes FIFO as selected Bits 31-16: Self Trigger - the interval in milliseconds (0 = off)
0x22 (R)	**Status Register READ:** Bit 1-0: EUDET FIFO Fill Level: 0b00: FIFO empty 0b01: FIFO contains data 0b10: FIFO half full 0b11: FIFO almost full Bit 3: FIFO overflow Bit 8: Wait flag (waits for clear) Bits 31-16: ms since last signal on trigger input (for end of burst detection)
0x22 (W)	**Status Register WRITE:** Bit 0: Clear wait flag Bit 1: Issue software trigger
0x23 (RW)	**Clock:** 32 bit POSIX time (unix std time)
0x24 (R)	**Number of words in EUDET FIFO**
0x25 (R)	**EUDET FIFO** (32-bit): Read data from FIFO depending of selection, each trigger places 32bit Timestamp (32bit), TPLL trigger ID (32bit), EUDET trigger ID (32bit)
0x26 (R)	**Number of accepted triggers** (32bit)
0x27 (R)	**Revision read as date** (0xYYMMDDxx)

Table 8.1.: Overview of the testbeam registers in the TPLL extensions



(a) Board layout



(b) Photograph of the board

(c) Crate

Figure 8.5.: Board layout and a photograph of the TPLL-to-TLU adapter board. The adapter embedded in the TurboDAQ crate is shown in (c)

To read the data of the EUDET FIFO, the user needs to read-in the value of register 0x25. As selected in register 0x21 the number of 32-bit words inside this FIFO per event varies between 0 and 4.

The number of accepted triggers since activating the testbeam mode is written to register 0x26. Register 0x27 gives a revision date of the EUDET extensions.

8.2.3. Adaption between the TPLL and the TLU

In order to allow a communication between the new TPLL trigger interface and the TLU, the signal standards need to be translated into each other. The standard of the TPLL is single-ended TTL[1], whereas the standard of the TLU is differential LVDS.

The figure 8.6 contains the schematic of the TPLL-to-TLU adapter used. The figures in 8.5 show the corresponding board layout and photographs. As one can see, the RJ-45

[1] **T**ransistor **T**ransistor **L**ogic

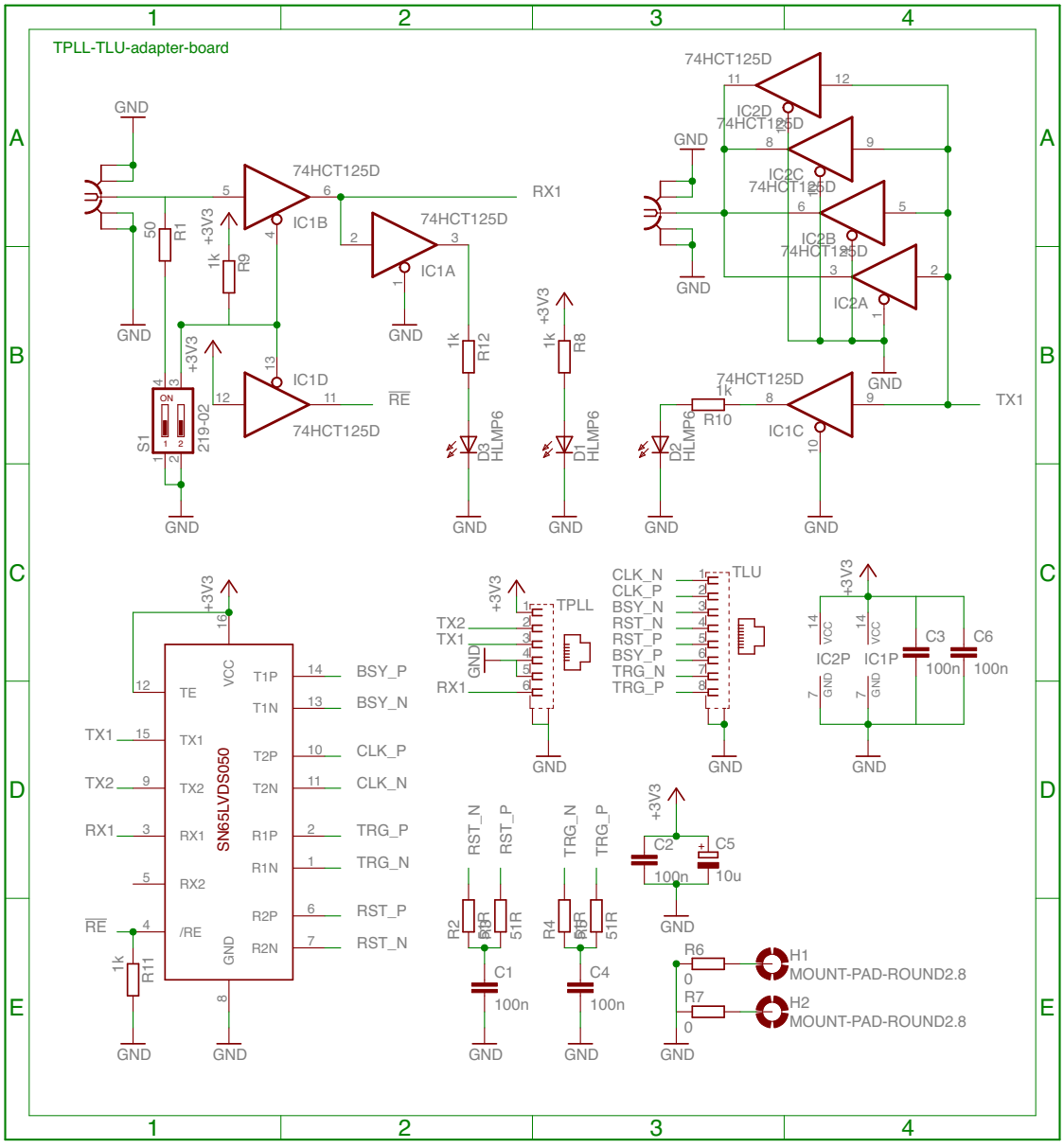


Figure 8.6.: Schematic of the TPLL-to-TLU adapter board

connector of the TLU and the RJ-12 connector of the TPLL can be connected in the adapter board. The board can be attached to an aluminium board to be plugged into the TurboDAQ VME crate right beside the TPLL (compare figure 8.5c). As the adapter board does not reach the VME backplane, the supply voltage needs to be routed from elsewhere. This is the reason why a microfuse was added to the TPLL (see figure 8.4b). The fuse enables the routing of the 3.3 V to the adapter board by using one of the signal cable connection. The fuse limits the maximum current to 100 mA to protect the TPLL from damages.

The commercial Dual Differential Transceiver Chip 65LVDS050 is used for the TTL-to-LVDS signal conversion and vice versa. This chip contains two converters for each direction. In order to use this adapter board for other purposes as well (for example a trigger interface with the BAT telescope) the board provides two optional LEMO connectors. With one of them a simple TTL trigger signal can be sent into the system. The other one returns the BUSY signal of the TPLL as TTL signal.

A micro switch is attached to the adapter board. Closing the switch at number 1 terminates the LEMO trigger-in with a 50 Ω resistor. The switch at number 2 changes the input trigger. The off-state (open) allows a usual communication between the RJ-12 and the RJ-45 connector. Closed state means that the LEMO trigger connector can be used to trigger the TPLL over the RJ-12 interface. In this case no trigger signals can be received via the TLU connector.

TTL driver chips are attached to the board, as the TTL signal termination by the users is unknown. For this the commercial chip 74LVTH125 is used. Two of them are used on the board. It contains four TTL drivers that allow currents up to ±128mA at 5 V per channel. To return the BUSY signal all four channels of one chip are combined. Because of this the line may be terminated with up to 10 Ω. The channels of the other chip are splitted. One is used to amplify the LEMO trigger-in signal. Two more are used to drive the LED[1]s showing activity on BUSY and the trigger line. The last one is used as an inverter to simultaneously switch between the LEMO and the TLU trigger-in by using only one micro switch. A third LED shows power on status.

8.2.4. MCC Board

The TPLL can either read out one full ATLAS pixel module containing 16 FE chips or one SCA with only one FE chip. So far (compare chapter 8.1.4) the parallel readout was realised by using one TPLL for each SCA. This was challenging as described in the chapter mentioned.

The reason why the TPLL can read out up to 16 FE on a module is the MCC. As described in chapter 4.5 it can distribute information to the individual FEs and merge the information coming from the FEs into a common data stream.

So, in order to connect more than one SCA to one TPLL, an MCC needs to become the interconnection. Exactly this is realised by the so called MCC board. Figure 8.7 shows a photograph of the MCC board. One can recognise the MCC ASIC attached in the board's

[1] **L**ight **E**mitting **D**iode

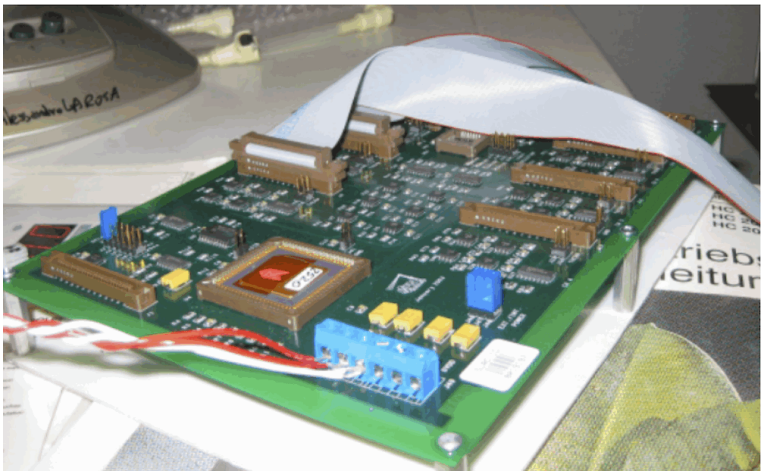


Figure 8.7.: Photograph of the MCC board

big socket. The connector to the left is the common ribbon cable going to the TPLL. The other connectors are used to attach the SCAs.

The MCC board can connect up to eight SCAs in parallel. The other eight possible connections are not routed out of the socket. Beside the well-known cable routings between the MCC and the SCA, which are principally a copy of the ATLAS pixel module's flex board, LVDS driver chips are attached to amplify the signals. Whereas the cable connections on a module are really short, the distances which need to be bypassed from the MCC board to the SCA are in the order of multiple meters. Therefore, these drivers are used to compensate voltage drops.

The MCC board offers plenty possibilities to connect the system to the supply voltages. VDD and VDDA can be attached to the TPCC and routed to the MCC board by using the ribbon connection. Alternatively, it is possible to attach it to the MCC board directly by using the screw joints shown in the photograph. The operating voltage for the MCC, usually VDD is used on a module, can be distributed from the TPLL. For all connections different ground connections can be chosen.

The different voltage supply schemes have been investigated in [75]. It has been shown that a well defined ground connection for all voltages is important in order to use the MCC board appropriately.

The board needs additional supply voltages of 3.0 V to operate the LVDS drivers. For operating the self trigger mode of the system, the hitbus signals coming from the SCAs need to be merged to a common OR and passed to the TPCC. To operate the OR gate, another -5 V supply voltage is necessary.

8.2.5. Description of the complete setup

Figure 8.8 shows the complete TurboDAQ setup and its cable connections with all extensions made for testbeam. In the upper right corner, one can see the four scintillators attached to the TLU. The TLU is connected to the TPLL through the level shifter board.

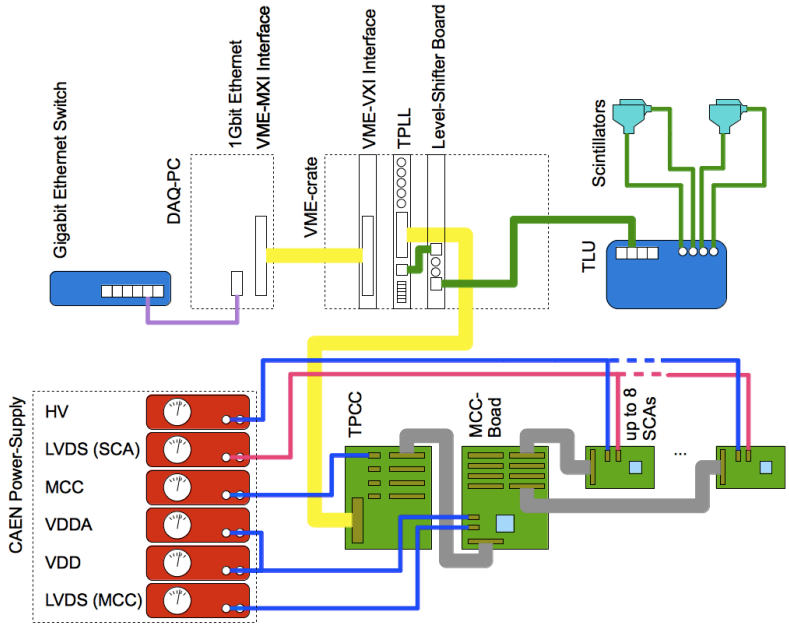


Figure 8.8.: Components and cable routings of a TurboDAQ Setup with the necessary extension for use in a testbeam environment

TPLL and TPCC are connected regularly. One of the TPCC channels is plugged into the MCC board.

Typically a CAEN power supply with exchangeable supply boards is used in the testbeam environment. With this power supply the ground was always well-defined. The MCC is powered via the ribbon cable from the TPCC. The supply voltages for the SCA's VDD and VDDA are distributed from the MCC board. The operating voltages for the LVDS drivers on the MCC board and the drivers on the SCA boards have individual power supplies. As the current consumption of the LVDS drivers on the board is much higher ($\approx 2\,\mathrm{A}$) than the consumption on the SCAs ($< 100\,\mathrm{mA}$), the splitting avoids accidental damage to the SCAs. The high voltage is passed to the SCAs directly. In case of unirradiated devices with uncritical operating bias voltages ($\approx 150\,\mathrm{V}$) and bias currents ($\approx 1\,\mu\mathrm{A}$) the supply lines can be interconnected. Irradiated devices should be operated at much higher voltages, because of this their leakage current is higher as well. 1 kV bias voltage and currents higher than $100\,\mu\mathrm{A}$ can occur. Therefore, these devices are operated with individual bias voltage supplies, which allow individual displaying and recording of the currents.

8.3. The Telescope-DAQ-Software EUDAQ

EUDAQ[1] is the software running the data-acquisition chain of the telescope and DUTs. EUDAQ is written in C++ and is available at hepforge [35]. Several threads of EUDAQ can run on machines communicating with each other by using sockets. These program parts have different functions. The most important parts are:

[1] **EU**DET **D**ata **AQ**uisition System

The Run Control is the master part of the software. All other threads in the system need to register to the run control first. Depending on their function, the Run Control handles them differently. Through a GUI[1] the user has the opportunity to start or stop a run. Furthermore, different configurations can be selected and sent to all connected threads.

A Producer is the data sending part in each DAQ system. As the system relies on the fact that all producers are connected to the same triggering system, one data stream is expected from each connected producer when a trigger occurs. The Producer needs to act on the signals distributed by the Run Control. This means if, for example, a Start Signal arrives, the system needs to enable the data taking of its connected sensors.

The Data Collector is the part of EUDAQ writing out the data streams of all producers into one common raw file. For each new run the Data Collector will create a new raw file.

The Log Collector has a similar function as the Data Collector. Each connected thread can send messages through the system. These are collected at a central place by the Log Collector.

A Monitor is a part of EUDAQ interpreting the raw data. For each Producer a Converter Plugin needs to be implemented. With this Plugin the byte stream can be converted into so called standard planes. These allow an online monitoring of all individual sensors.

A Listener is a special kind of a Producer that does not have the functionality to send data. This is useful if your system requires a thread that needs to know which run number is currently running without the necessity to send data streams to the Run Control. An example for this could be a temperature monitoring system that stores all data locally. In order to create a synchronisation between the temperature data and the run numbers the data could be stored run by run.

In order to allow a communication of all threads with each other, all program parts can be linked against the same C++ library, called libeudaq. In this library the functionalities of the different program parts are realised by using class inheritance.

The communication between the threads is technically allowed in both directions. However, the programming scheme of EUDAQ is a data push protocol. Therefore, the common way for communication is to push data into the Run Control. The return way of the information flow is limited. The Run Control sends only signals to the threads. These signals can include further information. If, for example, a signal is thrown because someone pushed 'Start Run', the OnStart method is called in every thread with an argument including the new run number.

The common way to distribute information from the Run Control to the connected threads is to use the configure method. In the Run Control window a configuration file can be selected and distributed to the connected threads. In listing A.3 such a configuration file is printed. The file is separated into sections; one section for each thread. The information is listed in key and value pairs. In the OnConfigure method of the different threads the interpretation of these pairs needs to be implemented individually.

[1] **G**raphical **U**ser **I**nterface

8.4. Configuration of the TLU

The TLU is the component responsible for the distribution of the trigger. It is described in chapter 8.1.1. The TLU can be configured by using a terminal program. This program is mainly used for debugging. A text line appears for each generated trigger with a lot of debugging information.

For the general use in testbeam, a TLU Producer is part of the EUDAQ framework. The Producer can be configured through the EUDET configuration file (compare listing A.3). As the TLU thread appears as a Producer in the Run Control, it needs to send data with each event in order to keep the event synchronisation. This data contains only the TLU trigger ID.

The next section will give an overview about the most important TLU configurations:

The BitFile parameter is the path to the filename containing the compiled VHDL[1] code for the TLU. It is essential and has to be set.

The AndMask parameter describes the bitmask which is used to create coincidences between up to all four scintillator inputs. The typical setting is a coincidence of all inputs, which leads to a value of 15. 15 in binary code is 0x1111. Each 1 in this string enables one coincidence.

The OrMask parameter can be used in the same way as the AndMask. But instead of building a coincidence by using a logical AND, any combination of enabled scintillators can generate a trigger by using a logical OR. The typical value in testbeam is 0. For debugging purposes this parameter was often set to 15, as a limited amount of muons can pass through the closed beamdump. In order to reach a higher rate the OR is very effective.

The VetoMask parameter can be used to set one or more scintillator inputs as veto inputs. The typical value in testbeam is 0, as vetos were never used.

The TriggerInterval parameter is always set to 0 in the common testbeam mode, as a self trigger can be initiated by using this parameter. The number corresponds to the interval time between two triggers in milli seconds. Therefore, a value of 1 initiates a 1 kHz self trigger. The self trigger mode is useful to test the DAQ chain without using a beam.

The DutMask parameter is used to configure the trigger distribution. Each active DAQ setup in the testbeam uses a trigger connection to the TLU. Up to six parallel setups can be connected to one TLU. Each setup that is connected needs to be activated in the TLU. The value describes a bit mask. When all six connectors are used, the value should be 63. It is important to set this value correctly, as an unset trigger line will not receive a trigger. Otherwise a set trigger line without an attached DUT system will lead to a BUSY state of the whole system. The reason for this is that the TLU distributes the trigger to the non-existing setup and waits for the BUSY high on that connector.

[1] **V**ery **H**igh Speed Integrated Circuit Hardware **D**escription **L**anguage

8.5. Software Extensions to TurboDAQ

In order to use the new implemented modifications of the TPLL, the DAQ Software, called TurboDAQ, needs to be changed as well. Furthermore, an interface needs to be set up to send the collected data to the common DAQ software EUDAQ. Both of these parts will be described in the following chapters.

8.5.1. TurboDAQ's interface to EUDAQ

An interface from TurboDAQ to EUDAQ is difficult to create as the platforms on which the programs are built differ very much. TurboDAQ is written in National Instruments LabWindows CVI [51], which is a kind of plain C Code. The underlying operating system is Microsoft Windows XP [89]. The amount of source code TurboDAQ is made of is roughly 100 000 lines.

EUDAQ is written in C++, which can be compiled on many operating systems like linux, Mac OS X and Windows (with and without cygwin) [19]. EUDAQ is written in roughly 16 000 lines of source code.

Therefore, the choice was to adapt EUDAQ to be running in a LabWindows CVI environment instead of doing vice versa. As the installed version of LabWindows was not able to handle object-oriented data structures, a work around was necessary. With a little effort EUDAQ compiled on Microsoft Visual Studio 2009 [86] using the embedded C++ compiler. The C++ classes have been wrapped into plain C functions, which were then put in a DLL[1]. This DLL could be imported into LabWindows CVI and used with all its features.

Listing A.1 shows the basic source code to explain how the wrapping from C++ to C was done. For a standard Producer, the connection to the Run Control is established as long as an object of the Producer exists. In C++ this can easily be realised by creating one instance on the heap (compare line 16). The wrapping from C++ to C is achieved in such a way that a subroutine creating such an instance is called; afterwards it is secured that this routine does not stop (compare line 18-20). A surrounding mutex ensures that the thread in which this subroutine lives is only called once. The thread can be initialised with a start function (compare line 1). The arguments are the hostname, or the IP[2] address and the name of the Producer. These parameters are passed into the thread that can be closed by calling a stop function (not shown), which simply sets the condition in the while loop (compare line 18-20) to false.

While the Producer thread is running, the collected data stream can be sent to the EUDAQ. The source code shown in listing A.2 shows the implementation of this method. The arguments passed to the function are the event object, the run number and the event number. The event object is defined in line 36. The common way to send data to EUDAQ is to separate the data streams collected from different sensors in this function[3], in order

[1] **D**ynamic **L**inker **L**ibrary
[2] **I**nternet **P**rotocol
[3] For example the EUDRBs run in such a way that the data streams are splitted at this point, in order to create one common byte stream conversion independent of the number of connected boards

to send it to EUDAQ in different blocks. For reasons, which will be explained next, the collected data is sent to EUDAQ in one block instead:

The primary reason is that a separation of the data at this point in the DAQ chain could cause permanent damage to the collected data. An accidental error in the implementation of the splitting mechanism could corrupt the data and would save it in its corrupted state. Therefore, a simple copying of the data stream is the more easy and reliable solution. The data will be saved by the Data Collector, to be interpreted later during the byte stream conversion (compare chapter 9.2).

The second reason is that such a conversion would be a conversion of the MCC format to the FE format. This would cause a lot of data overhead. Only the data part - without timing-information - of the FE consists of 16 EOE words of 32-bit length. A conversion would mean to copy this information for each connected FE.

As the data is not divided into blocks, the fourth parameter (compare listing A.2), which is the block number, is always 1.

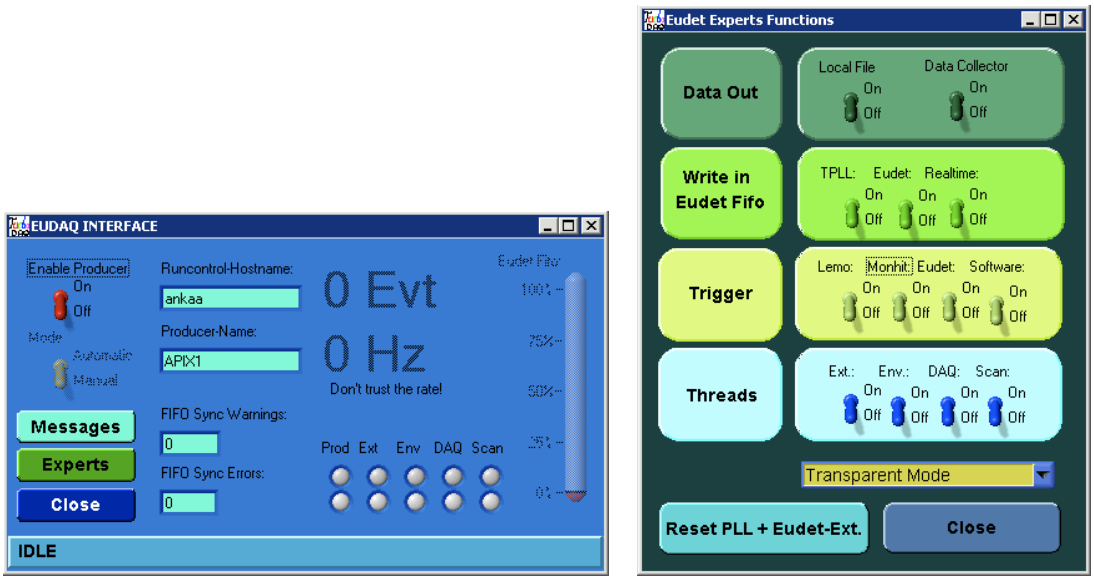
From the settings in TurboDAQ one can determine whether the connected board type is a Single Chip or a group of Single Chips attached via the MCC board as a Multi Chip. The kind of board type is part of the RawDataEvent, which is sent to EUDAQ. This information is used to allow an individual byte stream conversion of different connected detector types in the same run. As one can see in listing A.2 the data sent in the RawDataEvent consists sequentially of timing information and raw data.

The content of the timing information and the data part will be pointed out in chapter 9.2 Beside the event-wise raw data parts, the data of each run consists of a so called BORE[1] and an EORE[2]. These special events mark the first or the last raw data event. These kinds of events can be used to send dedicated information, so called tags, to the data collector. The function SendAPIXBORE (listing A.2, line 24) is the implementation of the TurboDAQ Producer for sending the BORE. As one can see, the tag PlaneMask is sent into the data stream. This PlaneMask contains the binary mask[3], which is a list of connected MCC channels. The mask is obtained from the configuration set in TurboDAQ. The implementation of the EORE (not shown) is the standard implementation for this function.

With the including of the compiled DLL, containing the wrapped source code of EUDAQ into TurboDAQ, the functions of the library can be called from the implementation of TurboDAQ. In order to allow a call of TurboDAQ's functions from the library, function pointers [41] were used. These so called call-back functions are used, for example, to allow a function of TurboDAQ to act when sending commands to the Producers like start, stop or configure. As the call back functions are not known during the compilation, they are implemented as void pointers to functions that point to NULL at the compilation time. Declare functions can be used to make these pointers point to functions embedded into the TurboDAQ source code.

Common call-back functions available are 'OnStartOfRun(int runnumber)', 'OnStopOfRun()' or 'OnConfigure(Config c)'. Further call-back functions are necessary in order to

[1] **B**egin **O**f **R**un **E**vent
[2] **E**nd **O**f **R**un **E**vent
[3] If all eight channels (0 to 7) of the MCC are connected, the mask is 255, a mask of 123 shows that the channels 0,1 and 3 to 6 are connected.



(a) Main EUDAQ interface (b) EUDAQ Experts window

Figure 8.9.: GUI windows in TurboDAQ representing the interface to EUDAQ as Producer

illustrate the behaviour of the system in the GUI of TurboDAQ. They are not important for the behaviour of the DAQ chain. As one can see, the corresponding functions are called in the implementation shown in listing A.1. In TurboDAQ the call of these functions is used as a control mechanism, in order to proof that the corresponding thread is active.

8.5.2. Software sided TPLL testbeam extensions

With the changes of the VHDL code of the TPLL, described in chapter 8.2, an adaption of the software is necessary as well. TurboDAQ needs to know how to handle the new implemented registers and how to readout data from the new FIFO.

Figure 8.9 shows two windows in the GUI of TurboDAQ which contain all functions of the TurboDAQ EUDAQ testbeam extension. The window shown in 8.9a is the main EUDAQ interface. The Producer thread (compare listing A.1) is activated with the red switch in the upper left. The text field can be used to set the hostname and the name of the Producer. In the lower right of this window one can determine ten GUI LED elements which are all inactive in their current state. By activating the Producer, the call back function OnStartOfProducer is called, which activates one green LED. When stopping the Producer, a red LED is turned on and the other one is disabled. The status bar in the bottom of the window is used to represent the different states of the Run Control like 'Starting Run' or 'Configuring'.

The experts window shown in 8.9b mainly illustrates the settings of the new testbeam registers described in table 8.1. With the switches titled 'Write in Eudet Fifo' the user can set the bits 5 to 7 in register 0x21. Depending on this setting, the EUDET FIFO will contain a TPLL trigger ID, a TLU trigger ID and/or a timestamp. The switches named 'Trigger' influence the bits 0 to 3 of register 0x21, which will change the trigger conditions

of TurboDAQ. With these settings it is possible to either trigger the readout of the FEs by a signal sent into the LEMO trigger connector, a signal sent from the monitored hitbus of the modules, a signal arriving the new implemented EUDET trigger connection or a software trigger. In most cases an activation of only one of these switches is useful. The combo box at the bottom of the experts window illustrates run or testmodes of the testbeam extensions (bits 8 to 15 register 0x21). The standard setting is the transparent mode in which the EUDET testbeam extensions will not appear. The testbeam mode is used for common data taking. Beside these modes, testmodes are available: 'On trigger write timestamp' will only write the 64bit timestamp into the EUDET FIFO, without doing anything more. 'Trigger feedback' will internally short the trigger and the BUSY lines of the trigger cable. This is often done manually to test features of the TLU. 'Trigger Handshake' will do a usual TLU handshake and will write out the data into the EUDET FIFO as selected above, but will go out of BUSY as soon as the handshake is over. In comparison to the testbeam mode, this setting will not take care of overfilled FIFOs or the latency time of the FEs.

By using the switches in the upper row of the experts windows, the destination of the raw data is set. The user can activate the dumping of the data into a local file or the sending of the data to EUDAQ's Data Collector. The last row represents a kind of main switch: The first switch will activate the EUDET testbeam settings in the TPLL. This means that the register 0x21 is set. The second switch will activate the environment thread, which will be described in chapter 8.5.6. The thread collecting the raw data from the FIFOs can be enabled or disabled with the switch titled DAQ. This will be described in chapter 8.5.5. The last switch will bring the FE into a scan mode that enables data taking (compare chapter 8.5.4).

8.5.3. Configuration Interface

During the testbeam the user can configure the connected threads by using a EUDET configuration file. A typical file like this is printed in listing A.3. The file is divided into sections; one section for each connected producer or any other thread. The configuration sent to the ATLAS pixel Producer is described in the listing from line 7 on. The settings called TriggerOnEudet, TriggerOnLemo, TriggerOnMonhit correspond to the switch positions in the EUDET experts window (compare figure 8.9b). A zero indicates 'off', everything else 'on'. This means that these switches can be manipulated via the EUDAQ interface by changing the configuration file. Furthermore, the content of the EUDET FIFO - which is again represented by the switches in the experts window - can be chosen by changing the setting of WriteOutEudetTrig, WriteOutPLLTrig and WriteOutTimestamp.

If the user wishes to configure the connected SCAs, Configure should be set to one. The configuration file for the SCAs, which needs to be saved on the TurboDAQ PC, can be chosen by writing it to ModuleConfigurationFile.

As it is possible to connect the MCC board to different channels on the TPCC and as this setting is not described by the configuration file of the SCA, the channel can be set separately with the parameter TPCCChannel.

For debugging purposes it is useful to dump the data locally, which can be configured as well by using SaveLocal and LocalFile.

Although the TurboDAQ configuration file contains the settings for the trigger delay and the latency it will be overwritten by the parameters Latency and TriggerDelay. The reason for this is that the configuration files are typically created in a setup without using a TLU. When using the TLU the trigger timings change. This offers a flexible solution which is easy to maintain.

In order to avoid accidental reconfigurations through the EUDAQ interface while a second user works at the TurboDAQ setup directly, the ability to do external configurations needs to be permitted globally. This can be done by setting the 'Mode' switch of the EUDAQ interface (figure 8.9a) from 'Manual' to 'Automatic'. By activating this switch, all windows of TurboDAQ, except for the interface itself, will grey out and cannot be changed anymore.

8.5.4. The Testbeam Scan

The origin of the so called testbeam scan is a typical source test as described in chapter 4.4. When TurboDAQ was used with the BAT Telescope, the original source scan was modified in a way that the data from the TPLL's FIFOs was not collected as part of the scan. Because the user has to specify the number of events that should be collected per FE before the test starts, a very large number was chosen. To stop the scan, a whole reset of the TPLL is applied. This procedure is only possible as the part of the scan which collects the data from the FIFO is separated.

The testbeam scan used in combination with the EUDET telescope is a copy of the original BAT telescope testbeam scan. Only few changes were necessary in order to make the scan work for the MCC board. These specialities will be described in chapter 11.3.

One part of the scan is its configuration. The settings in the testbeam scan require a trigger from the external trigger line which can be rerouted through the testbeam extensions to the TLU trigger interface. Furthermore, the TPLL will successively send 16 triggers to the FEs for each received trigger. With these settings it is possible to avoid timewalk effects (compare chapter 4.3).

8.5.5. Data Acquisition Thread

As already mentioned in chapter 8.5.4, a program needs to collect the data stored on the TPLL's FIFO while the scan is running. The program running this thread for the BAT telescope used the windows message queue system and ran in a standalone application. As the old program is incompatible[1] with the extended version of the TPLL, source code was developed to continuously read out this data from the FIFO. The code containing the DAQ thread was developed in Microsoft Visual C++, as it was done for EUDAQ. The binary was compiled as a second DLL to be available for TurboDAQ. The implementation of this DAQ thread is shown in listing A.5. One important method, the 'DataTakingLoop', is shown in listing A.4. The DAQ thread is framed by the callback functions to indicate the state of the thread in the TurboDAQ GUI. To guarantee that the thread is only running once, a mutex is locked with the start of the function. In lines 7 to 16, the variables used for the data taking are set to their initial values.

[1] The old program is incompatible because the BAT did not read out data from the EUDET FIFO containing the timing information

The typical case for a testbeam mode is that the DAQ thread starts before the scan does. In this way it is guaranteed that only the data belonging to the right run is acquired. So the first thing the thread does before starting with the acquisition is to drain all FIFOs, as only data from other scans can be part of the content.

The data collection method 'DataTakingLoop' is surrounded by two loops to allow a fast data acquisition as long as the FIFOs are not empty. It is important to slow down the acquisition speed[1] when it is not necessary to reduce the CPU[2]'s work.

When the 'DataTakingLoop' has downloaded a complete event from the TPLL, this event is passed to the EventHandler, which is a callback function which is availble in TurboDAQ. In TurboDAQ the event from the EventHandler is passed to the EUDAQ framework via the event sending function described in A.2. This long way around is necessary as DAQ thread and sending function are implemented in two independent DLLs.

The hard coding work was done in the DataTakingLoop[3]. This loop which is continuously called by the DAQ thread guarantees a synchronous readout of the TPLL's data FIFO and its EUDET FIFO. As described in the chapters 8.2.1 and 8.2.2, both FIFOs are filled simultaneously when running the testbeam scan. A synchronous readout is necessary in order to allow a correct merging of both data streams.

The necessity for the TPLL trigger ID in the EUDET FIFO is a feature of the code of the DataTakingLoop as the merging mechanism uses this number to ensure the synchronisation.

When the loop is called the first time the variable 'diff' will be set to zero, as it is made of equal initial values. Because of this, the function continues at the else case of line 14. One 32-bit word is loaded from the data FIFO by using the get_data_fifo() function. The byte stream in this FIFO can contain several kinds of words. In order to find a qualified beginning or end of an event, the uniqueness[4] of the TPLL trigger word is used. This word is the first word in a sequence belonging to one event. The FIFO is read out until the next TPLL trigger word is identified. As the read out trigger word cannot be loaded from the FIFO again it is saved in the data_back_buffer, which is a kind of second stage software FIFO.

Next, (compare line 28) the EUDET FIFO is read out. Just as the data FIFO, the EUDET FIFO uses the TPLL trigger word as identifier. The corresponding buffer is called eudet_back_buffer. As the sequence of words in the EUDET FIFO is known at this point, these words can be stored as TPLL trigger ID, TLU trigger, or timestamp directly.

As the testbeam scan and the DAQ thread are running completely independent from each other, it is possible that words belonging to one event are only read out partly as the remaining words have not been filled into the FIFO completely before. Therefore, a lot of checking mechanisms are included in order to complete the full event readout at the next loop.

In order to avoid a desynchronisation between the two FIFOs, which would mean that a wrong timing information including the trigger ID is stored to the raw data, the loop checks

[1] This is realised by using the sleep command in the while loop

[2] **C**entral **P**rocessing **U**nit

[3] The first version of this function was developed as part of this thesis. The version shown here was further improved by Ole Rhone

[4] This 32-bit word is always in the form 0x8n.nn.nn.n1, in which n can differ

for consistent TPLL trigger IDs in both FIFOs. Such a checking is done, for example, in line 12. In case a TPLL trigger ID word in the data FIFO is missing, the callback function 'callWarningFIFOSync' is called in order to make a note in TurboDAQ's GUI. In case there is a missing TPLL trigger ID word in the EUDET FIFO, another callback function 'callErrorFIFOSync' is called.

The naming of these functions is confusing as these callback functions have been used for other purposes before. A callback of these functions appears in TurboDAQ in that form that the numbers in the textfield FIFO Sync Warnings and FIFO Sync Errors increase. Additionally, a message is sent to the EUDAQ Log Collector.

8.5.6. Environment Thread

The environment thread is a gadget of the TurboDAQ's testbeam extensions that is not really necessary. With the modifications of the VHDL code of the TPLL the readout of the fill level of the EUDET FIFO became possible. The environment thread is a loop that continuously checks this fill level and illustrates it in the thermometer GUI element.

9. Online Software for Testbeam

This chapter describes the software used to do a quick analysis of the data taken in testbeam, while it is recorded. To allow an interpretation of the raw data stream of the ATLAS pixel FE chips, their raw data format needs to be described. Within the EUDAQ framework this can be enabled by using so called Converter Plugins. A description of the newly implemented plugins is given in chapter 9.2. In this chapter a description of the byte stream format is given as well.

When the interpretation of the byte stream is done, a tool is needed to graphically show some of the information obtained. This will be done in the so called Online Monitor which will be outlined in chapter 9.4.

9.1. Byte Stream Format

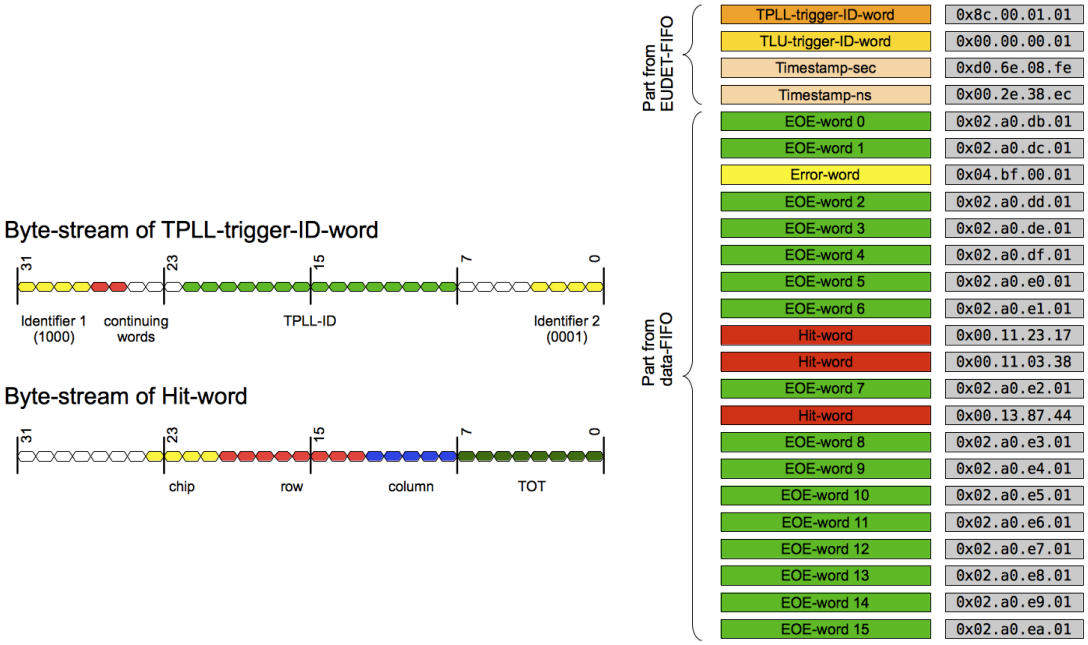
When the testbeam scan is configured in the FEs the chips back data. This data contains information about which pixels were hit with their corresponding TOT. The TPLL will collect the data and store it into the data FIFO. As the format of the data stream changes either by using different settings (e.g. dual link or single link) or by connecting a single FE or an MCC, the format in the data FIFO may change as well.

The only data format used during testbeam is the data format of an MCC connected to the TPLL configured for 40 MHz in single link mode.

The data stream received from the FIFOs for one event is sent in packages of 32-bit words. The splitting of the data stream into events has been realised in the DAQ thread (compare chapter 8.5.5) already. The event data stream will consist of two main parts: One being a copy of the words collected from the EUDET FIFO, the other being the copy of the data FIFO words. These parts are stored sequentially. As the TPLL trigger ID word from the data FIFO, which was stored in the data_back_buffer (compare listing A.4), is not send out as part of the data block, the ID word which is usually the first word of the part from the data FIFO will vanish. A graphical representation of the sequence of 32-bit words for one event is shown in figure 9.1b. Figure 9.1a shows the encoding format of the TPLL trigger ID word and the hit word.

The raw data part of the stream consists of different kinds of words. The beginning of the raw data part is indicated by the TPLL trigger ID word. Each event contains 16 EOE words, which indicate the end of the data stream for each of the 16 LVL1 triggers sent to the FEs. In between these EOE words, hit words can occur. By its position with respect to the EOE word the so called LVL1 value is defined. If an error occured either in the MCC or one of the FEs, an error word is sent right before the EOE word.

In figure 9.1a the encoding of the TPLL trigger ID word is illustrated: The word contains two identifiers: the most significant four bits are always 0b1000 and the least significant



(a) Byte stream format (b) Example of a byte stream sequence

Figure 9.1.: The byte stream format of the MCC as stored in the ATLAS Pixel TPLL. The least significant bit is always at the right position. The shown byte order is little endian. The order of the sequence illustrated in (b) is from up to down.

bits are always 0b0001. These identifiers are used to distinguish this kind of word from any other. The 15-bit TPLL trigger ID itself is stored in the bits 8 to 22, with the least significant at bit 8. As the origin of this word is the EUDET FIFO an extension to this format within the EUDET testbeam extensions was made. The two bits stored in the bits 26 and 27 contain the number of continuing words belonging to this event stored in the EUDET FIFO. So, if both bits are 1, this means that the content belonging to one event in the EUDET FIFO is made from the TPLL trigger ID word, the TLU trigger ID word and the 64-bit timestamp, which is stored in two words.

The encoding of the TLU trigger word and the timestamp is a simple binary number. The maximum number, which can be send from the TLU, is 15-bit long [17]; only the least significant bits in the FIFO word are used.

The identifiers for the EOE word and the error word are simple 1-bit markers: An enabled bit at position 24 indicates an EOE word; bit 25 indicates an error word. Further information can be obtained from the rest of the content of the error words.

The encoding of the hit words is shown in figure 9.1a as well. The least significant eight bits contain the TOT value. Further on, five bits are used to store the hit column; eight bits contain the position of the row. The last bits (21 to 24) indicate which chip was hit.

As shown in figure 9.1b two or more hit words can appear one after the other to show that two pixels were hit during the same LVL1 trigger. A sequence of EOE words, without intermediate hit words, indicate that no pixel was hit.

9.2. Byte Stream Conversion

In EUDAQ a so called converter plugin needs to be implemented for each kind of Producer. The function of this plugin is to convert the byte stream sequence of the arriving data into a so called standard plane. This plane is a very simple description of a sensor plane. The description uses variables like hit position in x and y and charge. Furthermore, a so called frame number can be stored in case a sensor is readout more than once in one event. As the TLU trigger ID can be stored in the raw data as well it can be stored here in addition.

By using the description of the byte stream format (chapter 9.1), the converter plugin was implemented. The column was converted into the x-position and the row into the y-position. Since the TOT value corresponds to a charge, it is stored in the corresponding variable. The TLU trigger ID was decoded from the second word of the EUDET part in the data stream and the frame number was used to distinguish between the hits which appeared at different LVL1 positions.

The standard frame requires the setting of a unique identifier to distinguish between separate sensors. In this case the so called sensor ID is derived by the connector position of the sensor at the MCC board starting with 10[1].

9.3. Root Monitor

With the help of the byte stream conversion (chapter 9.2), the data quality can be monitored in realtime. As shown in chapter 8.3, a thread class called Monitor is pre-implemented to fulfill functions that are common for an online monitor.

The program code of EUDAQ already contains an online monitor. This online monitor is called Root Monitor. It is based on the root library [13] including some graphical packages of that library in order to be independent from QT[2] [65]. Unfortunately, the Root Monitor has some disadvantages, so that a new online monitor (chapter 9.4) was developed.

9.4. Online Monitor

A redevelopment of the online monitor used for the data quality control during data acquisition was necessary. The basic functions of the program could be copied from the existing Root Monitor described in chapter 9.3. These functions are the connection to the EUDAQ framework via sockets and the inheritance of the monitor class. The mechanism to read in the data works in the same way as it worked for the Root Monitor: The events are not send via sockets, but by reading in the raw data file. The inheritance of the monitor class allows to implement the function 'OnEvent' which offers the possibility of using it as an event handler. The passed argument is an event class containing the

[1] It is necessary to use an offset to distinguish the DUT's ID from the ID of the telescope, which always starts with 0. 10 was chosen as this number is intuitive and independent from the number of connected telescope planes.

[2] **Q**uasar **T**echnologies

standard planes. Additionally, actor functions that are called by signals and commands from the Run Control are embedded.

A new GUI was implemented to illustrate histograms showing dedicated distributions of the data stream. As the amount of histograms is much too big to show everything in one window, the window is split into two parts. The left part contains a tree structure with elements which the user can select. Depending on the selection a histogram will appear in the right part. When a folder with subelements is selected, the histograms to each subelement are shown in parallel. This is the situation shown in figure 9.2. All histograms are being filled successively with the information from the incoming data stream. The active histograms are updated periodically.

Beside the two window parts in the GUI an area is reserved for a toolbar. The icons in the toolbar can be used either to save all histograms in their current state into a root file, to reset all histograms, or to quit the program. One more icon appears that stays in on- or off-state, which en- or disables an automatic reset of all histograms with the beginning of a new run. As the amount of data, depending on the beam intensity, can be so high that a realtime processing is not possible anymore, a prescaler can be selected in the toolbar in order to analyse only every n^{th} event. Setting this value to one will analyse everything.

This new online monitor does not need to be configured. The program obtains all necessary information from the data stream and will adapt its views based on the information sent.

With each event a simple data processing is achieved. Beside the filling of histograms this data processing uses the most processing time. In each event the pixels hit are clustered on each sensor . As the same clustering algorithm is used in the offline software, a description of this algorithm can be found in chapter 10.2.4. The clustering is important as a traversing particle can induce charge into more than one pixel. To obtain a charge signal as expected, the charges of all corresponding pixels need to be summed up. A further step in the data processing is the calculation of the hit occupancy.

The tree structure in the left part of the GUI window typically has - depending on the incoming data stream - three folders in the root level. The first folder is called MIMOSA-26, which is the name of the converter plugin. It contains several subfolders; one for each sensor of that kind. The subfolders are named after the IDs of the sensors, which are typically 0 to 5. These folders contain all kinds of information about the corresponding sensors. The second folder in the root level is called APIX and also contains subfolders for each sensor. As implemented in the converter plugin the IDs for the ATLAS pixel sensors start with 10. As the highest connector number on the MCC board is 7, IDs between 10 and 17 can be obtained. The ID folders contain information about the ATLAS pixel sensors. In comparison to the MIMOSA sensors more histograms are available. These were omitted for the MIMOSA sensors to reduce the processing time. The last folder in the root level is called Correlation. The meaning of a correlation is explained in chapter 9.5. All kinds of correlations between all sensors (also between the DUTs and the telescope) in x and y can be achieved here.

A particular version of the online monitor was developed for special purposes and shows a forth folder. This folder, called Conversion, offers histograms with real charges, instead of charges in TOT units. In order to obtain these information, charge conversion files need to be communicated to the online monitor to convert TOT into charge before doing the clustering. As this charge conversion is not really necessary for data quality control,

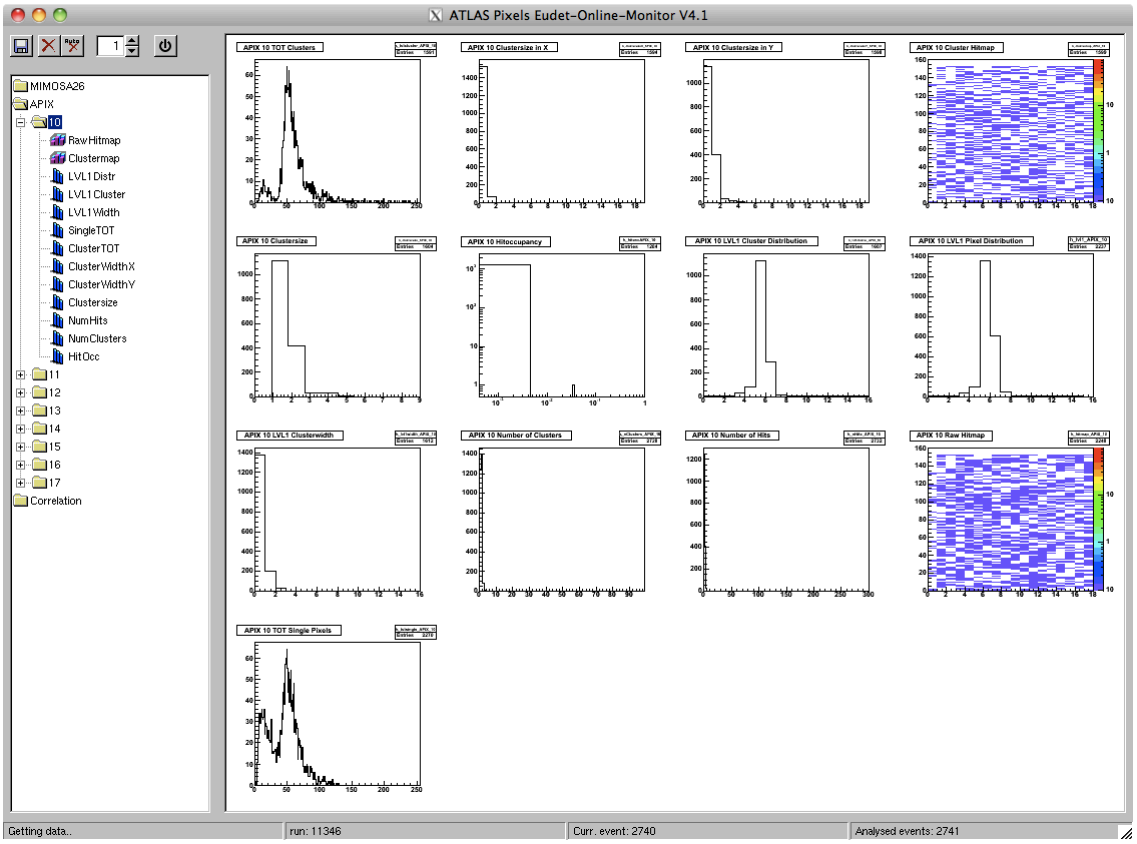


Figure 9.2.: Typical view of the ATLAS Pixel's EUDET Online Monitor Window

this particular version was never used during data acquisition, but for doing a preliminary analysis.

The histograms available for the ATLAS pixel FE-I3 sensors show the distributions of the following values:

Raw Hitmap This is a 2-dimensional histogram with the size of the sensor. One entry per pixel is filled at the dedicated position.

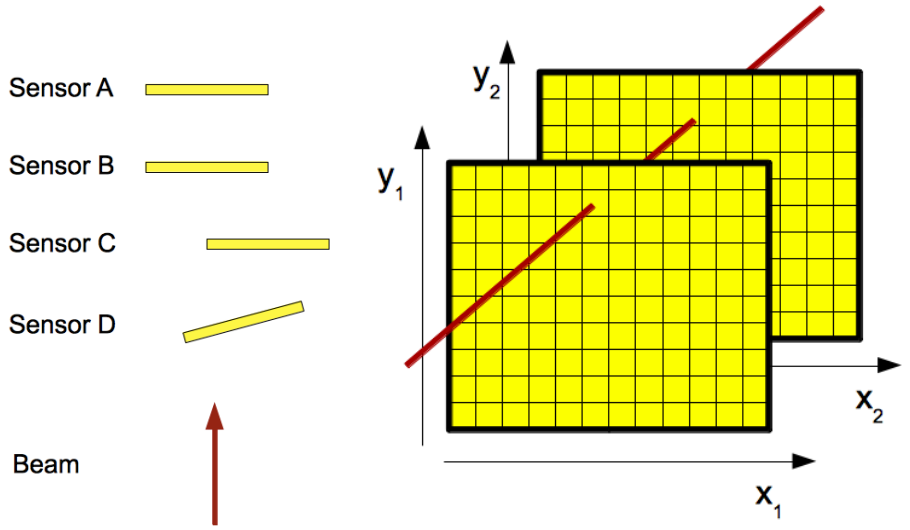
Cluster Hitmap This is a 2-dimensional histogram with the same size as the Raw Hitmap Histogram. Instead of filling one entry per pixel, one entry per cluster is filled. The filled bin is the integer of the mean value of the cluster, calculated from its dimension.

