Karlsruhe Institute of Technology

IEKP-KA/2012-15

A High-Rate Beam Test for the CMS Pixel Detector Phase I Upgrade

Simon Spannagel

Diplomarbeit

Fakultät für Physik
Institut für Experimentelle Kernphysik (IEKP)

Referent: Prof. Dr. Ulrich Husemann
Korreferent: Prof. Dr. Guido Drexlin

Karlsruhe, 2. November 2012

Deutsche Zusammenfassung

In den nächsten Jahren wird die Kollisionsenergie sowie die instantane Luminosität des Large Hadron Colliders (LHC) [EB08] stetig erhöht, um mehr Teilchenkollisionen in den einzelnen Experimenten erzeugen zu können. Besonders die höhere Energie verändert die Wirkungsquerschnitte verschiedener Reaktionen, z.B. der Higgs-Produktion. Dies ermöglicht den Experimenten, genauere Messungen von Eigenschaften der produzierten Teilchen durchzuführen. Nach der Konsolidierung des Beschleunigers im sogenannten Long Shutdown 1 (LS1) wird der LHC die doppelte Design-Luminosität von bis zu $\mathcal{L} = 2 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ bei einer Schwerpunktsenergie von 13 TeV liefern [Zim09]. Die höhere Luminosität zieht jedoch auch mehr gleichzeitig eintretende Teilchenkollisionen (Pileup) nach sich, die von den Experimenten separiert werden müssen. Momentan muss der Compact Muon Solenoid (CMS) Detektor [C$^+$08] bei einer instantanen Luminosität von $\mathcal{L} = 7.5 \times 10^{33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ die einzelnen Teilchenspuren von ungefähr 35 simultanen Kollisionen trennen; bei doppelter Design-Luminosität sowie höherer Schwerpunktsenergie und größeren Wirkungsquerschnitten wird diese Zahl jedoch auf ca. 100 Pileups ansteigen [CMS12]. Um diesen ständig steigenden Anforderungen gerecht zu werden, müssen Teile des CMS-Detektors verbessert oder sogar erneuert werden. Beispielsweise wird der Pixeldetektor im sogenannten *Phase I Upgrade* Ende 2016 vollständig ersetzt. Sowohl das Silizium-Sensormaterial muss aufgrund der Strahlungsschäden ausgetauscht werden, als auch der Auslesechip (Read Out Chip, ROC), der die Signalverarbeitung übernimmt. Der aktuelle ROC kann bei höherer Okkupanz nicht ohne erhebliche Datenverluste und Ineffizienzen betrieben werden, da zum einen nicht genügend Speicherzellen zum Zwischenspeichern der getroffenen Pixel und zum anderen nicht genug Bandbreite zum Übertragen des erhöhten Datenaufkommens vorhanden sind [Käs08, Mei11]. Aus diesen Gründen wurde ein neuer Auslesechip konstruiert. Der sogenannte *PSI46dig* ROC verfügt über einen größeren Zwischenspeicher und kann die Übertragung der anfallenden Datenmengen über eine digitale 400 MHz Schnittstelle bewältigen sowie die Auslese-Totzeit reduzieren. Auch der Token Bit Manager (TBM), der die Daten von mehreren ROCs bündelt und für die Trigger-Signalverarbeitung zuständig ist, muss entsprechend angepasst und erweitert werden.

Die Effizienz und das Verhalten des neuen *PSI46dig* ROCs müssen unter Betriebsbedingungen genau getestet und verstanden werden, bevor dieser in Serienproduktion gehen kann. Sogenannte *Strahltests*, also Detektortests in einem Teilchenstrahl, bieten einen geeigneten Weg, um neue Detektor-Komponenten zu qualifizieren. Der zu untersuchende Detektor (auch Device Under Test (DUT) genannt) wird dabei in einem Teilchenstrahl mit genau definierten Eigenschaften und Charakteristika platziert, der dazu dient, die Betriebsbedingungen im CMS-Detektor nach dem Phase I Upgrade zu simulieren. Üblicherweise kommen hierfür sogenannte Strahlteleskope zum Einsatz, die aus mehreren Sensorlagen bestehen. Teilchenspuren werden vom Teleskop aufgenommen und mit den Messungen im DUT verglichen. Dieser Vergleich erlaubt Rückschlüsse auf die Effizienz des untersuchten Detektors. Um die Effizienz des neuen CMS ROCs bei hohen Teilchenraten zu vermessen, wurde mehrere sogenannte Hochraten-Strahltest geplant. Der erste dieser Tests wurde im Juli 2012 durchgeführt, ein weiterer ist für Ende Oktober 2012 geplant. Für beide Tests wird der Strahlplatz *H4IRRAD* [BC11] des Super Proton Synchrotron (SPS)-Beschleunigers am CERN verwendet, der Protonen mit einer Energie von etwa 320 GeV bereitstellt. Der erste der beiden