Single Pixel TOT This is the histogram showing the distribution of the TOT of all pixels of the corresponding sensor.

Cluster TOT This histogram shows the distribution of the sum of all TOTs of one cluster.

Cluster Size These are three 1-dimensional histograms. One shows the distribution of numbers of pixels per cluster, the other two show the distribution of the cluster size as a projection to the x- and y-axis.

Number of Hits The distribution of the number of hits per plane and TLU trigger in a 1-dimensional histogram.



(a) 2d illustration (b) 3d illustration

Figure 9.3.: Drawing to illustrate the meaning of correlations: (a) Top view of four sensor planes with different positions with respect to each other, (b) Two sensor planes with pixel coordinates $(x_1, y_1), (x_2, y_2)$ and a traversing particle trajectory

Number of Clusters Showing the corresponding histogram with the number of clusters instead of hits.

LVL1 Distributions The LVL1 pixel histogram shows the appearance of pixel hits with respect to their LVL1 timing. The LVL1 cluster histogram shows the distribution of the earliest LVL1 appearance of each cluster. The LVL1 clusterwidth histogram shows the signal distribution within a cluster by filling the difference of the earliest and latest LVL1 appearance.

Hit Occupancy This is a histogram which can only be calculated by using a subset of events. It shows the distribution of the probability for a hit for each pixel. This is calculated from the raw hitmap and the total event number.

A further development on the online monitor is in planning. Even with the new algorithms, the online monitor is too slow to analyse the data stream in realtime if the data rate is high. A new concept is in preparation to do a realtime clustering and storing of the data into a root file. When a histogram is selected it will be calculated from the content of the root file. This concept, in contrast to the successive filling of all histograms, has the advantage that the calculation of the histograms requires less processing time. The final goal is to allow a complete realtime monitoring without the necessity of the event prescaler.

9.5. Correlations

A correlation between two sensors describes the positions of the two sensors with respect to each other. This chapter will give an overview of how a correlation is calculated, which

influencing parameters exist and how to interpret the resulting histograms.

Correlations are the common tool during real time data analysis to test if the sensors are overlapping each other. Figure 9.3b shows a typical situation of two overlapping sensor planes. A traversing particle can induce hits in the pixels of the sensor planes. These pixels are arranged in the coordinate system of the sensor. (x_1, y_1) describes the coordinate system of the first sensor; (x_2, y_2) of the second sensor. To make a correlation means to create two 2-dimensional histograms, one for the horizontal and one for the vertical axis. The number of bins corresponds to the number of pixels. x_1 and x_2 are the axes of the first histogram, y_1 and y_2 the axes of the second.

In the ideal theoretical case exactly one pixel is hit by each trigger in each sensor. In this case both histograms would be filled at one position. With n hits in the first plane and m hits in the second plane, the number of possible combinations is $n \cdot m$. Each of them will be filled into the histograms. During the online data quality control the number of combinations is reduced by using a pattern recognition. This is realised by doing clustering, as a traversing particle induces a cluster hit and not a pixel hit.

Assuming there are two sensors of similar properties, a correlation between these (sensors A and B), as shown in figure 9.3a, would create a perfect correlation. Because the sensors are completely overlapping, a diagonal line will appear in both histograms. The situation changes when looking at the correlation of the sensors A and C in the same figure: The planes are not overlapping completely and the diagonal line will be shifted. Sensors A and D are not overlapping and additionally D is tilted. The correlation will be shifted because of the sensor shift and the slope will be changed as the pixel pitch of sensor D in the virtual projection on sensor A is smaller than the pitch of sensor A.

In chapter 9.6 different scenarios are discussed and simulated plots are shown.

Further parameters like beam and sensor properties also influence the correlations. These will be shown in chapter 9.6 as well.

9.6. Examples of simulated correlations

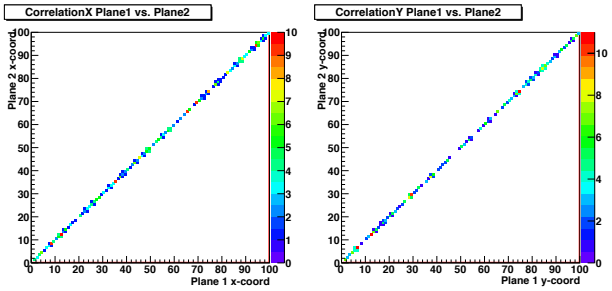
In order to give an overview about impacts on correlations between sensor planes, a C++ program was developed. The program, called CorrSim, can simulate different sensor geometries and beam properties.

The two sensor planes can be created with an individual number of pixels in both directions. The pixel's pitch can be varied as well. The user is able to define a beam geometry. All particle trajectories can be forced to be exactly parallel, or to be focussed to one point in the x- and the y-plane. An artificial jitter can be added to the track angulars to simulate uncertainties of the machine. A setting of the hit occupancy, which is naturally poisson distributed, can be forced to a dedicated number. Furthermore, the beam profile can be either set to a homogenous or a gaussian profile.

9.6.1. Influence of the sensor geometry

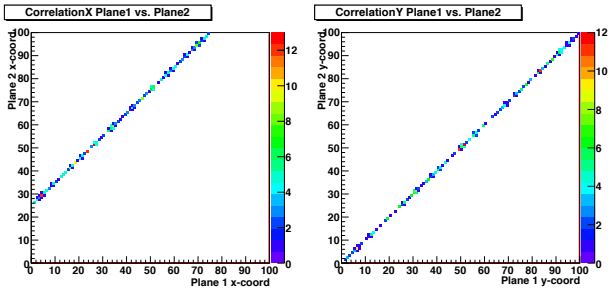
With the simulation program CorrSim, different scenarios have been simulated. In figure 9.4 an overview of different correlation plots created by changing the input parameters for

(a)



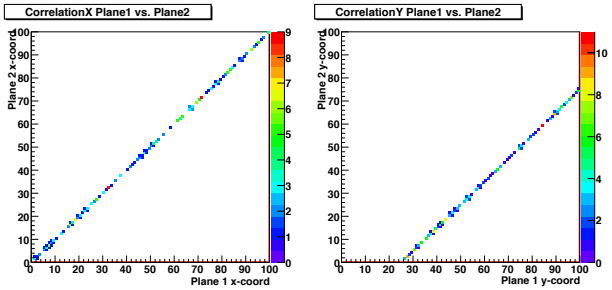
n pixels (1):	100×100
pitch (1):	$10 \times 10\,\mu\text{m}^2$
n pixels (2):	100×100
pitch (2):	$10 \times 10\,\mu\text{m}^2$
rel. position:	$0\,\mu\text{m}; 0\,\mu\text{m}$

(b)



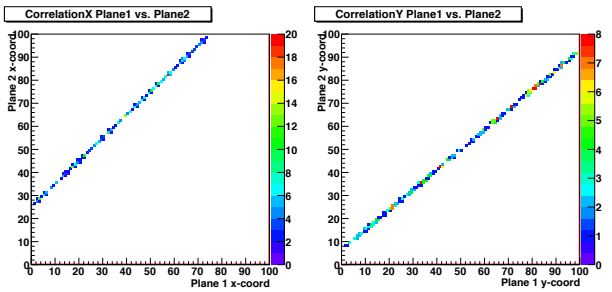
n pixels (1):	100×100
pitch (1):	$10 \times 10\,\mu\text{m}^2$
n pixels (2):	100×100
pitch (2):	$10 \times 10\,\mu\text{m}^2$
rel. position:	$250\,\mu\text{m}; 0\,\mu\text{m}$

(c)



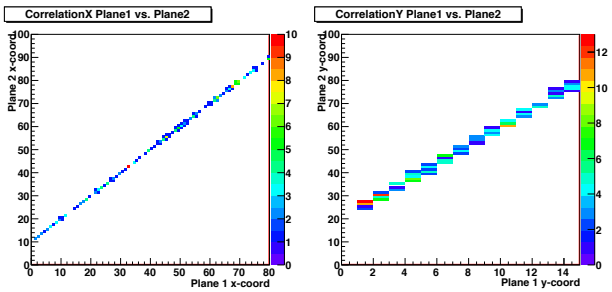
n pixels (1):	100×100
pitch (1):	$10 \times 10\,\mu\text{m}^2$
n pixels (2):	100×100
pitch (2):	$10 \times 10\,\mu\text{m}^2$
rel. position:	$0\,\mu\text{m}; -250\,\mu\text{m}$

(d)



n pixels (1):	100×100
pitch (1):	$10 \times 10\,\mu\text{m}^2$
n pixels (2):	100×100
pitch (2):	$10 \times 10\,\mu\text{m}^2$
rel. position:	$0\,\mu\text{m}; -250\,\mu\text{m}$
tilt (1) x:	$30\,^\circ$

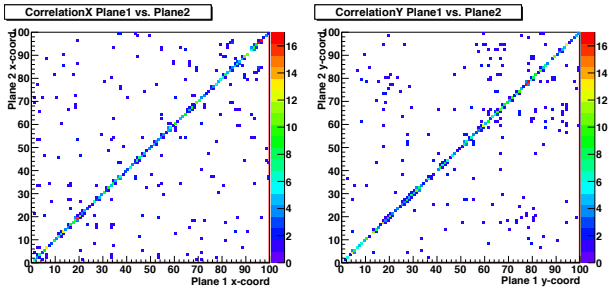
(e)



n pixels (1):	80×15
pitch (1):	$10 \times 40\,\mu\text{m}^2$
n pixels (2):	100×100
pitch (2):	$10 \times 10\,\mu\text{m}^2$
rel. position:	$0\,\mu\text{m}; 0\,\mu\text{m}$

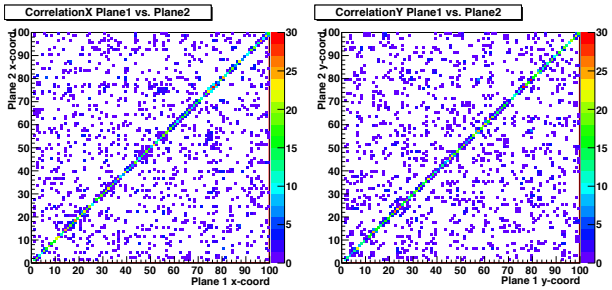
Figure 9.4.: Illustration of different relative sensor positions on the x- and y-correlations. Correlations are simulated with CorrSim. In the corresponding simulation the hit-occupancy is exactly 1 and all tracks are exactly parallel

(a)



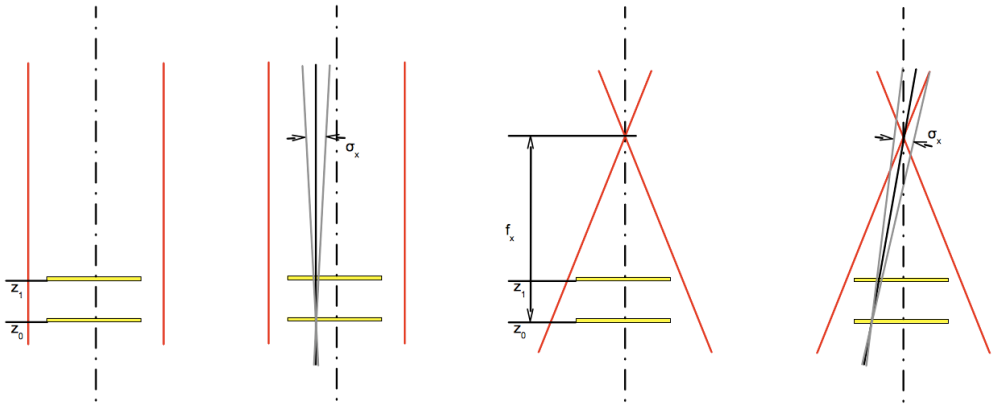
n pixels (1):	100×100
pitch (1):	$10 \times 10\,\mu\text{m}^2$
n pixels (2):	100×100
pitch (2):	$10 \times 10\,\mu\text{m}^2$
rel. position:	$0\,\mu\text{m}; 0\,\mu\text{m}$
$\langle$hit occ.$\rangle$:	2.0

(b)



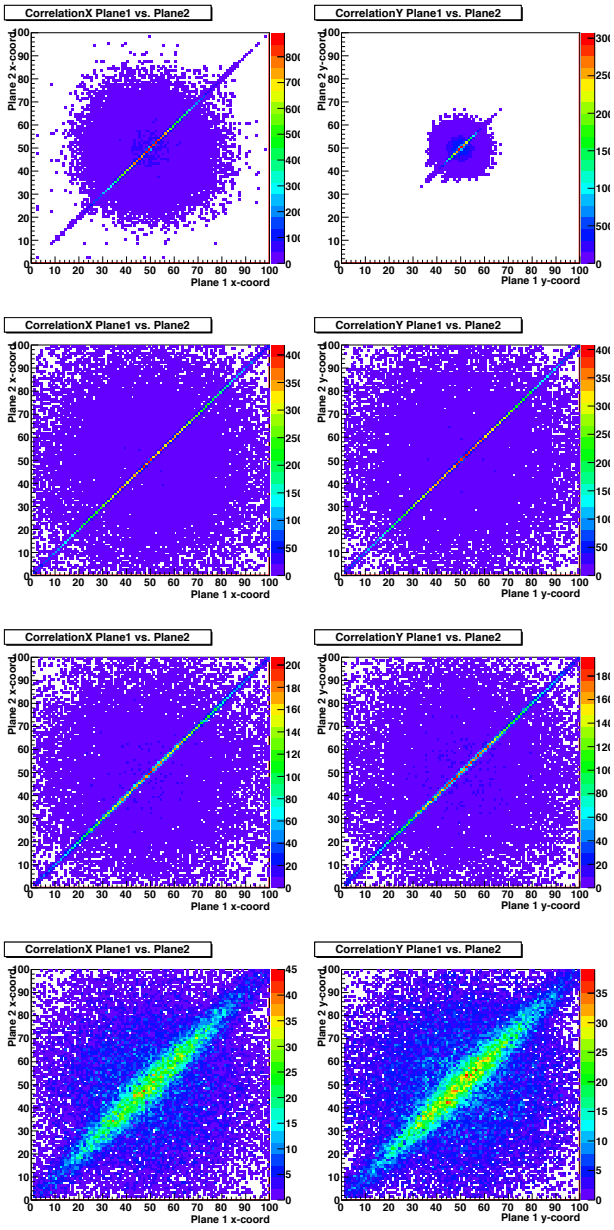
n pixels (1):	100×100
pitch (1):	$10 \times 10\,\mu\text{m}^2$
n pixels (2):	100×100
pitch (2):	$10 \times 10\,\mu\text{m}^2$
rel. position:	$0\,\mu\text{m}; 0\,\mu\text{m}$
$\langle$hit occ.$\rangle$:	5.0

Figure 9.5.: Illustration of different beam intensities on the x- and y-correlations. The investigated value is the mean value of the poisson distribution, describing the hit occupancy correlations are simulated with CorrSim. In the corresponding simulation the sensors are completely overlapping and all tracks are exactly parallel



(a) exactly parallel (b) parallel with jitter (c) exactly focussed (d) focussed with jitter

Figure 9.6.: Illustrations of different beam properties



(a)

$\langle$hit occ.$\rangle$:	2.0
beam shape:	$120\,\mu$m; $40\,\mu$m

(b)

$\langle$hit occ.$\rangle$:	2.0
beam shape:	$250\,\mu$m; $250\,\mu$m

(c)

$\langle$hit occ.$\rangle$:	2.0
beam shape:	$250\,\mu$m; $250\,\mu$m
jitter:	$1 \cdot 10^{-4}$; $1 \cdot 10^{-4}$

(d)

$\langle$hit occ.$\rangle$:	2.0
beam shape:	$250\,\mu$m; $250\,\mu$m
jitter:	$1 \cdot 10^{-3}$; $1 \cdot 10^{-3}$

Figure 9.7.: Simulated correlations. The histograms show different beam settings. In all cases the tracks are parallel, meaning not focussed. (a) and (b) show different beam shapes with a gaussian profile. (c) and (d) show different uncertainties of the track parallism, by applying a jitter. The number of simulated events is 10 000. The distance of the two sensors is 5 cm

the simulation are shown. Except for the last case, two artificial sensors with 100×100 pixels and a pitch of $10 \times 10\,\mu\text{m}^2$ are used in all cases. In case (a) the sensors are completely overlapping, which is indicated by the relative position; the resulting correlations are perfect. Case (b) shows the first sensor shifted $250\,\mu\text{m}$ in x-direction. The shifted x-correlation would indicate that in the data quality analysis. Case (c) shows the same effect on the correlations with a shift of the first sensor about $-250\,\mu\text{m}$ in the y-direction. In the fourth case the sensor is first shifted, as in case (b), and then turned $30\,^\circ$ around the x-axis. In this situation the slope of the y-correlation changes as the tilted sensor covers a smaller area of the perpendicular sensor in the y-direction. The last situation shown in figure 9.4 shows exactly aligned sensor planes, which means that they share the same center point in the xy plane. In contrast to the other cases shown, the pixel geometry of the two sensors differ in this case. The geometry of sensor 1 was changed to 80×15 pixels with a pitch of $10 \times 40\,\mu\text{m}^2$. As the first sensor covers a smaller area in both directions, the slope of the correlations is less high. A situation like this can be observed very often in the online monitor when looking at correlations between the telescope and a DUT. In order to guarantee a complete overlapping of the smaller DUT into the frame of the telescope, one looks at these correlations and moves the DUT when necessary.

9.6.2. Influence of the beam intensity

The correlations shown in figure 9.5 show different beam settings on the same setup. In comparison to case (a) in figure 9.4, in which the number of hits per event were forced to be 1, this number, also called hit occupancy, is now poisson distributed. The mean value of the poisson distribution in case (a) is 2.0. In case (b) the value was changed to 5.0.

With the increased number of hits per plane and event the number of fill operations in the histogram increases with the power of two. Because of this a huge number of entries can be found beside the correlation line. This so called combinatorial background is typical for data quality plots as well.

9.6.3. Influence of the beam shape

In CorrSim the beam optics can be varied. The figures shown so far were simulated under beam optic conditions as shown in figure 9.6a. The red lines indicate the beam envelope in one dimension. What is not shown is that the beam can be either gaussian distributed or homogenous. The two simulated sensors are shown in yellow.

In the simulated scenario (a) in figure 9.7 the selected beam optic is a forced parallel beam as drawn in figure 9.6a. The selected beam shape is gaussian with $\sigma_x = 120\,\mu\text{m}$ and $\sigma_y = 40\,\mu\text{m}$. Therefore, the beam has a flat shape which is wider in x and smaller in y. As the hit probability at the sensor edges is smaller due to the beam shape, the correlation histograms have more entries in the centre as well. The circles which begin to show in the histograms are a combination of the hit occupancy (which is greater than 1) and the gaussian shape. Case (b) in figure 9.7 shows a symmetrical beam shape in x and y. The shape is large enough to cover the sensors almost completely.

As illustrated in 9.6b, the typical situation in a real testbeam includes an uncertainty in the beam optic. This jitter is a fluctuation in the slope of the beam. In CorrSim a

jitter can be set to fluctuate the slope of the track in x and y. The parameter is the σ of a gaussian distribution, which describes the amount of the fluctuation. When applying this jitter (compare (c) and (d) in figure 9.7) the correlation becomes wider. In case (d) one can see that the resulting shape becomes ellipsoidal. The reason for this is that the number of possible combinations increases in the middle because of the geometrical arrangement of the sensors.

9.6.4. Influence of the beam focus

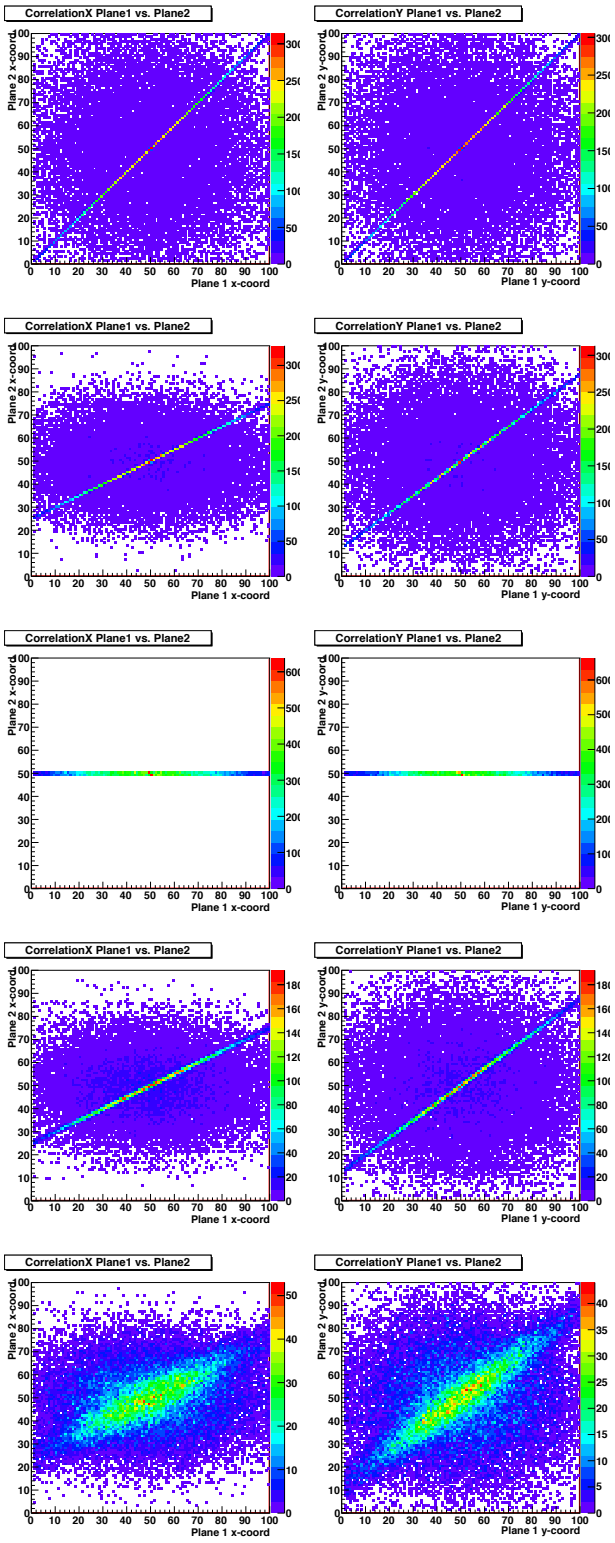
Situations with a focussed beam as illustrated in figure 9.6c and 9.6d were simulated and shown in figure 9.8. In (a) the focal point is so far away from the sensor that the beam appears to be almost parallel. Therefore, the correlation looks similar to figure 9.7b. When the beam is focused closely to the sensor the relative position of the sensors to each other plays an important role. In the simulated situations the distance was always set to 50 mm. In (b) the beam was simulated at focal points of $f_x = 100\,\text{mm}$ and $f_y = 200\,\text{mm}$. With different focal points in x and y the beam shape will become flat in vertical or horizontal direction at the dedicated positions. In the resulting correlations one can see that the shape changes. The projection of the first sensor on the second along the tracks changes its size because of the focussed beam. Case (c) is an usual and only theoretical case: When the focal point is directly in the second sensor, only one position on the sensor will be hit. In the resulting histograms one can see a correlation between each hit point of the first sensor and exactly one point in the second.

In reality the beam optic includes some uncertainties: Cases (d) and (e) show the correlations with an applied jitter of $\sigma_{x,y} = 1 \cdot 10^{-4}$ and $1 \cdot 10^{-3}$. The resulting effect looks similar to the one for the parallel tracks (compare figure 9.7c and d). In comparison to that the correlation ellipsoid appears wider. The reason for this is seen in the changed incident angle of the track in the sensor. With a less perpendicular incident, the possibility to hit more than one pixel increases.

9.6.5. Influence of Noise

Noisy or stuck pixels have influence on the correlation histograms. This is shown in figure 9.9. In CorrSim it is possible to flag a pixel noisy. The changed values in the simulations shown in that figure are the number of noisy pixels and their noise occupancy. The noise occupancy describes the probability that a pixels shows a noise hit. The position of the noisy pixels is randomly chosen. When applying noisy pixels only to one sensor, the resulting correlations will show horizontal or vertical lines beside the correlated entries. This is illustrated in the simulated scenarios (a) and (b). Case (c) is a combination of both; it shows horizontal and vertical lines.

When the amount of noisy pixels is higher (25 noisy pixels with 20 % noise occupancy in sensor 1, and 50 noisy pixel with 15 % noise occupancy in sensor 2) as shown in (d) the noise entries dominate. In order to see the correlated data, apart from the noisy data, the z-axis needs to be logarithmised. This is realised in (e) for which the same data set was used as for (d).



(a)

⟨hit occ.⟩:	2.0
beam shape:	250 μm; 250 μm
beam focus:	10 m; 10 m

(b)

⟨hit occ.⟩:	2.0
beam shape:	250 μm; 250 μm
beam focus:	100 mm; 200 mm

(c)

⟨hit occ.⟩:	2.0
beam shape:	250 μm; 250 μm
beam focus:	50 mm; 50 mm

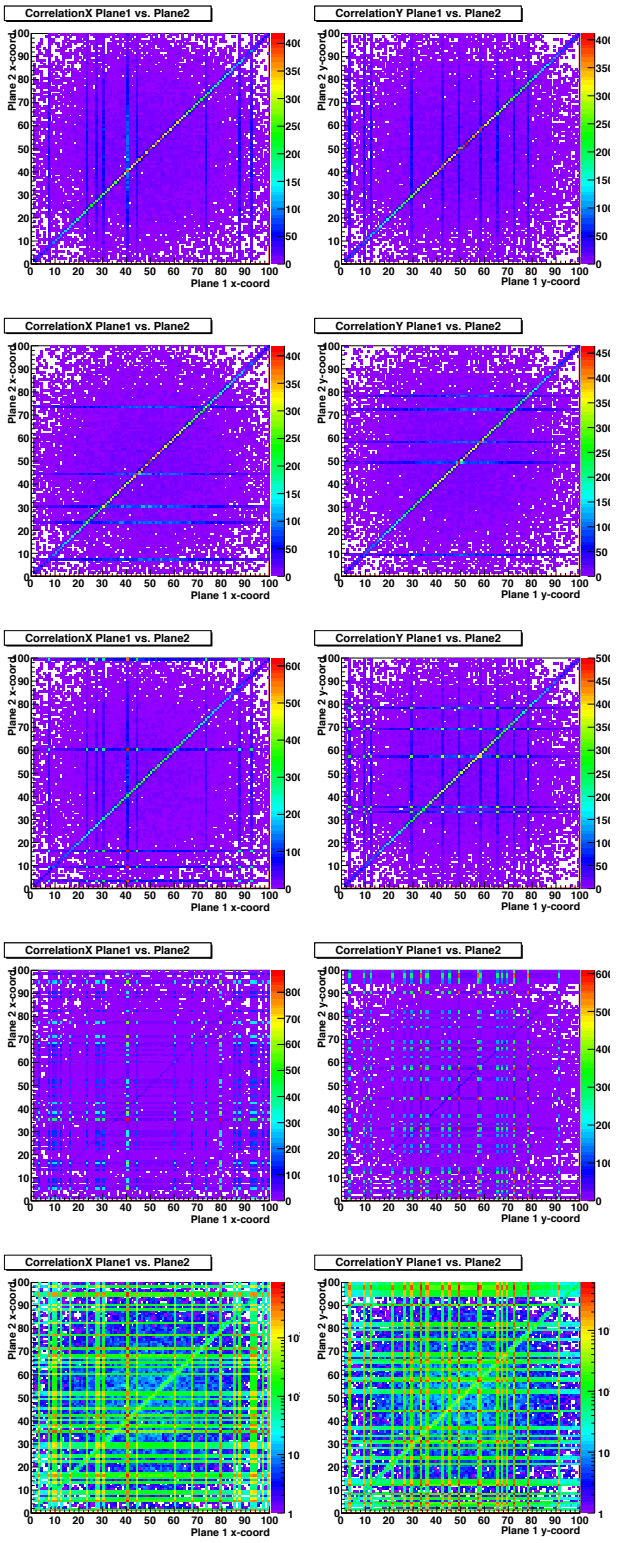
(d)

⟨hit occ.⟩:	2.0
beam shape:	250 μm; 250 μm
beam focus:	100 mm; 200 mm
jitter:	$1 \cdot 10^{-4}, 1 \cdot 10^{-4}$

(e)

⟨hit occ.⟩:	2.0
beam shape:	250 μm; 250 μm
beam focus:	100 mm; 200 mm
jitter:	$1 \cdot 10^{-3}, 1 \cdot 10^{-3}$

Figure 9.8.: Simulated correlations. The illustration shows the influence of different beam-focus points and jitter on the correlations. In the cases (a) to (c) all tracks are exactly focussed without jitter. (d) and (e) use the focus-setting of case (b) by applying jitter. In case (b) the focus-point of the beam in both directions in exactly in the second sensor. The distance between the simulated sensors in 50 mm. The number of event, used for the simulation is 10 000



(a)

⟨hit occ.⟩:	2.0
beam shape:	250 μm; 250 μm
noise (1):	10 pixels; 5 %
noise (2):	none

(b)

⟨hit occ.⟩:	2.0
beam shape:	250 μm; 250 μm
noise (1):	none
noise (2):	5 pixels; 10 %

(c)

⟨hit occ.⟩:	2.0
beam shape:	250 μm; 250 μm
noise (1):	10 pixels; 5 %
noise (2):	5 pixels; 10 %

(d)

⟨hit occ.⟩:	2.0
beam shape:	250 μm; 250 μm
noise (1):	25 pixels; 20 %
noise (2):	50 pixels; 15 %

(e)

⟨hit occ.⟩:	2.0
beam shape:	250 μm; 250 μm
noise (1):	25 pixels; 20 %
noise (2):	50 pixels; 15 %

Figure 9.9.: Simulated correlations. The illustration shows the influence of noisy pixels on the correlations. Varied is the number of noisy pixels and their occupancy. (a) and (b) shows noisy pixels either in the first or the second sensor; (c) in both. In (d) the number of noisy pixels is increased and also their occupancy. (e) shows the same histograms as (d), but with a logarithmic z-axis. The number of event, used for the simulation is 10 000.

10. Offline Software for Testbeam

In comparison to the online software which is used to check the quality of the data while it is being taken, the offline software is used to allow detailed analyses of the data at a later point in time. Before doing the analyses the datasets need to be reconstructed. Reconstruction means that the potential trajectories of the particles are calculated from the sensor hit positions.

The software used for reconstruction is called Eutelescope [34]. Eutelescope is written in C++ within the ILC framework. A description of the ILC framework can be found in 10.1. Eutelescope will be presented in chapter 10.2. After the reconstruction an analysis of the data can be achieved. A tool called TbTupleAna (compare chapter 10.3) was developed to allow such analyses.

10.1. ILC Framework

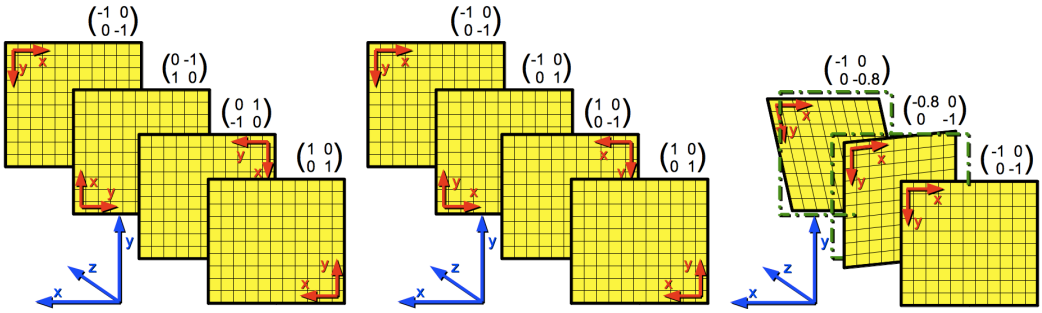
The ILC framework is the general software framework for ILC experiments. It offers a lot of possibilities to describe detectors, events and all kinds of objects related to high energy physics. These objects can be stored into a so called LCIO[1] file. This format provides an event data model and a persistency solution for ILC-soft [40]. It is used in the Eutelescope software.

The framework is built modularly. This means that an application written for the ILC framework is made of modules. These modules are called MARLIN[2] processors. To become such a processor a class in an application needs to inherit from a processor base class. Some abstract functions have to be implemented so that the MARLIN interface knows how to handle in different situations.

10.1.1. MARLIN processors

The processors can be activated and parametrised with an XML[3] steering file. An example of a file like this is shown in listing A.6. One or more steering files are used as parameters for the MARLIN application to run the code implemented in the modules. The steering file is divided into three main parts. The first part surrounded by the XML command 'execute' describes the flow of modules to be executed. Each processor which is going to run will appear in this section. The second part named 'global' is used to declare the input files of the LCIO format and also the GEAR[4] file which will be mentioned later in this

[1] A persistent framework and data model for **L**inear **C**ollider detector studies
[2] **M**odular **A**nalysis & **R**econstruction for the **LIN**ear Collider
[3] E**X**tensible **M**arkup **L**anguage
[4] **GE**ometry **A**pi for **R**econstruction



(a) 90° along z (b) 180° along x and y (c) small tilts

Figure 10.1.: Examples of different rotation matrices

chapter. Furthermore, the maximum number of events to be processed can be specified there. The last section, made of multiple parts using the XML command 'processor', is used to pass arguments to the individual processors.

10.1.2. GEAR

In order to allow a consistent description of the geometry of the testbeam setup the ILC library GEAR is used. GEAR uses an XML file to describe the properties and absolute positions of sensors in the testbeam. Whereas GEAR is usually used in the ILC framework to describe large 4π detectors for ILC applications, specialised classes of GEAR are provided for testbeam applications to describe sets of sensors arranged one after the other. An example GEAR file is printed in listing A.7. In this example the number of planes is reduced to provide a well-arranged view. Three telescope sensors and two ATLAS pixel FE-I3 sensors are included in this geometrical description. Each sensor is described with two items called 'ladder' and 'sensitive'. The item 'ladder' is used to keep the file consistent with the requirements of the GEAR library. These entries are usually used to describe the material amount in order to do a GEANT-4[1] simulation, which has not been implemented yet for testbeam applications. The declarations in the item 'sensitive' are used to identify the position of the sensitive matter. The number of pixels and their pitch is used to allow a transformation into a 3-dimensional space point. In this case a hit in a pixel on a special sensor is transformed into such a coordinate. The attributes 'rotation1' to 'rotation4' describe the relative orientation of one sensor in the global coordinate system of the telescope (compare chapter 6.4). These items form a rotation matrix m_{ij} of the xy plane

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \begin{pmatrix} x_\mathrm{p} \\ y_\mathrm{p} \end{pmatrix} \tag{10.1}$$

in which the coordinates x_p and y_p describe the local sensor coordinate system with their origin in the central position of the sensor. 'rotation1' is used for m_{11}, 'rotation2' for m_{12} and so on. Figure 10.1 shows different rotations of the sensors and their corresponding rotation matrix. In figure 10.1a rotations around the z-axis (the beam axis) are illustrated. The blue arrows represent the global coordinate system, the sensors' local coordinate

[1] **GE**ometry **AN**d **T**racking - Version 4.0

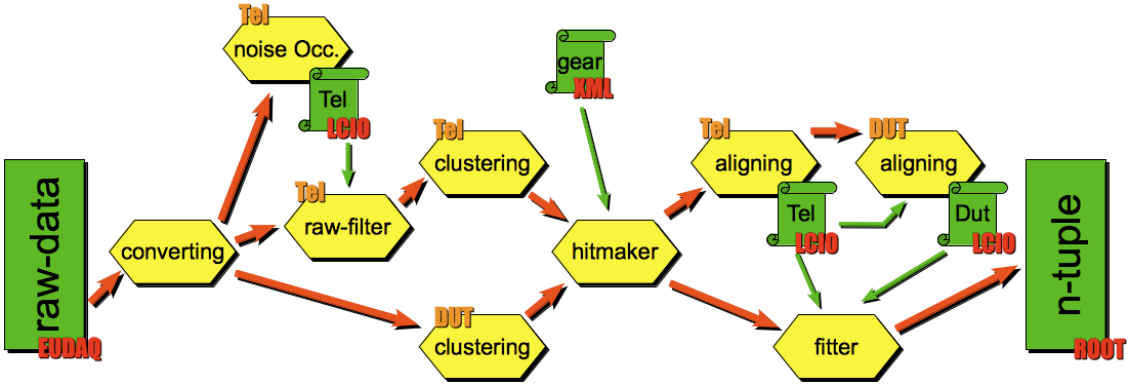


Figure 10.2.: Program flow of Eutelescope

systems are illustrated by the red arrows. The red arrows are always drawn in that corner of the sensor, in which the origin[1] of the pixels is. In figure 10.1b rotations around the x- and y-axis are done. As the matrices describe a projection of a 3-dimensional system onto a 2-dimensional plane, a 180° rotation around one of these axes mathematically equals a mirroring. Therefore, the determinant of these matrices is -1 (in figure (a) it is 1). The only exception is the fourth case, which describes two mirrorings, as it is a 180° rotation around both axes. When studying non-perpendicular incident angles, matrices as shown in figure 10.1c can be used. As the z-coordinate is not transformed with these matrices, the tilting of the sensors is a mathematical shrinking of the projected xy plane.

In the consistent description of GEAR the position $(0;0)$ for each sensor is the center position of the active region. Although not necessary for GEAR the coordinate system was always defined with the very first sensor placed to $z = 0\,\text{mm}$. By doing so the z-positions of all sensors are set as relative values to the first upstream sensor.

10.2. Eutelescope

Eutelescope is the data reconstruction software for testbeams with the EUDET telescope. The software is maintained by the EUDET community and organised in MARLIN processors (compare chapter 10.1). Figure 10.2 illustrates some of the processors which are needed to run the reconstruction chain in a simplified way. On the left hand side of the figure one can see the input file written out by EUDAQ and on the right hand side the output file which is in the root file format. The processors shown in the figure will be described in the following sections. Only few of them could be used in their original form, as modifications have been necessary in order to reconstruct data sets with ATLAS pixel sensors as DUTs.

10.2.1. Data Conversion

The first processor which needs to be called is the Data Converter. The implementation of this processor was not changed during this thesis.

[1] The pixel with row zero and column zero

Its function is the interpretation of the raw data and their conversion into standard planes. Because of this, it is quite similar to some functions of the online monitoring system (compare chapter 9.4). In fact, Eutelescope is linked against EUDAQ in order to provide the functionality of the byte stream conversion, which has already been implemented there for checking the data quality during the data acquisition.

The saving of the standard planes into the LCIO file format is not a part of the Data Converter. With every event the data will be dumped into standard plane objects, which will be saved to files by the EUTelOutputProcessor.

10.2.2. Occupancy Measurement

Before starting the reconstruction of the hits in the standard planes to tracks a data cleaning needs to be done. The cleaning is only applied to the data of the telescope in order to not reduce the efficiency of the DUTs by accident. To determine which parts of the data are valid and can be used for the reconstruction and which are invalid and need to be removed, the hit occupancy for each pixel in each sensor plane of the telescope is estimated. The hit occupancy is the measured probability for a pixel hit per event. The hit occupancy of each pixel in a sensor is stored in a database file. In a second processor (see next section) pixels are masked by applying a threshold to the distribution of the occupancy.

Instead of using the hit occupancy, it is also possible to use the noise occupancy, which can be estimated by using a self trigger during data taking without beam. The shape of both distributions is similar, although the shape of the hit occupancy is shifted to higher hit probabilities, of course.

10.2.3. Raw Data Filtering

By reading in the database file, created by the occupancy processor (compare section 10.2.2), the hit probability of each pixel in each sensor is known. The filter processor uses the fact that an increased hit or noise occupancy value of a pixel is a possible indication for a noisy or stuck pixel. Therefore, one defines a threshold value in the parameters of this processor, which masks out all pixels higher than this threshold value.

The standard setting for the threshold probability is 0.001. This means that if a pixel was hit more than once in 1 000 events it is dismissed.

The processor creates a new LCIO collection in the same format as the input collection. The content is a copy of this collection except from the data belonging to the masked pixels.

The concept of the raw data filtering on basis of an occupancy database is a new concept, which was developed as a part of this thesis.

10.2.4. Clustering

When a particle traverses a sensor it is typically determined by more than one pixel. Therefore, the coordinates of the pixels hit cannot be converted into geometrical space

Figure 10.3.: Illustration of the clustering algorithm. The left picture shows two clusters made of the pixels 1 to 4 and 5 to 6. The right picture shows the flow of the algorithm, running column by column from upper left to lower right.

points, as each hit pixel would become an individual hit position. In order to reduce the number of the hit positions a pattern recognition is implemented to build clusters made of pixels.

In the original code a clustering algorithm has already been implemented. This algorithm is called Fixed Frame Clustering. It uses a frame of, for example, 5×5 pixels surrounding each hit pixel to check for hits in that area. In case of a positive match these were combined as cluster.

Unfortunately, this code was performing very poorly and is unusual for the ATLAS pixel sensors[1]. Therefore, a new code was implemented during this thesis, which speeds up the processing time by a factor of roughly 60. Figure 10.3 illustrates the way the algorithm works by using an example with two clusters.

At the beginning, the algorithm handles every pixel hit as an individual cluster with a unique number. These are the shown numbers (1 to 6). In the algorithm's standard setting, direct pixel neighbours and diagonal neighbours indicate the cluster criteria. Successively all pixels are compared with each other. When two pixels fulfill the cluster criterion, the cluster value of the higher number is changed to the lower one. When comparing pixel 1 and 2 (upper left in figure 10.3), they do not behave as neighbours, so they keep their numbers. When comparing all pixels with the first pixel, the values of the pixels 3 and 4 will be changed to 1. Iteratively, pixel 2 will be compared with pixels 3 to 6, then 3 with 4 to 6. If the values are already equal the cluster criterion does not need to be checked of cause (case marked in yellow). At the end only two clusters with the numbers 1 and 3 will be left over. These numbers will be normalised to the series of natural numbers.

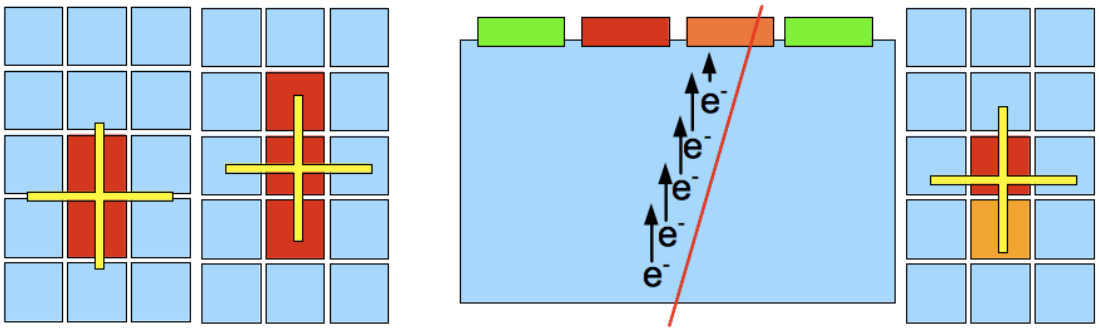
As one can see directly, the processing time of this algorithm per plane takes i steps, with

$$i = \frac{n(n-1)}{2}, \tag{10.2}$$

in which n is the total number of hit pixels in that sensor.

The clusters will be stored as cluster objects into an LCIO cluster collection. In order to keep a connection between the pixels belonging to one cluster and the cluster object

[1] Because of the rectangular pixels of the ATLAS pixel sensor a typical cluster is only one dimensional.



(a) dig. 2-cluster (b) dig. 3-cluster (c) sensor cross section (d) anal. 2-cluster

Figure 10.4.: Illustration of different cluster types and the calculation of the center of gravity. (a) and (b) show clusters made of two and three pixels and their digital center of gravity. In (c) a cross section through a sensor with a traversing particle track is shown. A charge weighted cluster is shown in (d). The center of gravity is shifted from the digital center.

itself, a new raw data collection will be created which is basically a copy of the original collection but with an additional cluster ID parameter.

A lot of parameters can be set to the new clustering algorithm. These are the settings for the cluster criterion by setting a maximum distance of two neighbouring pixels to fulfill the criterion. The horizontal, the vertical, and the diagonal case can be modified individually. Default values are 1 for all parameters, which indicates that only direct neighbours are allowed. Further parameters are the minimum number of pixels (default is 1), the minimum charge value[1] (default is 0) and the maximal allowed LVL1 trigger difference (default -1, meaning no cut) between two pixels in one cluster to be recognised as cluster.

10.2.5. Hitmaking

By doing clustering the triggered pixels belonging to one traversing particle have been combined to a cluster. After this the positions of the clusters in the 2-dimensional coordinate system of each sensor can be converted into space points in one global coordinate system. The coordinates of the xy plane can be estimated from the position of the cluster, whereas the z-position needs to be read-in from the geometric positioning data of the sensor. This data is passed to the so called hitmaker processor by using a GEAR file (compare section 10.1.2).

As tilts of the sensors around the x- or y-plane can only be described in GEAR by using a projection on the perpendicular plane, the resulting space points will share the same z-coordinate in any case as well[2].

The estimation of the x- and y-coordinates of the space points is done by calculating the center of gravity of the cluster. Figure 10.4 shows different cluster types and the

[1] Only the sum TOT of the cluster, not the charge in electrons, can be used here.
[2] This simplification was done as the description is sufficient enough as long as the tracks are almost parallel to the z-axis.

calculation of their center of gravity. In the figures in (a) and (b) the digital center of gravity is used. That means that independent from the charge of the clusters the center c_{xy} is the average value of all N pixel coordinates p_{xy}

$$c_{xy} = \frac{1}{N} \sum_{1}^{N} p_{xy}. \tag{10.3}$$

As shown in figure 10.4c, the amount of charge in the pixels hit depend on the incident position of a track with respect to the pixels. The figure is a simplification because the correct conversion into electron charge depends on plenty factors like trapping, irradiation and the field geometry. But in general the track is closer to the pixel with more charge. Therefore, the equation (10.3) can be modified by using a weighting factor. This factor is the charge. In case of ATLAS pixel FE-I3 the TOT can be used instead of the charge although the TOT is not directly proportional.

As will be explained in section 10.2.7 the calculation of the tracks is only done by the hit positions of the telescope. Therefore, the default setting for the hitmaker is digital clustering, as this conversion is sufficient enough.

For conversion of the telescope, digital clusters are used as well, as these kinds of sensors do not provide analogue information.

10.2.6. Alignment

The GEAR file provides information for the reconstruction algorithms to describe at which position the sensors are located in a global coordinate system. The fact is that the theoretical position of the sensors described in that file is only reached within practical limits.

For a better estimation of the positions of the sensors relative to each other, algorithms are used. The output of these algorithms are database files including correction parameters for the geometrical description of the setup. These corrections are shifts and rotations.

The original code for the alignment was realised by using the millipede program [59, 72]. Improvements were done [44] which include a kalman filter [47] for a better selection of hit candidates, which will be passed to the existing millipede program afterwards.

The kalman filter is a recursive formulation of the least square method. Within the algorithm the track estimation is improved by extrapolating the track to the next sensor plane and updating the track's predication with measurements [44]. The algorithm of the kalman filter can be used for track fitting as well.

Millipede is a program that can solve the linear least square program with a simultaneous fit of all local and global parameters irrespectively of the number of parameters. In this context the local parameters are only present in a subset of the data for a linear track model[1] and the global parameters are the alignment constants[2] [72]. Millipede is split into two parts:

[1] The number of free parameters per track is 4
[2] Shift and rotations of the sensors

Mille collects parameters of hit data as possible track candidates and saves them into a binary file. These are the biased residuals in x and y, their partial derivates with respect to local parameters $\partial x/\partial z, \partial y/\partial z$ and the derivates of the residuals with respect to global parameters such as translations and rotations.

Pede uses the input binary file of Mille to perform the alignment. It uses a matrix inversion method, so one can set a convergence limit and initial values. The default value for the limit is typically $1 \cdot 10^{-3}$, the calculation for the initial values is realised by mille.

As the integration time of the telescope is much higher than for the DUTs, the alignment is split, as the alignment of the telescope needs less data to work sufficiently. As one can see in figure 10.2 the telescope is aligned first and the DUTs afterwards.

10.2.7. Track Fitting

By using the alignment correction constants a more reliable description of the geometrical setup is possible. After the sensor positions are corrected, an algorithm is used to perform the track fitting. Originally, an algorithm which looked for the χ^2 of the tracks was used. A similar algorithm that was used for the determination of track candidates by using a kalman filter (compare last section) is used for the track fitting as well.

In order to obtain a track selection that does not depend on the DUTs, the track fitting is done without using the space points of the devices. The tracks are extrapolated into the planes of the DUTs and are stored as hit coordinates. Due to the higher integration time of the telescope, several tracks will be recognised by the telescope but cannot be seen by the DUTs. Therefore, one can demand a number of hits in the DUTs to reduce the number of tracks which are not intime in the DUTs and the telescope.

10.2.8. NTuple Dumping

The ntuple dumping is the last step in the reconstruction chain. Basically all variables estimated during the reconstruction and also the raw data are passed into a root file. Furthermore, a coordinate transformation is done to store the hit coordinates, estimated through the track fitting, in the local coordinate system of the sensor.

Parts of the ntuple dumper have been implemented during this thesis. An original version was available which did not dump all necessary data. Table 10.1 shows an overview of all accessible variables of the output root file.

The 'timings' branch will not be written when using the default settings in the ntuple dumper processor. Each branch contains vectors of variables and a variable containing the number of elements in the vector. In order to allow a mapping between the indices in the 'zspix' (raw data) and the 'euclusters' branch (clusters) the cluster ID, which corresponds to the index of the vector in the 'euclusters' branch, is saved.

The branch 'eutracks' contains estimated hit positions and further information like incidence angles, the sensor ID and track parameters like χ^2 and the degrees of freedom. In addition to that, the number of parameters[1] used for reconstructing the track is saved.

[1] This parameter can change because a minimum number of sensor planes can be set in the fitting processor.

Tree/Branch	Data type	Description
euhits		
nHits	int	Number of hits in this event
xPos	std::vector<double>	Global x coordinate [mm]
yPos	std::vector<double>	Global y coordinate [mm]
zPos	std::vector<double>	Global z coordinate [mm]
clusterId	std::vector<int>	ID of the corresponding cluster
sensorId	std::vector<int>	ID of the corresponding sensor
zspix		
nPixHits	int	Number of raw hits in this event
euEvt	int	Current event number
col	std::vector<int>	column of raw data hit
row	std::vector<int>	row of the raw data hit
tot	std::vector<int>	TOT of the raw data hit
lv1	std::vector<int>	LVL1 value of the raw data hit
iden	std::vector<int>	ID of the sensor
chip	std::vector<int>	ID of the sensor in the MCC board
clusterId	std::vector<int>	ID of corresponding cluster
eutracks		
nTrackParams	int	Number of parameters for estimation
euEvt	int	Event number
xPos	std::vector<double>	The fitted x position [mm]
yPos	std::vector<double>	The fitted y position [mm]
dxdz	std::vector<double>	The fitted derivate $\partial x/\partial z$
dydz	std::vector<double>	The fitted derivate $\partial y/\partial z$
trackNum	std::vector<int>	The track ID
iden	std::vector<int>	ID of the corresponding sensor
chi2	std::vector<double>	χ^2 of the track
ndof	std::vector<double>	tracks' degrees of freedom
euclusters		
euEvt	int	Event number
size	std::vector<int>	Number of pixels in a cluster
sizeX	std::vector<int>	Cluster width in x [pixels]
sizeY	std::vector<int>	Cluster width in y [pixels]
posX	std::vector<int>	Position of the cluster in x [pixels]
posY	std::vector<int>	Position of the cluster in y [pixels]
charge	std::vector<int>	Sum charge of the cluster [TOT]
iden	std::vector<int>	ID of the corresponding sensor
ID	std::vector<int>	ID of the cluster
timings		
NTimings	int	Number of timings in this event
SensorId	int	ID of the corresponding sensor
TluId	std::vector<int>	TLU ID of this event
TpllId	std::vector<int>	TPLL ID of this event
RealtimeSec	std::vector<int>	Realtime since clock reset [s]
RealtimeNs	std::vector<int>	Realtime since clock reset [ns]
RunNumber	std::vector<int>	Number of the run

Table 10.1.: Overview of content of the reconstruction output root file (tbtrack file)

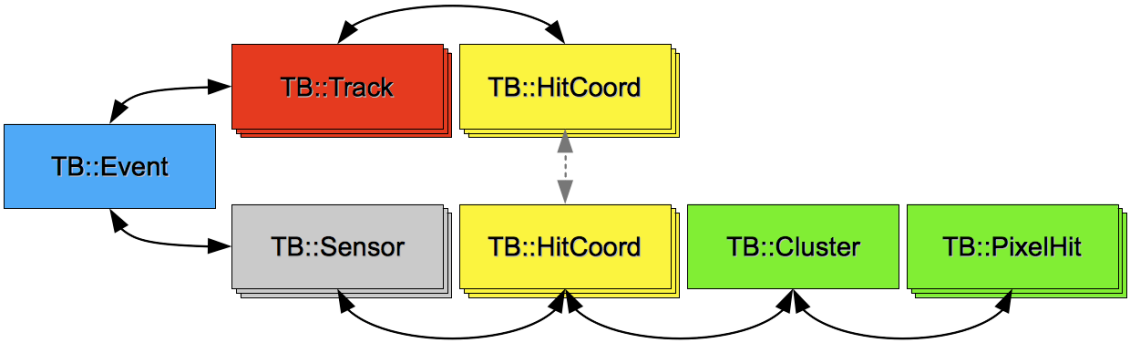


Figure 10.5.: Illustration of the object oriented TB::Event object and its children

10.3. TbTupleAna

In order to allow a detailed analysis of the reconstructed data, a tool called TbTupleAna was implemented in C++ as part of this thesis.