Strahltests am SPS wurde Ende Juli 2012 durchgeführt und diente hauptsächlich der Qualifizierung des Teilchenstrahls, der neu entwickelten Strahlteleskope sowie der Ausleseelektronik. Zudem wurde die Leistungsfähigkeit der Analysesoftware mithilfe der aufgenommenen Detektordaten überprüft und verbessert. Zwei verschiedene Teleskopgeometrien mit jeweils acht ROCs wurden entwickelt und gebaut. Während bei der ersten alle Teleskopebenen senkrecht auf dem eintreffenden Teilchenstrahl stehen, sind diese in der zweiten Version in zwei Richtungen gegen den Strahl geneigt, damit durchfliegende Teilchen nicht nur einzelne, sondern mehrere Pixel treffen. Diese können in der Analyse zu sogenannten Clustern verbunden und den einzelnen Teilchen zugeordnet werden können. Dies dient der Verbesserung der Ortsauflösung und simuliert die Bedingungen im CMS-Detektor, wo eintreffende Teilchen durch das Magnetfeld abgelenkt werden und auch die im Silizium erzeugte Ladungsträger in einem Winkel von etwa 25° zur Sensoroberfläche driften [CMS10]. Dieser Winkel wird entsprechend der verursachenden Kraft *Lorentzwinkel* genannt. Beide Teleskope werden ohne TBM betrieben und direkt von den Ausleseplatinen angesprochen, da der digitale TBM08 noch nicht produziert wurde. Im Juli-Strahltest wurden zwei gekoppelte Ausleseplatinen (auch *Testboards* genannt) verwendet. Das Altera Cyclone Testboard wurde vom Paul-Scherrer-Institut (PSI) für Labortests des analogen *PSI46v2* ROCs entwickelt und kann deshalb nicht ohne Modifikationen zur Auslese der digitalen Chips verwendet werden. Deshalb wurde zusätzlich ein auf dem leistungsstarken Xilinx Spartan-6-Prozessor beruhendes Testboard [Xil12] verwendet, um die Detektordaten aufzubereiten. Für den Strahltest werden die Teleskope mit 2 mm × 2 mm großen Szintillatoren vor und hinter der ersten bzw. letzten Teleskopebene ausgestattet, die es erlauben, die Auslese des Teleskops nach einem Teilchendurchgang zu starten. Diese Auslesesignale werden üblicherweise *Trigger* genannt und von der Trigger-Logikeinheit verarbeitet. Diese wurde für die Strahltests so konstruiert, dass Beginn und Ende des gepulsten Teilchenstrahls vom SPS erkannt und alle Triggersignale außerhalb dieses Zeitfensters ausgeblendet werden können. Die Datennahme wird mit jedem neuen Puls gestartet und dauert 7 s. Dies ermöglicht sowohl eine automatisierte Datennahme als auch eine einfache Korrelation zwischen gespeicherten Triggersignalen und Detektordaten.

Zur Analyse der Detektordaten wird die EUTelescope-Softwareumgebung eingesetzt [B$^+$07b]. EUTelescope ist ein Programmpaket, das speziell für die Analyse von Teleskop-Strahltests entwickelt wurde, in das ILCsoft Framework eingebettet ist und dessen Infrastruktur nutzt. Ein Vorteil von EUTelescope ist sein modularer Aufbau, in dem einzelne Softwareprozessoren dedizierte Aufgaben in der Analysekette übernehmen und die verarbeiteten Daten anschließend an den nächsten Prozessor weitergeben. Dies erlaubt einen schnellen und flexiblen Aufbau von vollständigen Teilchenrekonstruktionen und Analyseketten. Lediglich das Dekodieren und Konvertieren der nativen Detektordaten in das verwendete Linear Collider I/O (LCIO) Datenformat muss vom jeweiligen Anwender des Pakets implementiert werden, was im Rahmen dieser Arbeit vorgenommen wurde. Mit dem CMSPixelDecoder wurde eine einheitliche Programmbibliothek für schnelles Dekodieren der Datenströme von CMS Pixel-ROCs entwickelt. Dieser ist sowohl für das Dekodieren der Daten von einzelnen *PSI46* ROC in Laboraufbauten als auch der Datenströme von vollen Detektormodulen mit 16 ROCs oder den Strahlteleskopen ausgelegt. Der Dekodieralgorithmus wurde sowohl mit Daten aus Labortests als auch mit Daten, die während des Juli-Strahltests aufgenommen wurden, überprüft und die Dekodierstrategie für eine maximale Datenintegrität angepasst. Der CMSPixelDecoder kann sowohl Daten aus analogen *PSI46v2* Chips als auch digitale *PSI46dig* Daten dekodieren, beide ROCs können hierfür sowohl mit als auch ohne TBM betrieben werden. Dies erlaubt nicht nur die Nutzung in den Hochraten-Strahltests, sondern eröffnet zahlreiche Möglichkeiten des Einsatzes in den verschiedenen Labormessungen oder anderen Strahltests der *PSI46* ROCs. Um die Dekodierprozedur beeinflussen zu können, werden mehrere Parameter bereitgestellt. Diese ermöglichen beispielsweise die Auswahl von einzelnen Ereignissen basierend auf bestimmten Qualitätskriterien wie fehlerfreien ROC-Kennungen oder korrekter Datenlänge des Ereignisses. Des Weiteren werden detaillierte Statistiken über den Dekodiervorgang erhoben und ausgegeben. Dies umfasst sowohl die Gesamtzahl der