TbTupleAna is a library that provides classes that can be called from the root interpreter CINT[1] [13]. The program is separated into analysis steps. Different kinds of analyses can be attached into an analysis container. By using class inheritance and abstract functions, it is secured that three functions are implemented in each analysis step. These are 'start', 'end' and 'eventLoop'. 'Start' and 'Stop' have no parameters and no return values. 'EventLoop' needs a TB::Event object as an input parameter and returns such an object again.

When calling the 'run' method of the analysis container the following sequence of all analysis steps is called:

1. Call of the 'start' function of every analysis step.
2. Periodical call of the 'eventLoop' function of every analysis step as long as an event object is returned.
3. Call of the 'end' function of every analysis step.

In all cases the functions of every analysis step are called in the order they are attached to the analysis container.

Beside the analysis steps an easy description of the setup is necessary and needs to be provided for TbTupleAna. Each sensor is represented by the TB::SensorInfo class and can be attached to the TB::SensorInfoObjectCollection. By using this description a user friendly naming of the analyses plots can be realised very easily.

The TB::Event object is the object oriented representation of one detector event. An event in this context contains all information related to one TLU trigger. Figure 10.5 shows the dependencies of the event objects. The root object is of the type TB::Event. All sensor based information are available through the list of TB::Sensor objects. Each sensor object contains a list of hit coordinates represented by TB::HitCoord. Only one cluster is in relationship to one hit coordinate linked as TB::Cluster. The cluster is made of TB::PixelHit objects.

[1] **C** and C++ **INT**erpreter

From the root TB::Event object a list of TB::Track objects is available as well. A list of hits belongs to each track - one for each sensor. The connection between the hit coordinates of the tracks and the hit coordinates of the sensors is greyed out, as it is not a part of the reconstruction. A matching algorithm is used to estimate a connection.

The first analysis step which needs to be attached to the analysis container is the so called 'ReadInAnalysis'. This analysis step loops over all events of a list of ntuples and writes them into the object oriented structure as shown in figure 10.5. Two different kinds of analysis steps like this have been developed: one only reads in the raw data and the extrapolated hit positions from the tracks. That means that for the estimation of the hit coordinates of the sensor itself (lower branch in figure 10.5) a clustering and hitmaking needs to be applied afterwards. The other analysis step reads in all relevant information, so that an additional cluster and hitmaking is unnecessary. The first solution was necessary in order to cope with ntuples of older versions which did not include the mapping information between the indices of hit coordinates and clusters.

11. Testbeam data taking

This chapter gives a short overview of the testbeam periods, the available data and tunings of DUTs that were set during the data taking. A strange behaviour of the MCC board which limited the data acquisition in the first period will be discussed additionally.

11.1. Available Testbeam Periods

During this thesis five testbeam periods have been accomplished. The following list offers a brief summary of these periods:

CERN October 2009 was the first testbeam period ever achieved within the PPS community. It was very useful to obtain first experiences in planning and operating an experiment like this. In summary, the interface between EUDAQ and TurboDAQ worked very well, without greater problems. Also the byte stream conversion used to interpret the data taken in realtime worked sufficiently. Unfortunately, a lot of problems occured when more than two SCAs were connected to the MCC board simultaneously. More details about this can be found in chaper 11.3. Only because of the data taken in this period, the first reconstructions were possible. Table 11.1a gives an overview about the SCAs used during this period.

DESY March 2010 Additional studies [75] on the behaviour of the MCC board were necessary in order to solve the problem which was discovered during the last period. This testbeam period was used as a verification for the sufficient performance of the setup. As the motivation for this testbeam was not to understand the properties of the sensor, but the behaviour of the setup, only unirradiated ATLAS pixel devices were used.

CERN July 2010 With this period the debugging phase ended. Table 11.1b summarises the DUTs, and the bias voltages applied to the sensors during this period. By using an enhanced documentation system, more easy-to-use shift reports have been written, which make the mapping between data, devices and run properties and because of this the reconstruction less challenging. In addition to that, dedicated sensors with test patterns [83, 84] to allow specialised studies have been assembled to SCAs and used in a testbeam environment for the first time.

CERN October 2010 This period is the consistent continuance of the last testbeam. This time, further irradiated DUTs with higher fluences were available. Furthermore, the irradiation of the sensors with test patterns was completed, so that these devices could be investigated as well. When it became obvious that the temperature and the sensor currents are important variables in order to obtain a better understanding of the sensor's properties, the monitoring system for these values was improved. In table 11.1c the investigated devices are listed.

CERN November 2010 The main aspects of this period are bias voltage scans under different incidence angles. The studied angles were $0°, 15°$ and $30°$. The corresponding devices are summarised in table 11.1d. Part of the time was also used to validate a new testbeam setup for ATLAS pixel FE-I4 in combination with the USBpix[1] setup.

11.2. Parameter settings in the testbeam

During all testbeam periods the DUTs were tuned to a threshold of $3200\,e^-$ and the FDAC settings were adjusted to a value so that 60 TOT correspond to $20\,ke^-$. The time in between the testbeam data taking was used to perform intermediate scans of the DUTs such as threshold and TOT-scans.

The temperature of the DUTs varied over wide ranges (see chapter 12.6), as the dry ice cooling cannot be regulated easily and needed first experiences.

The devices investigated here were irradiated to varying fluences. Irradiations from 1 up to $20 \cdot 10^{15}\,n_{eq}\,cm^{-2}$ had been achieved to simulate the radiation damage of the ATLAS detector. The DUTs were irradiated either with neutrons[2] or protons[3]. Further information about the irradiation of the DUT can be found in [3].

11.3. Behaviour of the MCC board in October 2009

In the testbeam period of October 2009 one feature of the MCC board attracted attention. When running simple digital injection scans it occured that no or only little data was received from several connected SCAs. After some investigations it was recognised that the voltage drop caused by the two meter ribbon cable was high enough for the devices to loose their configuration. It only appeared when more than two devices were connected simultaneously, as the voltage drop increases with the higher current draw because of the greater number of devices.

It was not possible to cope with the problem during the first testbeam period. Additional studies [75] were necessary to find out that a well-defined ground connection of all components is required to guarantee a proper working.

In the testbeam periods beginning with July 2010, a CAEN power crate was used to apply voltages to all components. The power supply modules used in this crate distinguish themselves from the ones used before by the fact that they are more stable and more precisely defined with respect to ground. Although higher voltages (2.5 V at VDD and 2.1 V at VDDA) are needed to supply the SCAs via the MCC board, no problem like this occured again.

[1] ATLAS **PIX**el readout board with **USB** interface

[2] The irradiation with neutrons was done in the TRIGA Mark II nuclear reactor at the Jožef Stefan Institute in Ljubljana

[3] The proton irradiation was done either at IRRAD-3, CERN PS or at the Compact Cyclotron at KIT, Karlsruhe

device name		$\Phi_{eq}[n_{eq}cm^{-2}]$
1	2-6A	$1 \cdot 10^{15}$ (n)
2	11-5B	$2 \cdot 10^{15}$ (n)
3-6A	3-6A	unirradiated
DO1	10-5A	unirradiated
DO2	2-6B	unirradiated
DO7	7-6B	$1 \cdot 10^{15}$ (p)
DO28	7-6A	$0.5 \cdot 10^{15}$ (p)
DO29	8-6A	$1.5 \cdot 10^{15}$ (p)
DO30	6-6B	$2 \cdot 10^{15}$ (p)

(a) Devices used in testbeam October 2009

device name		$\Phi_{eq}[n_{eq}cm^{-2}]$	U[V]
DO1	10-5A	unirradiated	
DO3	9-10A	unirradiated	130
			100
DO6	9-10B	unirradiated	130
DO7	7-6B	$1 \cdot 10^{15}$ (p)	500
			600
			800
			1000
DO8	4-6B	$1 \cdot 10^{15}$ (n)	400
			600
			800
DO9	5-6B	$5 \cdot 10^{15}$ (n)	400
			600
			800

(b) Devices and their applied bias voltages, used in testbeam in July 2010

device name		$\Phi_{eq}[n_{eq}cm^{-2}]$	U[V]
DO1	10-5A	unirradiated	150
DO3	9-10A	unirradiated	130
DO6	9-10B	unirradiated	130
DO7	7-6B	$1 \cdot 10^{15}$ (p)	600
			800
DO8	4-6B	$1 \cdot 10^{15}$ (n)	400
			500
			600
			800
DO9	5-6B	$5 \cdot 10^{15}$ (n)	400
			600
			800
			1000
			1200
DO10	4-6A	$2 \cdot 10^{16}$ (n)	400
			600
			800
			1000
			1200
DO12	5-6A	$3.5 \cdot 10^{15}$ (p)	400
			600
			800
DO13	6-11A	$5 \cdot 10^{15}$ (n)	600
			1000

(c) Devices with their applied bias voltages, used in testbeam in October 2010

device name		$\Phi_{eq}[n_{eq}cm^{-2}]$	U[V]
DO1	10-5A	unirradiated	150
DO7	7-6B	$1 \cdot 10^{15}$ (p)	400
			600
			800
			1000
			1200
DO9	5-6B	$5 \cdot 10^{15}$ (n)	1000
DO10	4-6A	$2 \cdot 10^{16}$ (n)	500
			750
			1000
			1250
			1500
DO13	6-11A	$5 \cdot 10^{15}$ (n)	600
			1000
			1200
DO15	4-12A	$2 \cdot 10^{15}$ (n)	600
			1000
DO20	3-11B	$5 \cdot 10^{15}$ (n)	400
			600
			800
			1000
DO22	2-12B	$5 \cdot 10^{15}$ (n)	800
			1000
			1200
DO25	11-12B	$2 \cdot 10^{15}$ (n)	600
			1000

(d) Devices and their applied bias voltages, used in testbeam in November 2010

Table 11.1.: Overview of the used SCAs in the testbeam periods October 2009, July, October and November 2010

11.4. Further experiences

Successively the experiences from one testbeam period influenced the developments in all concerns to improve the situation the next time. Very positive reactions were registered concerning the developments of the DOBOX-3 (compare chapter 7.3.3) and the redevelopment of the online monitor (compare chapter 9.4).

In addition to that the mechanical building blocks used to mount the SCAs into the DOBOX are still state-of-the-art. Even for the new FE generation building blocks of similar kind were produced.

12. Preparations and Definitions for Analysis

Before actually starting the analysis of the recorded datasets some preparations need to be done, which are presented here. The necessary steps to perform the reconstruction are presented. Further on, some terms in the context of the sensor analysis are defined and the performance of the telescope is investigated.

In order to guarantee a correct analysis of the ATLAS pixel sensors some problems had to be solved beforehand, which will be discussed here. The process steps in the developed analysis framework TbTupleAna are shown as well. In addition to that, common illustrations which will be generated by the analysis tool will be explained.

At the end of this chapter already known limitations on sensor analysis studies like those presented in the next chapters will be given.

12.1. Running the reconstruction

By using the reconstruction software mentioned in chapter 10.2 datasets have been reconstructed. Each testbeam period was split into blocks in order to separate runs with individual DUT sets and DUT sets with different bias voltages. GEAR files were provided for each analysed block.

For the processing of the reconstruction, a set of python and shell scripts was developed in order to submit the jobs to the Dortmund hadoop cluster system. By using these scripts and some prepared template steering files the whole reconstruction can be done in a few steps:

Noise The first step calls the noise occupancy processor. It is called only once per block, as the noise occupancy is an effect of the sensor which does not change much. The generated LCIO database file is stored into an HDFS[1] directory automatically to be available for further reconstruction steps.

Raw to hit This step in the reconstruction chain automatically filters the telescope raw data on the basis of the noise occupancy file, performs the clustering and converts the clusters into hits.

Alignment on telescope This processor computes the alignment of the telescope. As described in chapter 10.2.6 the alignment is done in three steps: Kalman filtering to find track candidates, calling of mille to create binaries, call of pede to compute the alignment constants from the binaries. In this step pede is not called, because when doing so, multiple runs can be combined after being processed on several machines in parallel.

[1] **Ha**a**D**oop **F**ile **S**ystem

Pede on telescope As this step uses multithreading, which is not implemented for an hadoop environment, it needs to run directly. A program called 'pedemerge' was developed to merge the binary files. Furthermore, it computes the correct initial values for the matrix inversion by using the average value of the individual steering files.

Alignment on DUTs This step is similar to the one above, but it runs on the DUTs. Instead of using a Kalman filter, a DAF[1] fitter [44, 74] is used; this algorithm shows a better performance.

Pede on DUTs As for the telescope, the pede step can only be computed directly for the DUTs.

DAF Fitter The last step processes the DAF fitting and the dumping of all available data into a root ntuple. The necessary files, like the alignment constants are copied from an HDFS directory automatically.

12.2. Definitions used in the analysis

In the next sections common definitions used in the sensor analysis will be presented.

12.2.1. Definition of Hit Efficiency

The measurement of the hit efficiency is the determination of the detection efficiency. The hit efficiency can be estimated pixel by pixel or within one pixel. In both cases the hit efficiency q_{hit} is calculated by

$$q_{\text{hit}} = \frac{n_{\text{detected}}}{n_{\text{expected}}}, \tag{12.1}$$

where n is the number of hits. By using the EUDET telescope a particle trajectory can be reconstructed, so that a model point on a sensor through which the track traverses is estimated. This allows to determine n_{expected}. n_{detected} is received by the sensor information itself and a geometrical matching.

As the resolution of the telescope is higher than the resolution of the DUTs, it is possible to evaluate the hit efficiency in sub pixel resolution. As the statistic of traversing hits within one dedicated pixel is little, pixels of the same properties can be convoluted to increase the number of hits.

12.2.2. Definition of Charge Efficiency

The definition of the charge efficiency is more complex. As the ATLAS pixel SCA allows to measure the charge in the pixel, it is possible to compute the average charge either pixel by pixel or within a pixel. Therefore, the charge efficiency q_{charge} is defined as

$$q_{\text{charge}} = \frac{\text{charge}(n_{\text{detected}})}{n_{\text{detected}}}. \tag{12.2}$$

[1] **D**eterministic **A**nnealing **F**ilter

As the raw data unit for the charge in the sensors is measured in TOT, often q_{TOT} is used to illustrate the mean or the most probable value in units of TOT. Depending on the data illustration method, either the cluster TOT, meaning the sum of all TOT entries belonging to the corresponding cluster, or the individual pixel TOT is used.

To determine the hit position for the sub pixel resolution and to suppress noise hits, the hit positions estimated by the telescope are used.

12.2.3. Definition of Matching and Residual

When the root ntuple, which contains all available information about data taken and reconstructed, is read in into TbTupleAna, there is no direct connection between the hit coordinates of the reconstructed tracks and the hit coordinates of the sensor hits (compare figure 10.5). A connection between these different kinds of hit coordinates (grey dashed arrow) is essential for most of the following analysis steps.

Therefore, an algorithm is implemented in TbTupleAna as an individual analysis step to create such a connection. The algorithm uses a geometrical process to determine which hits belong together. One has to define two values which describe the maximum allowed distance between two partners in x and y. One has to take into account that the variation of this value from a theoretical zero value is caused by different factors:

1. As the pixel pitch of the ATLAS pixel sensors is much greater than the tracking resolution of the telescope, a natural width is obvious. In the distribution of the distances this effect causes the shape of a square function.

2. A sensor can be misaligned even after the alignment correction. In this case the shape in the distribution will be shifted.

3. The tracking resolution is limited. Therefore, the distribution will show a gaussian fraction.

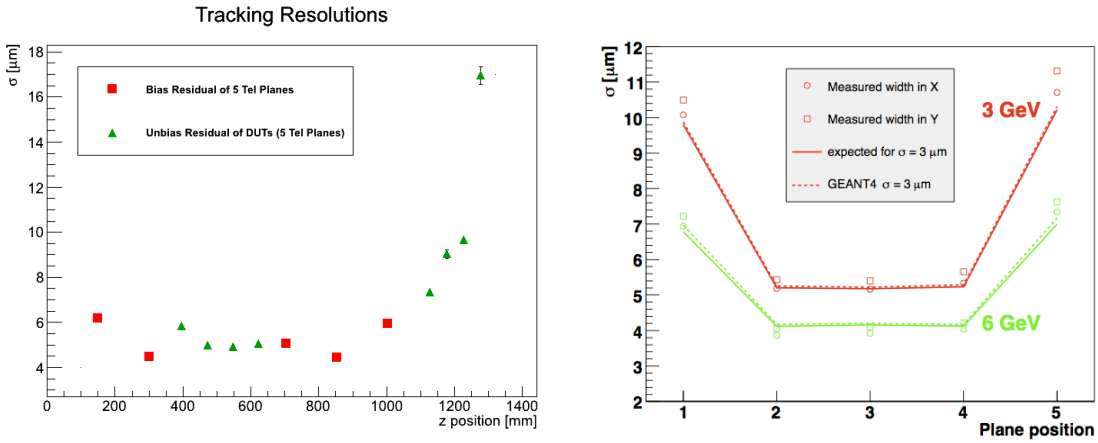
4. As the hit efficiency in the DUTs cannot be described digitally, especially in the region between two neighboured pixels, further gaussian-like fractions will be part of the distribution.

The algorithm compares the hit coordinates in x and y. If two coordinates appear within the mentioned limits they are connected to each other. Typically limits are 1.5 to 2.0 times the pixel pitch.

The distribution of the distances between the reconstructed hit position and the sensor hit position is called residual.

12.2.4. Definition of a Reference Hit and a Referenced Track

The number of tracks recognised by the telescope is higher than the number of tracks seen by the DUTs. The reason for this is on the one hand the higher integration time of the telescope (130 μs) in comparison to the DUTs ($16 \cdot 25\,\mathrm{ns} = 400\,\mathrm{ns}$) and the different sizes of both sensor types (telescope $2.25\,\mathrm{cm}^2$, FE-I3 DUT $0.64\,\mathrm{cm}^2$).



(a) Calculated telescope resultions

(b) Measurements for comparison [63]

Figure 12.1.: (a) Tracking resolution in dependency of the z-position, calculated from the biased residuals of the telescope (red) and the unbiased residuals of the DUTs. The error bars show the resulting error of σ calculated by the fit. The run 11250 of the July 2010 testbeam at CERN SPS was used for this analysis. Detailed residuals are printed in the figures B.1 to B.3,
(b) Telescope resolutions for the EUDET telescope demonstrator sensors at 3 and 6 GeV electron beam. The distance of the sensors is 2.5 cm [63]

This means when computing values like hit efficiency (compare chapter 12.2.1), not all tracks traversing a DUT may contribute to the denominator in equation (12.1), as their passing through the sensor happened at a point in time in which the sensor was blind.

An algorithm is used to distinguish between tracks that occured in the sensitive time of the DUT and those that did not. One uses one or more reference sensors to qualify the track as so called referenced track. A track can be referenced by a sensor by using a similar algorithm as used for the matching. For referencing other matching values can be set. Furthermore, individual reference sensors can be chosen for each DUT and cuts on the LVL1 trigger number can be applied.

12.3. Performance of the Telescope

For a better understanding of the tool used for the track determination the performance of the EUDET telescope is analysed. In order to achieve a performance attribute, so called biased and unbiased residuals can be used. When fitting a track by using, for example, all six telescope planes, one can calculate the residuals

$$\Delta x = x_{\text{Track}} - x_{\text{Plane}}, \tag{12.3}$$

$$\Delta y = y_{\text{Track}} - y_{\text{Plane}} \tag{12.4}$$

which describe the distribution of the deviation of the estimated track from the individual position at which a sensor was hit. These residuals are biased residuals because the observed plane is part of the fitting process. To obtain unbiased residuals of one specific plane, this plane must not be used for track fitting. Unbiased residuals allow the best

possible estimation on tracking resolutions. As the examination of all n planes needs n independent fittings, only biased residuals are used for the estimation of the tracking resolution at the positions of the telescope.

Because of their larger pitch the resolution of the DUTs is much worse than the resolution of the telescope. This is why they are not used for track fitting at all. Therefore, unbiased residuals can be estimated at the positions of the DUTs.

In figure B.1 on page 192 the biased residuals of the telescope of the run 11250 from the July 2010 testbeam period at CERN SPS are shown. At this time only five telescope planes, two in the first arm and three in the second arm, are used due to a failure in one sensor. The figures B.2 and B.3 on the following pages contain the unbiased residuals of the DUTs. The shoulders next side to the residual distribution of the long pixel direction appear as all pixels of the DUT, including the long pixels, were used for this kind of illustration.

In the residuals of the telescope a gaussian fit was used to determine the width of the distribution. As the pitch of the DUTs is much larger with respect to the tracking resolution of the telescope, the shape of the DUT's residuals will be different. These residuals will contain a rectangular fraction, as the traversing of the track within the area of the pixel will cause a homogenous distribution. Furthermore, a gaussian fraction will be visible due to the tracking uncertainty and the non-homogeneous hit efficiency behaviour of the pixels at their edges. Therefore, a fit function of the kind

$$y = c \cdot \frac{1}{\exp(\frac{\mu-\sigma/2-x}{p/2}) + 1} \cdot \frac{1}{\exp(\frac{x-\mu-\sigma/2}{p/2}) + 1} + b \tag{12.5}$$

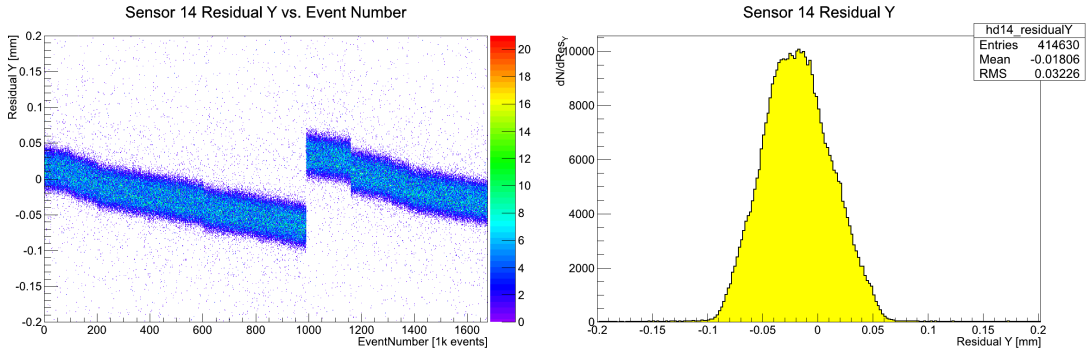
where

- c is a constant (varies with the amount of data),
- μ is the mean value of the residual (should be 0 when perfectly aligned),
- σ is the width of the gaussian fraction,
- p is the width of the rectangular part (corresponds to the pitch),
- b is the amount of data resulting from arbitrary coincidences (background)

is used to describe the residual of the DUT. This is called a double fermi function. The variable σ will contain the tracking uncertainty and also the pixels' behaviour at their edges. But it can be used as a worst case approximation for the tracking resolution.

The σ values of the residuals are plotted in figure 12.1a at their corresponding z-position. Only the residuals in the x-direction are used (y is comparable) for the telescope. As the DUTs located between the telescope planes are rotated 90 ° along the z-axis with respect to the DUTs located behind the telescope, the shape of the residuals in x and y differ in the mentioned figure. One can see that the width p of the residuals is compatible with the pixel pitch, which is 50 μm or 400 μm. For the plot in figure 12.1a only the residual containing the smaller pixel pitch is used, as the resolution is better due to the higher number of pixels in that direction.

One can see that the resolution of the telescope becomes worse for the outer telescope planes. In the second arm (containing three planes) the resolution is best at the middle plane with $4.435 \pm 0.014\,\mu$m. These results are compatible with those from [63], in which a demonstrator sensor of the EUDET telescope was used (compare figure 12.1b). In that case the pixel pitch was $30 \times 30\,\mu$m (instead of $18.4 \times 18.4\,\mu$m). As the distance of the



(a) The residual of y vs. the event number

(b) The residual of y

Figure 12.2.: Illustration of the so called running residual: (a) shows the residual in y of one sensor in dependency of event number. (b) is the residual distribution

sensors, their pitch, and their readout electronics is different[1] from the current EUDET sensor only a rough comparison between them is possible. The tracking resolution at the position of the DUTs located between the telescope arms is much better ($\approx 5\,\mu$m) than behind the telescope ($7.36 \pm 0.14\,\mu$m at 125 mm distance and $16.95 \pm 0.39\,\mu$m at 275 mm). One can see that at the position of the DUT closest to the telescope arm with only two sensors the resolution is slightly decreased ($5.85 \pm 0.12\,\mu$m).

As a consequence of this analysis, one sees that for studies which need a high resolution (for example sub pixel analysis) the DUTs should be mounted between the telescope arms, or very close behind the telescope.

12.4. Challenges of the analysis

During the analysis of the testbeam data some difficulties occurred, which had not been obvious before. These are presented here.

12.4.1. The Running Residual

In figure 12.2b the residual of the y coordinate (vertical) of one DUT is shown[2]. When running the matching algorithm it became obvious that this residual is much too wide in comparison to the pixel pitch. When looking at the development of this residual in units of events, which is printed in figure 12.2a, one sees that the residual moves as a function of the event number.

The reason for this is the sublimation of the dry ice used for cooling the DUTs. When sublimating it looses its weight and changes the stress distribution of the testbeam box, which was deformed before and then returns into its usual state. Furthermore, the step in figure 12.2a indicates a sudden movement of the box. The shift reports show that the dry ice was replaced at this point in time.

[1] The demonstrator sensor used an analogue readout to further improve the tracking resolution

[2] The underlying data are the runs 11250 to 11282 from July 2010. The sensor is DO8.

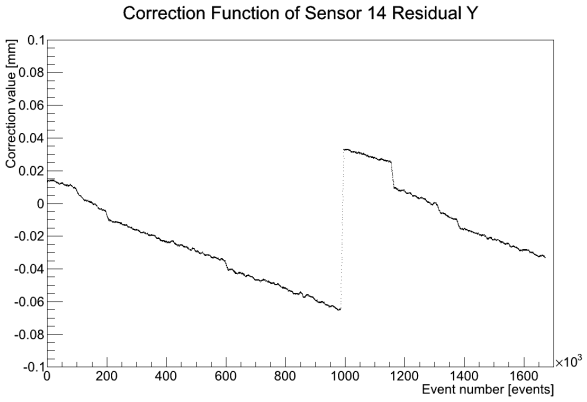


Figure 12.3.: Correction function of the running residual (compare figure 12.2)

The overall time from event number 0 in the figure to event number 1 000 000, where the step is located, is roughly six hours. The dimension of the shift is roughly 80 μm. With these values the typical speed of the moving devices induced by the melting dry ice can be calculated to 3.7 nm s^{-1}.

The running residual can be corrected by using two analysis steps in TbTupleAna. The first step determines the residual shift on a subset of events and the second step applies the correction values event by event.

The main difference of this alignment correction in contrast to the alignment correction of the reconstruction is the fact that not exactly one correction value is calculated per run or per set of runs, but per event or per set of events. Further alignment corrections such as rotations are not corrected here.

A so called moving average is used to calculate the offset for each sensor in x and y direction. The moving average $\bar{x}_i$ with the width n for the event number i is defined as

$$\bar{x}_i = \frac{1}{n} \sum_{a=i-\frac{n}{2}}^{i+\frac{n}{2}} x_a, \tag{12.6}$$

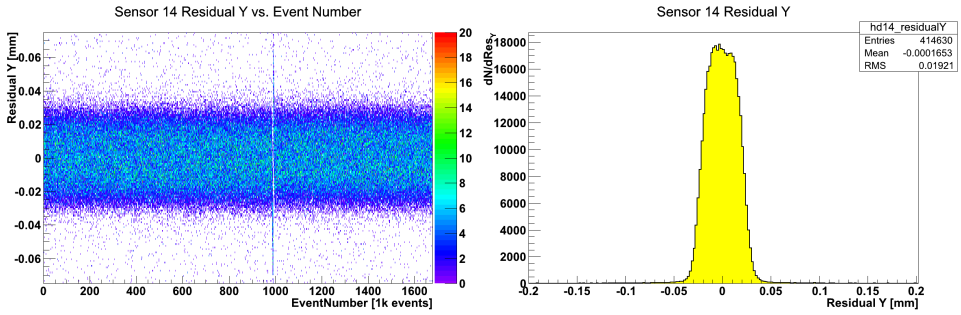
$$= \bar{x}_{i-1} - x_{i-\frac{n}{2}-1} + x_{i+\frac{n}{2}} \tag{12.7}$$

In this definition $\bar{x}_i$ can only be calculated for values $i > \frac{n}{2}$ and $i < N - \frac{n}{2}$, where N is the maximum available number of events. A definition for values of the first exception is

$$\bar{x}_i|_{i<\frac{n}{2}} = \frac{1}{n} \left(\sum_{a=0}^{i+\frac{n}{2}} x_a + \sum_{a=0}^{\frac{n}{2}-i} x_a \right). \tag{12.8}$$

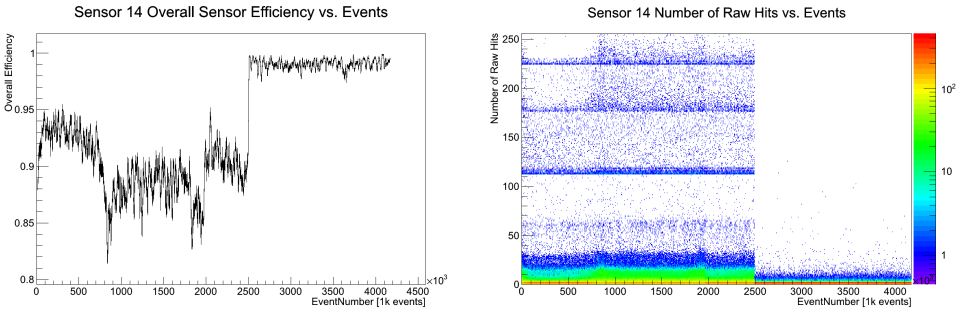
The other exception can be defined analogous.

Although it is possible to compute one alignment correction per event, only every 200$^{\text{th}}$ is stored into a root file, which is used as correction database to reduce the file size to a required minimum. The second analysis step reads in these values and uses a linear interpolation to apply individual alignment corrections for each event. The typical width of the moving average n is 2500 events.



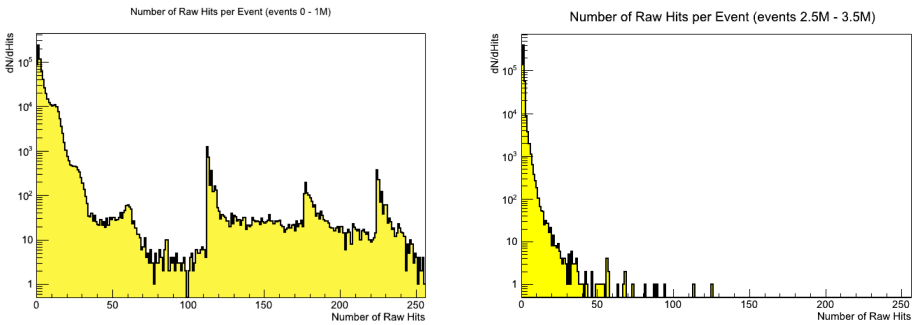
(a) The residual of y vs. the event number (b) The residual of y

Figure 12.4.: Illustration of the running residual after its correction (compare figure 12.2). (a) shows the residual in y of one sensor in dependency of event number. (b) is the residual distribution



(a) The Efficiency Puzzle (b) Number of raw hit vs. event number

Figure 12.5.: (a) The HitEfficiency of DO8 in dependency of the event number. The plot shows the so called Efficiency Puzzle, as the overall efficiency is not a constant. The data used here are the runs 11193 to 11282 of July 2010.
(b) Explanation for the Efficiency Puzzle. The number of raw hits vs. the event number is shown. The plot is split into two regions. The vertical projections are shown in figures 12.6a and 12.6b



(a) Raw hits in 1st region (b) Raw hits in 2nd region

Figure 12.6.: Explanation for the Efficiency Puzzle. The two reagions visualizied in figure 12.5b are shown in (a) and (b). The data used here are the runs 11193 to 11282 of July 2010

Figure 12.3 shows the correction function, which was calculated by the moving average for the running residual shown in figure 12.2. In figure 12.4 the sensor positions are corrected. One can see that the residual is almost independent from the event number. In addition to that the shape of the residual is as expected: The shape is given by the pixel pitch of 50 μm (rectangular fraction) and a gaussian fraction because of the tracking resolution and the non-homogenous hit efficiency.

The figure mentioned also shows the limits of this correction method. The step (at $\approx$ 1 000 000 events in figure 12.4a) is only corrected smoothly because of the moving average. This is the reason why the alignment constants in the reconstruction should not be used to create correction databases for each run. Although the individual residuals would be tighter, more steps would appear, which cannot be adapted correctly by this algorithm.

12.4.2. The Efficiency Puzzle

The development of the efficiency in units of event numbers is shown in figure 12.5a. One would expect that the hit efficiency is a constant number; in contrast to that the figure shows that it varies over a wide range (80 % to 95 %), shows steps and trends in its development.

In order to understand the behaviour of the DUTs many investigations were made. During this thesis a direct correlation to the distribution of the number of raw hits per event in one sensor and the hit efficiency was detected. Figure 12.5b shows the number of raw hits vs. the event number. One can immediately see that the plot can be split into two regions. The event 2.5 million seems to be the margin. To further illustrate the difference between the two regions, the histogram in figure 12.5b was projected to its y-axis by using events either from the first or the second region. Figures 12.6a and 12.6b show these projections by using the events 0 to 1 million and 2.5 to 3.5 million.

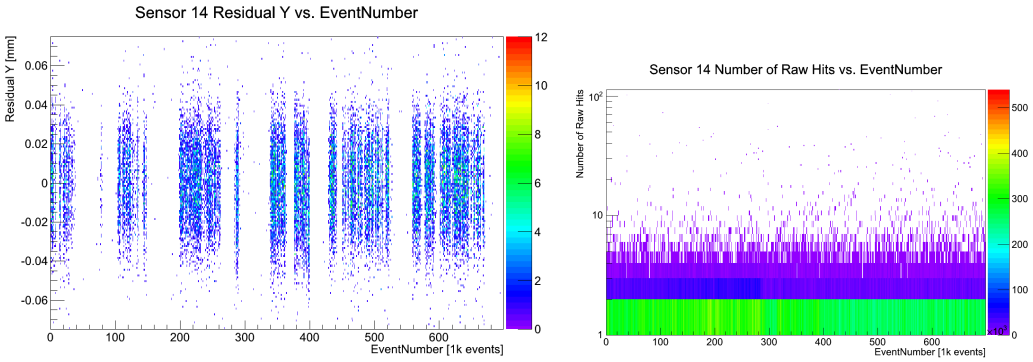
The number of raw hits per event and sensor is the number of pixels hit per ATLAS pixel FE-I3 at one TLU trigger. As explained in chapter 4.5 the maximum number of events that can be stored per FE in the ATLAS pixel MCC is 112. The first region (compare figure 12.6a) is distributed over the whole histogram with values much greater than 112. In contrast to this the values of the second region (compare figure 12.6b) occur only at small numbers lower than 112.

In addition to that the distribution shows peaks at the values $113 = 112 + 1$ and $225 = 2 \cdot 112 + 1$. This indicates that an event desynchronisation either in the FE or the MCC happened[1].

When comparing the figures 12.5a and 12.5b one also detects the margin in the efficiency plot. When using only the data of the second region the efficiency becomes constant.

Some effort was made in order to repair these corrupted data in the first region: The events with the number of raw hits greater than 112 were avoided, for example. Additionally, the further n events were neglected to see if this would improve the behaviour. But in all cases the development of the hit efficiency was still comparable to that shown in the mentioned figure. Furthermore, it was observed that sensors which do not show the

[1] If the content of a second or further events are copied or moved to another existing event accidentally, a copy of the shape from bins 0 to 112 to bin 113 and following and 225 and following appears.



(a) Residual holes (b) Development of the number of raw data

Figure 12.7.: (a) Development of the residual of y and (b) the number of raw hits of DO13. The matching algorithm cannot calculate the residuals interimly, which leads to the residual holes

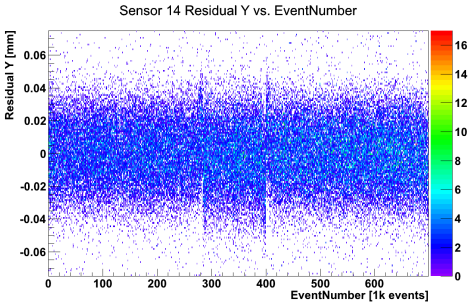
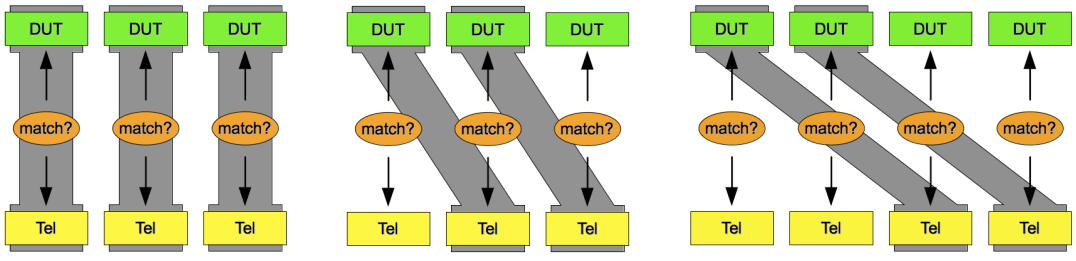


Figure 12.8.: Development of the residual of y of DO13 after having applied the merging algorithm



(a) No desynchronisation (b) Desynchr. offset 1 (c) Desynchr. offset 2

Figure 12.9.: Illustration to explain desynchronisation. The raw data parts that originally belong together are indicated by the grey background. The matching algorithm only works within one event

greater 112 occurrence change their behaviour when data corrupted region is used. This is one indication that the event desynchronisation happens in the MCC.

As it is not possible to repair the corrupted data it turned out to be very important to check the distribution of the number of raw hits during the data taking or at least before analysing the data.

12.4.3. Residual holes - Desynchronisation part 1

When the analysis of the testbeam data taken in October 2010 started, the phenomenon shown in figure 12.7a was recognised[1]. The illustration shows the development of a residual of a DUT as it was used before in chapter 12.4.1. Remarkable about this plot is the appearance of interim holes in the development of the residuals. In contrast to this, the amount data is constant (compare figure 12.7b). When comparing the residual developments of different sensors in the same runs it can be discovered that the holes appear in all cases, but at different times. For analyses which use DUTs as reference sensors (for example the hit efficiency analysis step does) this induces a big problem as it cannot be guaranteed anymore that reference sensor and DUT are able to detect the same track.

As the constant number of raw data in one sensor is in conflict with the residuals and furthermore the validity of analysis steps was not safe anymore, one started to investigate the phenomenon of the residual holes:

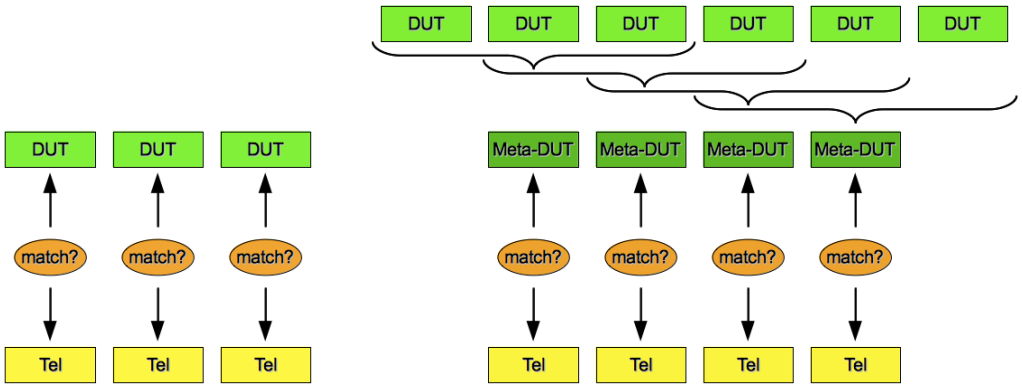
In figure 12.9 a theory is shown, which was used to explain this effect. The residuals are calculated within the matching algorithm (compare chapter 12.2.3). Origin of the extrapolated hit positions from the telescope tracks is the raw data collection of the telescope. In contrast to this, the origin of the hit positions on the DUTs, is the raw data collection of the DUTs. So, the matching algorithm uses data from both independent collections when computing event after event (compare figure 12.9a). If accidentally, for an unknown reason, the raw data of the telescope and of the DUTs are relatively displaced, as shown, for example, in figure 12.9b, the matching algorithm will not find hits which match, as the wrong datasets are compared with each other.

In order to verify this theory mentioned a merging analysis step was implemented in TbTupleAna. In figure 12.10 it is illustrated how this algorithm works. Figure 12.10a shows the well-known simple case: Exactly one DUT dataset is compared with one telescope dataset. In figure 12.10b the dataset of the DUT is replaced by a meta dataset consisting of multiple events. This means that the raw data of event n is merged with the raw data of events $n-1$ and $n+1$ for each sensor. The number of events to be merged is a parameter which needs to be set in the analysis step.

The probability that an arbitrary coincidence between the telescope tracks and the sensor hits causes a matching was calculated to be lower than $2.8 \cdot 10^{-5}$. Underlying values of these poisson calculations are an average number of tracks per telescope read-out frame (assumed to be not higher than 20) and the known sensor geometries. Furthermore, the matching values have influence.

After applying the merging analysis step in TbTupleAna to the data of this period, the development of the residual looks familiar, as shown in figure 12.8. The little spikes are

[1] The underlying data are the runs 20995 to 21016. The sensor used is DO13



(a) Simple matching (b) Matching on merged data

Figure 12.10.: Illustration to explain the function of the merging algorithm on the matching algorithm

debris of the alignment corrections (compare chapter 12.4.1). One has to keep in mind that in the fitter processor of the default reconstruction chain each track needs to be approved by at least one hit in a DUT in order to reduce the root file's size. The data investigated here is desynchronised which means that such a matching on reconstruction level leads to omissions of entire tracks. Therefore, this option needs to be disabled for data like this.

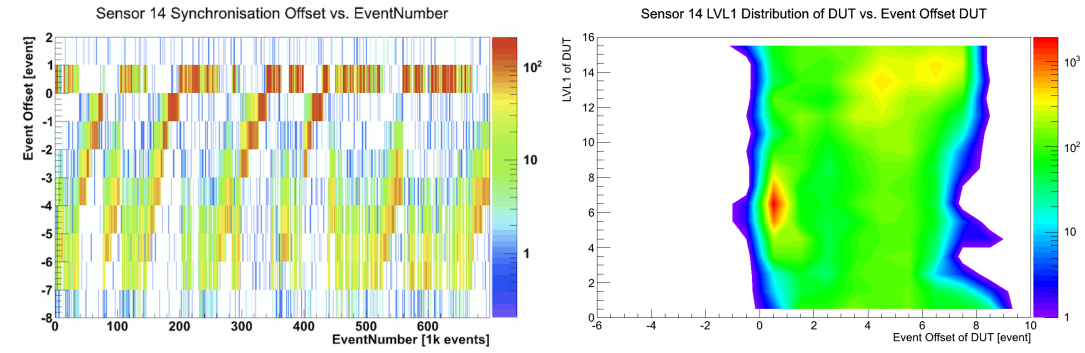
The amount of desynchronisation as a function of the event number can be investigated as well. Figure 12.11a shows such a plot. In this illustration, entries in the horizontal zero bin indicate no desynchronisation. Entries in the bin -1 or -2 correspond to the cases shown in figures 12.9b and 12.9c.

It can be discovered that the desynchronisation happens in periods. After a synchronised part, the DUT data is displaced 7 events too late but recovers itself from desynchronisation successively within roughly 50 000 events. The mechanism causing this desynchronisation is not completely understood. It seems to be related to features of the MCC board, as the raw data contains more errors indicating this in this testbeam period. A possible relation to the DAQ readout, especially the synchronous readout of the data FIFO and the EUDET FIFO (compare chapter 8.5.5), might explain this feature as well[1]. An old bug in the VHDL code of the TPLL might cause it, too.

12.4.4. LVL1 correlations - Desynchronisation part 2

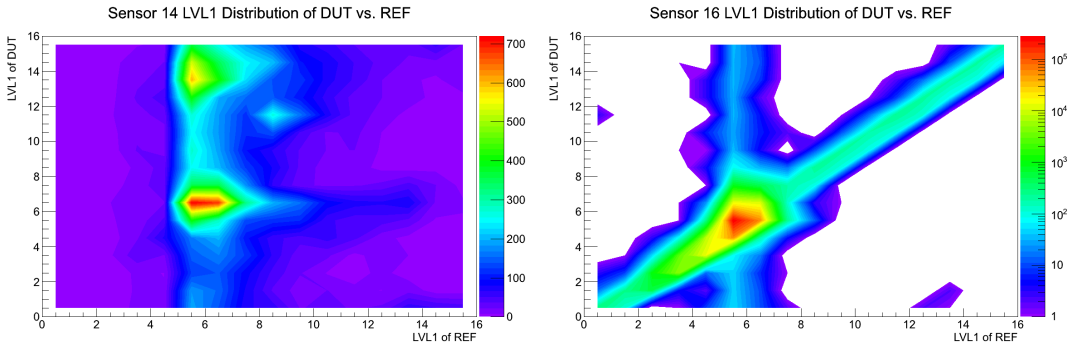
Usually the LVL1 trigger number of the cluster induced by a hit in the reference sensor is strongly correlated with the same value in the DUT. An example for a plot like this is shown in figure 12.12b. The data used to produce this plot is taken from the testbeam period at CERN July 2010[2]. In contrast to this the data of CERN October 2010[3] shows a correlation as illustrated in figure 12.12a. The merging algorithm, as explained in chapter

[1] In the October 2010 testbeam period a lot of FIFO synchronisation errors occured. These errors seemed to depend on the room temperature.
[2] The runs 11193 to 11250. The sensor used is DO9.
[3] The runs 20995 to 21016. The sensor used is DO13.



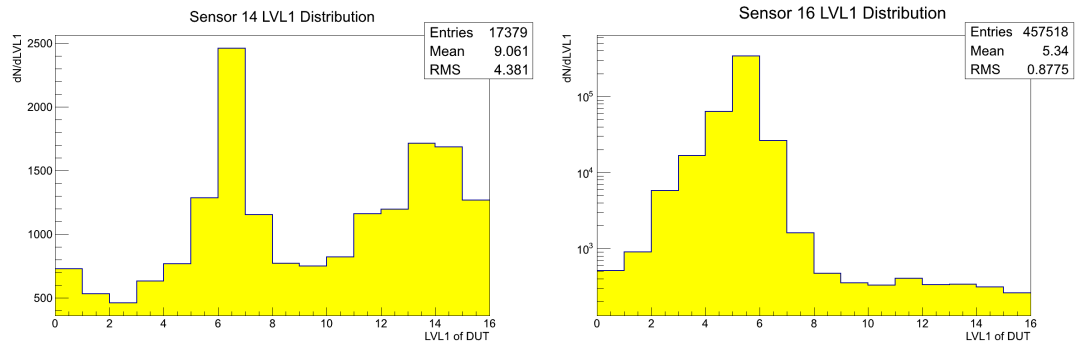
(a) Development of the desynchronisation (b) Desynchronisation vs. LVL1 of the DUT

Figure 12.11.: (a) The amount of desynchronisation as residual in time in units of events vs. the event number. (b) The desynchronisation offset vs. the LVL1 trigger number of the DUT



(a) Corrupted data (lin scale) (b) Good data (log scale)

Figure 12.12.: Correlations of the LVL1 trigger number of the reference sensor vs. the LVL1 trigger number of the DUT in different testbeam periods



(a) Corrupted data (lin scale) (b) Good data (log scale)

Figure 12.13.: LVL1 distributions of DUTs in different testbeam periods

12.4.3, was applied. The corresponding LVL1 distributions of the DUTs are shown in figures 12.13a and 12.13b.

In case of good data the LVL1 distribution peaks at two or three bins due to timewalk effects and is suppressed at least one order of magnitude beside these. In contrast to this the distribution shown in figure 12.13a peaks at two LVL1 trigger numbers and is almost flat. Usually the correlation of the DUT and the reference sensor shows one peak, as both LVL1 distributions look quite similar and a diagonal line which indicates that the track is recognised by different sensors intime. The correlation in figure 12.12a is almost flat. In addition to that, the histogram peaks at two positions. This indicates that the parallel readout of all sensors seems to be partly defected: Incidents that happened simultaneously are read out as if they occurred successively.

As one can see in figure 12.11b the second peak in the LVL1 distribution and the correlation is in direct relation to the desynchronisation described in chapter 12.4.3. The plot shows the amount of desynchronisation on the x-axis and the LVL1 distribution (figure 12.13a) on the y-axis. This means that the effect, which caused the displacement of raw data into wrong events, causes a second feature which moves hit occurences to wrong LVL1 trigger numbers. The resulting shape of the LVL1 distribution can be caused by two reasons: The data taking was working properly, but for an unknown reason the LVL1 values are not estimated correctly. Or the same problem causing the event desynchronisation causes a sending of the LVL1 triggers to the FE at wrong times during the data taking. In this case, possibly a number of hits occured at a moment in time, in which accidentally no trigger was sent to the FE. This means it was not recognised and not saved into the raw data. This feature would cause a systemic inefficiency of unknown magnitude. So far no possibility was found to determine the inefficiency of the system.

This feature has many consequences for dedicated analysis: From all kinds of analysis which needs to be referenced, absolute results cannot be used anymore. In particular this means that one cannot estimate absolute hit efficiency, neither globally, nor at a dedicated point on the sensor. Admittedly, relative hit efficiencies can be calculated anyhow when using the data including this feature.

12.5. Analysis in TbTupleAna

As already mentioned in chapter 10.3, TbTupleAna is organised in analysis steps. In section 12.5.1 some of the available analysis steps will be presented. The results can be illustrated in different ways. These are summed up in section 12.5.2.

12.5.1. Overview about Analysis Steps in TbTupleAna

The analysis steps in TbTupleAna are implemented individually from each other. Anyhow, all of these processors can influence the allocated event object, so that the order in which the analysis steps are called is important. The most important analysis steps available are:

ReadInAnalysis This processor is the first processor to be called. Its function is to read in the data from tbtrack files and pass it into the object oriented TB::Event structure.

Two processors of this kind are available. One is using the clustering information from the reconstruction to directly order the individual pixel hits into clusters, whereas the other one only passes pixel hits into the structure so that a further clustering algorithm needs to analyse the event again. This splitting was necessary, as the correct mapping between clusters and pixel hits was not possible when using tbtrack files reconstructed with older versions of the reconstruction software.

ClusterMakerAnalysis When using tbtrack files of the older generation the clustering needs to be done in a further step. The ClusterMakerAnalysis reorganises the TB::Event structure into the known structure with a possible mapping between clusters and pixel hits. The algorithm used here is a copy from the reconstruction software. Furthermore, the estimated hitposition on the sensor, by using the digital center of gravity of the cluster, is estimated and used as TB::HitCoord.

MatchingAnalysis The extrapolated hit position of the track traversing the sensor and the hit position of a cluster in the sensor are not connected to each other by default. The matching analysis searches for possible connections between these objects. The algorithm searches for a hit position of a cluster within an area surrounding each hit position of a track. When a connection is found, the position in which the track crossed the sensor and caused the cluster is identified with the precision of the telescope. Typical search limits are 1.5 times the pitch in x and y. When the alignment was not corrected by the PostShiftAnalysis/ApplyShiftsAnalysis these limits need to be set much higher, as they act as residual cuts in that case.

PostShiftAnalysis As mentioned in chapter 12.4.1 the alignment needs to be corrected after the reconstruction. This analysis creates a root file containing the alignment corrections for each sensor. The name was given as the processor only corrects the shifts and its running after the reconstruction.

ApplyShiftsAnalysis As known from chapter 12.4.1 the PostShiftAnalysis needs information from events happening after the event, which is currently investigated. As the event analysis in TbTupleAna is running successively over all events it needs to be called a second time after the alignment correction file was created. Because of this, TbTupleAna will be called the first time to create this root file. After the whole analysis over all events is completed it will be called again. The ApplyShiftsAnalysis will shift the track's position.

ReferencingAnalysis The sensitive integration time of the telescope is much higher than the integration time of the DUTs. Therefore, the number of tracks that will be reconstructed by the telescope will be higher than the number of tracks that were visible by the DUTs. The ReferencingAnalysis marks a track which should be visible for the DUTs. The algorithm used to differentiate the tracks uses a reference sensor. It is tested if the track is recognised by the reference sensor. A similar algorithm as for the matching is used here as well. For each DUT an individual reference sensor can be set. Furthermore, the search limits can be set individually from the MatchingAlgorithm. Additional cuts like maximum χ^2 of the track or limits on the LVL1 distribution of the reference sensor can be set.

MergeEventsAnalysis In chapter 12.4.3 it is mentioned that some runs could not be analysed easily, as the events were desynchronised. The MergeEventsAnalysis successively combines events in order to resynchronise the offsets.

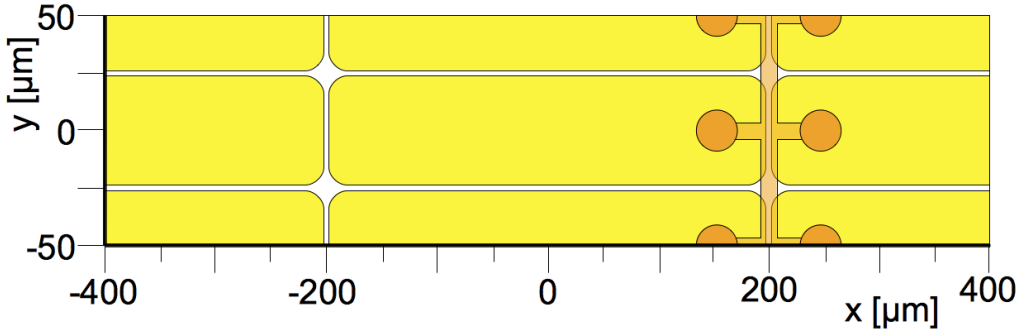


Figure 12.14.: Illustration to explain the sub pixels plots: 2×2 pixels of the DUT are shown, but arranged in a way, that one pixel is placed in the center. The two facing short pixel sides are shown with the bump bond contact on the left (at $-200\,\mu$m) and the bias grid network on the right hand side (at $200\,\mu$m)

HitEfficiencyAnalysis This processor generates all necessary plots to calculate the hit efficiency of each sensor. It loops over all tracks, which are referenced for this sensor, and checks whether the track was recognised or not.

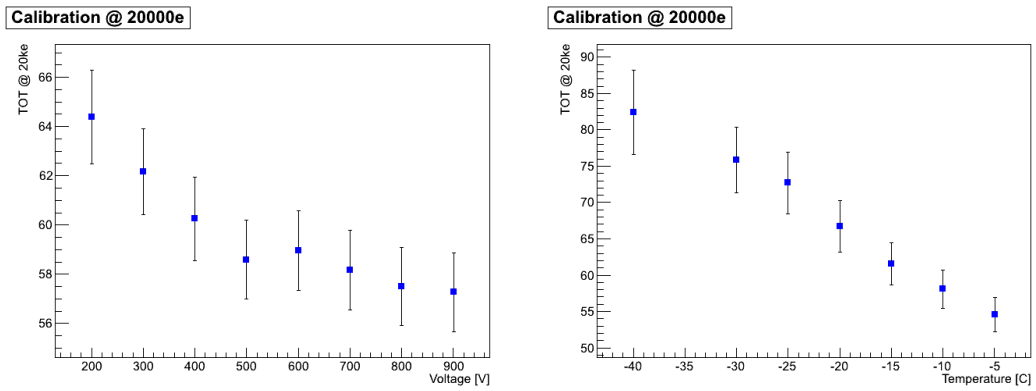
ChargeEfficiencyAnalysis The mean charge and also the mean TOT depending on the incident position in each sensor will be calculated in this processor. In order to allow a correct conversion of TOT into charges in units of electrons a charge conversion file needs to be provided.

12.5.2. Most important result illustrations

The main analysis steps which produce plots in order to acquire a quality statement of a sensor, a pixel, or a region of these are the HitEfficiencyAnalysis and the ChargeEfficiencyAnalysis. In both cases the illustrating plots show the same regions. The most important plots are:

Sensor Overview In this illustration a histogram represents the whole sensor. Each pixel corresponds to one bin. Therefore, it is useful to differentiate well-working pixels from bad ones. As the studies investigated here focus on the quality of the sensor and not the assembly made of sensor and FE electronics, it is common to ignore the content from obviously bad performing pixels. The reason that particular bad pixels occur is seen in bad bump bond connections between pixel and readout electronic or bad pixel cells in the FE.

Sub Pixel Overview By using the high precession tracking information of the telescope the exact traversing point through a pixel can be estimated. Therefore, a histogram representing one pixel with its surrounding pixels is common to illustrate properties of different pixel regions. An arrangement of such a histogram is shown in figure 12.14. The content of such sub pixel histograms typically consists of convoluted data collected from all pixels. As slightly different properties are expected in the edge pixels these are typically neglected. All pixels are equivalent to each other, as the



(a) dependency of the voltage (b) dependency of the temperature

Figure 12.15.: TOT response of an FE-I3 sample (DO9, irradiated to $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$). The number of injected electrons is constant at 20000. Investigated parameters are (a) the voltage and (b) the sensor temperature [82]

sensor producing masks have the same content. But because of the bias grid every second row is mirrored. This needs to be taken into account for the convolution.

Single Pixel Illustration This illustration uses the content of the same pixels than used for the sub pixel illustrations. But this time the center pixel appears as one single pixel in the whole sensor. In contrast to the sub pixel illustration the response of the pixel shown in the center with a track traversing beside the pixel range can be made visible. Therefore, charge sharing probabilities can be estimated, for example. A histogram showing charges or TOTs will show the pixel charge instead of the cluster charge.

12.6. Known limitations

In the following analysis of the testbeam data some limitations which lead to restrictions had to be taken into account. These are discussed here.

12.6.1. Uncertainty of the charge calculation

In [82] detailed studies of a fixed sensor tuning used in a modified sensor environment have been investigated. It is shown that the TOT-to-charge conversion depends on the applied sensor voltage and the sensor's environment. In the figures 12.15a and 12.15b the dependency on the voltage and the temperature is shown. The error bars indicate the σ of the gaussian distribution of the response from all pixels of the sensor.

When varying the voltage the maximum ΔTOT is roughly 7 TOT in the investigated bias voltage region. The changing of the sensor temperature has a larger influence: The maximum ΔTOT is 28 TOT.

For most of the testbeam periods only one tuning was used for all sensor voltages and temperatures and furthermore, if anything, only one TOT-to-charge calibration scan was

done for all changed parameters. The necessary scans had not been achieved, as this behaviour of the FE was not known at that point in time.

To roughly estimate the uncertainty coming from this effect, one can check the extent of the changes in temperature and bias voltage. The changing of the bias voltage for the irradiated sensors can be limited to roughly $\pm 600\,\mathrm{V}$. The changing of the temperature is harder to estimate. In order to obtain a gaussian σ value one could probably agree on the fact that most of all measurements had been achieved between -30 to $-10\,^{\circ}\mathrm{C}$. One obtains (only rough and conservative estimation):

$$\sigma_{\mathrm{U}} \approx 4\,\mathrm{TOT}, \tag{12.9}$$

$$\sigma_{\mathrm{T}} \approx 6\,\mathrm{TOT} \tag{12.10}$$

The combined uncertainty is $7.2\,\mathrm{TOT}$, which is about $1.2\,\mathrm{ke}^{-}$.

As the impact of this uncertainty is so very high[1] no direct TOT-to-charge conversion will be used in this thesis, by using the obtained fit values of the TOT scan. Instead of this only the measured TOT values will be given, which are still exact within their context.

During the commissioning of the ATLAS detector a linear approximation was used to do back-of-the-envelope estimations of charge from the TOT [46]. A linear function using the crossing points at $(20000\,\mathrm{e}^{-}, 30\,\mathrm{TOT})$ and $(0\,\mathrm{e}^{-}, 3.5\,\mathrm{TOT})$ was used to determine the charge. As the TOT is non-linear at low values this non-physical parameterisation[2] was chosen. The fit point at $(20000\,\mathrm{e}^{-}, 30\,\mathrm{TOT})$ was used, as the modules were tuned to this parameter.

For the testbeams done in this study another tuning was used (compare chapter 11). The first value needs to be changed with the modified tuning point: $(20000\,\mathrm{e}^{-}, 60\,\mathrm{TOT})$. As the threshold is reduced from $4000\,\mathrm{e}^{-}$ to $3200\,\mathrm{e}^{-}$ the second value will be moved to $(0\,\mathrm{e}^{-}, 4\,\mathrm{TOT})$. The resulting conversion function, which may only be used for rough estimations, is:

$$q(q_{\mathrm{TOT}}) = 0.36\,\frac{\mathrm{ke}^{-}}{\mathrm{TOT}} q_{\mathrm{TOT}} - 1.43\,\mathrm{ke}^{-}. \tag{12.11}$$

Especially at low TOT values the precision of this function is very limited.

12.6.2. Uncertainty of the bias voltage

Beside the dependency of the collected charge on the bias voltage the bias voltage itself is also only known within some limits. The reason for this is that a so called bias resistor is soldered in series to the sensor. The intended use was to protect the sensor in case of an accidental current breakdown.

The measured sensor bias voltage presented here is always the voltage applied to sensor and bias resistor. To obtain the sensor's bias voltage the voltage drop caused by the resistors needs to be subtracted. This drop is proportional to the current of the sensor and the value of the resistor. For all DUTs bias resistors of values between $100\,\mathrm{k}\Omega$ to $1\,\mathrm{M}\Omega$

[1] The uncertainty is in the same order of magnitude as the threshold

[2] Using this parameters at low TOTs means negative charges.

were used. Furthermore, the sensor currents depend highly on the temperature (compare chapter 3.2.2). For a typical bias resistor of 820 kΩ and a bias voltage of 1000 V a current of 200 μA is an average value for $2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$. The resulting voltage drop will be 160 V.

This effect will cause a systematic overestimation of the sensor bias voltage which will increase with higher fluences and higher voltages. A sensor irradiated to $1 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ has a typical current of 20μA, therefore the voltage drop will be decreased one order of magnitude.

12.6.3. Uncertainty of the fluence

Further important is the fact that the fluence to which the DUTs had been irradiated is not exactly known. The uncertainty of the fluence will not be higher than 5 to 10 % [4, 85].

13. Analysis and Interpretation of Testbeam data

Most runs of the testbeam data taken in all periods (compare chapter 11.1) were reconstructed and analysed. Because of many effects, such as these described in chapter 12.4, not all kinds of analysis were possible with all datasets. Therefore, the data quality needs to be checked for all runs. In chapter 13.1 a general analysis of the overall sensor hit- and charge efficiency of selected devices and fluences will be presented. Further on, a subpixel analysis is performed. A certain region in the sensor performs worse than expected and a modification of the sensor layout to improve this behaviour will be recommended in chapter 13.4.

In addition to that a sensor analysis at non-perpendicular incidence will be performed.

13.1. General sensor analysis

To investigate the hit efficiency and the collected charge as a general sensor property, available testbeam data from three different periods have been reconstructed and analysed. The data shown is a selected subset from all reconstructed data. Table 13.1 summarises all results into one common illustration. As an unirradiated reference two independent devices were taken. Furthermore, SCAs which were neutron irradiated to fluences of $\Phi_{\mathrm{eq}} = 1$, 2 and $20 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ were used to illustrate the behaviour of the sensors at the end of the lifetime of the IBL and a possible SLHC detector.

Detailed results of the individual sensor analyses can be achieved from the analysis plots shown in the figures C.1 to C.17 (page 196ff). In the upper left these plots show the residual distribution in x and y. A double fermi fit, as shown in equation (12.5) on page 115 is used to determine the gaussian fraction of the distribution. Its σ value is in the same order as the spacial resolution of the telescope at the investigated sensor plane. Further information in the upper part of the plot are the name and the fluence of the SCA and the testbeam period in which the corresponding data had been taken. The mean value of the hit efficiency is shown below. In addition to that, the time-dependent development of the hit efficiency is shown.

The rest of the sheet is divided into left and right. The left part shows information related to the hit efficiency and the right part information to the collected charge. A sensor overview is shown and also plots with subpixel resolution as explained in chapter 12.5.2. On the right hand side the TOT distribution of the clusters is shown with an additional landau fit.

As explained in chapter 12.4 a systematic inefficiency of the TurboDAQ readout system was detected in some runs. The subset of data used here was selected from a greater

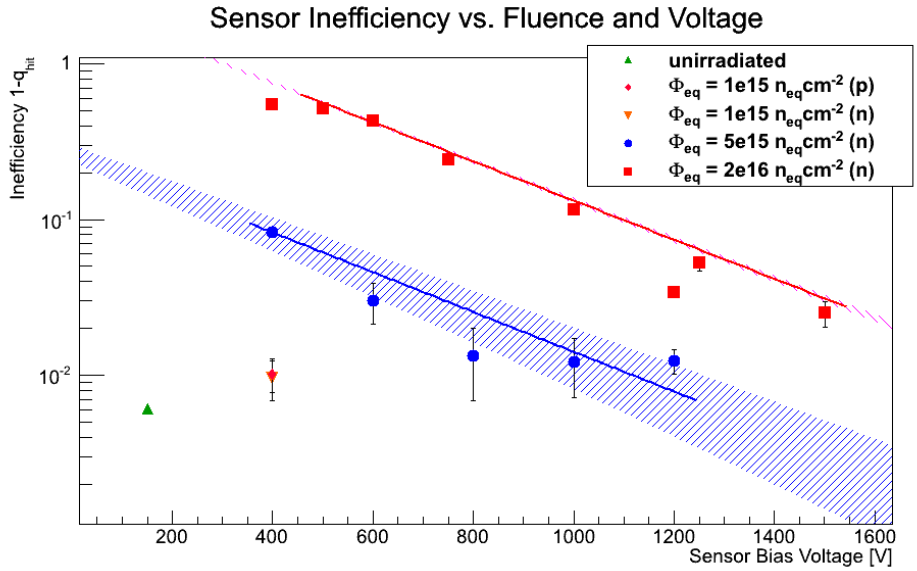


Figure 13.1.: Inefficiency $1-q_{\text{hit}}$ vs. bias voltage for $\Phi_{\text{eq}} = 1 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (orange), $\Phi_{\text{eq}} = 5 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (blue), $\Phi_{\text{eq}} = 2 \cdot 10^{16}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (red) and an unirradiated (green) device. The irradiation was done with neutrons

amount in order to reduce the systematic inefficiency to the most possible minimum. Criteria from explained sensor data sheets were used to select datasets.

For example, a stable behaviour of the hit efficiency was taken into account. Although the effect is minimal, completely inefficient pixels were masked in some cases in order to be ignored by the analysis. The maximum amount of pixels that are investigated are $14 \times 127 = 1778$. Therefore, the masking of one completely inefficient pixel raises the hit efficiency 0.056 % in average. In all cases the number of reference planes was exactly one. As the DUT and the chosen reference sensor were not completely overlapping (compare chapter 9.5) in all cases, the amount of investigated pixels might be smaller, as the further pixels cannot be referenced by the chosen sensor.

The values, summarised in table 13.1, are plotted against applied bias voltage. The developments of the hit efficiency and the collected charge are shown in the figures 13.1 and 13.2.

Instead of showing the hit efficiency q_{hit}, the hit inefficiency $1 - q_{\text{hit}}$ is plotted in figure 13.1, because this illustration shows more details at high efficiencies. The y-axis uses a logarithmic scale. The calculation of the statistical error was done by assuming a poisson distribution. In the graphic the error bars indicate the σ environment of the poisson error.

The collected charge values used in table 13.1 and figure 13.2 show the most probable value of the landau distribution. These values were taken from the landau fit in the mentioned sensor data sheets in the appendix. The error bars show the width σ of the landau distribution.

In the logarithmic plot in figure 13.1 the hit inefficiencies are arranged linearly. In particular for $\Phi_{\text{eq}} = 2 \cdot 10^{16}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ this behaviour is conspicuous. In an efficiency illustration this would mean that the efficiency approximates 100 % asymtotically, with a decaying

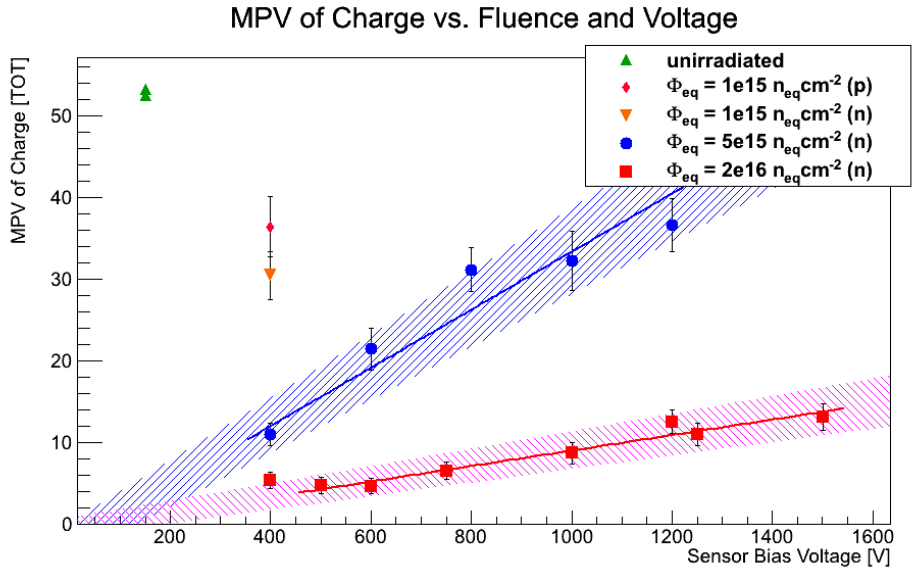


Figure 13.2.: Most probable value of the TOT distribution of the collected charge vs. bias voltage for $\Phi_{\mathrm{eq}} = 1 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ (orange) $\Phi_{\mathrm{eq}} = 5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ (blue), $\Phi_{\mathrm{eq}} = 2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$ (red) and an unirradiated (green) device. The irradiation was done with neutrons. The error bars indicate the σ values of the landau distribution

exponential slope. Two data points attract attention: Those at 400 and 1200 V, which stay outside the linear behaviour. For this fluence the sample used was always the same (DO10). But mentioned data points and one more (600 V) were recorded in another period with slightly different environmental behaviour. This explains the offset.

For $\Phi_{\mathrm{eq}} = 5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ data points from different SCAs and testbeam periods had to be taken. This explains why these do not fit to each other as well, as those for the other fluence did.

Assuming a very simple exponential model, the development of the hit efficiency can be fitted. One obtains:

$$q_{\mathrm{hit}}(U)\Big|_{\Phi_{\mathrm{eq}}=5\cdot10^{15}\,\mathrm{n_{eq}\,cm^{-2}}} = 1 - (0.269 \pm 0.034)\exp\left[(-2.94 \pm 0.30)\frac{1}{\mathrm{kV}} \cdot U\right] \tag{13.1}$$

$$q_{\mathrm{hit}}(U)\Big|_{\Phi_{\mathrm{eq}}=2\cdot10^{16}\,\mathrm{n_{eq}\,cm^{-2}}} = 1 - (2.388 \pm 0.037)\exp\left[(-2.893 \pm 0.026)\frac{1}{\mathrm{kV}} \cdot U\right] \tag{13.2}$$

The resulting uncertainty of these functions are visualized by the hatched areas in figure 13.1.

Unfortunately, only one data point is available with an SCA irradiated to a fluence of $\Phi_{\mathrm{eq}} = 1 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$. As the slope of both fits at the higher fluences is very similar, one can assume that only the constant factor is influenced by the irradiation and not the slope. As the error of the slope of the highest fluence is the smallest, it will be used to investigate the dependency of the fluence. In a very simple model, the constant factor

U [V]	Name	Period	Run number	q_{hit}	q_{TOT} [TOT]
$\mathbf{\Phi_{eq} = 0\,n_{eq}\,cm^{-2}}$**:**					
150	DO6	Jul 2010	11250 – 11282	0.9939 ± 0.0028	53.27 ± 4.56
150	DO3	Jul 2010	11250 – 11282	0.9940 ± 0.0020	52.60 ± 4.48
$\mathbf{\Phi_{eq} = 1 \cdot 10^{15}\,n_{eq}\,cm^{-2}\,(n)}$**:**					
400	DO8	Jul 2010	11250 – 11282	0.9904 ± 0.0027	30.49 ± 2.96
$\mathbf{\Phi_{eq} = 1 \cdot 10^{15}\,n_{eq}\,cm^{-2}\,(p)}$**:**					
400	DO7	Jul 2010	11250 – 11282	0.9898 ± 0.0025	36.47 ± 3.68
$\mathbf{\Phi_{eq} = 5 \cdot 10^{15}\,n_{eq}\,cm^{-2}\,(n)}$**:**					
400	DO9	Jul 2010	11250 – 11282	0.9169 ± 0.0022	11.02 ± 1.33
600	DO13	Oct 2010	20995 – 21016	0.9698 ± 0.0089	21.51 ± 2.54
800	DO9	Oct 2010	20860 – 20867	0.9866 ± 0.0065	31.20 ± 2.65
1000	DO9	Nov 2010	21481 – 21511	0.9878 ± 0.0050	32.29 ± 3.61
1200	DO9	Oct 2010	20869 – 20895	0.9876 ± 0.0023	36.63 ± 3.24
$\mathbf{\Phi_{eq} = 2 \cdot 10^{16}\,n_{eq}\,cm^{-2}\,(n)}$**:**					
400	DO10	Oct 2010	20860 – 20867	0.4545 ± 0.0028	5.42 ± 0.96
500	DO10	Nov 2010	21770 – 21795	0.4875 ± 0.0024	4.83 ± 1.01
600	DO10	Oct 2010	20668 – 20749	0.5676 ± 0.0009	4.70 ± 0.96
750	DO10	Nov 2010	21691 – 21715	0.7588 ± 0.0046	6.59 ± 1.11
1000	DO10	Nov 2010	21669 – 21690	0.8838 ± 0.0054	8.74 ± 1.26
1200	DO10	Oct 2010	20869 – 20895	0.9658 ± 0.0022	12.60 ± 1.41
1250	DO10	Nov 2010	21637 – 21664	0.9477 ± 0.0053	11.02 ± 1.41
1500	DO10	Nov 2010	21481 – 21511	0.9750 ± 0.0046	13.14 ± 1.65

Table 13.1.: Overview of different SCAs used to investigate their hit efficiency and the most probable value of their TOT distribution under varying voltages

can be set to be linear dependent on the fluence. One obtains:

$$q_{\text{hit}}(\Phi_{\text{eq}}, U) = 1 - \left[(-0.7700 \pm 0.090) + (1.567 \pm 0.070)10^{-16}\frac{\text{cm}^2}{\text{n}_{\text{eq}}}\Phi_{\text{eq}}\right] \exp\left[(-2.893 \pm 0.065)\frac{1}{\text{kV}} \cdot U\right] \tag{13.3}$$

In this assumption the datapoint of $\Phi_{\text{eq}} = 1 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ is used as well, and the same slope as for the highest fluence is presumed.

The efficiency decreases with higher fluence and lower voltage as the collected charge at the pixel implants decreases as well. This can be cross-checked by using the plot in figure 13.2. In addition to that the inefficiency is plotted as a function of the collected charge in figure 13.3.

Using the data points from table 13.1 the most probable values of the TOT distribution can be fitted:

$$q_{\mathrm{TOT}}(U)\big|_{\Phi_{\mathrm{eq}}=5\cdot10^{15}\,\mathrm{n_{eq}\,cm^{-2}}} = (-2.15 \pm 2.41)\,\mathrm{TOT} + (35.57 \pm 3.65)\,\frac{\mathrm{TOT}}{\mathrm{kV}}U \tag{13.4}$$

$$q_{\mathrm{TOT}}(U)\big|_{\Phi_{\mathrm{eq}}=2\cdot10^{16}\,\mathrm{n_{eq}\,cm^{-2}}} = (-0.43 \pm 1.28)\,\mathrm{TOT} + (9.49 \pm 1.40)\,\frac{\mathrm{TOT}}{\mathrm{kV}}U \tag{13.5}$$

A fit function crossing the origin is physically reasonable and compatible with the uncertainties. The resulting uncertainties of these functions are visualised in figure 13.2 by the hatched areas.

This shows that a higher applied voltage leads to more collected charge. That means that the TOT distribution will be shifted to higher values. When the sensor voltage is so little that parts from the left side of the landau distribution are not recognised because of the threshold the hit efficiency decreases. The closer the most probable value is to the threshold the higher is the impact to the hit efficiency, as the slope of the landau distribution is high there and with this the amount of charge lost below the threshold. Because of this the decrease of the hit efficiency has a very high impact at low TOT values. This can be seen in figure 13.3 as well:

The plot shows the hit efficiency in dependency of the collected TOT. In the logarithmic plots it becomes obvious that lower TOT values lead to an exponential drop of the efficiency. The number of data points shown here is very limited, because of this there is no clear result whether the shown dependency between hit efficiency and collected charge depends on the fluence as well. One tends to say that a shift of the TOT distribution has greater influence when using higher fluences.

As explained in chapter 12.6 the uncertainty of the collected charge is approximately $1.2\,\mathrm{ke^-}$. The roughly linear approximation model (12.11) will cause further uncertainties, especially at low TOT values. Therefore, the uncertainty will be doubled here in order to stay conservative. In addition to that, the bias voltage is systematically too high, due to a bias resistor in series. The bias voltage will be corrected by using common values (see the chapter mentioned). An uncertainty of 10 % will be added. Using these uncertainties and the linear approximation, the collected charges can be calculated to

$$\Phi_{\mathrm{eq}} = (5 \pm 0.5) \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}} \tag{13.6}$$

$$U_{\mathrm{bias}} = (1200 - 41 \pm 4)\,\mathrm{V} \tag{13.7}$$

$$\rightarrow q(\Phi_{\mathrm{eq}}, U_{\mathrm{bias}}) = (11.8 \pm 2.4)\,\mathrm{ke^-} \tag{13.8}$$

for the IBL fluence. And

$$\Phi_{\mathrm{eq}} = (2 \pm 0.2) \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}} \tag{13.9}$$

$$U_{\mathrm{bias}} = (1500 - 200 \pm 20)\,\mathrm{V} \tag{13.10}$$

$$\rightarrow q(\Phi_{\mathrm{eq}}, U_{\mathrm{bias}}) = (3.3 \pm 2.4)\,\mathrm{ke^-} \tag{13.11}$$

for the SLHC fluence.

As the hit efficiency for the value at the highest fluence is almost 100 %, this is a strong indication that the last charge value is underestimated, as this value is too close to the threshold which would usually cause an inefficiency.

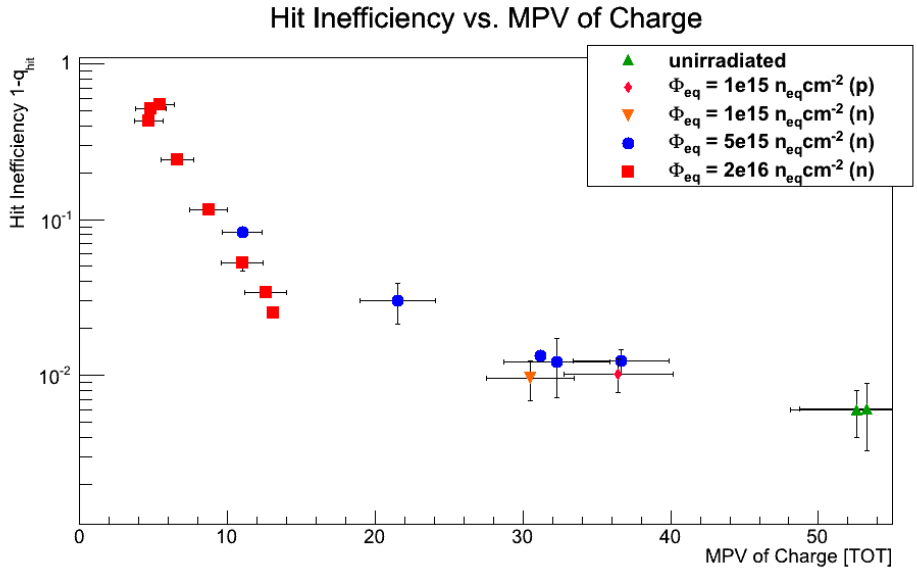


Figure 13.3.: Hit inefficiency $1 - q_{\text{hit}}$ vs. the most probable value of the TOT distribution for sensors irradiated to $\Phi_{\text{eq}} = 1 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (orange) $\Phi_{\text{eq}} = 5 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (blue), $\Phi_{\text{eq}} = 2 \cdot 10^{16}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (red) and an unirradiated (green) device. The irradiation was done with neutrons. The error bars indicate the poisson error (x) and the σ values of the landau distribution (y)

Comparing these results with the maximum estimated collected charge values by using the trapping model (compare table 5.1), it becomes obvious that the collected data shows charges higher than the predicted value. Especially at $5{\cdot}10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$, where the predicted charge collection value is $< 4\,\text{ke}^-$ a significant deviation can be seen. This is a strong indication for the charge amplification effect visible in planar pixel sensors as well.

For cross-checking, one can control if the estimated charge fits into the plots in figure 5.2. The data point for the IBL fluence is in perfect congruence with the plot shown in 5.2a. The data point for the SLHC fluence is at too low values, which can be explained by the fact that this value was said to be underestimated due to the linear approximation model before.

13.2. Subpixel sensor analysis

In contrast to the general sensor analysis (chapter 13.1) the results shown here will not summarise the global behaviour of a DUT neither in units of pixels nor in single values describing the properties of the whole sensor, but will show in detail the features of a pixel at its different positions.

As the surface of one pixel is three orders of magnitude smaller than the surface of the whole sensor, a special technique needs to be used to obtain enough statistics: The amount of data from all pixels of the sensor[1] is used and convoluted into one common histogram. As the orientation of pixels with an odd row is mirrored with respect to pixels with an

[1] Except pixels at the margin of the sensor and specially masked pixels

even row the estimated hit positions are mirrored as well. By using a factor of four more in statistics one can illustrate the behaviour of 2×2 pixels. This plot has the advantage that effects occurring in the region of the gap between two pixels can be illustrated in a better way. Figure 12.14 shows a plot illustrating the same amount of data. The only difference is that the plot is shifted half a pitch in each direction. In this way only one pixel is shown in the center surrounded by its direct neighbours (compare figure 12.14). When illustrating only one pixel the alignment needs to be sufficient enough with a resolution in the order of a bin width. Otherwise the mirroring of the pixel rows is done at a wrong position which will lead to a worse illustration. When the plot is done in the way as shown in figure 12.14 these effects do not appear. Therefore, this is the preferred illustration in this thesis.

The same datasets as used in chapter 13.1 are analysed here as well. In figure 13.4 plots as explained in figure 12.14 showing the space resolved subpixel hit efficiency $d^2q_{\mathrm{hit}}/dxdy$ are drawn for two unirradiated DUTs (DO6 and DO3) and two DUTs irradiated to $1 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ either with protons (DO7) or with neutrons (DO8). The bias voltages are 150 V for the unirradiated and 400 V for the irradiated devices. Further on, the horizontal and vertical projections of the histogram are shown above and beside in the figure. The histograms in figure 13.5 use only the horizontal projection of the hit efficiency dq_{hit}/dx, which is printed in blue. The same datasets are used here. In addition to that the space resolved mean charge dq_{TOT}/dx and mean cluster size in x dCS_x/dx are shown in red and green. As these values have different magnitudes they all use their own scale, which is shown in the corresponding colour.

When trying to interpret these histograms it is very important to take into account that the illustration strongly depends on the telescope's resolution. As the used datasets analysed here were obtained from different testbeam periods with different telescope geometries the resolution will not be comparable. Especially the z-position of the DUTs has a strong impact on the spatial resolution. As the available statistics was of varying amount for the different datasets the binning was changed between the datasets. But within one figure the individual scales are constant.

In case of the two unirradiated devices one can see that the hit efficiency does not depend on the incidence position. Its constant value is statistically compatible with 100 %. When looking at the mean collected charge the entries are constant within the area of the pixel, but slightly drop at $x = -200\,\mu$m and drop even more at $x = 200\,\mu$m (figure 13.5). The bump bond contact is located at the position on the left, whereas the bias grid is placed on the right. Especially in the upper most plot in figure 13.5 (DO6) one recognises that the drop has a more complex structure. This structure is caused by the bias dots, which can be proven when looking at the two-dimensional charge plot in figure C.1. The slight drop to the left is caused by the gap between two neighboured pixels. This can be shown by looking at the mean cluster size (green). As the cluster size rises, the probability that the charge will be collected by two independent pixels rises as well. The slight drop can be explained by two effects: At first, the fraction of charge collected in one pixel needs to cross the readout electronics threshold and the possibility that it will be neglected is higher for the charge fraction than for the complete amount of charge. Secondly, the TOT is not a linear value but is handled linearly because the TOTs of all pixels of a cluster are simply added to the cluster TOT shown in the illustration.

The sub structure in the bias grid region of the second unirradiated DUT is not visible

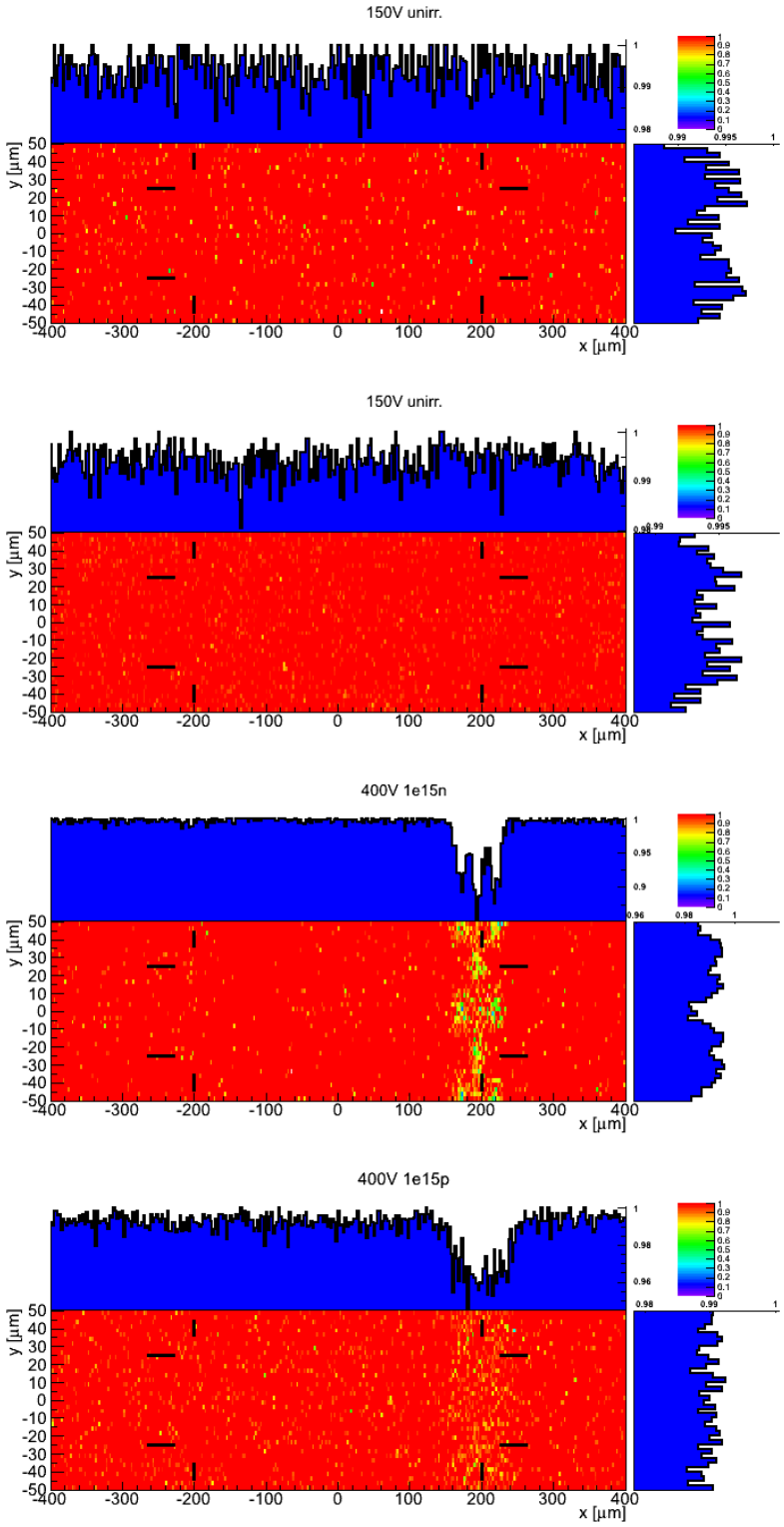


Figure 13.4.: Subpixel hit efficiency for unirradiated sensors (1) and (2) and $1 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons (3) and protons (4). The outer dimension of the center pixel is marked. The bias grid is at $x = 200\,\mu\mathrm{m}$. The horizontal and vertical projections of the histogram are shown as well

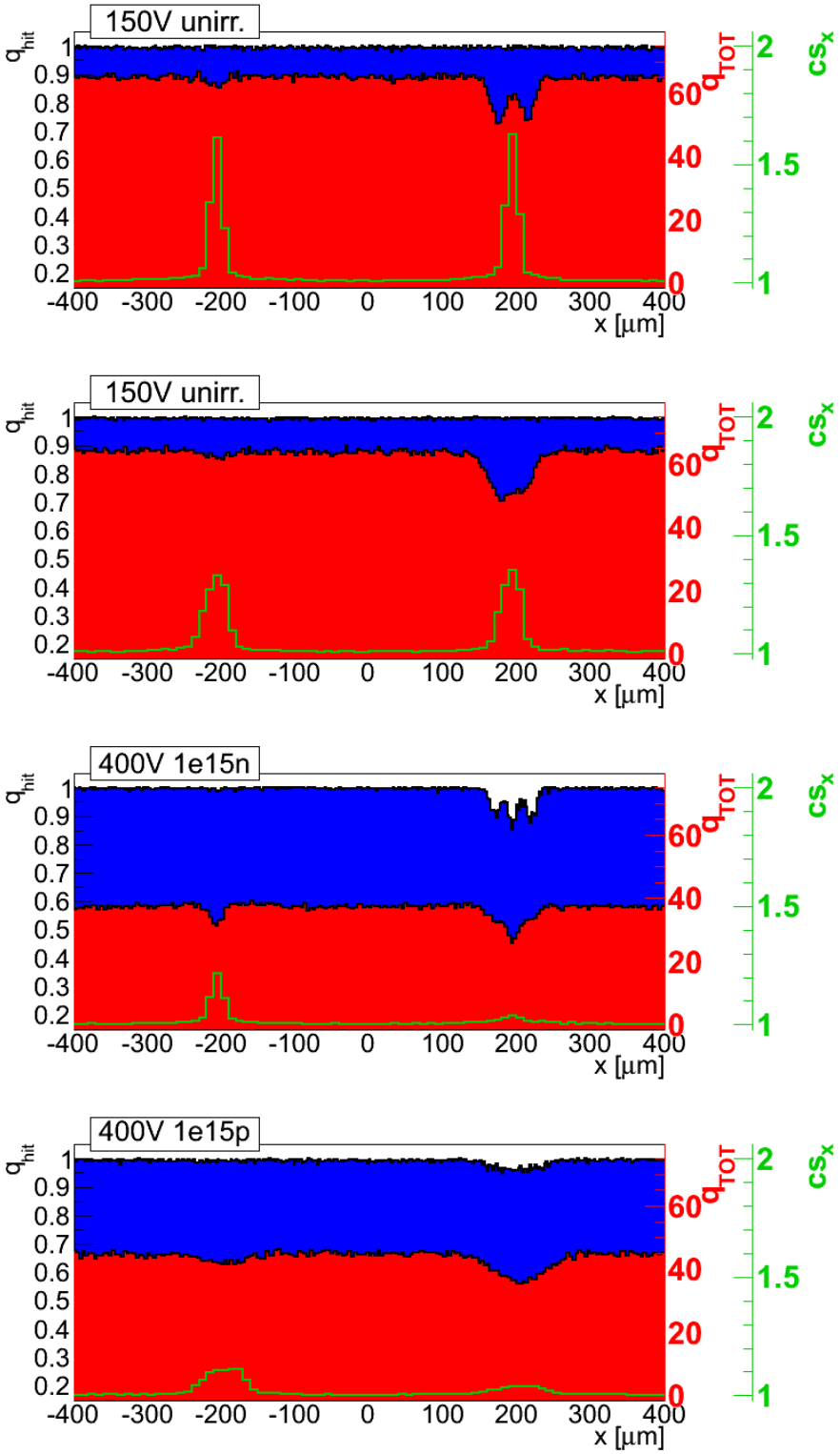


Figure 13.5.: Subpixel hit efficiency (blue), mean TOT (red) and the mean cluster size in x (green) as a function of the incidence point in x (different scales) for unirradiated sensors (1) and (2) and $1 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons (3) and protons (4). The outer dimension of one pixel is from $x = -200$ to $200\,\mu$m. The bias grid is at $x = 200\,\mu$m

because of the worse resolution of the telescope. In the figures C.1 and C.2 one can see that the resolution is $\sigma_x = 8.9\,\mu$m for the first device and $\sigma_x = 16.3\,\mu$m for the second. This also explains the wider cluster size distribution in the pixel gap for the second device.

The lowest fluence investigated is $1 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$. In comparison to the unirradiated devices one suddenly notices that the mean charge collected in the central part of the pixels is reduced to 37 TOT (unirradiated: 65 TOT). The drop of the mean collected charge in the region of the bump bond contact has increased to 8 TOT (unirradiated: 3 TOT). The telescope resolution for the neutron irradiated DUT was calculated to $\sigma_x = 9.3\,\mu$m and $\sigma_x = 31.5\,\mu$m for the proton irradiated DUT.

In the region of the bias grid the charge is reduced so much that the hit efficiency decreases as well. This means that the charge collected at the pixel implants, when a particle traverses through the bias grid, is not sufficient enough to cross the threshold in all cases. Very interesting is the fact that the sub structure caused by the bias dots is now visible in the hit efficiency but not in the collected charge. As the amount of data investigated for the charge efficiency is only the effective fraction of the hit efficiency (compare equations (12.1) and (12.2)) very inefficient areas will not contribute or only little to the charge efficiency. Therefore, the collected charge in the area of the bias dots is not visible anymore. In the two-dimensional plots (figure 13.4) one sees that the hit efficiency in the bias dot region is lower than 80 %. In addition to that the charge sharing in the bias dot region happens less often than in the bump bond region. This is also caused by the lost charge carriers which are collected by the bias dot implants.

As the resolution is very bad for the proton irradiated DUT, only a superficial analysis can be done here. An efficiency drop in the bias dot region can be confirmed as well, just as a lower amount of charge sharing in the same region. The mean collected charge in the center pixel area seems to be little higher than for the neutron irradiated sample (45 TOT).

Figures 13.6 and 13.7 show a voltage scan of DUTs irradiated to $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ arranged in the same way as done before in this chapter. In the two-dimensional illustration one can see that the gaps between the pixels cause efficiency drops especially when running at voltages (400 V) lower than the depletion voltage.

The efficiency drop in the bias grid region is increased in comparison to the fluence investigated before, but becomes less important with increased voltage. At voltages higher than 600 V the efficiency drop in the bump bond region is neglectable. Differences between these two regions can also be identified when investigating the cluster size. The mean cluster width in x is comparable with 1 in the bias grid region for almost all voltages. At voltages higher than 800 V the cluster size might be slightly increased. With much more significance the cluster size increases in the bump bond region. At 400 V the value is comparable with 1 and it reaches a maximum at 800 V (1.3 pixels). At higher voltages it becomes smaller again.

Although there are physical effects in the sensor which could explain this[1] behaviour, the more apparent cause is the changed telescope resolution. In the mentioned fig-

[1] With a higher bias voltage in the sensor the drifting time is reduced. Therefore, diffusion effects which cause a lateral movement of the charge carriers in the sensor have less time to take effect. The result is a smaller charge cone and less charge sharing.

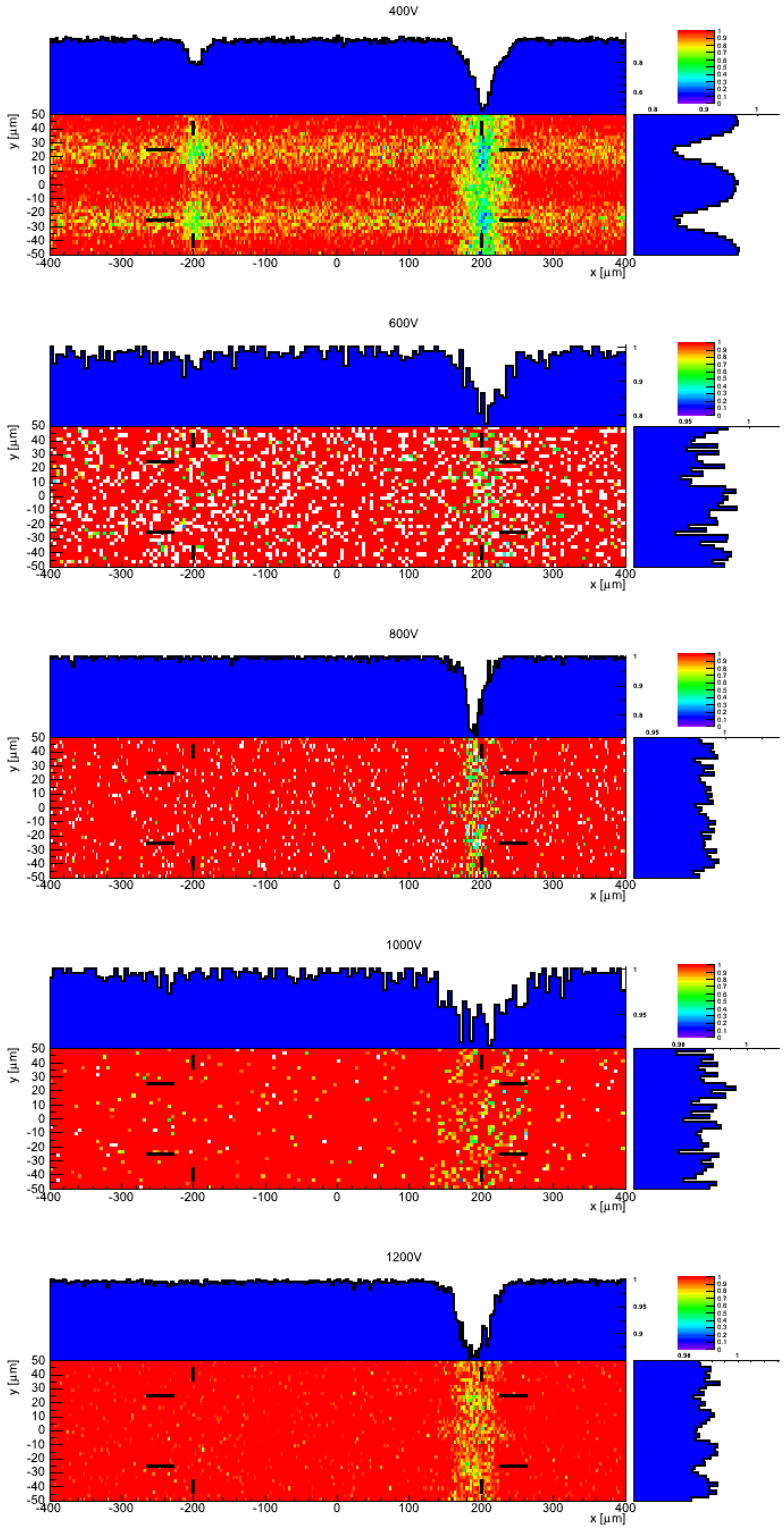


Figure 13.6.: Subpixel hit efficiency for sensors irradiated to $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons under varying voltages. The outer dimension of the center pixel is marked. The bias grid is at $x = 200\,\mu\mathrm{m}$. The horizontal and vertical projections of the histogram are shown as well

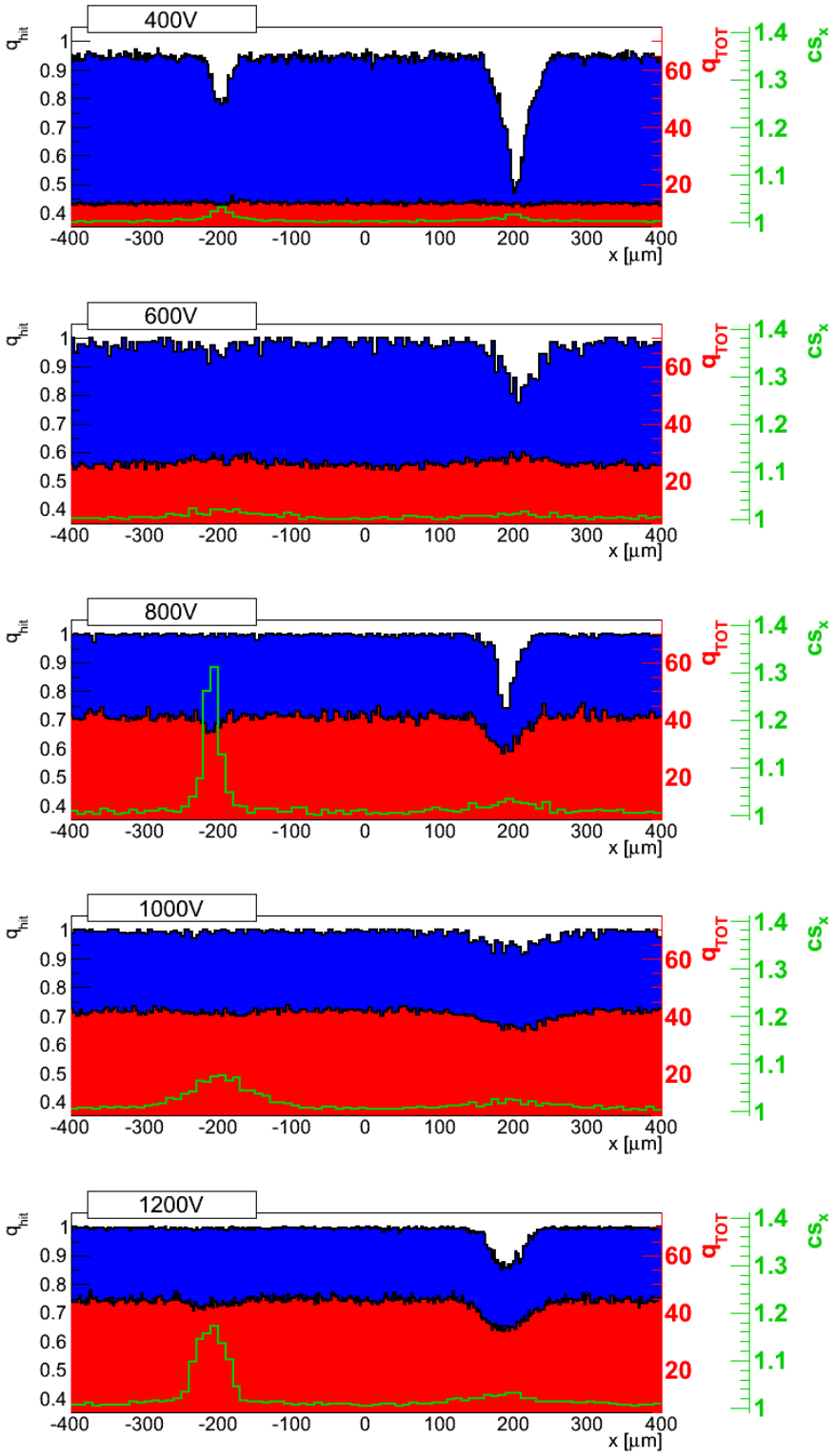


Figure 13.7.: Subpixel hit efficiency (blue), mean TOT (red) and the mean cluster size in x (green) as a function of the incidence point in x (different scales) for sensors irradiated to $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons under varying voltages. The outer dimension of one pixel is from $x = -200$ to $200\,\mu$m. The bias grid is at $x = 200\,\mu$m

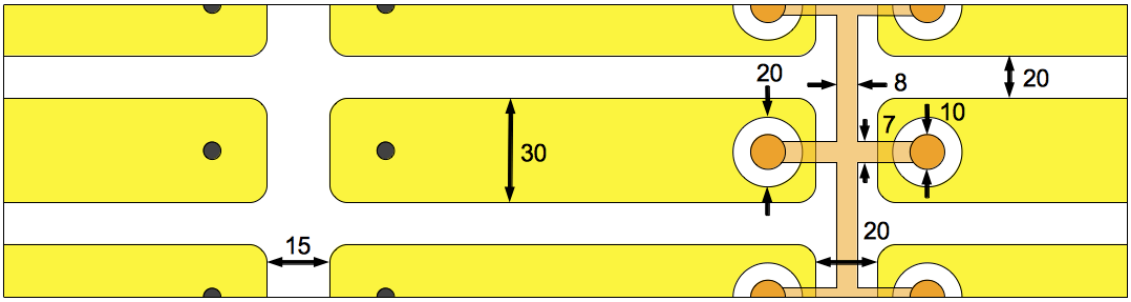


Figure 13.8.: Illustration of the pixel pattern of the ATLAS pixel sensor. The pixel n-implants, the bias grid network and all important distances are shown in units of μm

ures from $U_{\text{bias}} = 400\,\text{V}$ to $1200\,\text{V}$ the corresponding calculated resolutions are $\sigma_x = 13\,\mu\text{m}, 36\,\mu\text{m}, 12\,\mu\text{m}, 49\,\mu\text{m}$ and $24\,\mu\text{m}$ (compare figures C.5 to C.9).

When looking at the two-dimensional hit efficiency illustrations (figure 13.6) with sufficient resolution (for example 400, 800 and 1200, V) one sees that the less efficient areas are only limited to very small places in the pixel. At the bias grid edge of the pixel especially the corners are extremely inefficient. At 400 V this area is less efficient than 50 %. In comparison to the bump bond edge the collected charge is reduced significantly at the whole bias grid edge of the pixel. This feature is not only limited to the bias dots.

When looking at the dimension of a pixel implant and the gaps between the pixels (compare figure 13.8) in more detail one sees that the gap between two pixel implants at the bump bond side is $15\,\mu\text{m}$, whereas the gap at the bias grid side is $20\,\mu\text{m}$. So one possible explanation could be that charge is lost due to the higher distance of the n-implants. But as the gap between two pixel implants in y-direction is also $20\,\mu\text{m}$ this is easy to check: By using the 400 V dataset one can confirm that the resolution in both directions is comparable ($\sigma_x = 13.7\,\mu\text{m}, \sigma_y = 10.7\,\mu\text{m}$, compare figure C.5).

In the two-dimensional plot one can see that the left short pixel side ($15\,\mu\text{m}$ gap) has the fewest efficiency loss at the pixel gap ($\approx 5\,\%$). At the long pixel side ($20\,\mu\text{m}$ gap) the inefficiency is slightly larger ($\approx 10\,\%$). The highest amount of inefficiency is at the right short pixel side ($20\,\mu\text{m}$ gap), which also comprises the bias grid network. Here the efficiency drops down to $\approx 50\,\%$.

Because of this the greater implant distances cannot be the main cause for the lost charge in the region of the bias grid.

The interpretation is that the lower hit efficiency at the pixel's edges is caused by two independent effects. The one is charge sharing, which causes smaller signals per pixel which might not cross the threshold. A possible explanation for the other effect causing the heavy efficiency loss under the bias grid network is shown in figure 13.9. The illustration shows two very simplified pixel gaps. One with bias grid network and the other without. In figure (a) the potentials are printed, whereas possible very simplified streamlines of the field are drawn in figure (b). The potential of the bias grid network can be set to either ground or floating which would cause a potential difference of the punch-through voltage ($\approx 2\,\text{V}$). In both cases the potential of the bias grid network and the implants is almost the same. Due to this fact the streamlines of the electric field inside the sensor should be

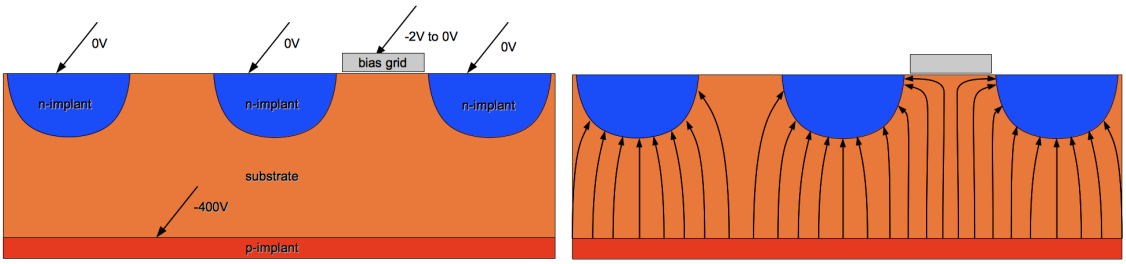


Figure 13.9.: Possible explanation for the low efficiency under the bias grid network. As the potential of the network is almost identical to the potential of the pixel n-implants, the charge carriers drift almost directly to the sensor surface under the bias grid network instead of being pulled directly to the implants

almost parallel under the bias grid network. As the substrate material between the pixel implants is highly resistive due to the p-spray implant (not shown), the streamlines of the field are not effected at the pixel gap shown on the left.

It is not possible to identify what exactly happens to a charge carrier drifting to the sensor surface below the bias grid network. The aluminum trace of the bias grid network is insulated from the sensor bulk with silicon oxid. Therefore, the coupling is AC. As not the charge carrier itself but only the image charge in the n-implants contributes to the charge signal, only the passage in which the charge carrier heads towards the implants - this means laterally and directly underneath the aluminum trace - will contribute to the signal. If this charge carrier, hidden in the gap, has the chance to drift laterally is impossible to say. It might be possible that the charge enters the oxid and is lost. A possible decoupling through the implants at time scales larger than the sensitive read out time is possible as well.

Another possible option [57] might be that the ionising dose in the silicon oxid caused by the irradiation causes conductive paths between the aluminum trace of the bias grid network and the sensor bulk.

A very easy improvement of the sensor geometry will be shown in chapter 13.4. A more detailed analysis of the inefficiency differences at the gap of the short pixel side and the bias grid network region is given in chapter 13.3.

In figures 13.10 and 13.11 the plots arranged in the same way as before are shown for sensors irradiated to $2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons. The plots show a voltage scan with values between 400 V and 1500 V. The telescope resolutions of the corresponding sensor planes for all voltages are listed in table 13.2. One can see that the resolutions are very poor in comparison to the DUTs irradiated to $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$. The reason for this is that

$\sigma_{x400\,\mathrm{V}}$	$\sigma_{x500\,\mathrm{V}}$	$\sigma_{x600\,\mathrm{V}}$	$\sigma_{x750\,\mathrm{V}}$	$\sigma_{x1000\,\mathrm{V}}$	$\sigma_{x1200\,\mathrm{V}}$	$\sigma_{x1250\,\mathrm{V}}$	$\sigma_{x1500\,\mathrm{V}}$
8 μm	39 μm	26 μm	41 μm	39 μm	20 μm	39 μm	42 μm

Table 13.2.: Telescope resolutions in x for the voltage scan at $2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$. The values were calculated by using a double fermi fit on the residual distributions in figures C.10 to C.17

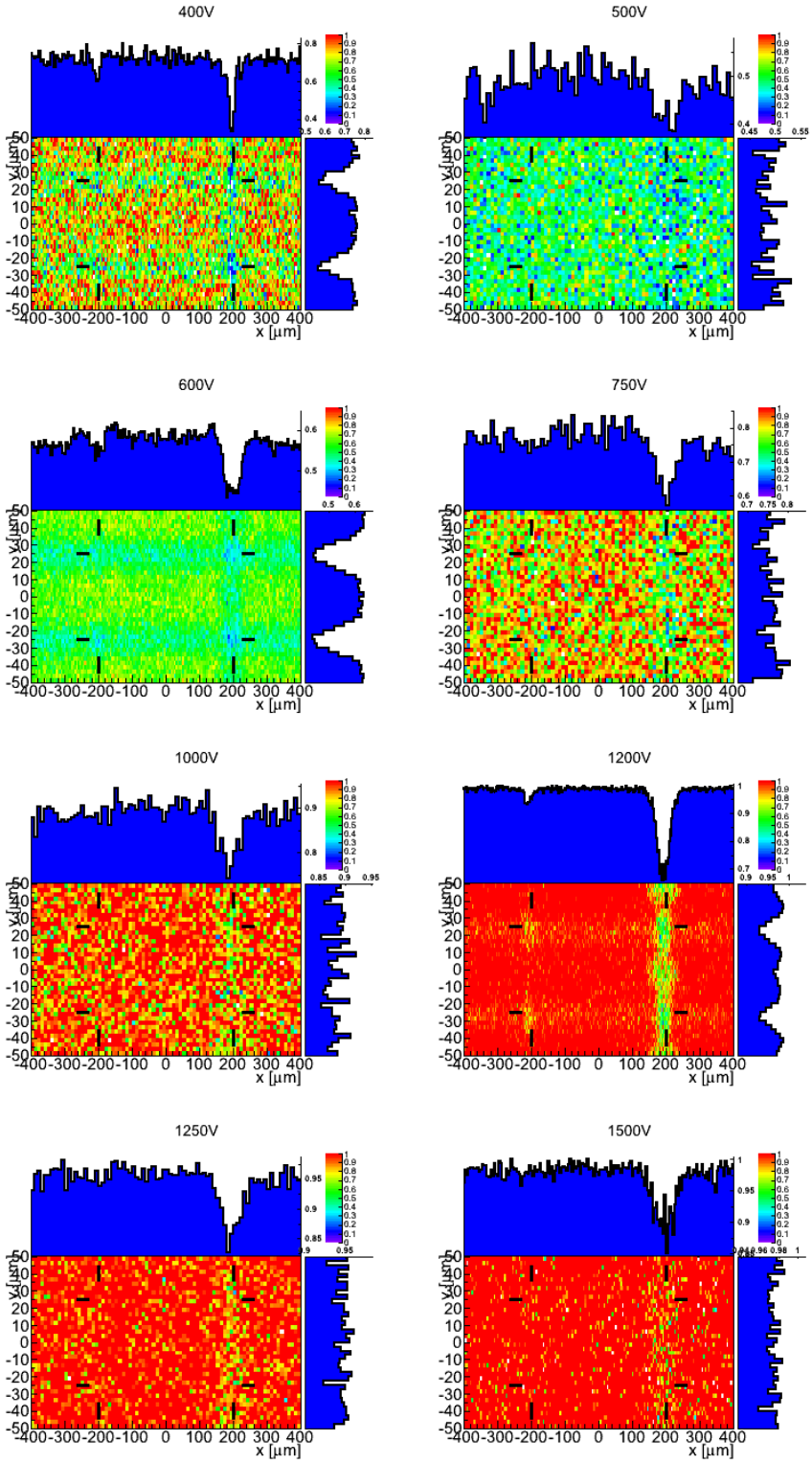


Figure 13.10.: Subpixel hit efficiency for sensors irradiated to $2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons under varying voltages. The outer dimension of the center pixel is marked. The bias grid is at $x = 200\,\mu\mathrm{m}$. The horizontal and vertical projections of the histogram are shown as well

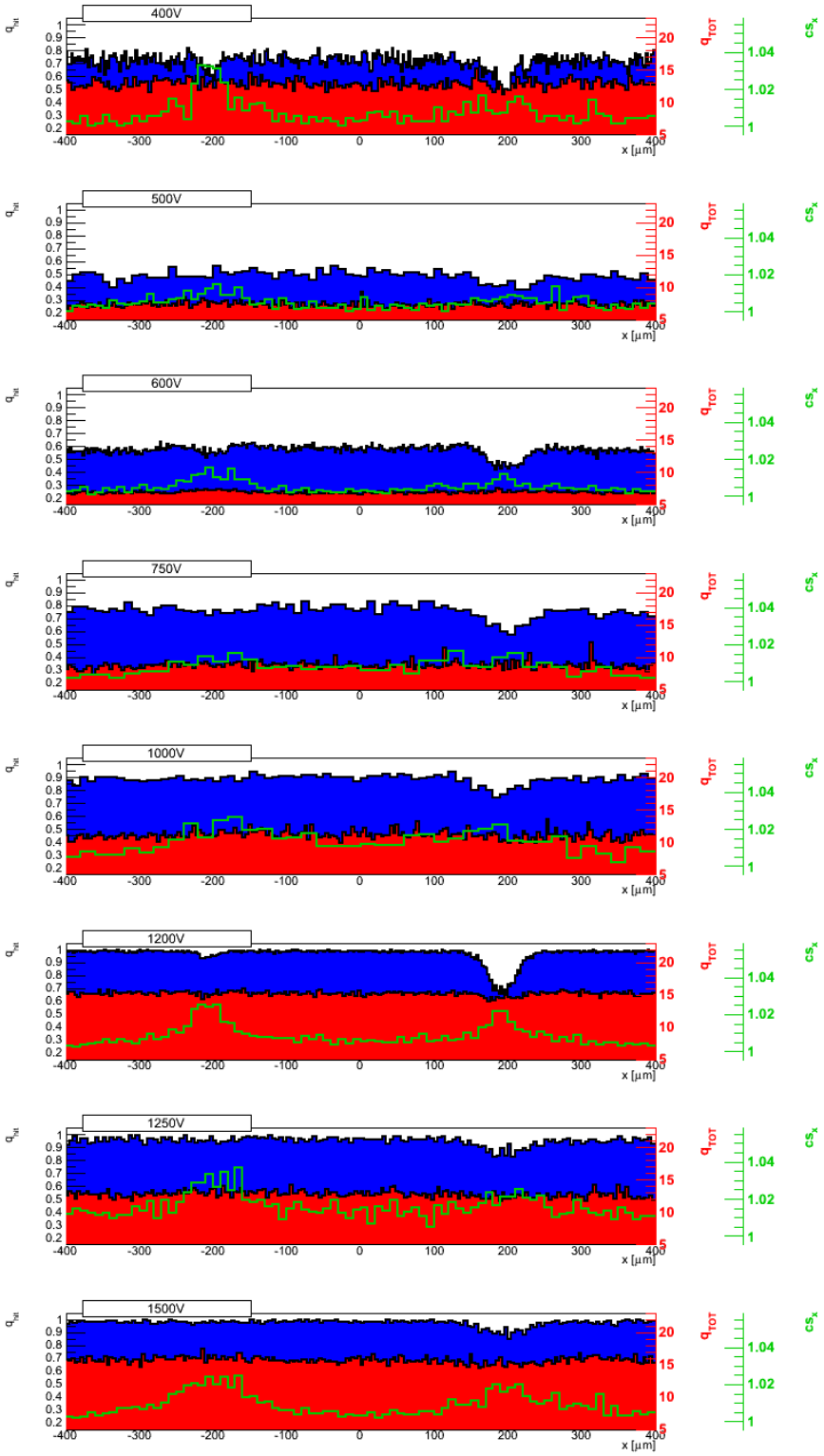


Figure 13.11.: Subpixel hit efficiency (blue), mean TOT (red) and the mean cluster size in x (green) as a function of the incidence point in x (different scales) for sensors irradiated to $2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons under varying voltages. The outer dimension of one pixel is from $x = -200$ to $200\,\mu$m. The bias grid is at $x = 200\,\mu$m

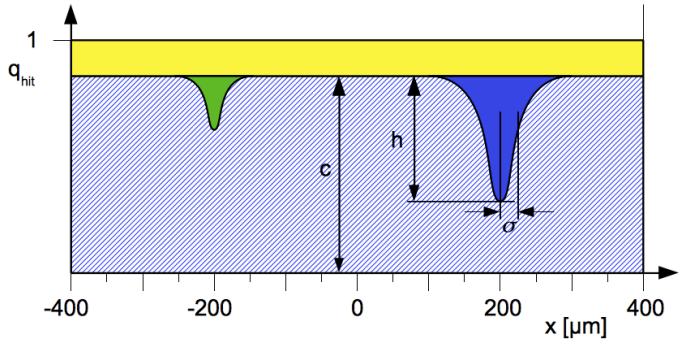


Figure 13.12.: Illustration showing the qualitative behaviour of the sub pixel efficiency as a projection to the long pixel side. The pixel gap at $x = -200\,\mu$m is located close to the bump bonds and the gap at $x = 200\,\mu$m is close to the bias grid

during the periods in which the runs with the bad resolutions were taken, only five out of six telescope planes were working properly. Unfortunately, the resolution is not sufficient enough to make the substructure of the bias grid network and the bias dots visible.

A few things strike out immediately: The collected charge at 400 V is higher than at, for example, 1000 V As there is no physical reason for this, the behaviour can only be caused by other environmental settings. Maybe the tuning of the device or the sensor's temperature were changed. This information could not be discovered from the shift logs. In comparison to the DUTs irradiated to $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$, the charge sharing seems to be reduced a lot. Where a mean cluster width in x of 1.3 pixels is common for $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ (see 800 V), the values for $2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$ are placed around 1.00 and 1.03 pixels.

Very interesting is the fact that the shape of the cluster width distribution at the pixel gap at the bump bond side and the gap at the bias grid network look very similar, whereas the effect is very different for $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$. This might be related to the decreased effective sensor zone. Due to trapping only a very thin area underneath the pixel implants will be effective. For further investigations of the bias grid area this information might be useful.

Looking at the different shapes of the sub pixel plots in the progress of the voltage scan, one sees that the inefficiency caused by the pixel gap to the left is neglectable for voltages higher than 1200 V. An efficiency gap at the bias grid region is still visible even at 1500 V.

13.3. Inefficiencies at the short pixel gap and the bias grid network

In the chapter 13.2 it was shown that the shape of the sub pixel hit efficiencies depends heavily on the telescope's resolution. Although the resolution is not sufficient enough to identify a substructure which might be caused by the bias grid network or the bias dots it is sufficient enough to quantify the degree of inefficiency in the dedicated regions.

In figure 13.12 a qualitative illustration is shown to distinguish three different parts of inefficiencies. Most of the sub pixel efficiencies investigated here look similar and a de-

scription of their behaviour is possible with the model shown in the figure mentioned. First, there is a fraction of inefficiency which is constant for all areas of the pixel (yellow). The physical cause for this fraction is caused by trapping. In addition to that particular areas on the pixel are more inefficient. Due to the limited tracking resolution of the telescope these areas appear gaussian shaped. The first area shown to the left (green) is caused by the gap on the short pixel side, at which the bump bond contacts are located. The drifting charge of a track traversing in this area will be individually collected by two pixels. As the charge is shared the threshold is effectively higher for these tracks. This is the interpreted reason for the efficiency loss in this region. The second area shown to the right (blue) is located at the position of the other short pixel gap. Furthermore, this area comprises the bias grid network which is an aluminum trace with a floating potential of roughly $-2\,\mathrm{V}$ AC coupled to the sensor. As already discussed in chapter 13.2 the exact mechanism is not completely understood. But a combination of the AC coupled aluminum trace and the charge sharing caused by the pixel gap might explain the greater magnitude of inefficiency in this region.

Further on, in this chapter the dimension of inefficiency in the mentioned areas and their relative fraction to the sensor inefficiency is investigated.

The x-projection of the sub pixel efficiency is used as a basis for this analysis. One uses a model as shown in figure 13.12 to quantify the amount of the related area. At each data point investigated in chapter 13.2 a gaussian function

$$q_{\mathrm{hit}} = c - h \cdot \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{13.12}$$

was fitted to both pixel gaps. By using these fits it is easy to calculate the area covered by the different regions as shown in figure 13.12. These are

$$1 - q_{\mathrm{hit}}|_{\mathrm{BG}} = \frac{\sqrt{2\pi}}{2p_x} h_{\mathrm{BG}} \sigma_{\mathrm{BG}}, \tag{13.13}$$

$$1 - q_{\mathrm{hit}}|_{\mathrm{BP}} = \frac{\sqrt{2\pi}}{2p_x} h_{\mathrm{BP}} \sigma_{\mathrm{BP}}, \tag{13.14}$$

$$1 - q_{\mathrm{hit}}|_{\mathrm{Const}} = 1 - c, \tag{13.15}$$

in which the indices show the region close to the bias grid network (BG), close to the bump pads (BP) and the constant part. Actually, the constant part was calculated twice: Once for each gap to obtain better fit results. So, for the calculation of the constant fraction the average value of both was used.

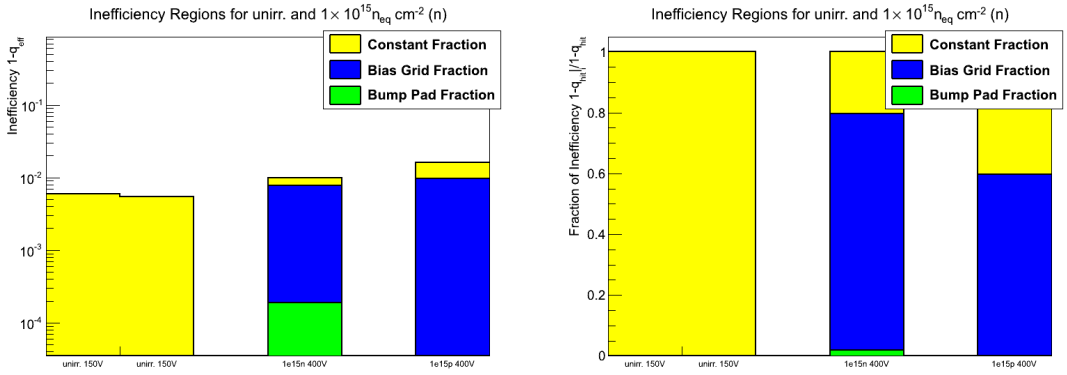
The tables 13.3 and 13.4 contain all necessary results. The fit values were shown on the left hand side and the calculated inefficiencies from these values are shown on the right. From the values in the brackets one can see the relative amount of this region in comparison to the total sensor inefficiency. In some cases the efficiency gaps could not be identified. This means that these regions are as efficient as the constant part. Therefore, the inefficiency of this region was set to zero with the calculated uncertainty of the constant part.

	gap at BPs	gap at BG	$1 - q_{\text{hit}}\vert_{\text{BP}}$	$1 - q_{\text{hit}}\vert_{\text{BG}}$	$1 - q_{\text{hit}}\vert_{\text{const}}$
$\Phi_{\text{eq}} = 0\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$, U = 150 V:					
c	$99.4 \pm 0.5\,\%$	$99.4 \pm 0.4\,\%$	$0.00 \pm 0.46\,\%$	$0.00 \pm 0.46\,\%$	$0.60 \pm 0.5\,\%$
h	-	-	$(0.0 \pm 0.5\,\%)$	$(0.0 \pm 0.5\,\%)$	$(100.0 \pm 0.5\,\%)$
σ	-	-			
$\Phi_{\text{eq}} = 0\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$, U = 150 V:					
c	$99.4 \pm 0.3\,\%$	$99.5 \pm 0.4\,\%$	$0.00 \pm 0.32\,\%$	$0.00 \pm 0.39\,\%$	$0.73 \pm 0.42\,\%$
h	-	-	$(0.0 \pm 0.3\,\%)$	$(0.0 \pm 0.4\,\%)$	$(100.0 \pm 0.3\,\%)$
σ	-	-			
$\Phi_{\text{eq}} = 1 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (n), U = 400 V:					
c	$99.7 \pm 0.5\,\%$	$99.9 \pm 0.6\,\%$	$0.02 \pm 0.13\,\%$	$0.76 \pm 0.16\,\%$	$0.25 \pm 0.48\,\%$
h	$0.6 \pm 1.9\,\%$	$9.9 \pm 1.4\,\%$	$(1.9 \pm 13.7\,\%)$	$(77.7 \pm 16.4\,\%)$	$(25.6 \pm 48.3\,\%)$
σ	$10.0 \pm 64.3\mu$m	$24.7 \pm 3.9\,\mu$m			
$\Phi_{\text{eq}} = 1 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (p), U = 400 V:					
c	$99.3 \pm 0.4\,\%$	$99.4 \pm 0.6\,\%$	$0.00 \pm 0.42\,\%$	$0.35 \pm 0.18\,\%$	$0.73 \pm 0.42\,\%$
h	-	$9.9 \pm 1.4\,\%$	$(0.0 \pm 0.4\,\%)$	$(34.5 \pm 17.4\,\%)$	$(72.5 \pm 41.9\,\%)$
σ	-	$31.0 \pm 11.7\,\mu$m			
$\Phi_{\text{eq}} = 5 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (n), U = 400 V:					
c	$94.5 \pm 0.4\,\%$	$94.2 \pm 0.5\,\%$	$0.55 \pm 0.08\,\%$	$2.13 \pm 0.09\,\%$	$5.46 \pm 0.41\,\%$
h	$16.8 \pm 1.4\,\%$	$43.1 \pm 0.9\,\%$	$(6.7 \pm 0.9\,\%)$	$(25.9 \pm 1.1\,\%)$	$(66.6 \pm 5.0\,\%)$
σ	$10.5 \pm 1.2\mu$m	$15.7 \pm 0.6\,\mu$m			
$\Phi_{\text{eq}} = 5 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (n), U = 600 V:					
c	$98.0 \pm 1.8\,\%$	$98.2 \pm 1.9\,\%$	$0.17 \pm 0.96\,\%$	$1.08 \pm 0.51\,\%$	$2.01 \pm 1.76\,\%$
h	$3.7 \pm 6.2\,\%$	$16.3 \pm 4.6\,\%$	$(5.6 \pm 32.0\%)$	$(35.9 \pm 17.1\,\%)$	$(66.8 \pm 58.7\,\%)$
σ	$14.7 \pm 79.6\mu$m	$21.1 \pm 8.1\,\mu$m			
$\Phi_{\text{eq}} = 5 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (n), U = 800 V:					
c	$99.7 \pm 0.8\,\%$	$99.4 \pm 0.8\,\%$	$0.01 \pm 0.33\,\%$	$0.85 \pm 0.18\,\%$	$0.33 \pm 0.80\,\%$
h	$0.3 \pm 10.6\,\%$	$25.1 \pm 3.0\,\%$	$(0.6 \pm 26.0\,\%)$	$(65.9 \pm 14.1\,\%)$	$(25.7 \pm 62.2\,\%)$
σ	$10.0 \pm 47.2\mu$m	$10.8 \pm 1.9\,\mu$m			
$\Phi_{\text{eq}} = 5 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (n), U = 1000 V:					
c	$99.4 \pm 0.8\,\%$	$99.4 \pm 1.5\,\%$	$0.00 \pm 0.83\,\%$	$0.71 \pm 0.47\,\%$	$0.62 \pm 0.83\,\%$
h	-	$5.8 \pm 2.3\,\%$	$(0.0 \pm 0.8\,\%)$	$(56.6 \pm 37.6\,\%)$	$(49.8 \pm 66.2\,\%)$
σ	-	$39.2 \pm 20.7\,\mu$m			
$\Phi_{\text{eq}} = 5 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$ (n), U = 1200 V:					
c	$99.6 \pm 0.4\,\%$	$99.6 \pm 0.5\,\%$	$0.01 \pm 0.12\,\%$	$0.85 \pm 0.12\,\%$	$0.43 \pm 0.41\,\%$
h	$0.2 \pm 3.5\,\%$	$14.0 \pm 1.3\,\%$	$(0.5 \pm 9.2\,\%)$	$(67.9 \pm 9.9\,\%)$	$(34.5 \pm 32.4\,\%)$
σ	$10.0 \pm 57.7\mu$m	$19.5 \pm 2.2\,\mu$m			

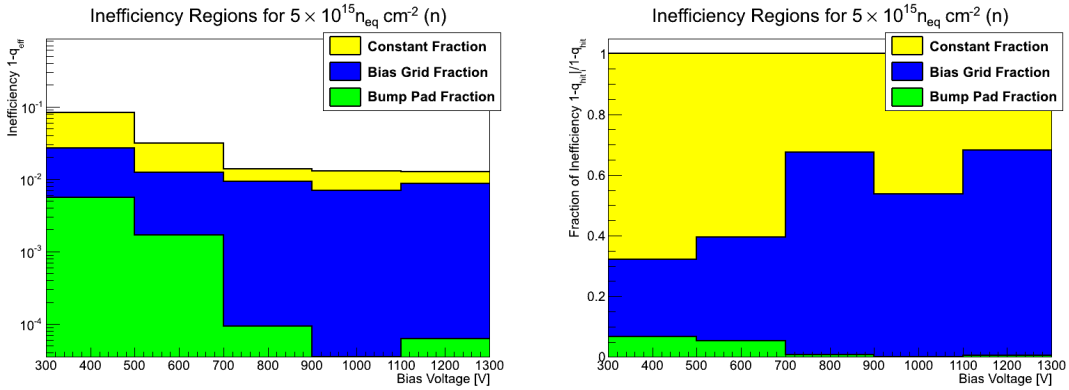
Table 13.3.: Values of the gaussian fit (c is the constant efficient part, h is the amplitude and σ the width of the gap) in the region of the bump bond pad (BP) and the bias grid (BG) for unirradiated sensors and sensors irradiated to 1 and $5 \cdot 10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$. To the right the inefficiency parts of the dedicated areas and their fractions (in brackets) of the whole sensor inefficiency are shown for all regions

	gap at BPs	gap at BG	$1 - q_{\text{hit}}\|_{\text{BP}}$	$1 - q_{\text{hit}}\|_{\text{BG}}$	$1 - q_{\text{hit}}\|_{\text{const}}$
$\mathbf{\Phi_{eq} = 2 \cdot 10^{16}\, n_{eq}\, cm^{-2}\ (n), U = 400\, V:}$					
c	$72.7 \pm 0.5\,\%$	$71.5 \pm 0.6\,\%$	$0.28 \pm 0.02\,\%$	$1.08 \pm 0.36\,\%$	$27.39 \pm 0.54\,\%$
h	$8.8 \pm 2.3\,\%$	$25.7 \pm 1.4\,\%$	$(0.9 \pm 0.7\,\%)$	$(3.7 \pm 0.4\,\%)$	$(94.0 \pm 1.9\,\%)$
σ	$10.0 \pm 6.9\mu$m	$13.4 \pm 1.1\,\mu$m			
$\mathbf{\Phi_{eq} = 2 \cdot 10^{16}\, n_{eq}\, cm^{-2}\ (n), U = 500\, V:}$					
c	$50.4 \pm 0.5\,\%$	$49.7 \pm 0.6\,\%$	$0.46 \pm 0.26\,\%$	$1.09 \pm 0.21\,\%$	$49.60 \pm 0.52\,\%$
h	$6.5 \pm 1.6\,\%$	$9.7 \pm 1.1\,\%$	$(0.9 \pm 0.5\,\%)$	$(2.1 \pm 0.4\,\%)$	$(96.6 \pm 1.0\,\%)$
σ	$22.5 \pm 11.4\mu$m	$35.9 \pm 5.5\,\mu$m			
$\mathbf{\Phi_{eq} = 2 \cdot 10^{16}\, n_{eq}\, cm^{-2}\ (n), U = 600\, V:}$					
c	$59.0 \pm 0.2\,\%$	$58.0 \pm 0.2\,\%$	$0.22 \pm 0.04\,\%$	$1.02 \pm 0.05\,\%$	$41.00 \pm 0.26\,\%$
h	$5.0 \pm 1.7\,\%$	$14.9 \pm 0.4\,\%$	$(0.5 \pm 0.1\,\%)$	$(2.4 \pm 0.1\,\%)$	$(94.8 \pm 0.6\,\%)$
σ	$14.0 \pm 1.8\mu$m	$21.9 \pm 0.7\,\mu$m			
$\mathbf{\Phi_{eq} = 2 \cdot 10^{16}\, n_{eq}\, cm^{-2}\ (n), U = 750\, V:}$					
c	$77.7 \pm 1.0\,\%$	$77.2 \pm 1.1\,\%$	$0.32 \pm 0.25\,\%$	$1.47 \pm 0.30\,\%$	$22.25 \pm 0.95\,\%$
h	$3.8 \pm 2.3\,\%$	$17.6 \pm 2.1\,\%$	$(1.3 \pm 1.0\,\%)$	$(6.1 \pm 1.2\,\%)$	$(91.8 \pm 3.9\,\%)$
σ	$26.5 \pm 13.0\mu$m	$26.8 \pm 4.4\,\mu$m			
$\mathbf{\Phi_{eq} = 2 \cdot 10^{16}\, n_{eq}\, cm^{-2}\ (n), U = 1000\, V:}$					
c	$90.6 \pm 1.7\,\%$	$89.4 \pm 1.2\,\%$	$0.26 \pm 0.37\,\%$	$1.07 \pm 0.35\,\%$	$9.36 \pm 1.65\,\%$
h	$2.9 \pm 2.8\,\%$	$13.0 \pm 2.6\,\%$	$(2.2 \pm 3.2\,\%)$	$(9.2 \pm 3.0\,\%)$	$(80.1 \pm 14.1\,\%)$
σ	$29.3 \pm 29.3\mu$m	$26.4 \pm 6.8\,\mu$m			
$\mathbf{\Phi_{eq} = 2 \cdot 10^{16}\, n_{eq}\, cm^{-2}\ (n), U = 1200\, V:}$					
c	$98.6 \pm 0.4\,\%$	$98.5 \pm 0.4\,\%$	$0.17 \pm 0.10\,\%$	$1.80 \pm 0.10\,\%$	$1.37 \pm 0.40\,\%$
h	$5.0 \pm 1.7\,\%$	$31.6 \pm 1.0\,\%$	$(5.1 \pm 2.9\,\%)$	$(53.2 \pm 3.0\,\%)$	$(40.4 \pm 11.8\,\%)$
σ	$11.0 \pm 5.1\mu$m	$18.2 \pm 0.8\,\mu$m			
$\mathbf{\Phi_{eq} = 2 \cdot 10^{16}\, n_{eq}\, cm^{-2}\ (n), U = 1250\, V:}$					
c	$96.0 \pm 0.9\,\%$	$95.4 \pm 1.2\,\%$	$0.06 \pm 0.46\,\%$	$0.94 \pm 0.37\,\%$	$4.0 \pm 0.91\,\%$
h	$1.9 \pm 3.6\,\%$	$10.6 \pm 2.6\,\%$	$(1.1 \pm 8.6\,\%)$	$(17.7 \pm 7.0\,\%)$	$(76.1 \pm 17.2\,\%)$
σ	$10.0 \pm 72.8\mu$m	$28.2 \pm 8.8\,\mu$m			
$\mathbf{\Phi_{eq} = 2 \cdot 10^{16}\, n_{eq}\, cm^{-2}\ (n), U = 1500\, V:}$					
c	$98.3 \pm 0.8\,\%$	$98.2 \pm 1.1\,\%$	$0.05 \pm 0.37\,\%$	$0.87 \pm 0.34\,\%$	$1.65 \pm 0.79\,\%$
h	$1.4 \pm 3.1\,\%$	$9.2 \pm 2.2\,\%$	$(1.8 \pm 14.4\,\%)$	$(33.7 \pm 13.1\,\%)$	$(64.1 \pm 30.6\,\%)$
σ	$10.0 \pm 79.2\mu$m	$30.2 \pm 9.1\,\mu$m			

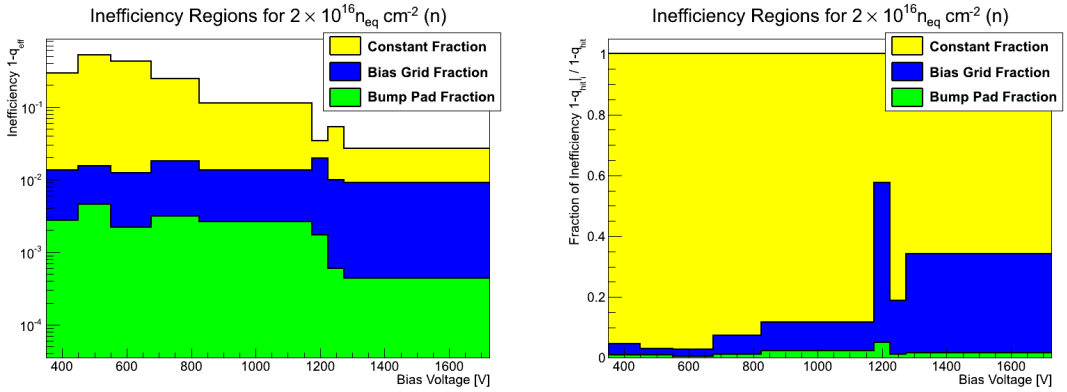
Table 13.4.: Values of the gaussian fit (c is the constant efficient part, h is the amplitude and σ the width of the gap) in the region of the bump bond pad (BP) and the bias grid (BG) for sensors irradiated to $2 \cdot 10^{16}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}$. To the right the inefficiency parts of the dedicated areas and their fractions (in brackets) of the whole sensor inefficiency are shown for all regions



(a) unirr. and $1 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$



(b) $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$



(c) $2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$

Figure 13.13.: Stacked histograms illustrating inefficiencies caused by the pixel gap at the bump pad region (green), the pixel gap at the bias grid region (blue) and a constant inefficiency of the whole sensor (yellow). The plots on the left show the absolute magnitude of inefficiency and those on the right show the respective relative fractions of the inefficiency

A better illustration of these results can be found in figure 13.13. The histograms shown in (b) and (c) contain the data from the voltage scan of the DUTs irradiated to 5 and $20 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$. In (a) a combination of the unirradiated sensors and those irradiated to $1 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ is shown. The histograms shown on the left, are stacked[1], so that the uppermost line shows the total sensor inefficiency again. The relative fractions of the inefficiency regions are plotted on the right hand side in the figure mentioned. As a linear scale is used here the fractions can be visualised in a better way.

Using the figures mentioned the plots can be analysed: As no inefficiency gap could be identified for the unirradiated devices only a contribution of the constant inefficiency is shown. When comparing these with the DUTs irradiated to the lowest fluence, one determines that the bump pad pixel edge's fraction of inefficiency is still vanishing, whereas the bias grid gap contributes the most (roughly 75% of the inefficiency).

A voltage scan is available for the DUTs irradiated to higher fluences. Looking at $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$, one can see that with higher voltage the inefficiency gap in the bump bond region decreases with the highest slope. The constant part also decreases. In contrast to this the contribution of the bias grid region seems to be very stable over a wide voltage range (600 to 1200 V). In this region the fraction of inefficiency is $1-q_{\mathrm{hit}}|_{\mathrm{BG}} = 0.85 \pm 0.10\,\%$. The only exception is the inefficiency at the data point of 400 V which is slightly higher ($2.13 \pm 0.09\,\%$). A possible explanation might be that indeed the inefficiency is caused by two effects, as mentioned before. The amount of inefficiency caused by charge sharing is neglectable for voltages higher than 800 V, which can be controlled by using this value of the other pixel gap. By considering the higher inefficiency at 400 V at the bias grid the whole concept of the two independent effects is even more consistent.

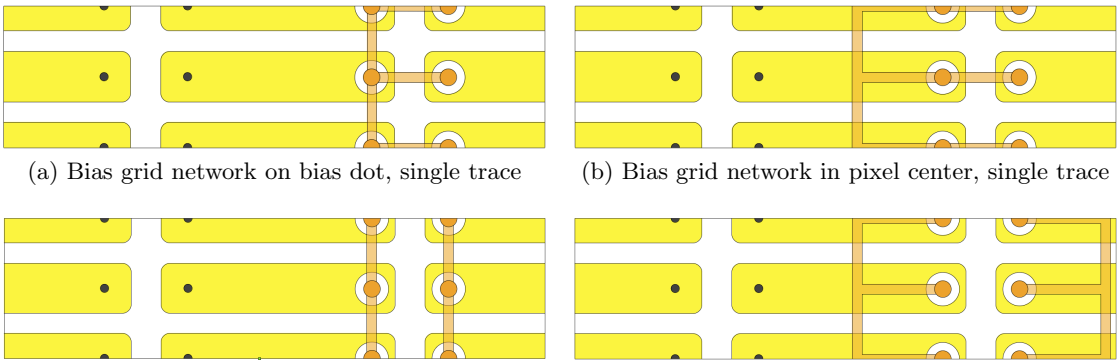
A similar behaviour as shown for $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ can be discovered when analysing the DUTs irradiated to $2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$. A special point at which the contribution of inefficiency caused by the bias grid stabilises cannot be estimated. For all voltages the inefficiency caused by the bump bond pixel gap is very low ($< 0.5\,\%$). By using a higher voltage an improvement can be seen here. A similar behaviour can be identified for the constant fraction. The data point at 400 V seems to contain a non-typical behaviour which was also detected before.

Remarkable is the fact that the sensor investigated at a voltage of 1200 V, which attracted attention before due to a too high sensor efficiency in contrast to other ones, has a much higher inefficiency in the bias grid region than common in the other datasets. In that case the bias grid's contribution to the global sensor inefficiency is higher than 50 %.

One important result of this study is the determination of a region causing an effect on the sensor, which is more inefficient than expected. In addition to this, the effect cannot be repaired by applying more voltage to the sensor. The most apparent explanation for this effect is the position of the bias grid network in exactly this region. The differences in the behavior of a pixel gap with a bias grid and without were analysed in detail to confirm these studies.

In chapter 13.4 a suggestion is proposed in which a changed bias grid layout could further improve the sensor inefficiency.

[1] Set on top of each other



(a) Bias grid network on bias dot, single trace

(b) Bias grid network in pixel center, single trace

(c) Bias grid network on bias dot, double trace

(d) Bias grid network in pixel center, double trace

Figure 13.14.: Changed bias grid network geometries as possible improvements for the sensor efficiency

13.4. Recommendation to improve the sensor with a changed layout

With the studies made here it was possible to suggest a change of the sensor layout. More precisely, it was determined that the location of the bias grid network directly on the pixel gap causes an inefficient region on the sensor which contributes more to the inefficiency than the other pixel gap without the bias grid network on top does. Furthermore, it was shown that the bias grid network's contribution to this effect cannot be repaired by applying more voltage to deplete the sensor.

In figure 13.14 four possible solutions for an improved bias grid network layout are presented. In the solutions (a) and (c) the aluminum trace is moved from the region between the pixels exactly on top of one of the bias dots. These solutions have the advantage that the length of the aluminum trace does not change in total. As the trace is not located over the gap anymore, the two effects causing less charge collection are separated from each other, which allows more collected charges to cross the threshold of the read out electronics. The only disadvantage is that the crosstalk between neighboured pixels under the bias grid trace might be increased as the AC coupling between these pixels is slightly higher than before. Solution (a) uses a single aluminum trace to connect a double column. Here the trace is still placed on top of the short pixel gap. By using two independent traces to connect the columns individually the trace on top of the pixel gap can be avoided. This is realised in solution (c).

In the original layout the aluminum trace was placed between two pixels. But as the pixels are also segmented in y, charge sharing happens not only within two pixels but also within four pixels. By moving the trace it does not overlap with a pixel corner anymore and therefore should increase the efficiency. The overlapping of the trace with the bias dot cannot be predicted by using the existing data. Principally, the bias dot is a separated pixel whose charge collection does not contribute to the read out signal.

In figure 13.14b and 13.14d even more enhanced layouts are shown. The trace is moved away from the region between the bias dots to the center of the pixel. In this layouts a possible mixing of the effects caused by the bias dot and the aluminum trace can be

prevented. One layout uses a single trace to connect a double column, the other individual traces. These sensor layouts should be used to investigate the behaviour of the bias grid network with the possibility of neglecting the effects of charge sharing. The longer aluminum traces will cause more crosstalk between the pixels.

As the n-implants of the pixels are geometrically located under the aluminum trace, the charge should be collected by the implant without being affected by the trace. Only in the region between the pixels (the pixel gap in y) the trace is directly AC coupled to the sensor bulk. To investigate the magnitude of the pixel gap in y the projection in this direction was investigated as well.

In order to assume a worst case situation the geometry shown in figure 13.14b is used: If the area in which the aluminum trace is not shielded from the bulk by n-implant is completely non-efficient the magnitude of this impact can be estimated:

$$A_{\text{unshielded}} = l_{\text{gap}} \cdot w_{\text{trace}} = 20\,\mu\text{m} \cdot 8\,\mu\text{m} = 160\,\mu\text{m}^2 \qquad A_{\text{pixel}} = 2 \cdot p_x \cdot p_y = 2 \cdot 50\,\mu\text{m} \cdot 400\,\mu\text{m} = 40000\,\mu\text{m}^2$$

The quotient of this values (0.4 %) is the impact of this worse case situation.

Sensors with changed bias grid layouts similar to those shown here were currently in production. Further testbeams with these DUTs are necessary to investigate the behaviour of the changed layout.

In addition to that, completely different concepts are under discussion as well. The most obvious idea is to omit the whole bias grid network and the bias dots. Unfortunately, this would prohibit the possibility to test the sensor without a connected read out electronics. Removing the bias grid network after a first testing of the sensor is also a possibility. But other concepts such as connecting the pixels with a high resistive polysilicon layer are in discussion as well.

13.5. Analysis at non-perpendicular incident angle

During the testbeam period of Nov 2010 at CERN SPS a DUT was mounted in the beam at an incidence angle of $\varphi = 30\,^\circ$. A rotation in this direction means that the sensor is tilted along the rows. This DUT, called DO20, is a special device because the bias grid is omitted. Table 13.5 gives an overview of the runs and their applied voltages. In addition to that an unirradiated reference device was also mounted in the same way.

In figures D.1 to D.5 on pages 214ff the sensor overview plots of the sensors at the varying voltages as used in the previous chapters can be found.

Very interesting about these kinds of studies is the fact that properties of the sensor happening in different depth regions can be made visible. For studies like this the single pixel hit efficiency illustration (compare chapter 12.5.2) is very useful. It shows the reaction of one pixel in dependency of different incidence positions. As the track traverses non-perpendicular a track can cross a pixel without traversing its surface at the pixel n-side (compare figure 13.16b). This kind of illustration for the reference device and DO20 at different bias voltages is provided on the left hand side of figure 13.15. The right hand

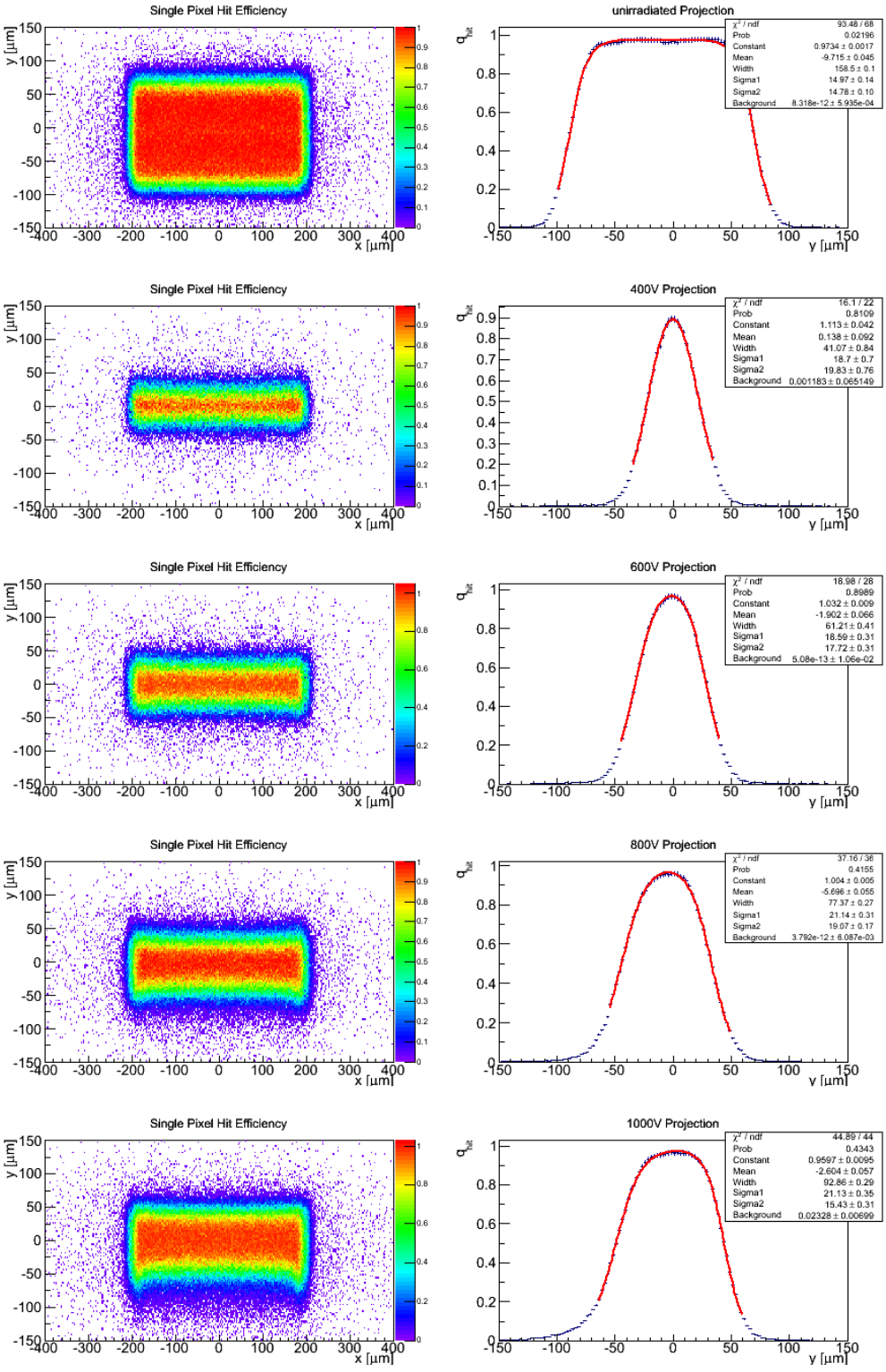


Figure 13.15.: Left: Single pixel hit efficiencies at 30 ° incidence angle. Right: The projection in y-direction (limited from -150 to $150\,\mu$m). The first row shows an unirradiated reference DUT. The others are DO20, irradiated to $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons with varying voltages. The data used for this study was taken in the Nov 2010 testbeam period at CERN SPS

U [V]	Run number	p_{eff} [μm]	d_{hit} [μm]
$\mathbf{\Phi_{eq} = 0\,n_{eq}\,cm^{-2}}$**:**			
150	22271 − 22284	158.52 ± 0.11	260.68 ± 0.17
$\mathbf{\Phi_{eq} = 5 \cdot 10^{15}\,n_{eq}\,cm^{-2}}$ **(n), DO20** :			
400	22413 − 22436	41.33 ± 1.02	57.73 ± 1.77
600	22356 − 22406	61.13 ± 0.41	92.03 ± 0.71
800	22311 − 22355	77.37 ± 0.27	120.16 ± 0.47
1000	22271 − 22284	92.86 ± 0.29	146.99 ± 0.50

Table 13.5.: Datasets used for the 30 ° incidence angle analysis. All data was taken in the testbeam period of Nov 2010 at CERN, SPS. p_{eff} is the effective pitch which is the width calculated from the histograms shown on the right hand side of figure 13.15. d_{hit} is the effective sensor depth using the threshold model explained in figure 13.17b

side of this figure shows the projection of the single pixel hit efficiency within the area of $x = -150\,\mu$m to $x = 150\,\mu$m.

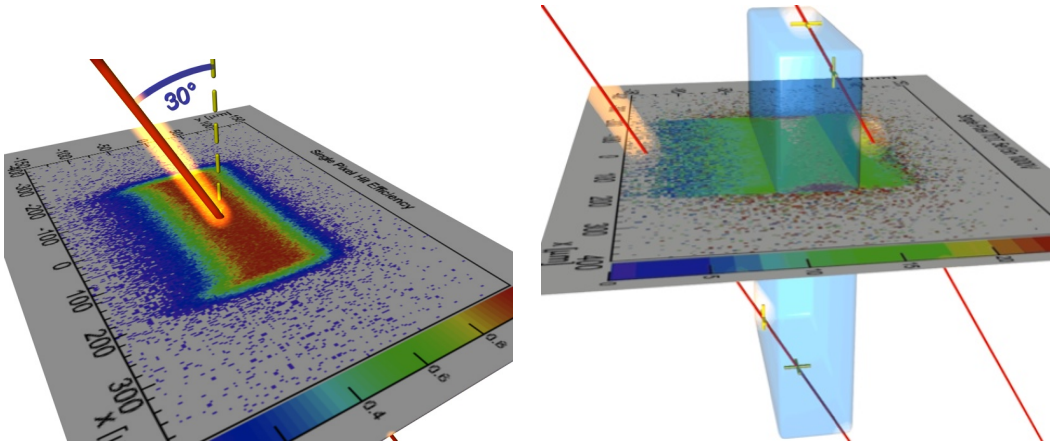
Looking at these illustrations the incidence angle of $\varphi = 30\,^\circ$ means that the track traverses the pixel in viewing direction in an 30 ° angle with respect to the perpendicular line starting at low y values towards higher y values. This is illustrated in figure 13.16a.

One suddenly recognises that the active pixel size in y in the single pixel hit efficiency illustration increases with higher voltage. At the unirradiated device this value is even greater. Furthermore, one detects frayings at the pixel edges, especially in the regions at which the track traverses the pixel only in distances very far away from the n-side. In order to neglect these frayings, first of all, a projection in y excluding the pixel edges is shown in the same figure. A slightly modified fit function as used before in equation (12.5) is used to fit these projections. The modification is that individual sides of the double fermi function obtain independent σ values, as the non-perpendicular incidence causes different shapes. Because of the non-perpendicular incidence the width p, which was compatible to the pixel pitch before, becomes p_{eff} now. The values which were fitted are shown at the corresponding datapoints in table 13.5.

In figure 13.17 modelised illustrations are given in order to convert the estimated effective pixel width p_{eff} into an efficient sensor thickness d_{hit}. In the simple model (figure 13.17a) the lines $\overline{A_1B_1}$ and $\overline{C_1D_1}$ show the maximum limits of a traversing particle, as a particle traversing outside this region does not touch the pixel. Of course, in this model diffusion and lateral drifts of the charge carriers are neglected. The lines $\overline{A_3B_3}$ and $\overline{C_3D_3}$ indicate the region in which the maximum charge can be collected as the number of charge carriers is proportional to the distance of the track inside the pixel. This distance is marked as i and depends on the incidence position. The lines $\overline{A_2B_2}$ and $\overline{C_2D_2}$ indicate the middle position of the maximum and minimum efficiency of the yellow pixel.

Using this model the conversion can be done:

$$d_{\text{hit}} = \frac{p_{\text{eff}}}{\tan\varphi}. \tag{13.16}$$



(a) Incidence angle (b) Single pixel TOT

Figure 13.16.: Computer aided illustration to explain (a) the incidence direction in the plots shown in figure 13.15, (b) the meaning of the single pixel TOT illustration in figure 13.19. The blue box indicates the 3-dimensional volume of a pixel

In this case p_{eff} corresponds to the width estimated from the fits in figure 13.15. The width is calculated by using the 50 % efficiency points.

Figure 13.17b shows a threshold model for this study. It is more precise than the simple model. The idea is that at the threshold of 3200 e^-, the hit efficiency is just 50 % because this is the definition of the threshold. It is marked by the lines $\overline{A_2B_2}$ and $\overline{C_2D_2}$. In comparison to the simple model p_{eff} becomes larger.

In 250 μm silicon a MIP produces roughly 19400 e^- [4, 85]. This means that a MIP needs to cross $s_{\text{min}} = 41.2\,\mu\text{m}$ in order to cross the threshold in 50 % of all cases. Because of this, the relative position of line $\overline{A_2B_2}$ in figure 13.17b with respect to the edge of the pixel is $x_{\text{th}} = 41.2\mu\text{m} \cdot \sin\varphi = 20,6\mu\text{m}$ at an incidence angle of $\varphi = 30\,°$. Assuming the same threshold behaviour at the sensor's front and backside d_{hit} in the threshold model can be calculated to

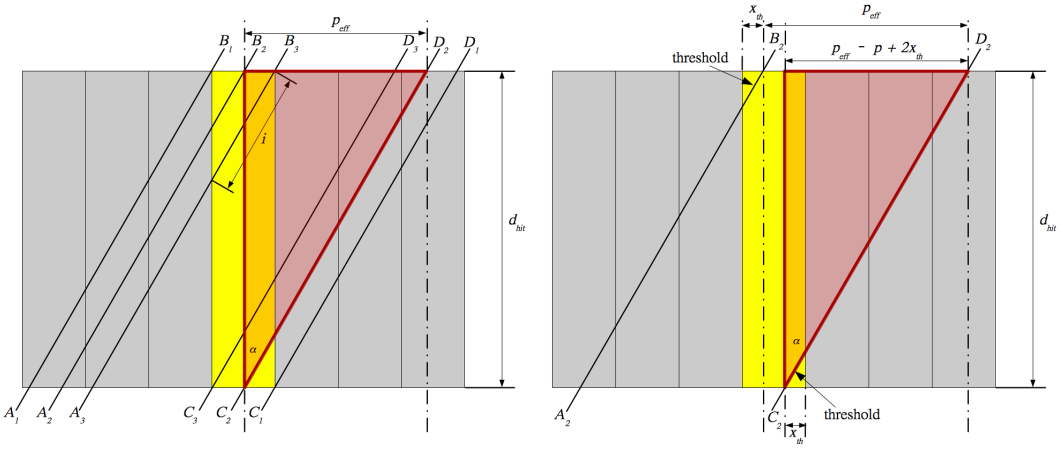
$$d_{\text{hit}} = \frac{p_{\text{eff}} - p + 2x_{\text{th}}}{\tan\varphi} \tag{13.17}$$

where p is the pixel's pitch.

The corresponding values calculated from the data points shown in figure 13.15 are listed in table 13.5. The unirradiated reference device shows an effective depth of $260.68 \pm 0.17\,\mu$m, which is compatible with the value of the sensor's thickness ($250 \pm 10\,\mu$m [4, 49, 84]).

Figure 13.18 shows a graphical illustration of the efficient depth d_{hit} in dependency of the bias voltage. One can see that all points of the irradiated DUT are arranged linearly. A linear fit using these data points gives an approximation of the effective depth of

$$d_{\text{hit}}(U)\Big|_{\Phi_{\text{eq}}=5\cdot10^{15}\,\text{n}_{\text{eq}}\,\text{cm}^{-2}} = (140.42 \pm 1.88)\,\frac{\mu\text{m}}{\text{kV}} \cdot U + (7.16 \pm 1.58)\,\mu\text{m}. \tag{13.18}$$



(a) Simple model (b) Threshold model

Figure 13.17.: Cross-sections through a pixelated sensor to explain non-perpendicular incidence models. (a) shows a very simple model with geometrical limits. (b) shows a theoretical threshold model

In the figure mentioned the maximum collection distance at saturation velocity (compare table 5.1) which is the theoretical limit for d_{hit} is shown in a dashed line. One can directly see that all data points are above this limit which is a clear indication that the simple trapping model explained in chapter 3.2.4 does not work anymore for the fluences used here. Even when using the simple model without the threshold correction the highest values are well above this line.

In order to get a better understanding of the frayings mentioned (compare figure 13.15), the same analysis as done in chapter 13.2 was applied on these datasets as well. The plots produced are visible on the left hand side of figure 13.19.

One sees that the cluster TOT of the unirradiated reference device breaks in little in the regions between two pixels. This corresponds to the increased charge sharing probability which is indicated by the higher cluster size values there.

At the irradiated DUT it becomes obvious that the cluster TOT is increased in the inter-pixel region which is extremely untypical and not discovered at perpendicular incidence. In comparison to the cluster size the TOT is increased roughly 25 % in this region, whereas the mean cluster size is only increased by 5 %. This means that a further effect independent from charge sharing increases the number of collected charge carriers in the inter-pixel region.

For a better understanding of this phenomenon, so called single pixel mean TOT plots have been arranged on the right hand side of figure 13.19. This challenging plot can be understood more precisely by using the explanation shown in figure 13.16b. The plot shows the collected TOT of one pixel in dependency of the incidence position of a track traversing a surface through the pixel. The track line shown to the left at very low y-values passes crosses only the very bottom region of the pixel - therefore, the charge is only little. The line to the right crosses only curtly the upper side of the pixel. The TOT response of the pixel will be filled at the position in the histogram where the track crosses its surface.

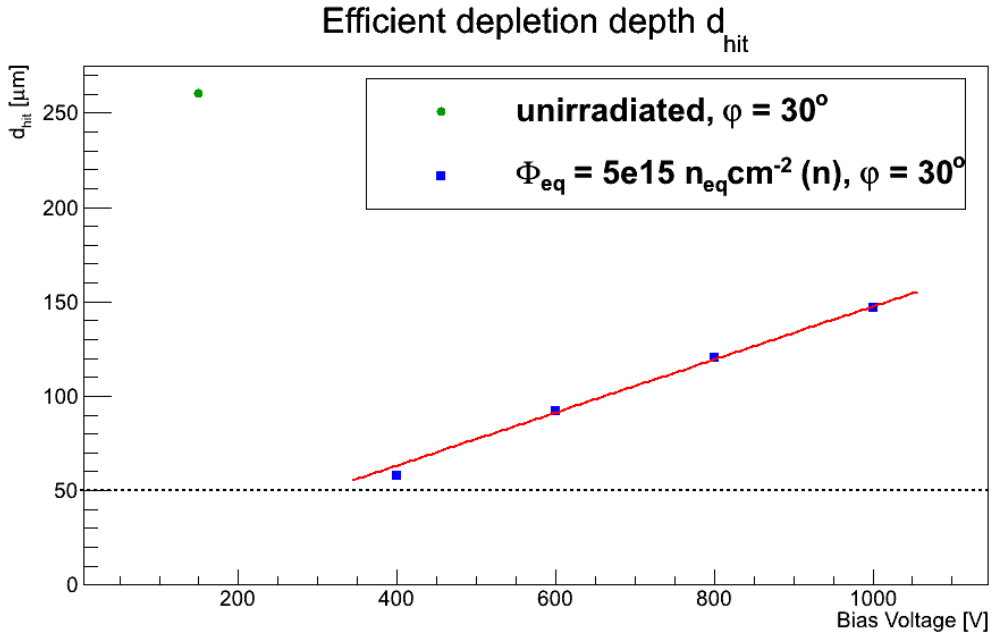


Figure 13.18.: Efficient depletion depth d_{hit}, of an unirradiated reference DUT (green) and DO20, irradiated to $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ (blue), calculated from the single pixel hit efficiency width at 30 ° incidence angle. The dashed line is the theoretical collection distance at saturation velocity for the corresponding fluence (compare table 5.1)

In difference to the hit efficiency the mean TOT does not express probabilities. For example, the frequent very high entries at the pixel sides ($x < -200\,\mu$m and $x > 200\,\mu$m) only seem to have a high impact. Looking at the hit efficiency one finds out that they appear only very seldom ($< 0.1\,\%$). Their appearance might be related to delta rays, SEUs or simple cross-talk.

One basically sees - especially at high voltages - that less charge is transported from the lower pixel regions. In the area in which the hit efficiency is lower than 50 % only 10 TOT or less is collected.

In addition to that, very high contributions can be recorded at the short pixel edges. When comparing these with the projections, they seem to be the cause for the mentioned higher TOT values. At this moment there is no clear explanation for their appearance. The only difference to the other regions is the fact that the field geometry here is slightly different due to the p-spray pixel isolation. The tracks traversing between the short pixel edges in a non-perpendicular incidence seem to produce more charge or propagate more charge to the pixels. As the single pixel mean TOT plots show the TOT values of single pixels and not the sum of the total cluster the effect cannot be related to charge sharing.

A relation between the higher TOT contributions at the short pixel edges and the frayings visible in the single pixel hit efficiency (figure 13.15), which make the pixel's hit efficiency appear bone-shaped, cannot be excluded and seem to be a reasonable explanation. The higher charge collection leads to a higher hit efficiency, as the threshold is crossed with a higher probability.

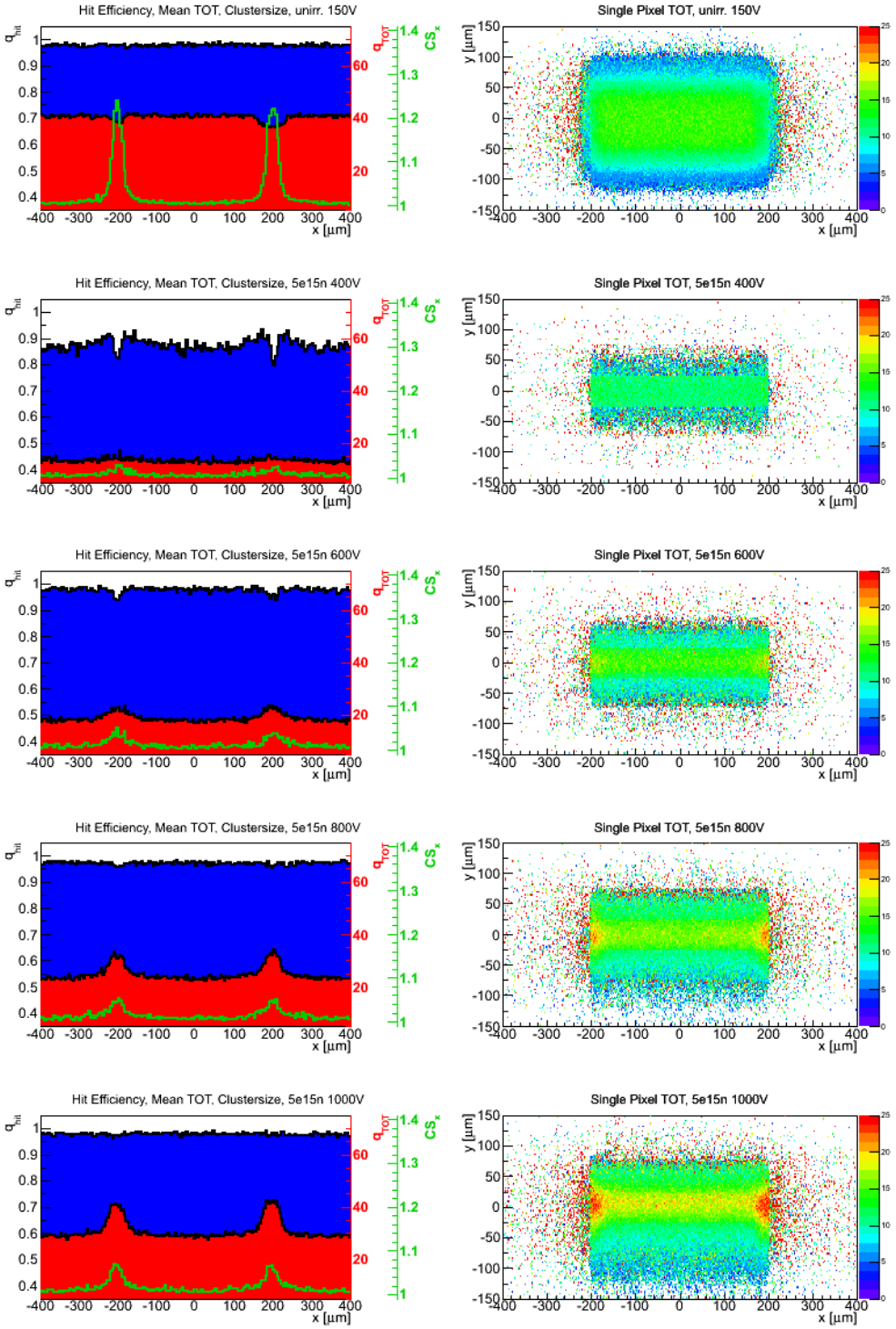


Figure 13.19.: Left: Subpixel hit efficiency (blue), mean TOT (red) and the mean cluster size in x (green) as a function of the incidence point in x (different scales) for an unirradiated sensor (1) and $5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons (2) to (5) at an incidence angle of $\varphi = 30\,^{\circ}$. Right: Mean TOT of a single pixel in dependency of the incidence position. As the incidence angle is $\varphi = 30\,^{\circ}$, an incidence point at lower y-values hits the pixel only in its lower region

14. Conclusions and Outlook

The ATLAS detector at CERN, LHC has been collecting data for three years. Upgrades for the detector are under way. The installation of a fourth pixel layer, called IBL, is scheduled for 2013. It raises the specifications in terms of radiation hardness, ineffective sensor edges and readout time. As a further upgrade a replacement of the pixel detector is under evaluation and a replacement of the whole inner detector is in preparation. For these future detectors a radiation hardness of all components up to $\Phi_{\mathrm{eq}} = 2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$ is required.

In contrast to the trapping model, which was used to design the current detector, charges were collected from silicon strip sensors irradiated to very high fluences, which are much higher than predicted. The detectors were operated at voltages beyond the limits possible in the current pixel detector. This effect, called charge amplification, has not been verified at pixel sensors so far but is a very welcome effect as it allows to operate the sensors at fluences higher than their design limits.

In order to allow the measurement of quality attributes of ATLAS pixel sensors, components for a fixed target testbeam experiment were designed, composed and tested. Existing components, which had been used for other purposes, have been adapted and extended to be able to interface the existing setup of the EUDET testbeam telescope.

Furthermore, mechanics used to mount the DUTs in the testbeam were developed. Particular for this mechanics is the possibility to operate the DUTs at very low temperatures ($\approx -40\,^{\circ}\mathrm{C}$), which is possible by using a dry ice cooling.

During the datataking an online analysis of the datastream is important to verify its quality. An online monitor was developed to provide this functionality. All relevant sensor data can be inspected in a graphical user interface. Furthermore, correlations between sensor planes are illustrated. By using these plots the relative position of the sensors can be controlled. As their interpretation is challenging, simulated correlations with changed environmental settings are presented in this thesis as hints for a later interpretation of real data.

Four testbeam periods were achieved in which data of unirradiated DUTs and DUTs irradiated to different fluences was recorded. These fluences correspond to the actual design limits of the current detector, an IBL detector and an SLHC detector.

Parts of the reconstruction chain for the telescope data, called Eutelescope, have been improved or completely redeveloped. In particular, a hot pixel killer, masking out very noisy pixels has been implemented as well as a clustering algorithm which processes the data a factor of ≈ 60 faster than the original algorithm. At last a processor writing out the reconstructed data into a common root file for a later offline analysis has been developed.

For the offline analysis which is going to be performed on the reconstructed data an analysis tool called TbTupleAna was developed within the root framework. Several analyses

steps, such as hit efficiency analysis or charge efficiency analysis, have been implemented and were used to analyse the data.

Several challenges which occurred during the analysis were identified and solutions were developed. These difficulties were related to different aspects of the testbeam setup or the environment of the testbeam. For example, the sublimating dry ice used for cooling caused a slow shifting of the DUTs which needed to be compensated.

A general sensor analysis was performed to estimate the global sensor efficiency and the collected charge in dependency of the bias voltage and the amount of irradiation. One important result is that all sensors (irradiated up to $\Phi_{\mathrm{eq}} = 2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$) can be operated at efficiencies higher than $97.5 \pm 0.5\,\%$. The very high efficiency is caused by the fact that the charge collected at sensors irradiated to $\Phi_{\mathrm{eq}} = 5$ and $20 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ is significantly higher than predicted by the trapping model. This is a strong indication for the charge amplification effect, which was only visible in silicon strip detectors before.

With the high resolution of the EUDET testbeam telescope a track reconstruction of significantly higher resolution than the pixel pitch of the DUT is possible. This makes sub pixel analyses possible. Analyses like these have been achieved with the result that especially one particular part of the pixel is less efficient. In this region, which contains the bias grid network, less charge is collected and therefore the pixel's threshold is crossed with less probability. Although the detailed physical origin of this effect is not completely understood, the part in the layout of the sensor causing this feature could be identified. Small changes in the sensor's geometry have been recommended, which might help to avoid this effect. They have been presented and sensors with such modified geometries are currently in production and will be analysed in further testbeams in the future.

Testbeam data with non-perpendicular incidence was analysed as well. These studies offer the possibility to expose effects which happen in different depths of the sensor. An equation to calculate the effective sensor thickness from this testbeam data has been set up and tested at an unirradiated reference device. At a DUT irradiated to $\Phi_{\mathrm{eq}} = 5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ the efficient sensor thickness in dependency of the bias voltage was analysed. In all cases an active sensor thickness higher than predicted by the trapping model has been identified. In addition to that, dedicated regions in the inter-pixel regions seem to collect higher charges than the other pixel areas. This effect is not understood and shows the necessity for further testbeams. A better understanding of this effect might help to improve the sensor's charge collection. Sensors with test patterns might amplify this effect and would verify it. Such devices are already produced but no testbeam data has been taken so far.

List of Tables

List of Figures

List of Acronyms

AC **A**lternating **C**urrent 21
AGND **A**nalog **G**rou**ND** 25
ALICE **A** **L**arge **I**on **C**ollider **E**xperiment 3
ASIC **A**pplication **S**pecific **I**nterated **C**ircuit 22
ATLAS **A** **T**oroidal **L**HC **A**pparatu**S** 3
BAT **B**onn **A**tlas **T**elescope 63
BBCB **B**lue **B**ox for **C**lock **B**uffering 61
BBLA **B**lue **B**ox **L**evel **A**dapter 61
BCID **B**unch **C**rossing **ID**entificator 29
BORE **B**egin **O**f **R**un **E**vent 75
CCK **C**ommand **C**loc**K** 28
CERN **C**onseil **E**uropéen pour la **R**echerche **N**ucléaire 3
CINT **C** and C++ **INT**erpreter 104
CK LHC bunch crossing **C**loc**K** 27
CMOS **C**omplementary **M**etal **O**xide **S**emiconductor 29
CMS **C**ompact **M**uon **S**olenoid 3
CNGS **C**ERN **N**eutrino Beam to **G**ran **S**asso 39
COMPASS **CO**mmon **M**uon and **P**roton **A**pparatus for **S**tructure and **S**pectroscopy 39
CPU **C**entral **P**rocessing **U**nit 79
DAC **D**igital to **A**nalog **C**onverter 23
DAF **D**eterministic **A**nnealing **F**ilter 112
DAO **D**at**A** **O**utput (FE-port) 29
DAQ **D**ata **AQ**uisition 38
DC **D**irect **C**urrent 19
DESY II **D**eutsches **E**lektronen **SY**nchrotron II - accelerator at DESY 42
DESY **D**eutsches **E**lektronen **SY**nchrotron 42
DGND **D**igital **G**rou**ND** 25
DICE **D**ual **I**nterlock storage **CE**ll 25
DLHC **D**ouble Energy **LHC** 5
DLL **D**ynamic **L**inker **L**ibrary 74
DOBOX-2 **DO**rtmund testbeam **BOX** second generation 51
DOBOX-3 **DO**rtmund testbeam **BOX** third generation 53
DOBOX **DO**rtmund testbeam **BOX** 49
DORIS German: **DO**ppel **RI**ng **S**peicher - double-storage-ring - accelerator at DESY 42
DSM **D**eep **S**ub **M**icron 22
DTI0 **D**a**T**a **I**nput - channel **0** (FE-port) 28
DTI15 **D**a**T**a **I**nput - channel **15** (FE-port) 28
DTI **D**a**T**a **I**nput (module-port) 27

List of References

[1] A. Affolder. **Presentation at ATLAS PPS-Meeting: Enhanced efficiency of segmented silicon detectors of various thicknesses after hadron irradiations up to** $2.2 \cdot 10^{16}\,\mathbf{n_{eq}\,cm^{-2}}$. University of Liverpool, 2009.

[2] Personal correspondance with **Alessandro LaRosa, CERN**.

[3] André Rummler. **PhD-Thesis: Investigation of radiation hard n^+-in-n planar pixel sensors irradiated up to HL-LHC fluences using lab and testbeam measurements**. Universität Dortmund, Germany, in preparation.

[4] Personal correspondance with **André Rummler, TU Dortmund**.

[5] ATLAS Colaboration. **Insertible B-Layer Technical Design Report**, volume ATLAS TDR 19, CERN/LHCC 10-xx. CERN, 2010.

[6] ATLAS Collaboration. Public webpages of the ATLAS collaboration. Accessible at: `http://atlas.ch`.

[7] ATLAS Collaboration. **Liquid Argon Calorimeter Technical Design Report**. CERN, 1996, CERN/LHCC/96-41.

[8] ATLAS Collaboration. **Barrel Toroid Technical Design Report**. CERN, 1997, CERN/LHCC/97-19.

[9] ATLAS Collaboration. **End-Cap Toroids Technical Design Report**. CERN, 1997, CERN/LHCC/97-20.

[10] R. Beccherle. MCC-I2 Design Summary. In **ATL-IP-EM-0115, Geneva**. ATLAS Pixel Week Jun. 2003, 17.06.03„ 2003.

[11] L. Blanquart. **Design improvements for the FEI2 Engineering Run**. AT-IP-EM-0084, Marseille, 2002.

[12] A. Borowski. **Diploma thesis: Verarmungseigenschaften des ATLAS-Pixelsensors vor Typkonversion**. University of Dortmund, Exp. Physik IV, 2001.

[13] CERN. Homepage des objekt-orientierten Analyse-Frameworks Root. Accessible at: `http://root.cern.ch`.

[14] CERN. Public webpages of CERN. 2011. Accessible at: `http://public.web.cern.ch`.

[15] A. Chilingarov. **Presentation at ATLAS Upgrade Week: Bulk current temperature dependence**. Lancaster University, 2011.

[16] Claus Grupen. **Teilchendetektoren**. BI Wissenschaftsverlag, 1993.

[17] D. Cussans. **Description of the JRA1 Trigger Logic Unit (TLU), v0.2c**. University of Bristol, UK, 2009.

[18] D. Barberis, A. Rosanov, M. Olceses, M. Gilchriese. Pixel Detector Active Area Layout. **Geneva, ATL-IP-EP-0004 2002, Rev. 3.1**.

[19] Daniel Haas. **The EUDET Pixel Telescope Data Acqusition System**, volume EUDET-Report 2009-03. Université de Genève, 2009.

[20] Personal correspondance with **Daniel Muenstermann, CERN**.

[21] G. Darbo. Overview of MCC Updates Since FDR, PRR of the pixel on- detector electronics. 2003. Accessible at: `http://www.ge.infn.it/ATLAS/Electron-ics/MCC-I2/MCC-PRR/02.Giovanni_MCC_Overview.pdf`.

[22] DESY. Webpage of the DESY II accelerator. Accessible at: `http://desy2.desy.de/`.

[23] DESY. Webpages of DESY. Accessible at: `http://hasylab.desy.de/images/content/e81/e83/imageobject147/DESY-Beschleuniger_ger.jpg`.

[24] DESY. Webpages of DESY. Accessible at: `http://min.desy.de/linac_ii__pia/`.

[25] DESY. Webpages of DESY. Accessible at: `http://www.desy.de/forschung/forschungsbereiche/beschleuniger/uebersicht_beschleuniger/hera/index_ger.html`.

[26] D. A. Dobos. **Diplome thesis: Production accompanying testing of the ATLAS Pixel module**. Lehrstuhl für Experimentelle Physik IV, TU Dortmund, 2004.

[27] Personal correspondance with **Edda Gschwendtner, CERN**.

[28] K. Einsweiler. **Change List for FE-I2 Engeneering Run**. ATL-IP-EM-0100, Geneva.

[29] K. Einsweiler. **Change List for FE-I3 Engineering Run**. ATL-IP-EM-0117, Geneva.

[30] K. Einsweiler. **On-detector Electronics Architecture, Draf V3.0**, 2003.

[31] K. Einsweiler. **Overview of the FE-I3 Pixel Front-end Chip**. ATLAS FE-I PRR, Geneva, 2003.

[32] G. L. et al. Streifendetektor, Patentoffenlegungsschrift. **OS 19620081A121.11.97.**, 1997.

[33] L. A. et al. 3rd RD49 Status Report - Study of the Radiation Tolerance of ICs for LHC. **CERN/LHCC/2000-003, Geneva**, 2002.

[34] Eudet Collaboration. Homepage of EUTelescope at hepforge. Accessible at: `http://projects.hepforge.org/eudaq/Eutelescope/`.

[35] Eudet Collaboration. Homepage of the EUDAQ DAQ System. Accessible at: `projects.hepforge.org/eudaq`.

[36] Eudet Collaboration. Project homepage of EUDET. Accessible at: `www.eudet.org`.

[37] Eudet Collaboration. **EUDET Pixel Telescope Data Taking Manual - Updated Version for M26**. DESY, 2010.

[38] Fischer. **Readout of Pixel detectors – some thoughts on the next chip generation**. Accessible at: `http://ltp.web.psi.ch/VER-TEX2001/slides/fischer.pdf`.

[39] P. Fischer. **AT-IP-EM-0084, Marseille**.

[40] Frank Gaede. **Presentation at LC Software Workshop CERN: LCIO Persistency Format for Detector Simulation and Reconstruction - towards LCIOv2**. DESY, 2009.

[41] Homepage of cprogramming.com. Accessible at: `http://www.cprogramming.com/tutorial/function-pointers.html`.

[42] G. Kramberger, V.Cindro, I, Mandić, M.Mikuz, M.Mavrtanik. Determinator of effective trapping times for electrons and holes in irradiated silicon. **NIM A 476 (2002) 645-651**, 2002.

[43] E. Geschwendtner. **Presentation: CNGS and North Area Operation**. CERN, AB/ATB/SBA, 2008.

[44] H. Gjersdal. **Track fitting and alignment in the Eutelescope framework**. Universty of Oslo, 2010.

[45] B. Hayes. **How to count**. American Scientist 89, Number 2 p. 110, 2001.

[46] Personal correspondance with **Jens Weingarten, University of Göttingen, Germany**.

[47] Homepage of the University of North Calolina - The Kalman Filter. Accessible at: `http://www.cs.unc.edu/~welch/kalman/`.

[48] Homepage of KEL. Accessible at: `www.kel.jp/english`.

[49] J. Klaiber-Lodewigs. **PhD-Thesis: The ATLAS Pixel Sensor - properties, characterization and quality control**. Lehrstuhl für Experimentelle Physik IV, TU Dortmund, 2005.

[50] Konrad Kleinknecht. **Detektoren für Teilchenstrahlung**. Teubner, 2005.

[51] Homepage of National Instruments. Accessible at: `http://www.ni.com/lwcvi/`.

[52] Homepage of LEMO. Accessible at: `www.lemo.de`.

[53] LHC Collaboration. **CERN-faq - LHC the guide**, 2006.

[54] Marc Winter. **Presentation at STAR Meeting: Status Report of the Laoratory Tests of MIMOSA-26**. IPHC/Strasbourg - IRFU/Saclay collaboration, 2009.

[55] Martin Maß. **ATLAS pixel testbeam results**. Universität Dortmund, Germany, 2004.

[56] M. Maß. **ATLAS Pixel Module - der Aufbau und deren Tests im Labor und im Pionenstrahl**. Dissertation, Experimentelle Physik IV, Universität Dortmund, 2005. Accessible at: `http://hdl.handle.net/2003/20163`.

[57] Personal correspondance with **Mathieu Benoit, CERN**.

[58] Michael Moll. **PhD-Thesis: Radiation Damage in Silicon Particle Detectors - Microscopic Defects and Macroscopic Properties**, 1999.

[59] Homepage of the Helmholtz Alliance - Millipede II. Accessible at: `https://www.wiki.terascale.de/index.php/Millepede_II`.

[60] Olaf Krasel. **PhD-Thesis: Charge Collection in Irradiated Silicon-Detectors - A study of the Operation Conditions of Silicon Sensors in the ATLAS Pixel Detector.** University of Dortmund, Exp. Physik IV, 2004.

[61] Homepage of Orange Tree Technologies. Accessible at: `http://www.orangetreetech.com`.

[62] Phil Allport. **Presentation at ATLAS PPS-Meeting: Process with Planar Silicon Technology for Pixel Layers in ATLAS at sLHC**. University of Liverpool, 2009.

[63] Philipp Roloff. **The EUDET High Resolution Pixel Telescope**. DESY, 2008.

[64] Prof. Dr. Bogdan Povh, Prof. Dr. Klaus Rith, Dr. Christoph Scholz, Dr. Frank Zentsche. **Teilchen und Kerne**. Springer Verlag, Berlin, Heidelberg, 5 edition, 1999.

[65] Homepage of Nokia - QT - crossplatform application and UI framework. Accessible at: `http://qt.nokia.com/`.

[66] R. Bechherle, G. Darbo. MCC-I2.1 Specifications - Event Builder, Draft V0.1. Accessible at: `http://www.ge.infn.it/ATLAS/Electronics/MCC-I2/Specs/evb.pdf`.

[67] R. Bechherle, G. Darbo. MCC-DSM Specifications – Reciever, Draft V0.1. 2002. Accessible at: `http://www.ge.infn.it/ATLAS/Electronics/MCC-I2/Specs/rec-I1.pdf`.

[68] R.Beccherle. Overview of MCC design specification updates since FDR, PRR of the pixel on-detector electronics. 2003. Accessible at: `http://www.ge.infn.it/ATLAS/Electronics/MCC-I2/MCC-PRR/03.Roberto_MCC_SpecChanges.pdf`.

[69] RD11 collaboration. Webpage of RD11 at CERN. Accessible at: `http://rd11.web.cern.ch/RD11/rkb/PH14pp/node62.html`.

[70] Renate Wunstorf. **PhD-Thesis: Systematische Untersuchungen zur Strahlenresistenz von Siliziumdetektoren für die Verwendung in Hochenergiephysik-Experimenten**. Fachbereich Physik, University of Hamburg, 1992.

[71] Communication with **R.H. Richter**.

[72] P. Roloff. **Introduction to Alignment**, volume EUTelescope Tutorial, DESY 27/05/2010. DESY and University of Hamburg, 2010.

[73] A. Rummler. **Diploma thesis: Design and Commissioning of a Setup for Charge Collection Efficiency Measurements of Highly Irradiated ATLAS Pixel Sensors**. Technische Universität Dortmund - Experimentelle Physik IV, 2009.

[74] Sebastian Fleischmann. **Diploma thesis: Track Reconstruction in the ATLAS Experiment - The Deterministic Annealing Filter**. Bergische Universität Wuppertal, 2006.

[75] Sebastian Müller. **Bachelor Thesis: Aufbau und Untersuchungen an einer ATLAS Pixel Teststrahlanordnung**, 2010.

[76] Silke Herbst. **Diploma thesis: Charakterisierung von ATLAS SingleChip-Sensoren vor und nach Bestrahlung**. University of Dortmund, Exp. Physik IV, 2009.

[77] H. Spieler. **Semiconductor Detector Systems, Series on Semi- conductor Science and Technology**. oxford science publications, 3 Aufl., 2006.

[78] SPS Page 1. Accessible at: `http://op-webtools.web.cern.ch/op-webtools/vistar/vistars.php?usr=SPS1`.

[79] T. Calin, M. Nicolaidis, R. Velazco. **Upset hardened memory design for submicron CMOS technology**. IEEE Trans. Nucl. Sci. 43 p. 2874, 1996.

[80] T. Stockmanns. **Diploma thesis: Messungen an der PIRATE Front-End-Elektronik fpr den ATLAS Pixeldetektor**. Univertity of Bonn, 2000.

[81] Private photolibrary of **Tobias Lapsien**.

[82] Tobias Lapsien. **Diploma thesis:**. University of Dortmund, Exp. Physik IV, 2011.

[83] Tobias Wittig. **Diploma thesis: Entwurf von Sensorprototypen und Teststrukturen für den ATLAS Upgrade Pixel Detektor**. University of Dortmund, Exp. Physik IV, 2009.

[84] Tobias Wittig. **PhD-Thesis: Design and Quality Control of Planar Pixel Sensors for the ATLAS Insertible B-Layer based on Slim Edge studies**. University of Dortmund, Exp. Physik IV, in preparation.

[85] Personal correspondance with **Tobias Wittig, University of Dortmund, Germany**.

[86] Homepage of Microsoft. Accessible at: `http://www.microsoft.com/germany/visualstudio/`.

[87] J. Weber. **Diploma thesis: Production Accompanying Measurements on the ATLAS Pixel Sensor**. Technische Universität Dortmund - Experimentelle Physik IV, 2004.

[88] J. Weingarten. **The Eudet Telescope**. Presentation at the ATLAS Planar Pixel / 3D Testbeam reconstruction tutorial, 2010.

[89] Homepage of Microsoft. Accessible at: `http://windows.microsoft.com/de-DE/windows/products/windows-xp`.

A. Most important parts of source code and examles of configuration files

```
1  void StartAPIXEudetHandler(char* name, char* hostname) {
2   mutex1 = PTHREAD_MUTEX_INITIALIZER;
3   ProducerArgs args;
4    args.name = name;
5    args.hostname = hostname;
6   producer_thread_return = pthread_create(&producer_thread, NULL, ProducerRun, (void
        *) &args);
7   Sleep((unsigned)1000);
8  }
9
10 void *ProducerRun(void * arg) {
11  if(declared_on_start_of_prod) {callOnStartOfProd();}
12  pthread_mutex_lock(&mutex1);
13  ProducerArgs *Args;
14  Args = (ProducerArgs *) arg;
15  running = 1;
16  APIXProducer *prod = new APIXProducer(Args->name,Args->hostname);
17  prodObj = prod;
18
19  while(running == 1) {
20   Sleep((unsigned)500);
21  }
22  delete prod;
23
24  pthread_mutex_unlock(&mutex1);
25  if(declared_on_stop_of_prod) {callOnStopOfProd();}
26  return 0;
27 }
```

Code Listing A.1: The C++ to C wrapping of TurboDAQ to EUDAQ-interface

```
void SendAPIXEvent(Event ev, int run, int n_ev, int block) {
 std::string det = "";
 if (board_type == APIX_SC) {
  det = "APIX-SC";
 } else {
  det = "APIX-MC";
 }
 RawDataEvent raw_ev(det, run, n_ev);
 std::vector<unsigned int> data;

 data.push_back(ev.PLL_Trig);
 if (ev.Eudet_Trig != 0) { data.push_back(ev.Eudet_Trig); }
 if (ev.Timestamp1 != 0) {
  data.push_back(ev.Timestamp1);
  data.push_back(ev.Timestamp2);
 }
 for (unsigned int i = 0; i< ev.max_data;i++) {
  data.push_back(ev.data[i]);
 }
 raw_ev.AddBlock(block,data);
 prodObj->SendEvent(raw_ev);
}

void SendAPIXBORE(int run, int mask) {
 std::string det = "";
 if (board_type == APIX_SC) {
  det = "APIX-SC";
 } else {
  det = "APIX-MC";
 }
 RawDataEvent bore(RawDataEvent::BORE(det,run));
 bore.SetTag("PlaneMask", eudaq::to_string(mask));
 prodObj->SendEvent(bore);
}

struct Event {
 unsigned int data[1000];
 unsigned int max_data;
 unsigned int PLL_Trig;
 unsigned int Eudet_Trig;
 unsigned int Timestamp1;
 unsigned int Timestamp2;
};
```

Code Listing A.2: Event-Sending mechanism in TurboDAQ

```
[DataCollector]

[LogCollector]
PrintLevel = ALL
SaveLevel = ALL

[Producer.APIX1]
Configure = 1
Latency = 134
LocalFile = nowhere.txt
ModuleConfigurationFile = MPPS-TBOct2010/configs/2010-10-16-D09-D010-MPP1-MPP2-D08-
    D07_800V-LVP1-LVP2.cfg
SaveLocal = 0
TPCCChannel = 0
TriggerDelay = 108
TriggerOnEudet = 1
TriggerOnLemo = 0
TriggerOnMonhit = 0
WriteOutEudetTrig = 1
WriteOutPLLTrig = 1
WriteOutTimestamp = 1

[Producer.EUDRB1]
AdcDelay = 3
Board0.Slot = 3
Board1.Slot = 5
Board2.Slot = 7
ClkSelect = 2
Det = MIMOSA26
IDOffset = 0
Mode = ZS2
NumBoards = 3
PostDetResetDelay = 4
ResetBusy = 0
Unsynchronized = 0
Version = 3

[Producer.EUDRB2]
AdcDelay = 3
Board3.Slot = 3
Board4.Slot = 5
Board5.Slot = 7
ClkSelect = 2
Det = MIMOSA26
IDOffset = 3
Mode = ZS2
NumBoards = 3
PostDetResetDelay = 4
ResetBusy = 0
Unsynchronized = 0
Version = 3

[Producer.TLU]
AndMask = 15
BitFile = ../tlu/TLU2_Toplevel-65_40MHz.bit
DutMask = 14
OrMask = 0
TimestampPerRun = 0
TrigRollover = 0
TriggerInterval = 0
VetoMask = 0

[RunControl]
NoTrigWarnTime = 10
RunSizeLimit = 100000000
```

Code Listing A.3: Typical EUDET-configuration-file. Beside the TurboDAQ-System two MVME-DAQ systems, two read out two times 3 EUDRBs in parallel, are being configured

```
void DataTakingLoop() {
  unsigned int data = 0;
  // Compare trigger numbers
  int diff = (next_PLL_Trig - next_Eudet_PLL_Trig);
  // Fill in ev_data
  ev_data.data[ev_data.max_data++] = data_back_buffer;
  ev_data.PLL_Trig = eudet_back_buffer;
  ev_data.Eudet_Trig = 0xffffffff;
  ev_data.Timestamp1 = 0xffffffff;
  ev_data.Timestamp2 = 0xffffffff;
  // Missing data? Flag error if true
  if ((0 < diff && diff < 0x3800) || diff <= -0x3800) {
    if(Declared_extern_error_fifo_sync) callWarningFIFOSync(1);
  } // Read data fifo
  else while (1) {
    while (daq_running && is_data_fifo_empty()) Sleep(1);
    if (!daq_running) break;
    data = get_data_fifo();
    if ((next_PLL_Trig = is_PLL_Trig(data)) != 0xffffffff) {
      data_back_buffer = data;
      break;
    }
    ev_data.data[ev_data.max_data++] = data;
  }
  // Missing Eudet? Flag warning if true
  if ((-0x3800 < diff && diff < 0) || 0x3800 <= diff) {
    if(Declared_extern_warning_fifo_sync) callErrorFIFOSync(1);
  } else while (1) { // Read Eudet fifo
    if (eudet_back_buffer & 0x004000000) { // Read Eudet_Trig?
      while (daq_running && is_eudet_fifo_empty()) Sleep(1);
      if (!daq_running) break;
      data = get_eudet_fifo();
      if ((next_Eudet_PLL_Trig = is_PLL_Trig(data)) != 0xffffffff) {
        eudet_back_buffer = data;
        break;
      }
      ev_data.Eudet_Trig = data;
      eudet_back_buffer &= ~0x004000000;
      continue;
    } // Read Timestamp?
    if (eudet_back_buffer & 0x008000000) {
      while (daq_running && is_eudet_fifo_empty()) Sleep(1);
      if (!daq_running) break;
      data = get_eudet_fifo();
      if ((next_Eudet_PLL_Trig = is_PLL_Trig(data)) != 0xffffffff) {
        eudet_back_buffer = data;
        break;
      }
      ev_data.Timestamp1 = data;
      while (daq_running && is_eudet_fifo_empty()) Sleep(1);
      if (!daq_running) break;
      data = get_eudet_fifo();
      if ((next_Eudet_PLL_Trig = is_PLL_Trig(data)) != 0xffffffff) {
        eudet_back_buffer = data;
        break;
      }
      ev_data.Timestamp2 = data;
      eudet_back_buffer &= ~0x008000000;
      continue;
    }  // Should be at next header
    if (eudet_back_buffer | 0x00c000000) {
      while (daq_running && is_eudet_fifo_empty()) Sleep(1);
      if (!daq_running) break;
      data = get_eudet_fifo();
      if ((next_Eudet_PLL_Trig = is_PLL_Trig(data)) != 0xffffffff) {
        eudet_back_buffer = data;
        break;
      }
      continue;
} } }
```

Code Listing A.4: Implementation of the TurboDAQ's data taking loop

```
void * DAQThread(void * arg) {
  OnStartOfDaqRun();
  pthread_mutex_lock(&DAQMutex);
  daq_running = 1;

  /* Everytime you restart the DAQ-Thread these values should be initialized */
  next_PLL_Trig = 0xFFFFFFFF;
  next_Eudet_PLL_Trig = 0xFFFFFFFF;
  plain_next_PLL_Trig = 0xFFFFFFFF;
  data_back_buffer = 0xFFFFFFFF;
  eudet_back_buffer = 0xFFFFFFFF;

  last_eudet_type = eudet_unknown;
  next_expected_eudet_type = eudet_unknown;
  last_data_type = data_unknown;
  sync_status = unknown;

  // Drain fifos
  while (!is_data_fifo_empty()) get_data_fifo();
  while (!is_eudet_fifo_empty()) get_eudet_fifo();
  clear_event_buffer();

  while(daq_running == 1) {
    while((is_eudet_fifo_empty()==0) && (is_data_fifo_empty()==0) && (daq_running
        == 1)) { //Must be data in both fifos
      DataTakingLoop();
      if (ev_data.PLL_Trig != 0xFFFFFFFF) {
        HandleOneEvent(ev_data);
        clear_event_buffer();
    } }
    Sleep((unsigned int)50);
  }
  pthread_mutex_unlock(&DAQMutex);
  OnStopOfDaqRun();
  return 0;
}
```

Code Listing A.5: Implementation of TurboDAQ's testbeam DAQ-tread

```
<?xml version="1.0" encoding="us-ascii"?>
<marlin xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="http://ilcsoft.desy.de/marlin/marlin.xsd">
  <execute>
    <processor name="AIDA"/>
    <processor name="UniversalNativeReader"/>
    <processor name="FilterZSDataM26"/>
    <processor name="M26Clustering"/>
    <processor name="APIXClusteringUnfiltered"/>
    <processor name="Save"/>
  </execute>
  <global>
    <parameter name="LCIOInputFiles"> </parameter>
    <parameter name="GearXMLFile" value="gear.xml"/>
    <parameter name="MaxRecordNumber" value="1100000"/>
    <parameter name="SkipNEvents" value="0"/>
    <parameter name="SupressCheck" value="false"/>
    <parameter name="Verbosity" value="MESSAGE4"/>
  </global>
  <processor name="AIDA" type="AIDAProcessor">
    <parameter name="Compress" type="int" value="1"/>
    <parameter name="FileName" type="string" value="run021120-converter-histo"/>
    <parameter name="FileType" type="string" value="root"/>
  </processor>
  <processor name="UniversalNativeReader" type="EUTelNativeReader">
    <parameter name="EUBRDRawModeOutputCollection" type="string" lcioOutType="
        TrackerRawData"> rawdata </parameter>
    <parameter name="EUDRBZSModeOutputCollection" type="string" lcioOutType="
        TrackerData"> zsdata </parameter>
    <parameter name="EUDRBSensorIDVec"> 0 1 2 3 4 5 </parameter>
    <parameter name="SyncTriggerID" type="bool" value="true"/>
    <parameter name="GeoID" type="int" value="56"/>
    <parameter name="InputFileName" type="string" value="run021120.raw␣␣"/>
  </processor>
  <processor name="FilterZSDataM26" type="EUTelFilterZSDataProcessor">
    <parameter name="_filteredZSDataCollectionName" type="string" value="
        filtered_zsdata_m26"/>
    <parameter name="ZSDataCollectionName" type="string" value="zsdata_m26" />
    <parameter name="NoiseOccupancyCollection" type="string" value="noiseocc" />
    <parameter name="NoiseOccupancyDBFile" type="string" value="run21092-noise-db.
        slcio"/>
    <parameter name="Threshold" type="double" value="0.001" />
  </processor>
  <processor name="M26Clustering" type="EUTelM26ClusteringProcessor">
    <parameter name="ZSDataCollectionName" type="string" lcioInType="TrackerData">
        filtered_zsdata_m26 </parameter>
    <parameter name="HistoInfoFileName" type="string" value="histoinfo.xml"/>
    <parameter name="ClusterCollectionName" type="string" lcioOutType="TrackerPulse
        "> cluster_m26_filtered </parameter>
  </processor>
   <processor name="APIXClusteringUnfiltered" type="EUTelAPIXClusteringProcessor">
    <parameter name="ZSDataCollectionName" type="string" lcioInType="TrackerData">
        zsdata_apix </parameter>
    <parameter name="HistoInfoFileName" type="string" value="histoinfo.xml"/>
    <parameter name="ClusterCollectionName" type="string" lcioOutType="TrackerPulse
        ">cluster_apix_unfiltered </parameter>
    <parameter name="OriginalZsDataCollectionName" type="string">
        original_zsdata_apix_unfiltered </parameter>
  </processor>
  <processor name="Save" type="EUTelOutputProcessor">
    <parameter name="DropCollectionNames" type="StringVec"> firstFrame secondFrame
        thirdFrame </parameter>
    <parameter name="LCIOOutputFile" type="string" value="run021120-cluster.slcio"/
        >
    <parameter name="LCIOWriteMode" type="string" value="WRITE_NEW"/>
    <parameter name="SkipIntermediateEORE" type="bool" value="true"/>
  </processor>
</marlin>
```

Code Listing A.6: Example Steering-file for for converting a raw-file into a cluster-file

```
<gear>
  <global detectorName="EUTelescope" />
  <BField type="ConstantBField" x="0.0" y="0.0" z="0.0"/>
  <detectors>
    <detector name="SiPlanes" geartype="SiPlanesParameters">
      <siplanesID ID="100" />
      <siplanesType type="TelescopeWithoutDUT" />
      <siplanesNumber number="5"/>
      <layers>
        <!--Eudet-Plane 0 -->
        <layer>
          <ladder ID="0" positionX="0.00" positionY="0.00" positionZ="0.00" sizeX="
              21.2" sizeY="10.6" thickness="0.686" radLength="93.660734" />
          <sensitive ID="0" positionX="0.00" positionY="0.00" positionZ="0.00"
              sizeX="21.2" sizeY="10.6" thickness="0.02" npixelX="1152" npixelY="
              576" pitchX="0.018402778" pitchY="0.018402778" resolution="0.003"
              rotation1="-1.0" rotation2="0.0" rotation3="0.0" rotation4="-1.0"
              radLength="93.660734" />
        </layer>
        <!--Eudet-Plane 1 -->
        <layer>
          <ladder ID="1" positionX="0.00" positionY="0.00" positionZ="150.00" sizeX
              ="21.2" sizeY="10.6" thickness="0.686" radLength="93.660734" />
          <sensitive ID="1" positionX="0.00" positionY="0.00" positionZ="150.00"
              sizeX="21.2" sizeY="10.6" thickness="0.02" npixelX="1152" npixelY="
              576" pitchX="0.018402778" pitchY="0.018402778" resolution="0.003"
              rotation1="-1.0" rotation2="0.0" rotation3="0.0" rotation4="-1.0"
              radLength="93.660734" />
        </layer>
        <!--Eudet-Plane 2 -->
        <layer>
          <ladder ID="2" positionX="0.00" positionY="0.00" positionZ="300.00" sizeX
              ="21.2" sizeY="10.6" thickness="0.686" radLength="93.660734" />
          <sensitive ID="2" positionX="0.00" positionY="0.00" positionZ="300.00"
              sizeX="21.2" sizeY="10.6" thickness="0.02" npixelX="1152" npixelY="
              576" pitchX="0.018402778" pitchY="0.018402778" resolution="0.003"
              rotation1="-1.0" rotation2="0.0" rotation3="0.0" rotation4="-1.0"
              radLength="93.660734" />
        </layer>

        <!--APIX-Plane 0 -->
        <layer>
          <ladder ID="10" positionX="0.00" positionY="0.00" positionZ="401.95"
              sizeX="7.2" sizeY="8.0" thickness="0.686" radLength="93.660734" />
          <sensitive ID="10" positionX="0.00" positionY="0.00" positionZ="401.95"
              sizeX="7.2" sizeY="8.0" thickness="0.025" npixelX="18" npixelY="160"
              pitchX="0.40" pitchY="0.05" resolution="0.003" rotation1="-1.0"
              rotation2="0.0" rotation3="0.0" rotation4="1.0" radLength="93.660734"
               />
        </layer>

        <!--APIX-Plane 1 -->
        <layer>
          <ladder ID="11" positionX="0.00" positionY="0.00" positionZ="455.65"
              sizeX="7.2" sizeY="8.0" thickness="0.686" radLength="93.660734" />
          <sensitive ID="11" positionX="0.00" positionY="0.00" positionZ="455.65"
              sizeX="7.2" sizeY="8.0" thickness="0.025" npixelX="18" npixelY="160"
              pitchX="0.40" pitchY="0.05" resolution="0.003" rotation1="-1.0"
              rotation2="0.0" rotation3="0.0" rotation4="1.0" radLength="93.660734"
               />
        </layer>
      </layers>
    </detector>
  </detectors>
</gear>
```

Code Listing A.7: Example gear-file with three MIMOSA-26 sensors and two ATLAS pixel SCAs

B. Typical residuals of the telescope and the DUTs

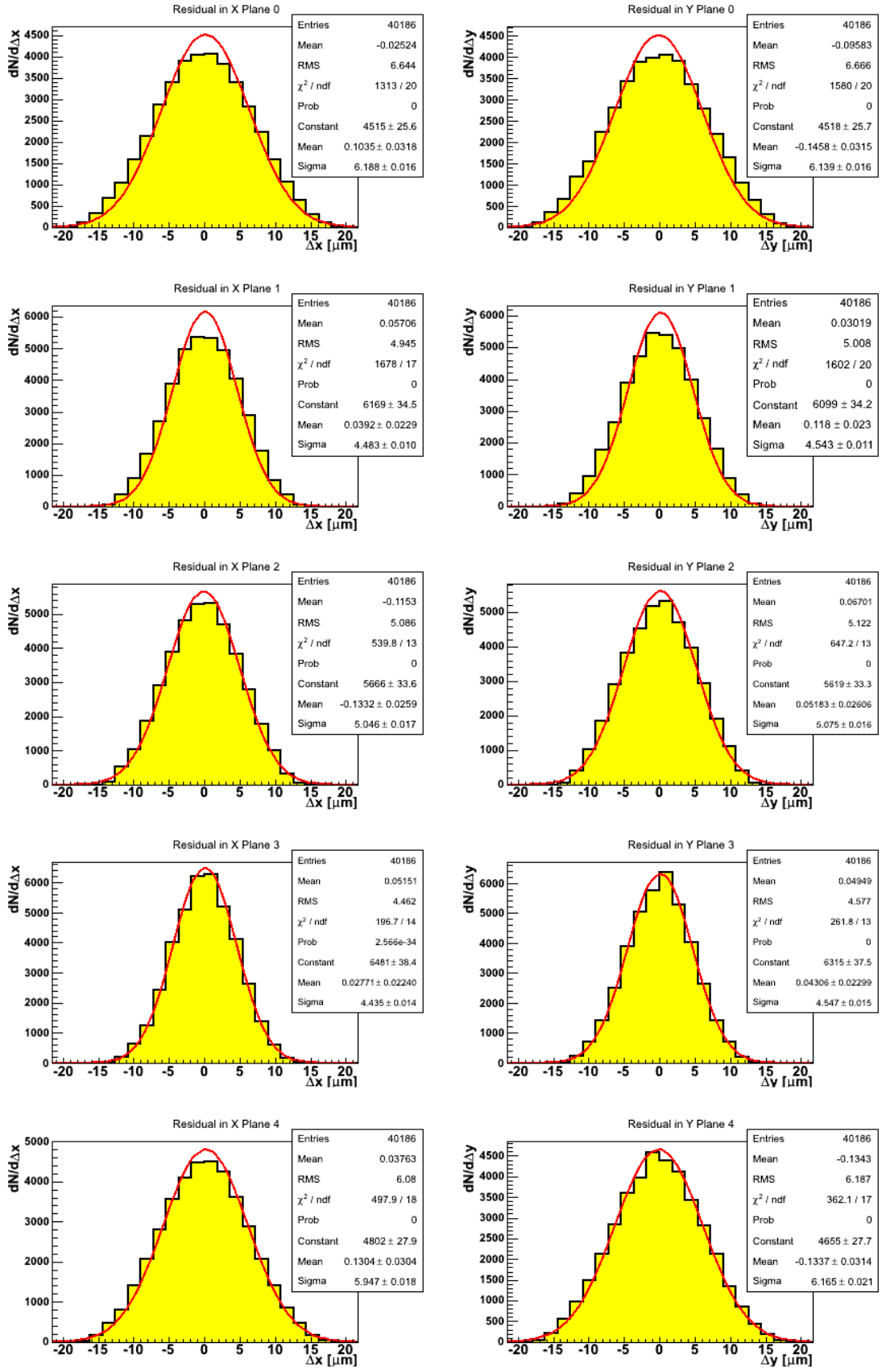


Figure B.1.: Biased residuals of the EUDET telescope planes. Only five planes out of six are used for track fitting. The underlying data is the run 11250 of the July 2010 testbeam at CERN

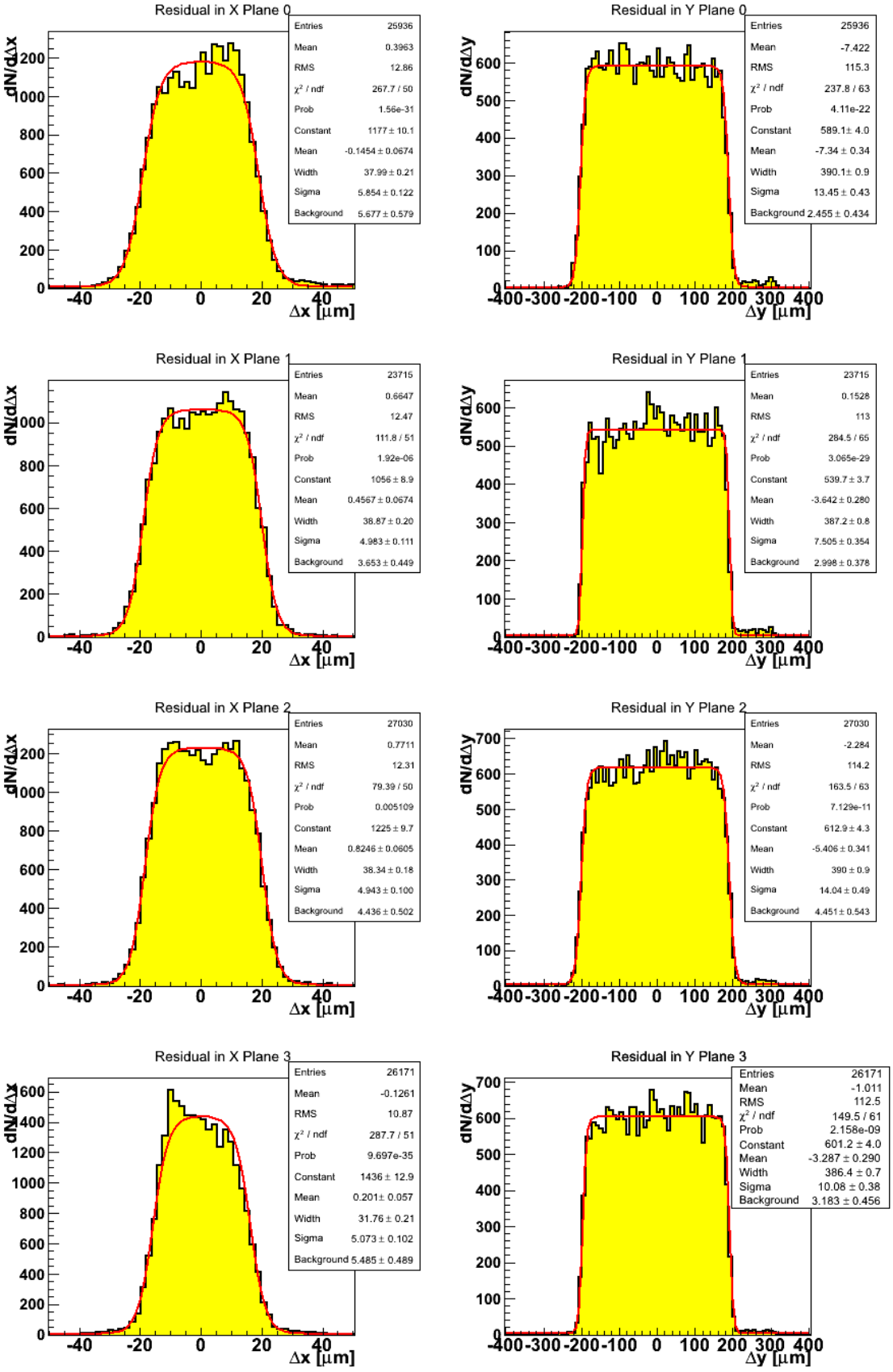


Figure B.2.: Unbiased residuals of DUTs 0-3. The track fitting was done using the EUDET telescope planes. Only five planes out of six are used for track fitting. The data from the DUTs are not used for fitting. The underlying data is the run 11250 of the July 2010 testbeam at CERN

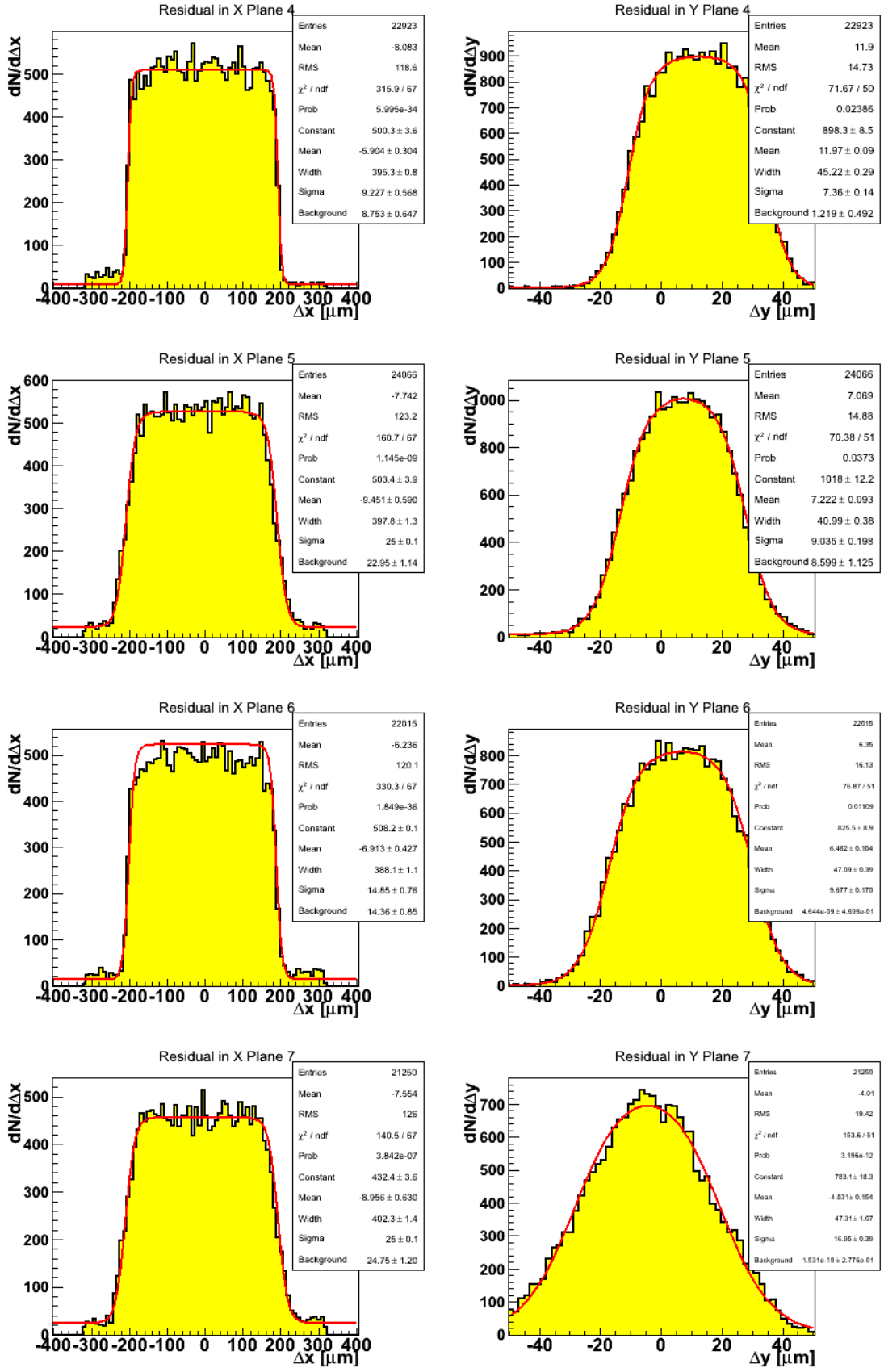


Figure B.3.: Unbiased residuals of DUTs 4-7. The track fitting was done using the EUDET telescope planes. Only five planes out of six are used for track fitting. The data from the DUTs are not used for fitting. The underlying data is the run 11250 of the July 2010 testbeam at CERN

C. Sensor overview plots at perpendicular incidence

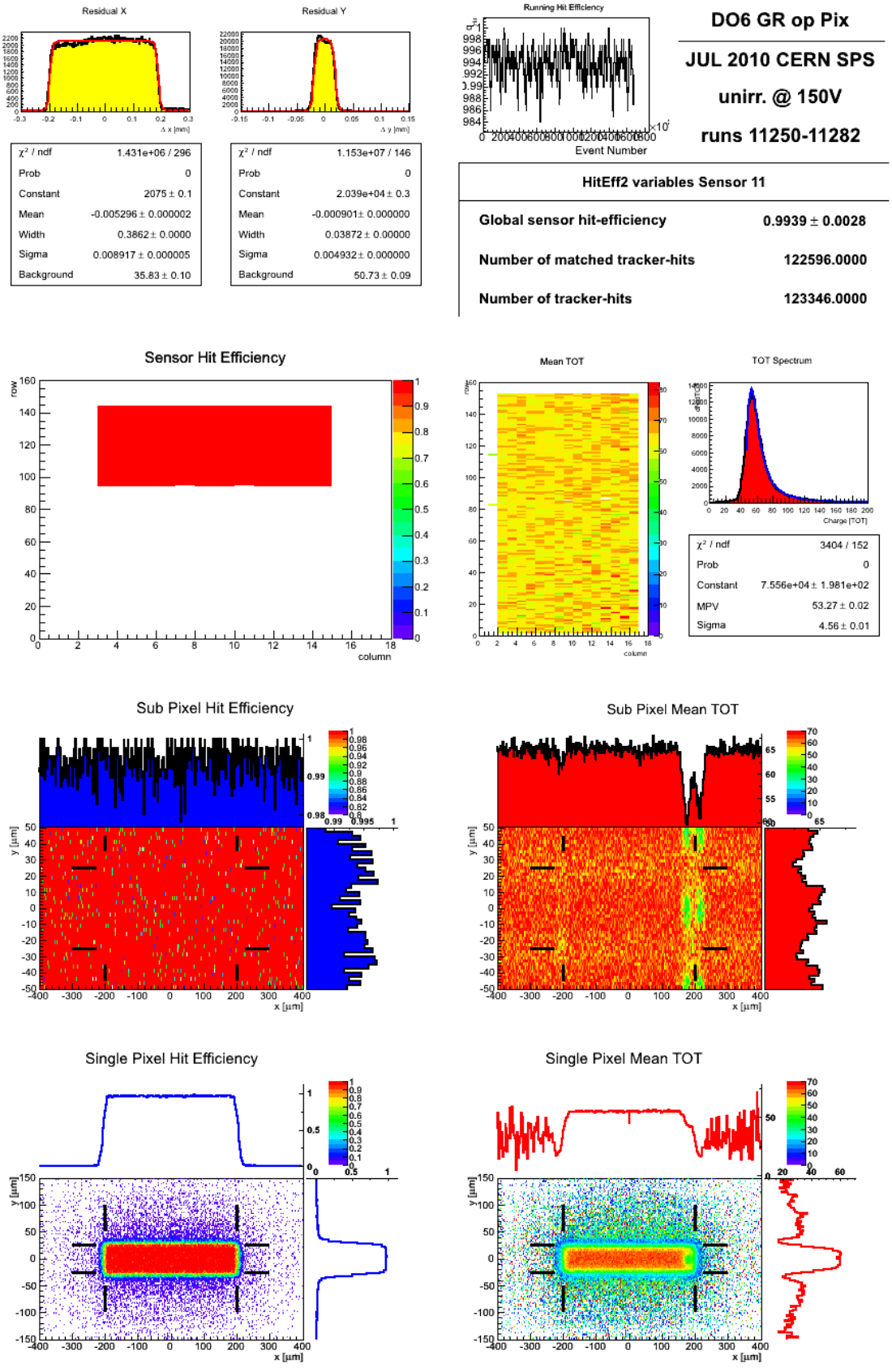


Figure C.1.: Sensor overview of sensor DO6 (not irradiated), running at 150 V

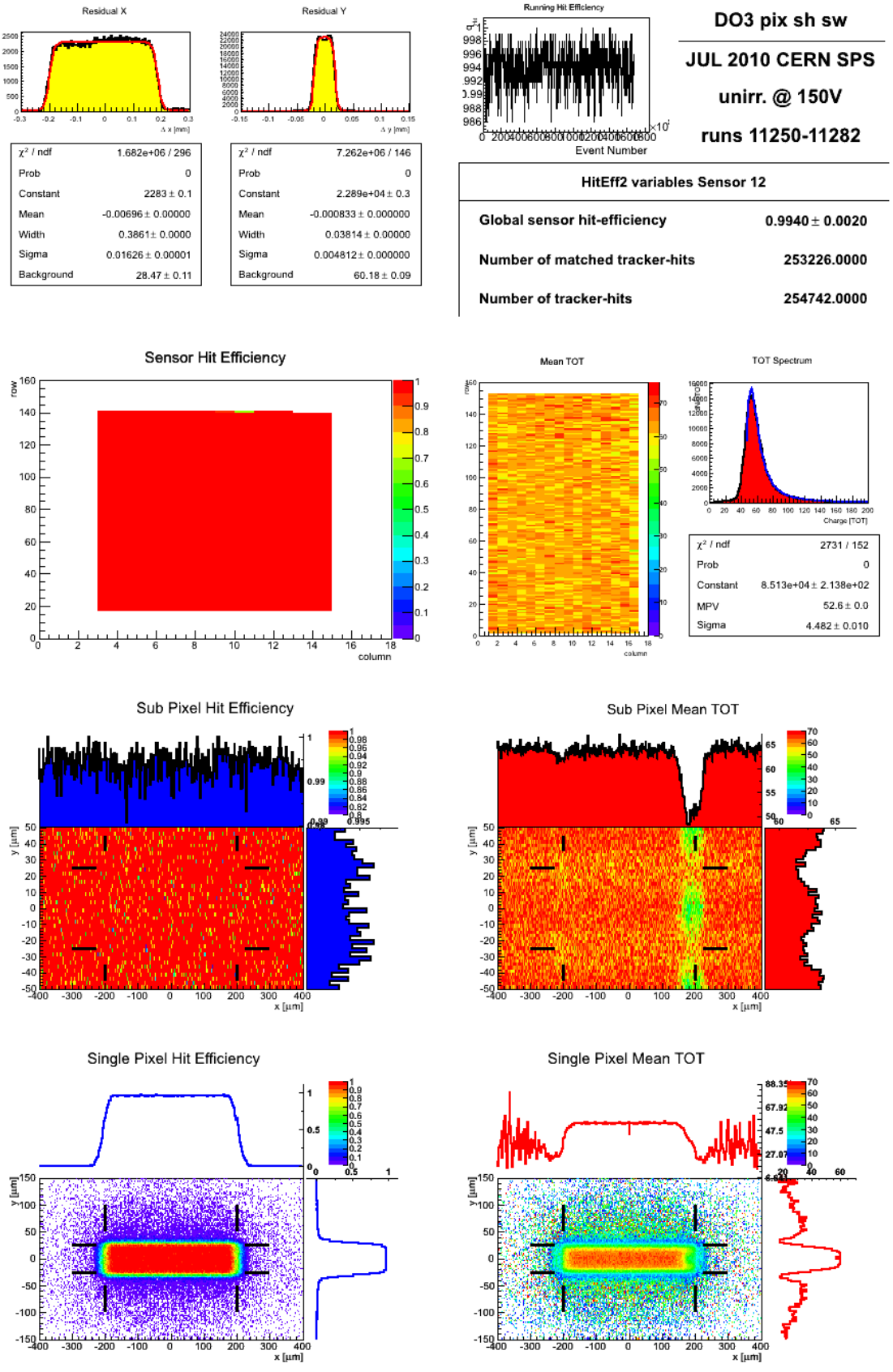


Figure C.2.: Sensor overview of sensor DO3 (not irradiated), running at 150 V

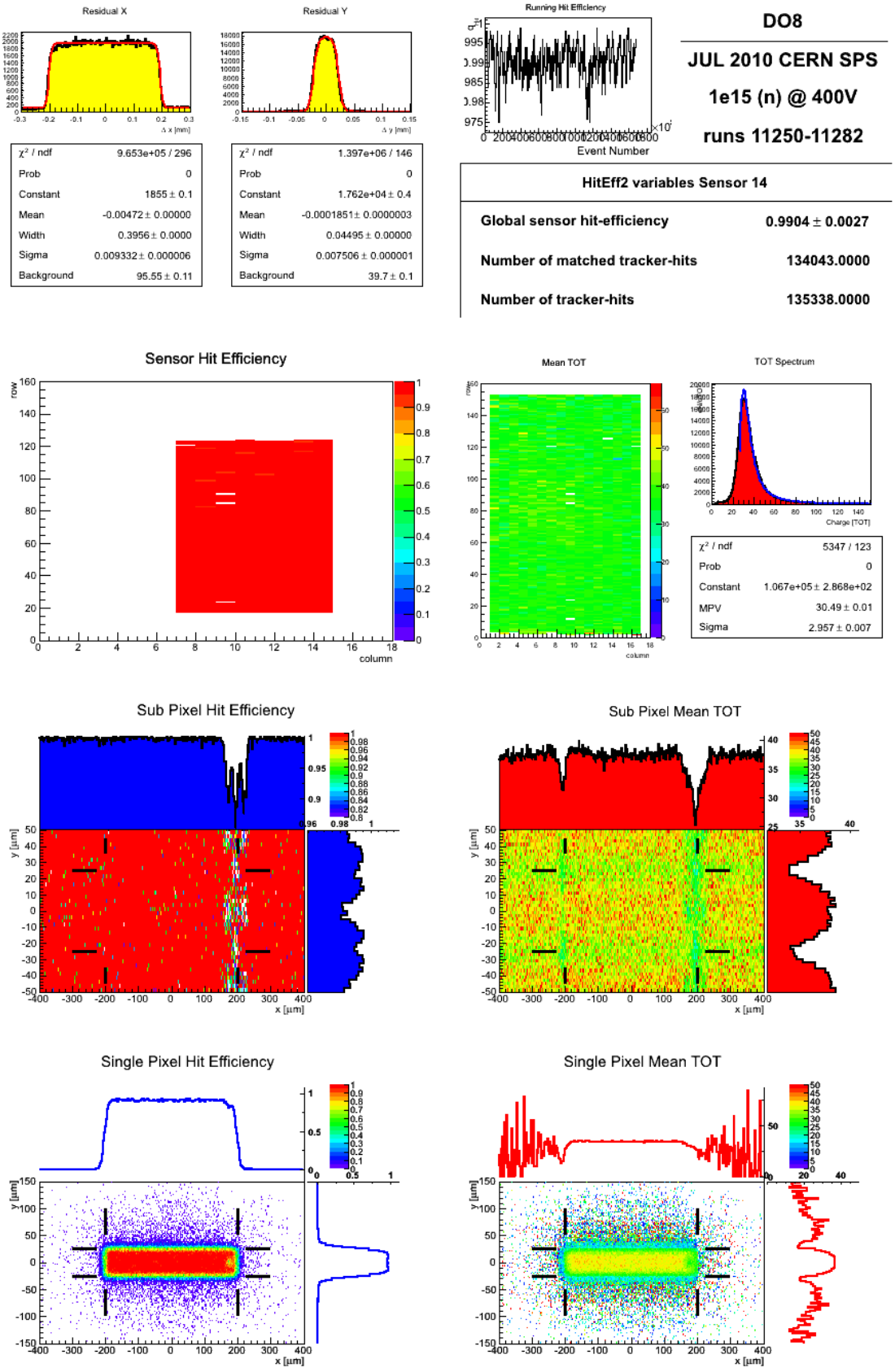


Figure C.3.: Sensor overview of sensor DO8 irradiated to $\Phi_{eq} = 1 \cdot 10^{15}\,n_{eq}\,cm^{-2}$ with neutrons, running at 400 V

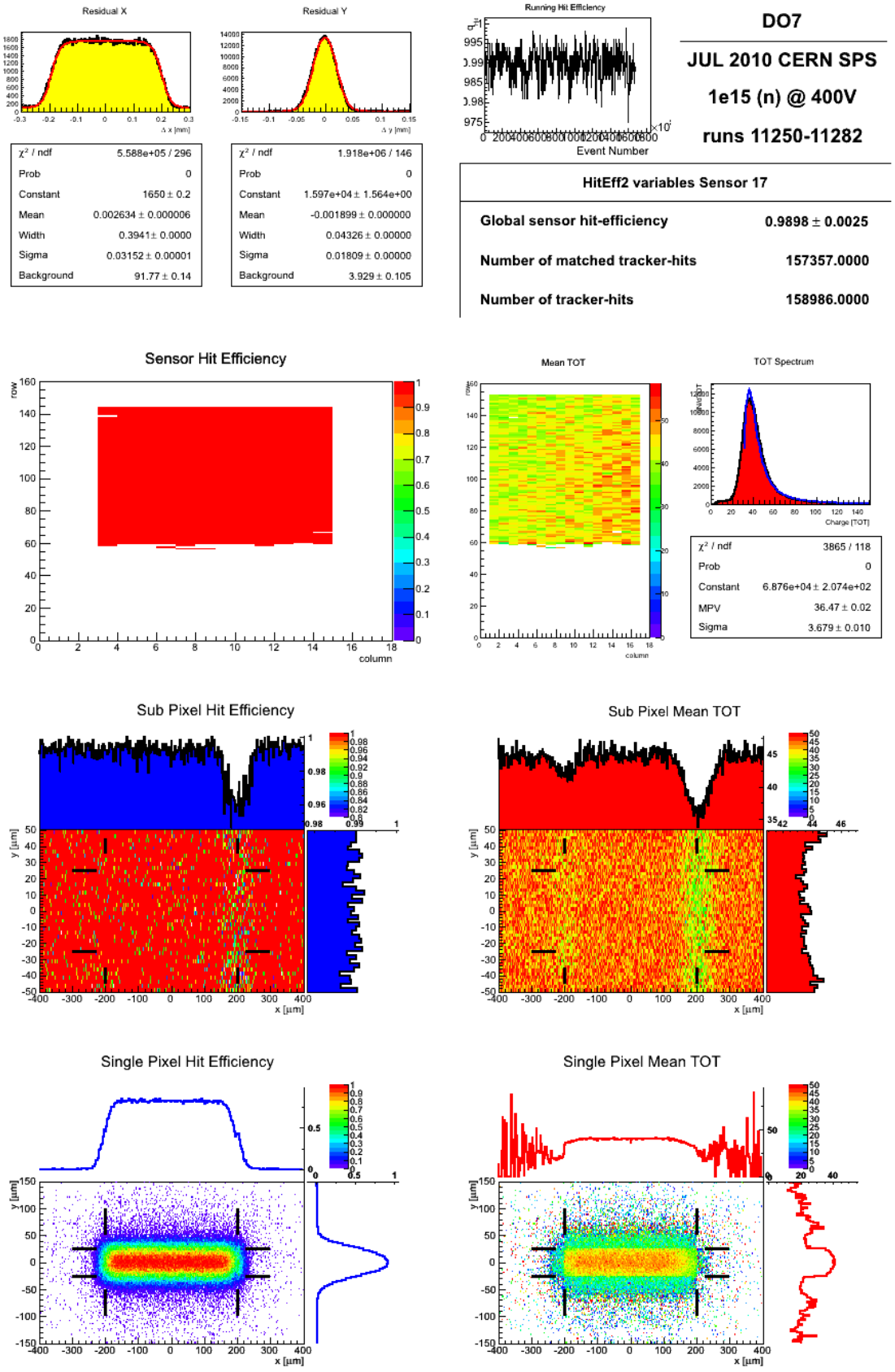


Figure C.4.: Sensor overview of sensor DO7 irradiated to $\Phi_{eq} = 1 \cdot 10^{15}\,n_{eq}\,cm^{-2}$ with protons, running at 400 V

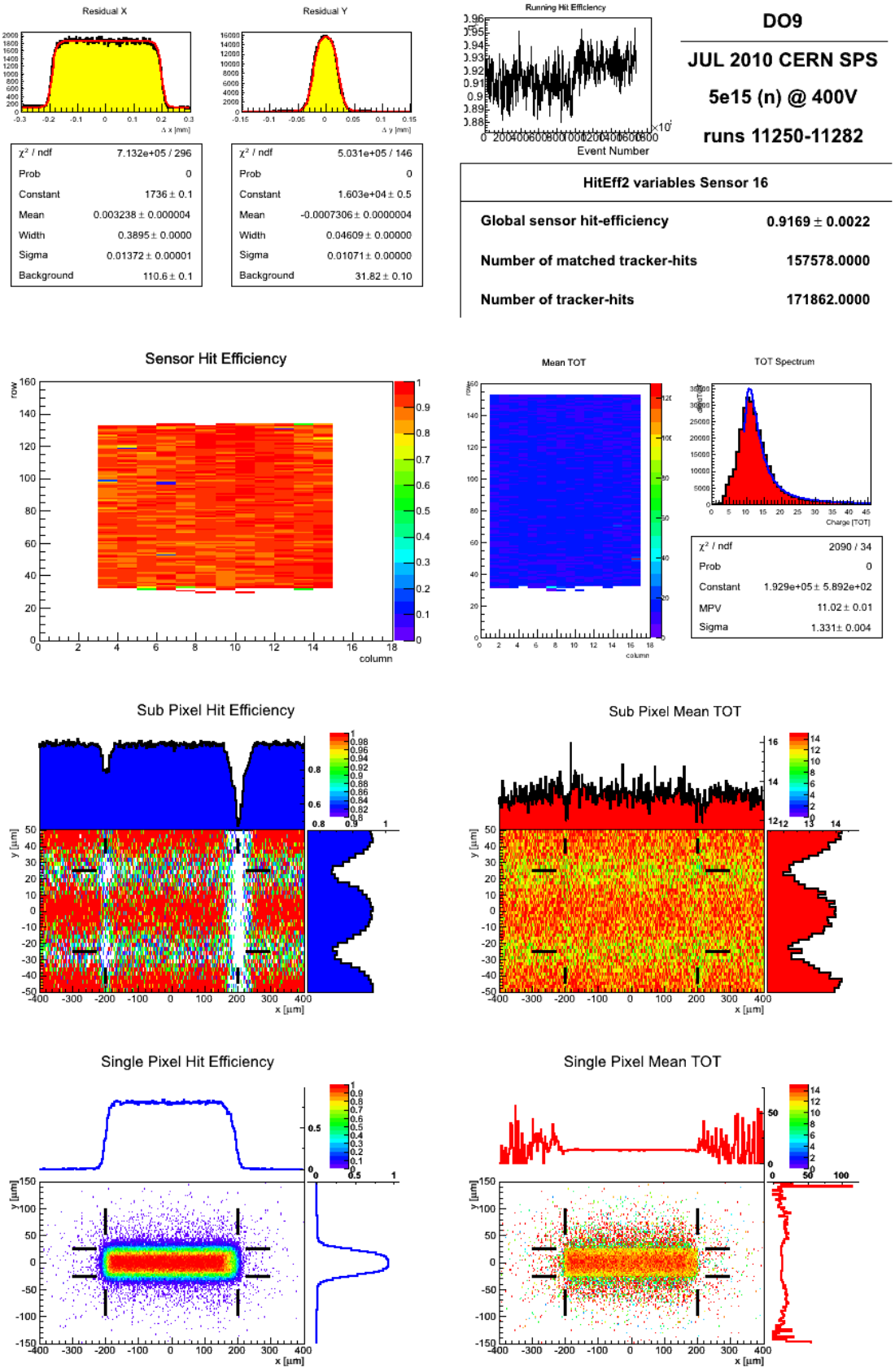


Figure C.5.: Sensor overview of sensor DO9 irradiated to $\Phi_{eq} = 5 \cdot 10^{15}\,n_{eq}\,cm^{-2}$ with neutrons, running at 400 V

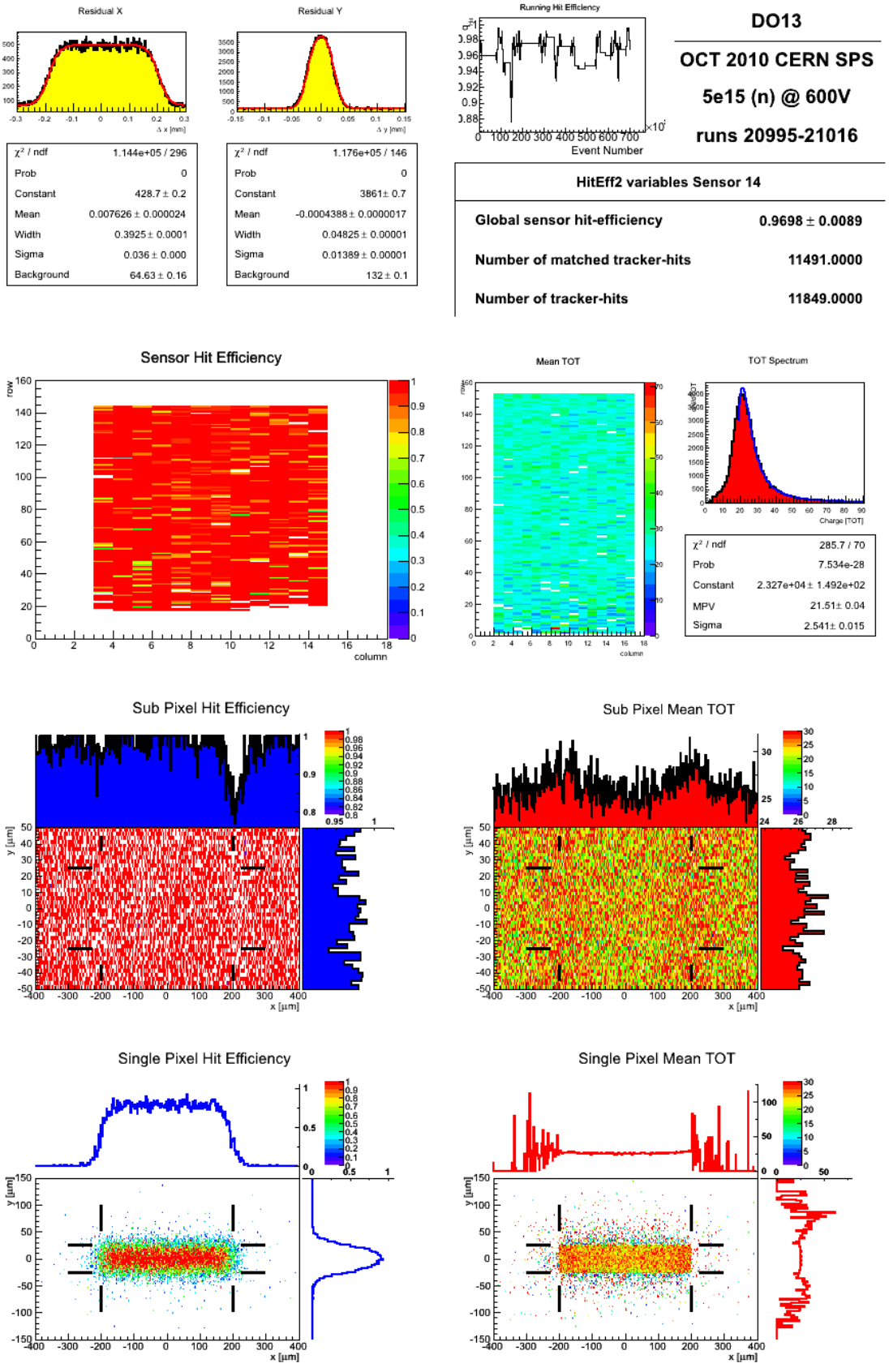


Figure C.6.: Sensor overview of sensor DO13 irradiated to $\Phi_{\mathrm{eq}} = 5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons, running at 600 V

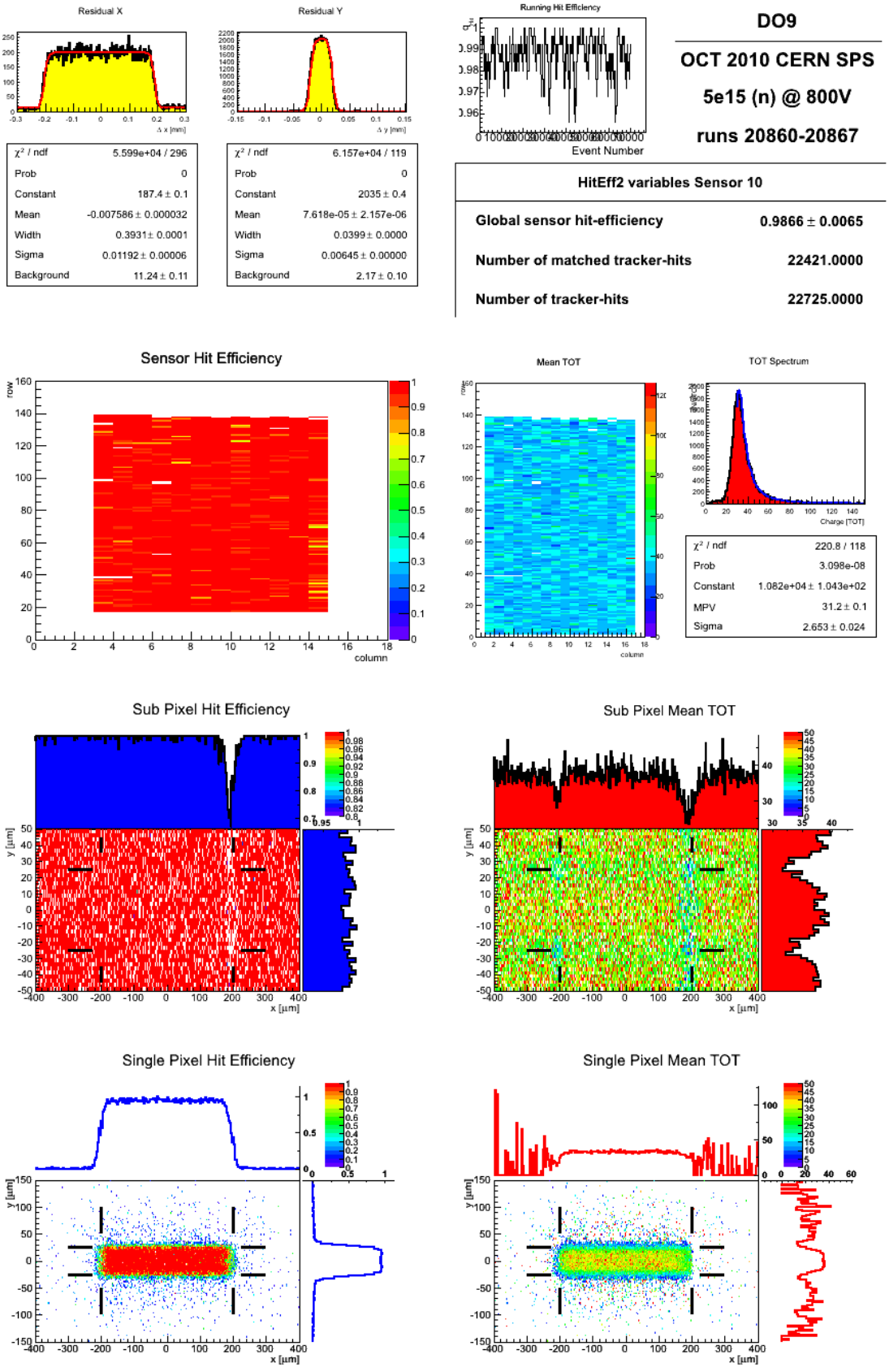


Figure C.7.: Sensor overview of sensor DO9 irradiated to $\Phi_{eq} = 5 \cdot 10^{15}\,n_{eq}\,cm^{-2}$ with neutrons, running at 800 V

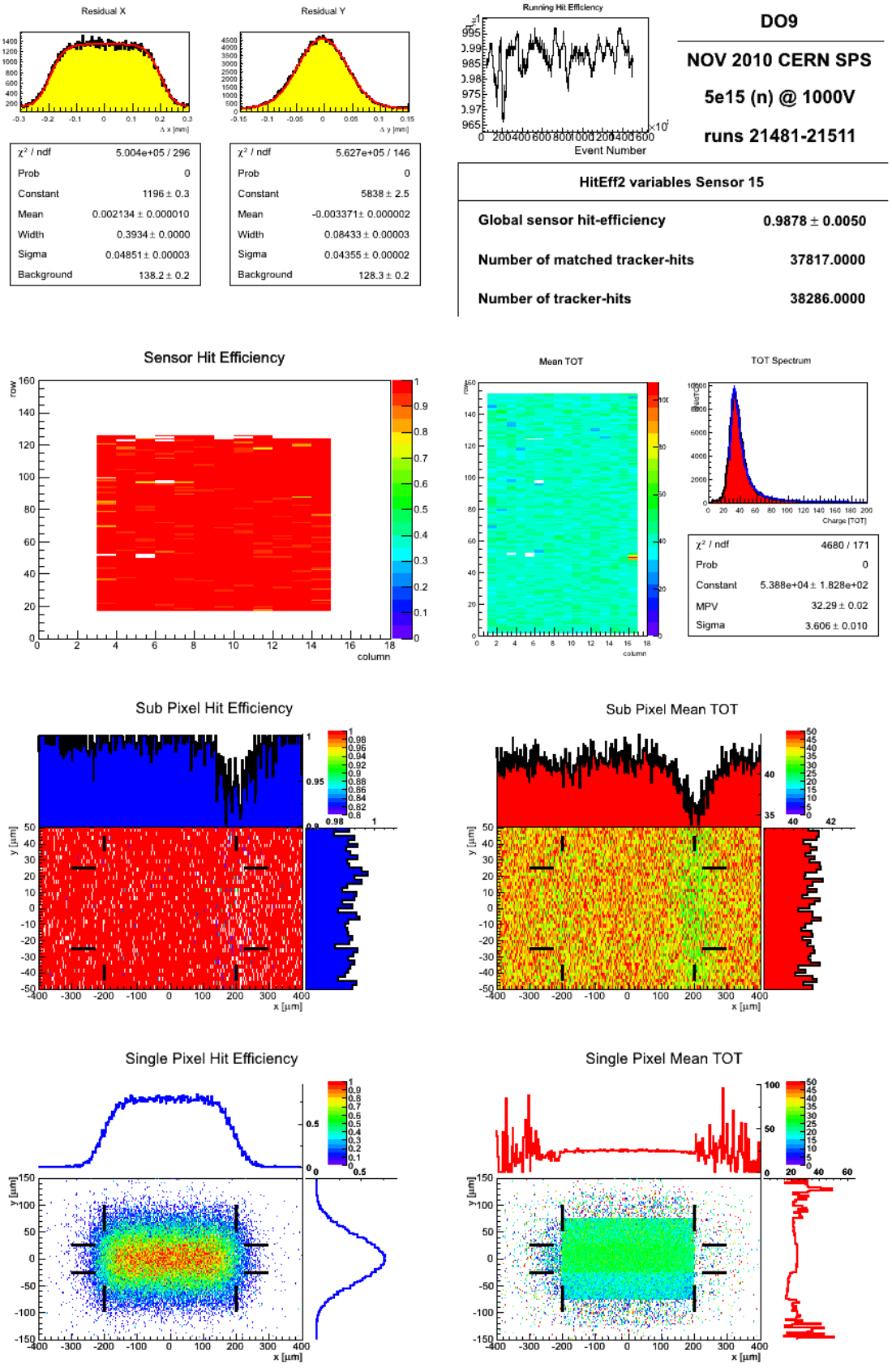


Figure C.8.: Sensor overview of sensor DO9 irradiated to $\Phi_{eq} = 5 \cdot 10^{15}\,n_{eq}\,cm^{-2}$ with neutrons, running at 1000 V

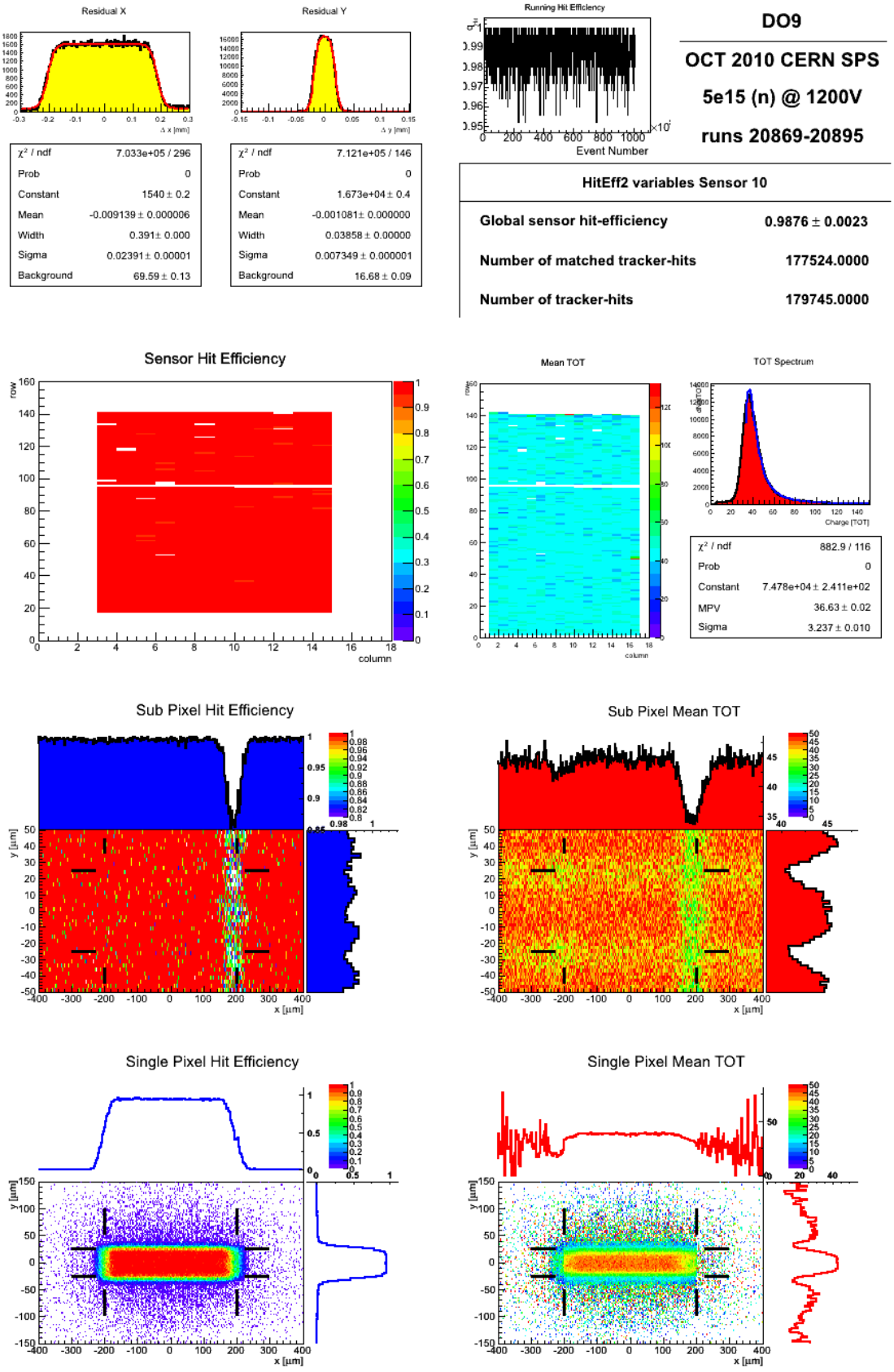


Figure C.9.: Sensor overview of sensor DO9 irradiated to $\Phi_{\mathrm{eq}} = 5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons, running at 1200 V

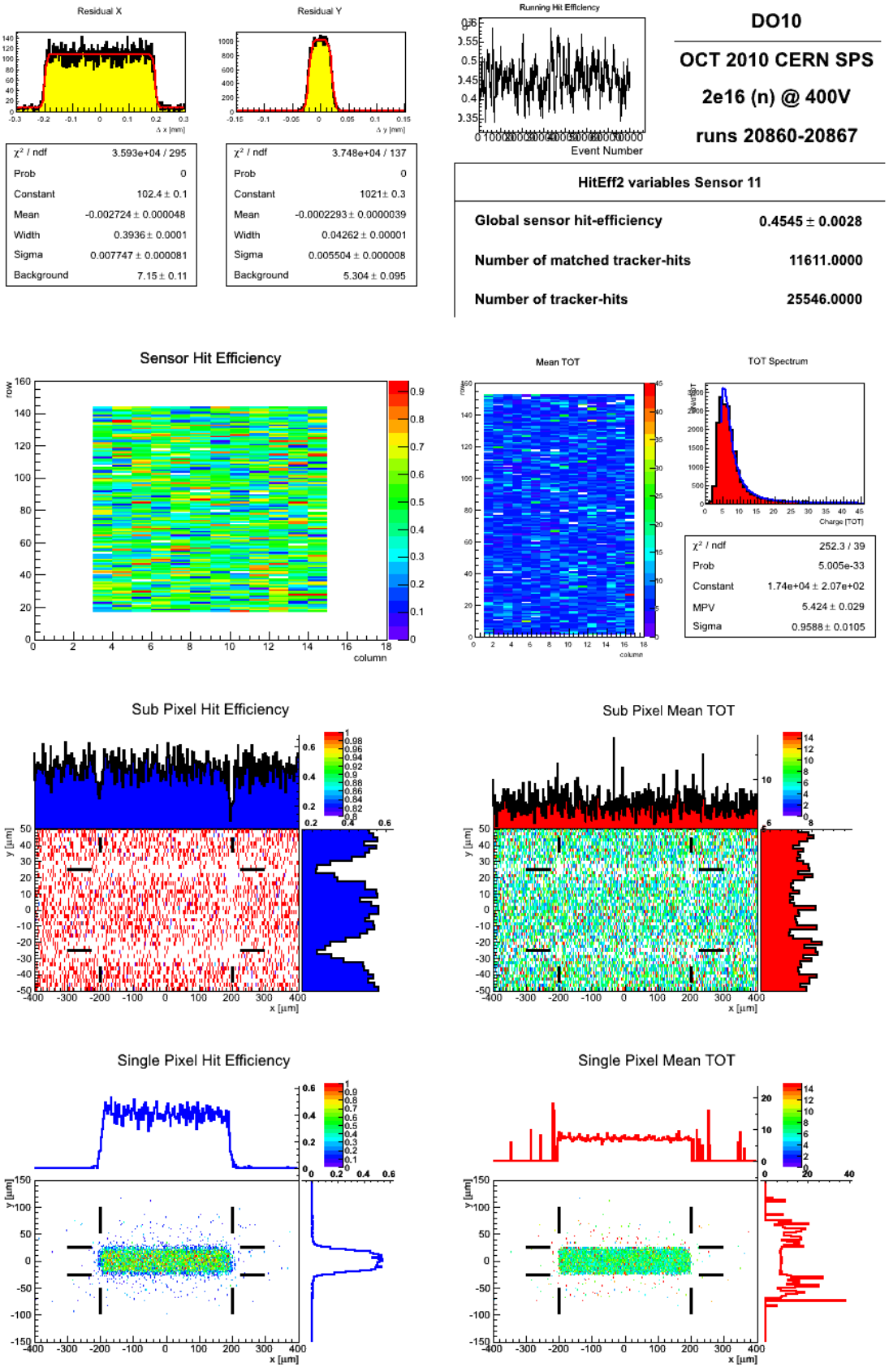


Figure C.10.: Sensor overview of sensor DO10 irradiated to $\Phi_{\mathrm{eq}} = 2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons, running at 400 V

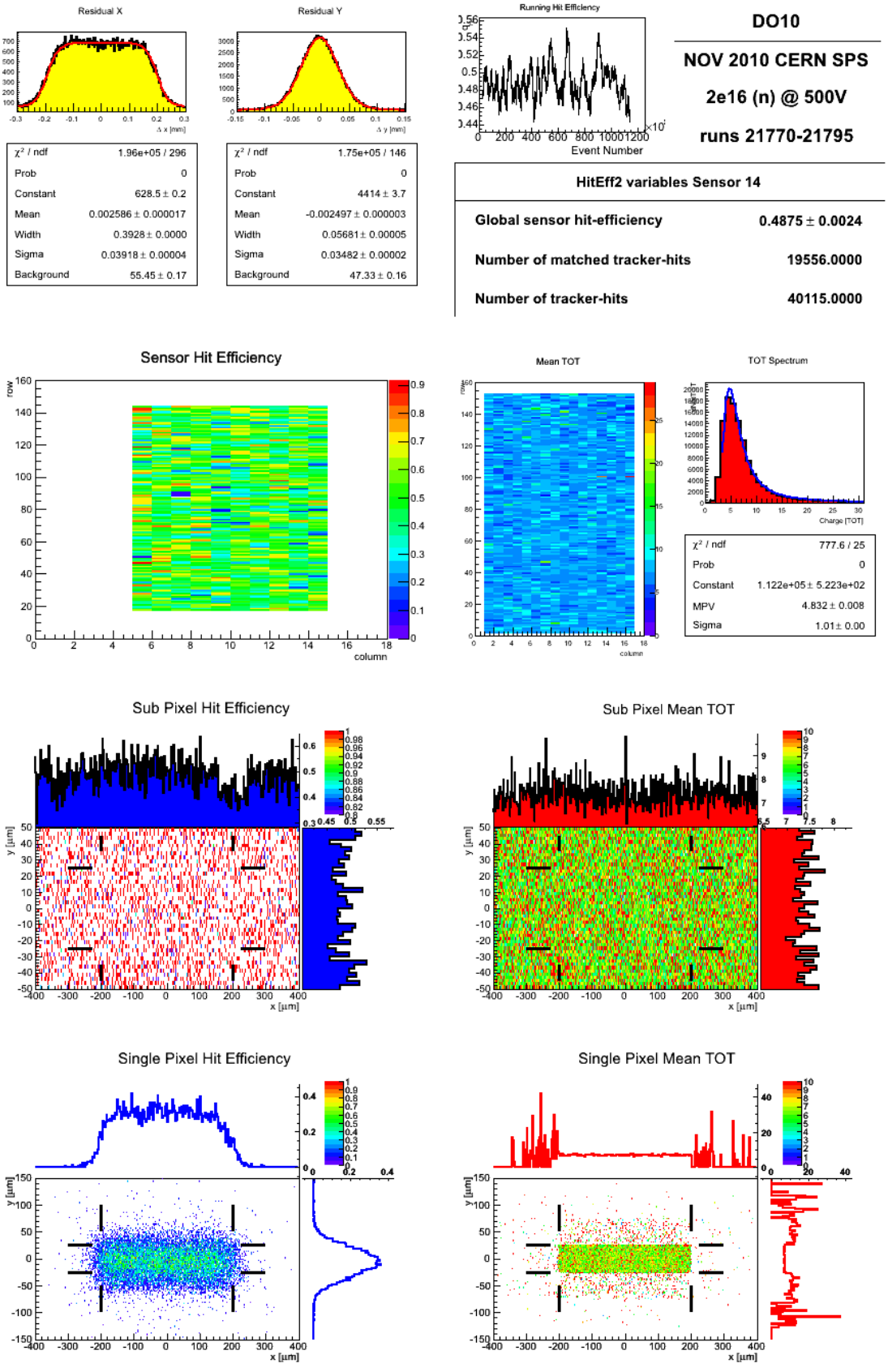


Figure C.11.: Sensor overview of sensor DO10 irradiated to $\Phi_{eq} = 2 \cdot 10^{16}\,n_{eq}\,cm^{-2}$ with neutrons, running at 500 V

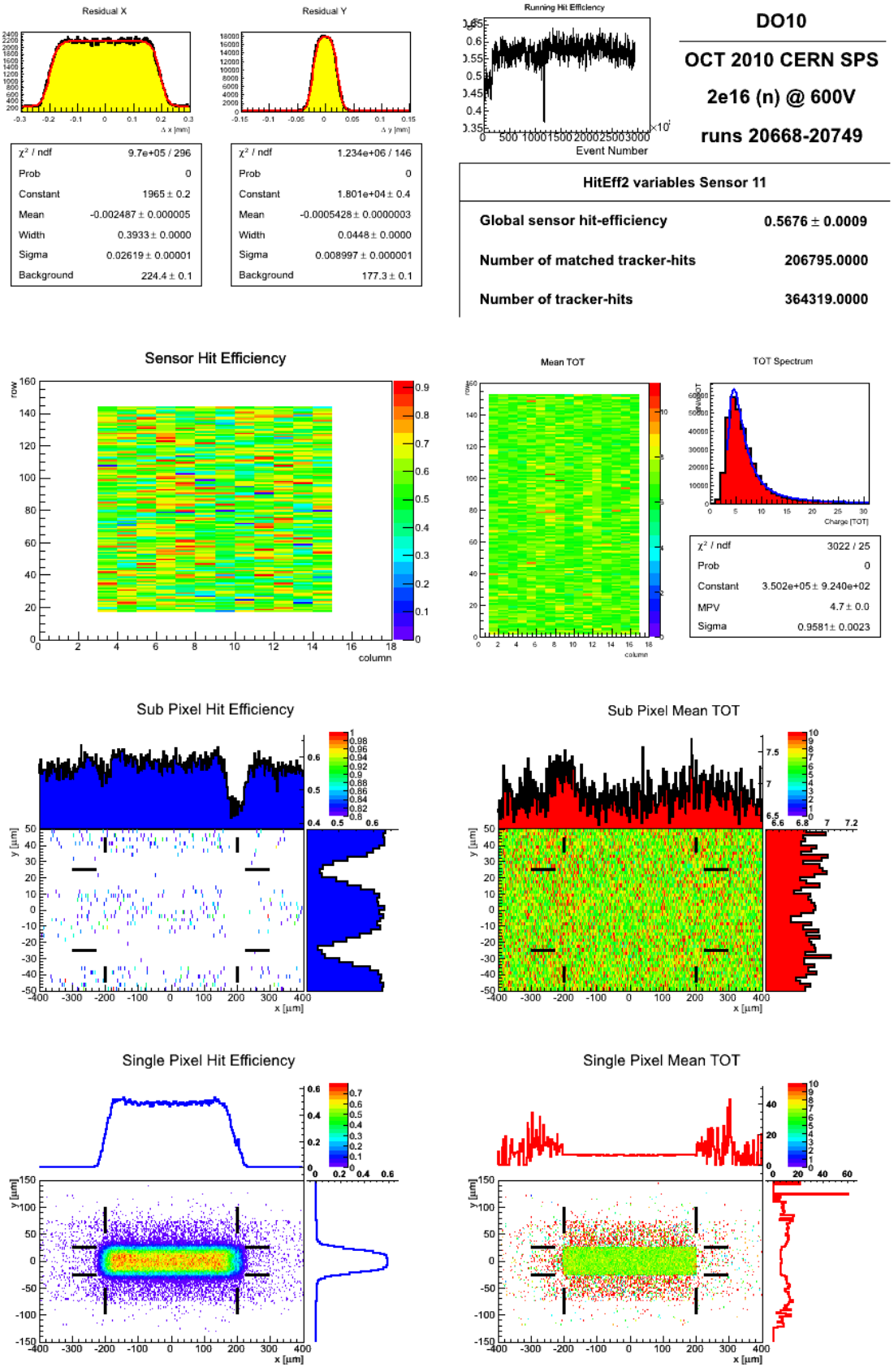


Figure C.12.: Sensor overview of sensor DO10 irradiated to $\Phi_{eq} = 2 \cdot 10^{16}\,n_{eq}\,cm^{-2}$ with neutrons, running at 600 V

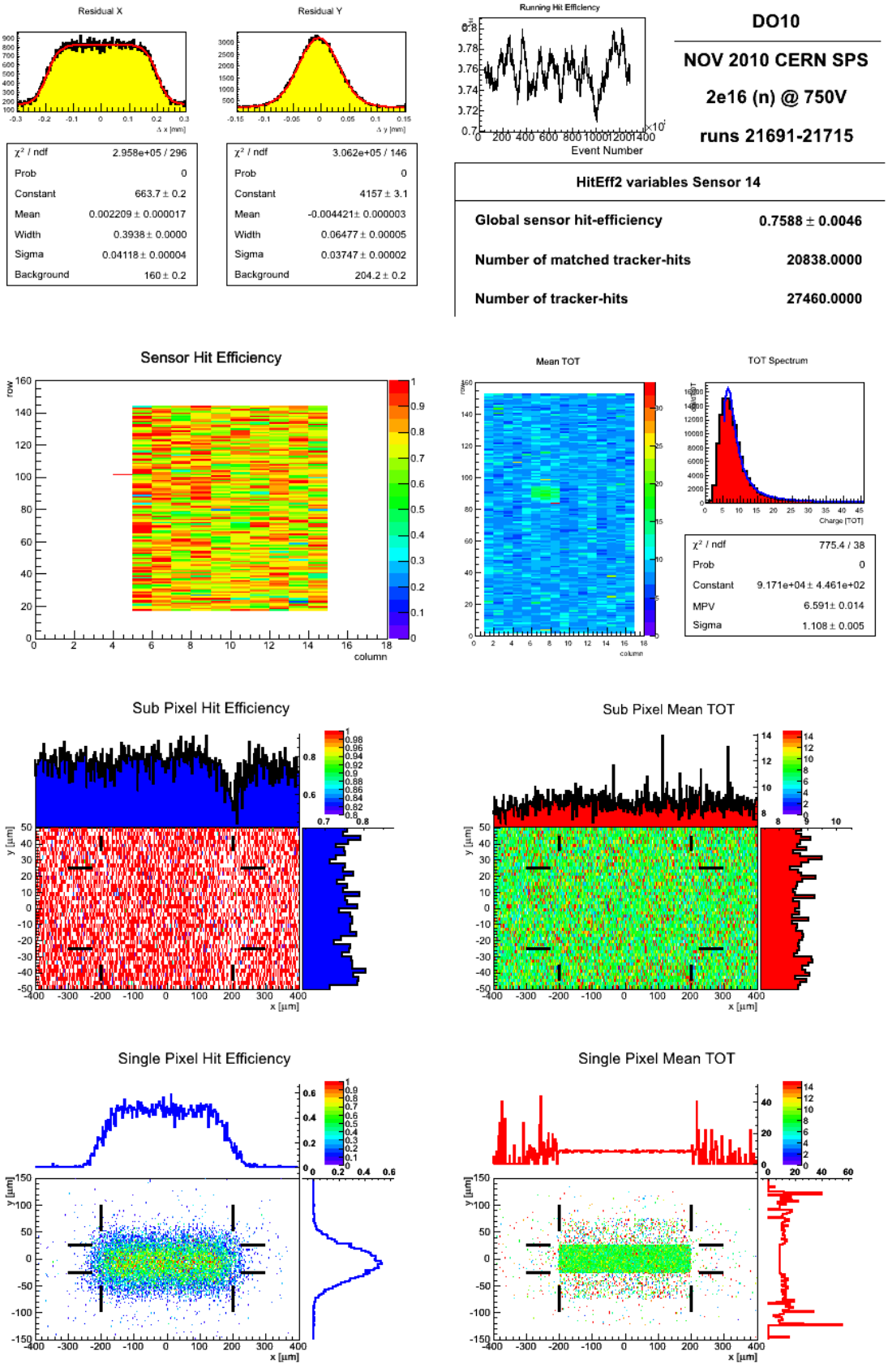


Figure C.13.: Sensor overview of sensor DO10 irradiated to $\Phi_{eq} = 2 \cdot 10^{16}\,n_{eq}\,cm^{-2}$ with neutrons, running at 750 V

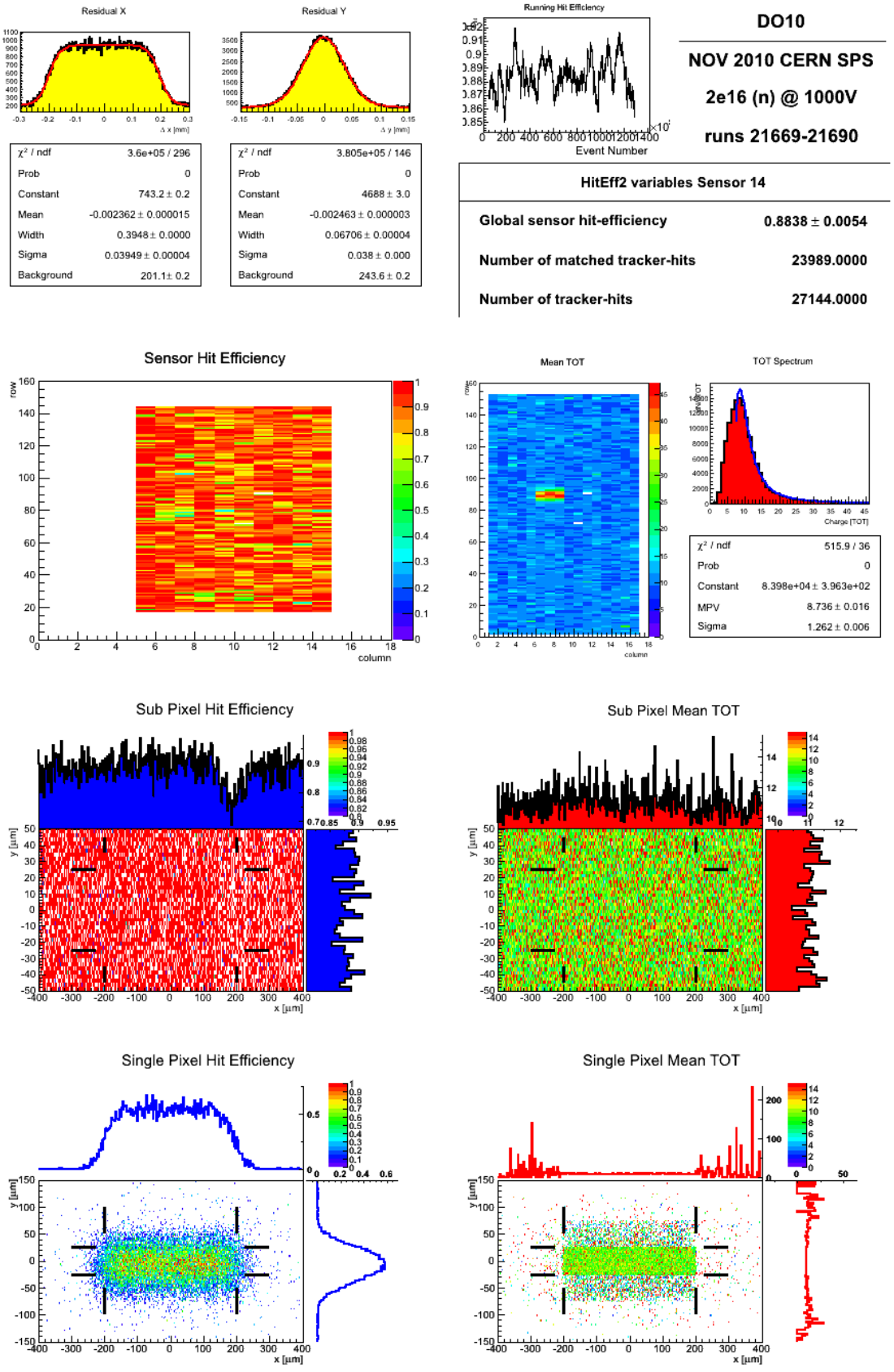


Figure C.14.: Sensor overview of sensor DO10 irradiated to $\Phi_{\mathrm{eq}} = 2 \cdot 10^{16}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons, running at 1000 V

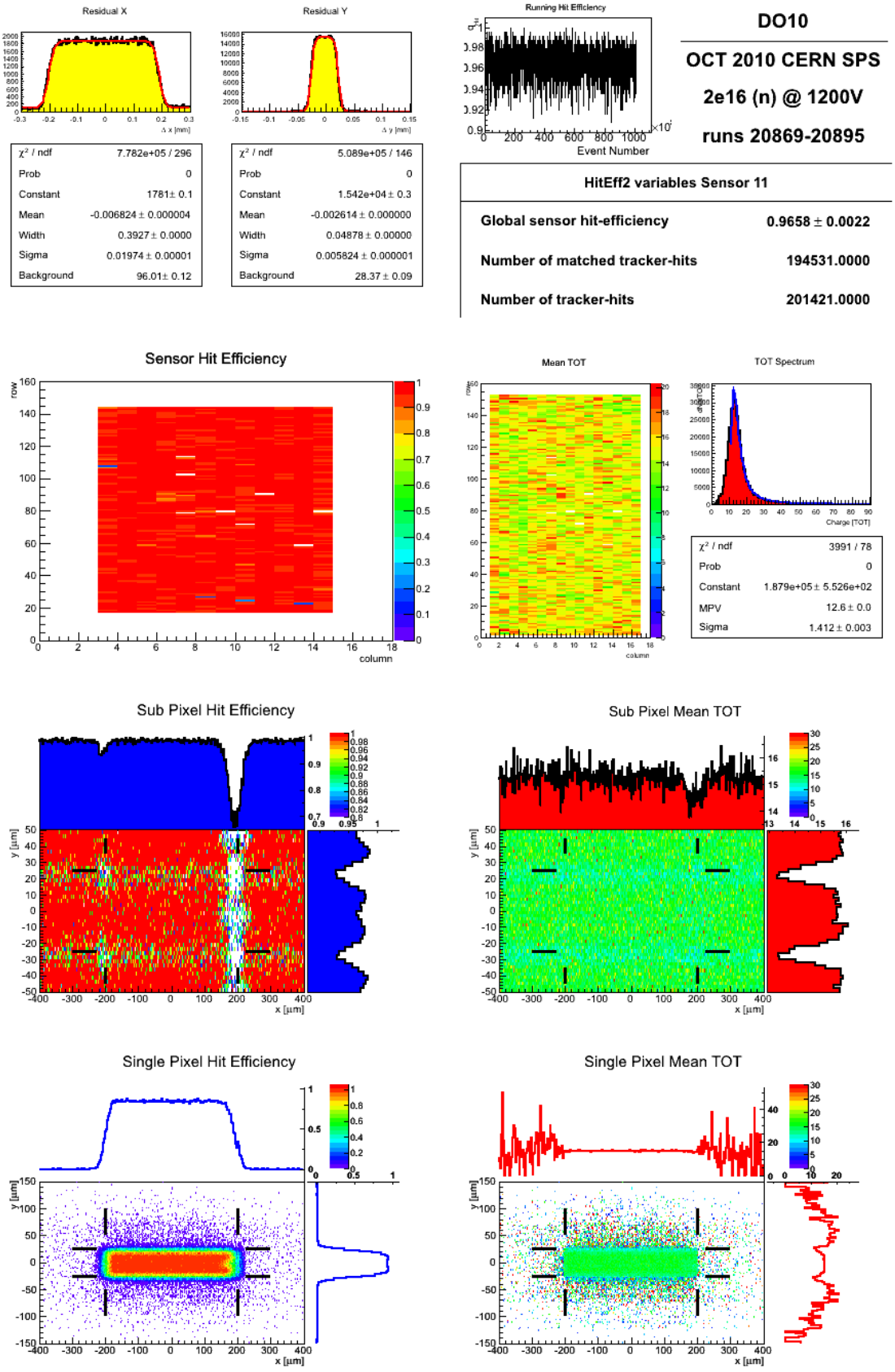


Figure C.15.: Sensor overview of sensor DO10 irradiated to $\Phi_{eq} = 2 \cdot 10^{16}\,n_{eq}\,cm^{-2}$ with neutrons, running at 1200 V

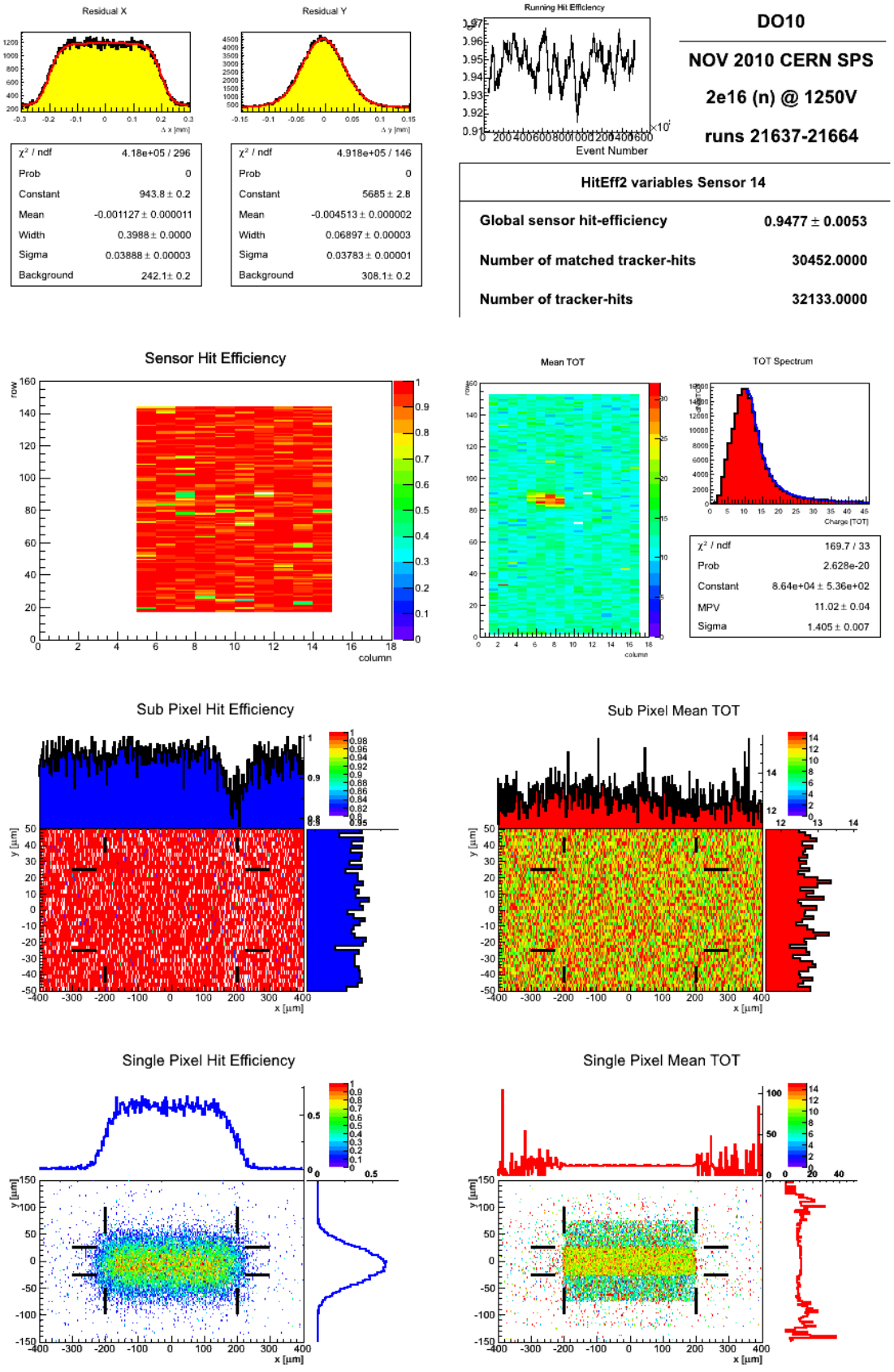


Figure C.16.: Sensor overview of sensor DO10 irradiated to $\Phi_{eq} = 2 \cdot 10^{16}\,n_{eq}\,cm^{-2}$ with neutrons, running at 1250 V

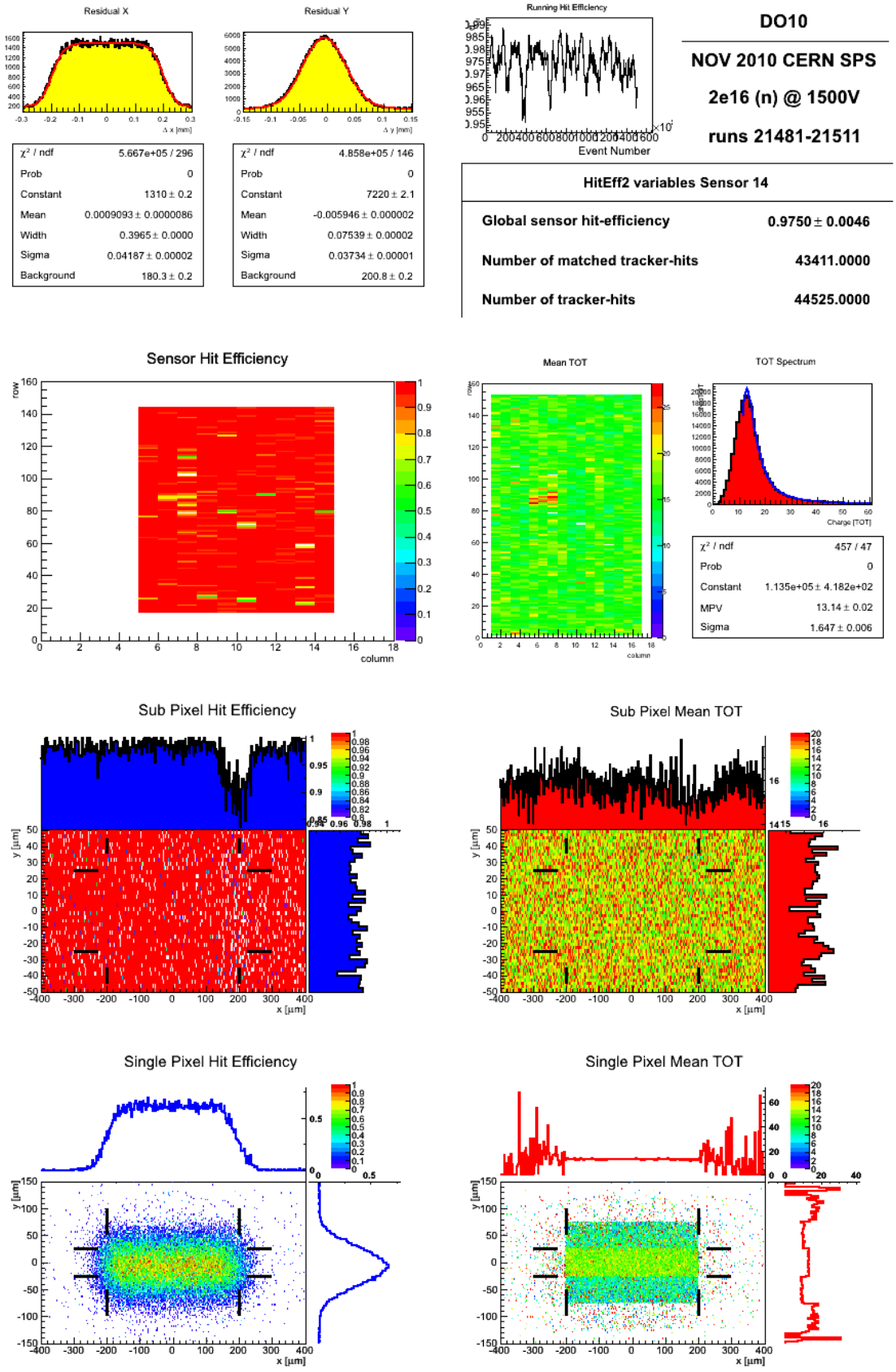


Figure C.17.: Sensor overview of sensor DO10 irradiated to $\Phi_{eq} = 2 \cdot 10^{16}\,n_{eq}\,cm^{-2}$ with neutrons, running at 1500 V

D. Sensor overview plots at non-perpendicular incidence

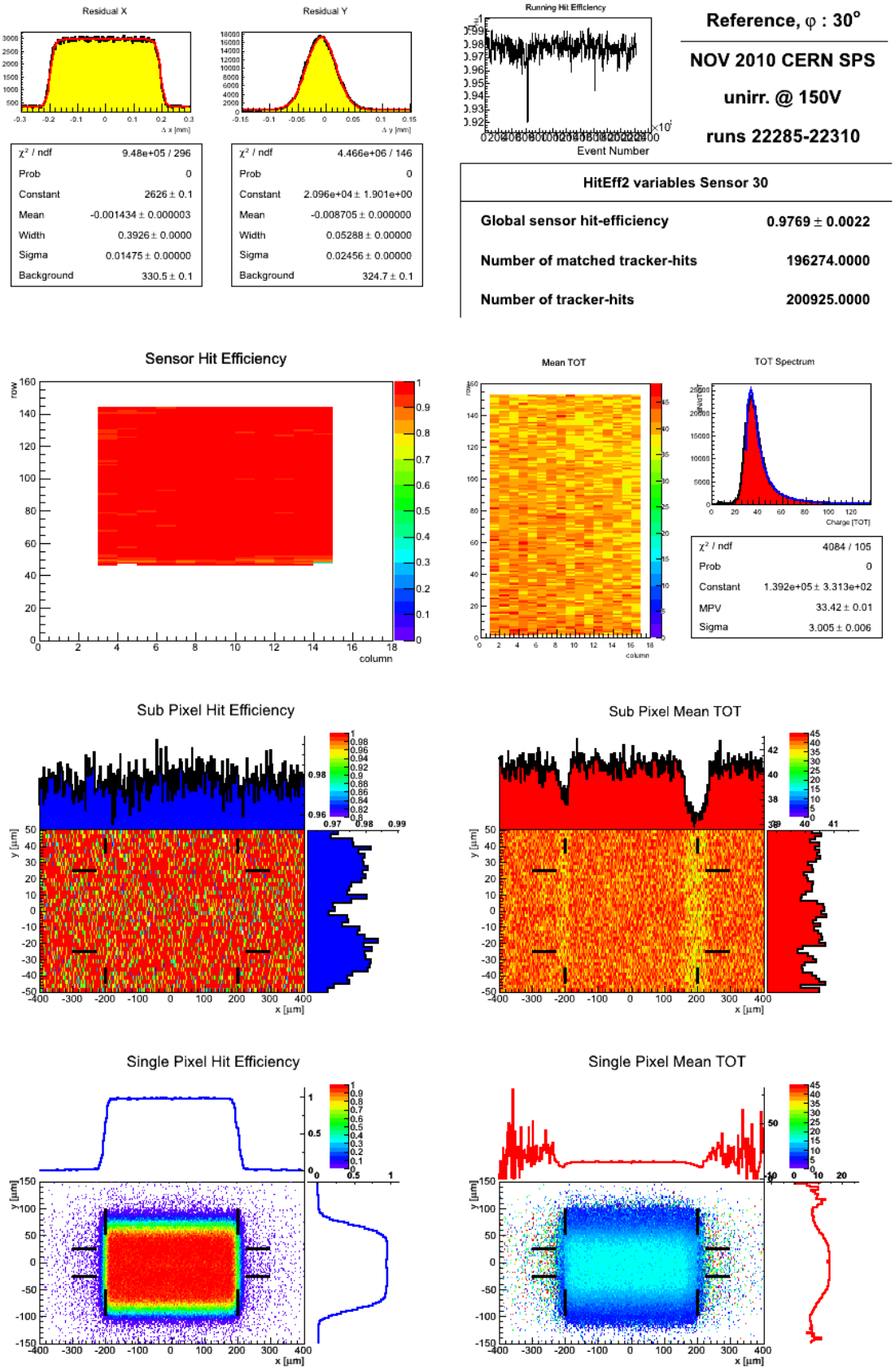


Figure D.1.: Sensor overview of non-irradiated reference sensor, running at 150V and an incidence angle of $\varphi = 30\,^{\circ}$

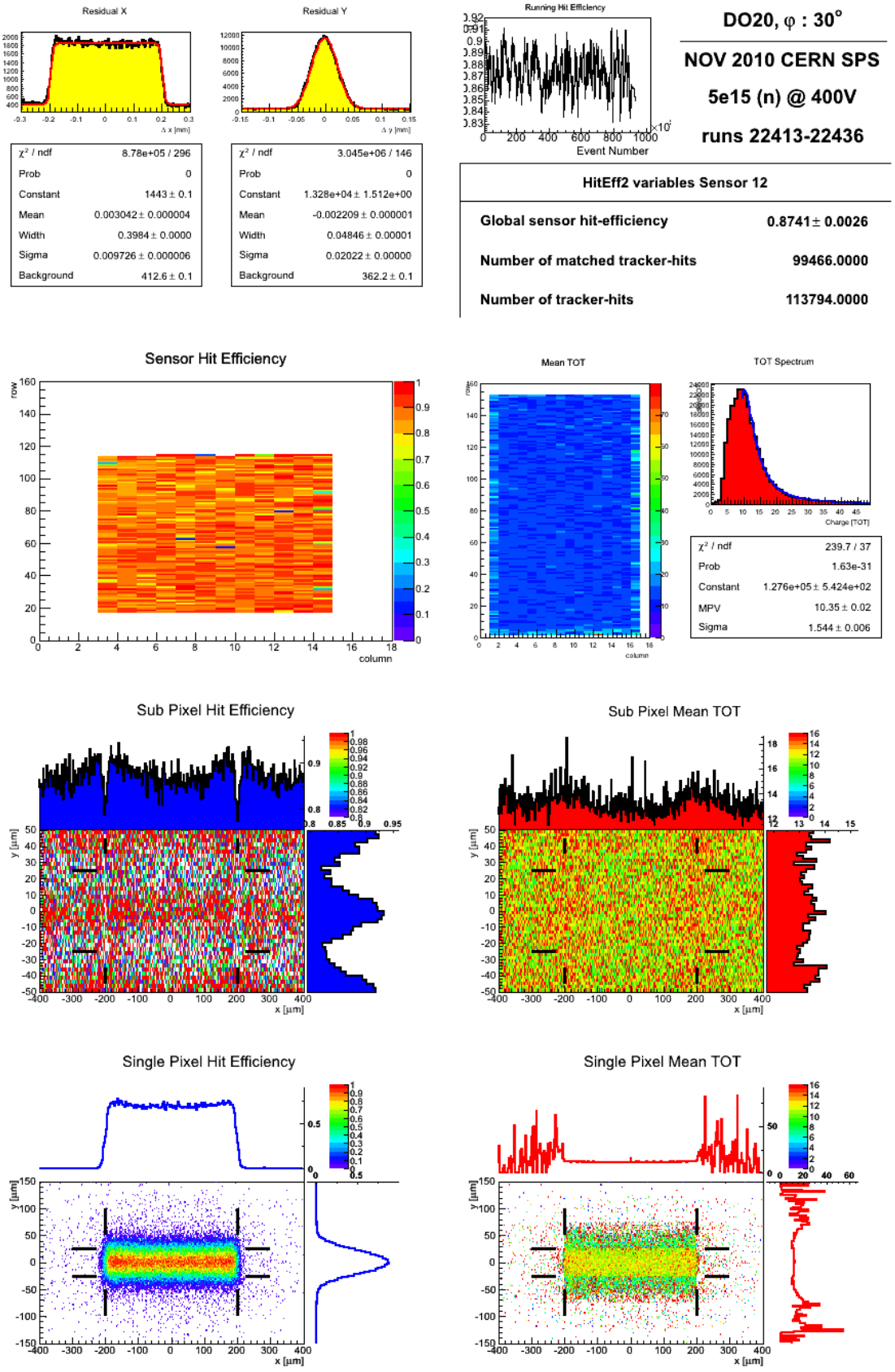


Figure D.2.: Sensor overview of sensor DO20 irradiated to $\Phi_{eq} = 5 \cdot 10^{15}\,n_{eq}\,cm^{-2}$ with neutrons, running at 400V and an incidence angle of $\varphi = 30\,°$

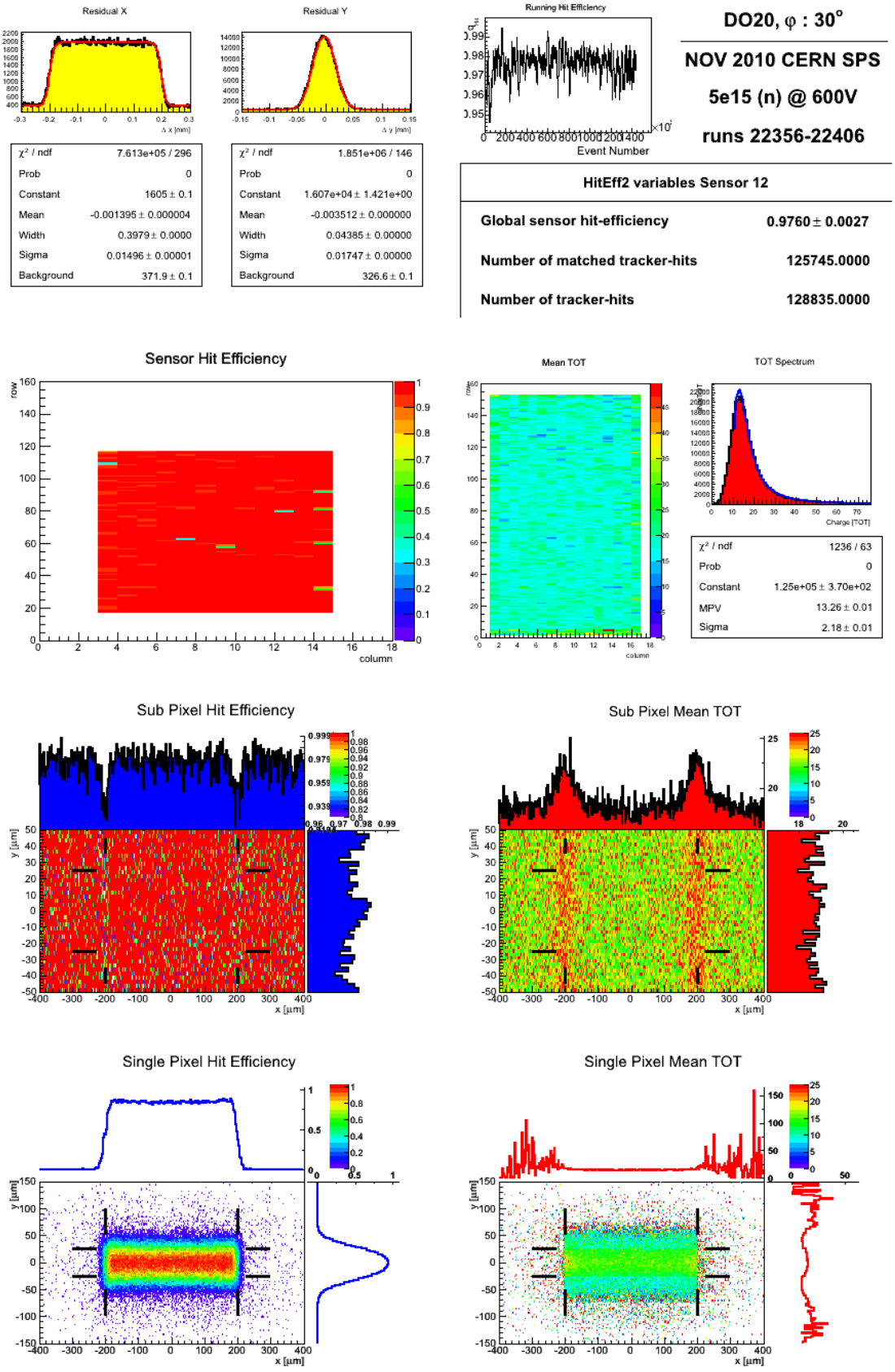


Figure D.3.: Sensor overview of sensor DO20 irradiated to $\Phi_{\mathrm{eq}} = 5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons, running at 600V and an incidence angle of $\varphi = 30\,^{\circ}$

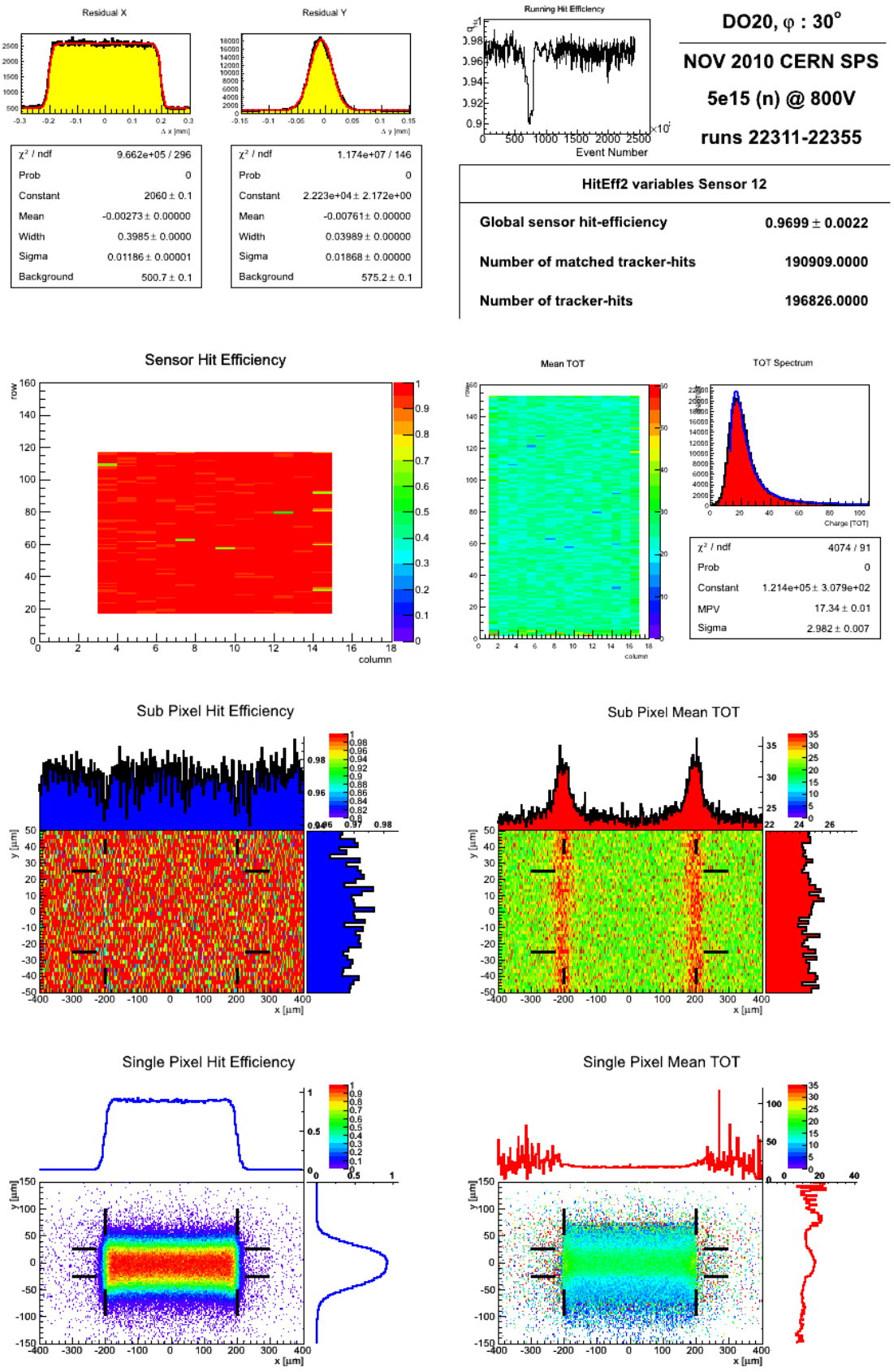


Figure D.4.: Sensor overview of sensor DO20 irradiated to $\Phi_{\mathrm{eq}} = 5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons, running at 800V and an incidence angle of $\varphi = 30\,^{\circ}$

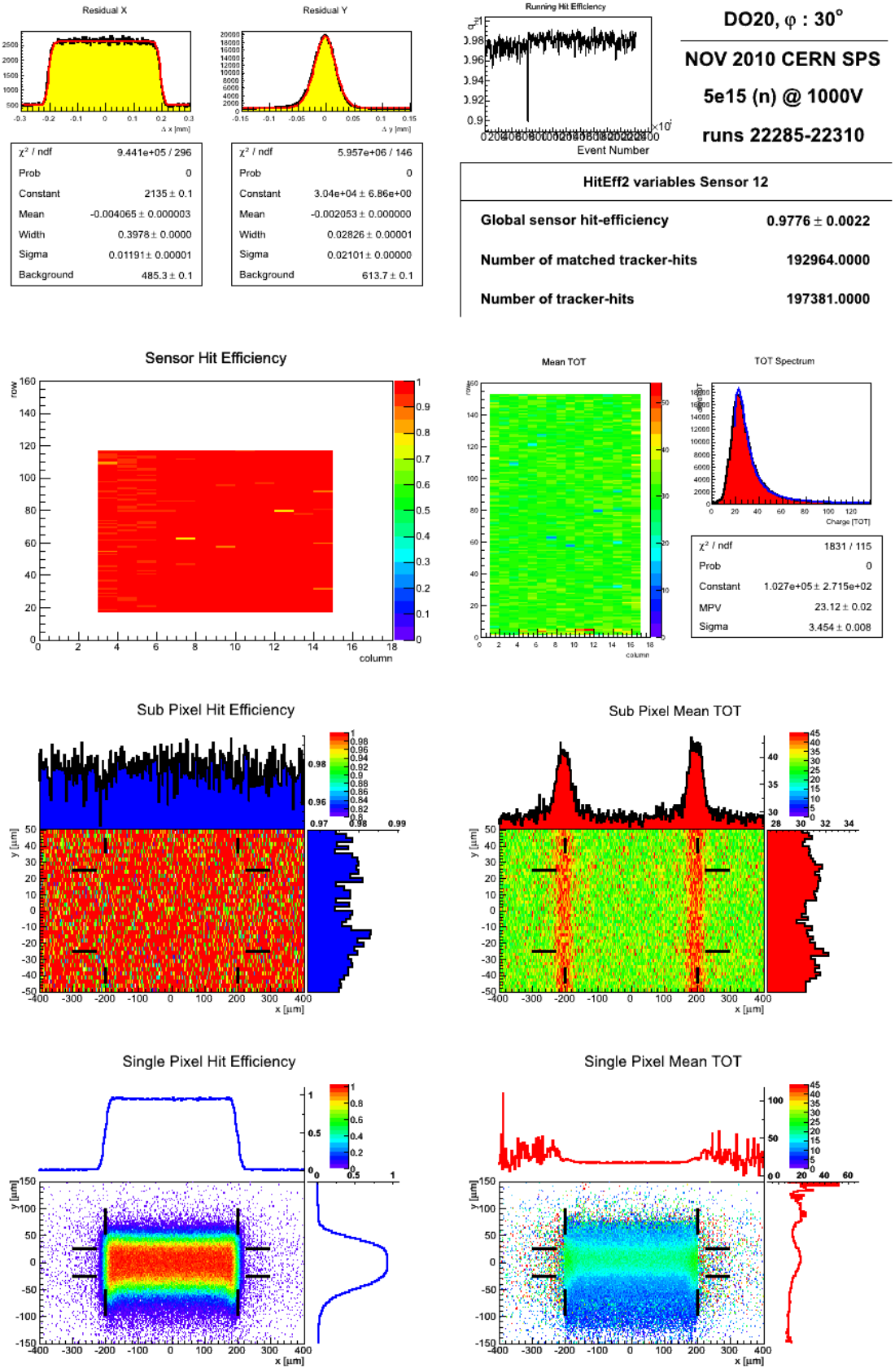


Figure D.5.: Sensor overview of sensor DO20 irradiated to $\Phi_{\mathrm{eq}} = 5 \cdot 10^{15}\,\mathrm{n_{eq}\,cm^{-2}}$ with neutrons, running at 1000V and an incidence angle of $\varphi = 30\,^{\circ}$

Index

Acknowledgements

These last pages of my thesis are addressed to all people who made this work possible:

First, I would like to express my special thanks to Prof. Dr. Claus Gößling, who was much more than a Ph.D. supervisor to me. It's due to him that it was possible for me to write this thesis. You gave me the chance to travel to CERN in the first years of my studies where I gained a lot of experience. Thank you very much for the friendly support (and some kicks in the butt) during the past years.

Thanks to Prof. Dr. Dr. Wolfgang Rhode and Dr. Bärbel Siegmann for having volunteered to be additional examiners.

My special thanks are directed to the current and former members of the ATLAS Upgrade group in Dortmund, namely Reiner Klingenberg, Jens Weber, Jens Weingarten, Markus Mathes, Silke Altenheiner, Jennifer Jentzsch, Tobias Lapsien, Till Plümer, Stephan von Kleist, Dave Sutherland and my office collegues André Rummler and Tobias Wittig. You were the everyday-environment that gave me support in the last years. Thank you very much for many discussions, meetings, rehearsals and conferences. In particular thanks to Jenny and Tobi for the proof reading.

I am grateful to all the other members of the working group Experimental Physics IV, TU Dortmund, for their excellent team work, assistance and the friendly working atmosphere.

My best wishes to Oliver Schulz for many brainstormings, discussions and assistance in teaching me how to write better source code. The intermediate talks, most often off-topic, always cheered me up.

Thank you very much to Markus Mathes, who gave me much support for setting up the first connections between TurboDAQ and EUDAQ. He taught me many of the things I know today about testbeam. Most parts of the changes at the FPGA code of the TurboDAQ's TPLL were done by him.

My gratitudes to all members of the EUDET collaboration, especially Ingrid-Maria Gregor, Emlyn Corrin, Daniel Haas, Jörg Behr, Vladislav 'Slava' Libov and Igor Rubinski. Thanks for the support, the many shifts and so much fun we had together in the testbeam area. I will never forget the newly-created life form next to the coffee machine.

Further thanks to the members of other institutes contributing the PPS and IBL testbeams. In particular I would like to thank Hâvard Gjersal, Per 'Pelle' Hanson, Ole Rhone and Alessandro LaRosa. Hâvard spent much time improving the alignment procedure of the reconstruction chain and set up easy-to-use scripts for submitting the reconstruction jobs. The first testbeam setup with an MCC board by using a ^{90}Sr source was created in team work with 'Pelle'. Ole debugged my first version of the FIFO merging algorithm, which was of course faulty at the first testbeam. When I came to CERN to modify

TurboDAQ Alessandro offered me working space in his lab and many tools - thanks for this.

I cannot send enough acknowledgements to the team of the mechanical workshop of the institute of physics at the TU Dortmund, with whose help it was possible to build the many constructions for testbeam. In addition to that, my thanks to Markus Alex and Theo Villet, for their brave missions telling the members of the mechanical workshop that we need it as soon as possible!

I would like to express my thankfulness to my family. Thank you for your support and that you are always there for me when it really matters. Thanks to my brother Christoph for the additional proof reading. And thanks to my Dad for teaching me so much about mechanics and for building many very urgent constructions in his fully equipped CNC basement workshop.

Tanja, thanks for the proof reading, but actually more thanks for the place at your side. I know it was hard for you to teach me that there is only one correct way to write 'necessary' in all this boring physics stuff. Thank you very much, I love you.

In particular, I would like to send my best wishes and thanks to the department for travel expenses at the TU Dortmund. You taught me that something is not necessarily allowed only because it's cheaper. By now even foreign people at CERN know the word 'Reisekostenstelle' and its evil meaning. Thank you very much for hundreds of Euros I paid when travelling and never got back.

Thank you to Tim Berners-Lee for offering me a possibility to do investigations so quickly. Thanks to Linus Torvalds for making better operating systems. Thanks to Steve Jobs for designing better computers. I would like to thank Alexander Bell and Thomas Watson for inventing the fundamentals for phone conferences.

I would like to send my special thanks to Dr. Samantha Carter from Stargate Command SG-1 for protecting the world so often against the Goa'uld and for telling me the practical use of an Einstein-Rosen bridge.

Thank you very much to MacGyver who inspired me to study physics in the first place.