Ereignisse als auch der Zahl der einzelnen Pixeltreffer, fehlgeschlagener Adressdekodierungen oder detektierter ROC-Kennungen. Dies kann besonders nützlich sein, um Fehler in der Detektorauslese oder der Testboard-Firmware aufzuspüren.

Viele der bereits vorhandenen EUTelescope-Prozessoren konnten für die Analysekette verwendet werden wobei lediglich kleine Änderungen vorgenommen werden mussten. Der CMSPixelReader-Prozessor zum Konvertieren des nativen Datenformats in das LCIO-Format wurde im Rahmen dieser Arbeit von Grund auf entwickelt. Er verwendet intern den CMSPixelDecoder zum Dekodieren der Rohdaten. Nach der Konvertierung werden die Daten mit Hilfe des CMSPixelCalibrateEvent-Prozessors kalibriert. Dieser nimmt eine Kalibrierung der einzelnen Signalhöhen anhand einer aufgenommenen Kalibrationskurve für jedes einzelne Sensorpixel vor. Das Gruppieren der einzelnen Pixeltreffer (auch *Clustering* genannt) wird von einem sogenannten *sparse clustering*-Algorithmus vorgenommen, der alle zusammenhängenden Pixeltreffer findet und zu Clustern zusammenfasst. Der Prozessor wurde für den Hochraten-Strahltest angepasst und erweitert und kann beispielsweise Pixel, die zu oft oder dauerhaft ansprechen (*Hot Pixels*) von seiner Clustersuche und dem Ausgabedatensatz ausschließen. Native EUTelescope-Prozessoren übernehmen alle folgenden Analyseschritte, wie zum Beispiel Energiegewichtung der Cluster und Bestimmung der genauen Trefferposition sowie die eigentliche Spurfindung. Dies erlaubt die Verwendung von komplexen, gut getesteten Algorithmen, wie beispielsweise dem Millepede II-Paket für die präzise Ausrichtung der einzelnen Teleskoplagen gegeneinander. Monte Carlo-Simulationen wurden verwendet, um die Funktionalität der einzelnen Schritte zu prüfen und zu optimieren sowie das Verhalten der gesamten Analysekette unter verschiedenen Bedingungen zu untersuchen. Dies beinhaltete zum Beispiel Vorbereitungen für die Effizienzmessung am DUT ebenso wie die Simulation von Hot Pixels.

Durch die neue und weitestgehend ungetestete Hard- und Software entstanden während des Juli-Strahltests einige Probleme. Beispielsweise konnten aufgrund eines Fehlers in der Firmware des Altera-Testboards keine Triggersignale der Szintillatoren verarbeitet werden und es musste auf den internen Triggergenerator des Testboards zurückgegriffen werden. Dieser erzeugt Triggerimpulse mit einer festen Frequenz, die jedoch nicht mit dem Durchgang von Teilchen korreliert sind (*Zufallstrigger*). Das Hauptproblem des Juli-Strahltests stellen jedoch die aufgetretenen Auslesefehler in den Rohdaten dar. Abhängig von den gewählten Qualitätskriterien im CMSPixelDecoder konnten bis zu 6 % der Pixeltreffer innerhalb einer Datennahme nicht korrekt dekodiert werden. Die Fehler traten sowohl in den ROC-Kennungen als auch in den einzelnen Datensequenzen der Pixeltreffer auf. Während die ROC-Kennungen jedoch aufgrund der einheitlichen Deformation in vielen Fällen rekonstruiert werden konnten, ist dies für die Pixeltreffer nicht möglich. Die Datenqualität der analysierten Ereignisse konnte verbessert werden, indem ausschließlich Ereignisse, die festgelegten Kriterien (keine Bitfehler in den ROC-Kennungen, richtige Anzahl an ROCs) entsprachen, ausgewertet und alle anderen verworfen wurden. Hierdurch konnte die Fehlerrate auf etwa 2 % gesenkt werden. Es wird angenommen, dass die meisten Auslesefehler von falschen Phaseneinstellungen und unzureichender Taktsynchronisierung zwischen den beiden eingesetzten Testboards herrühren. Dies muss jedoch mit den Daten des Oktober-Strahltests überprüft werden. Mit der momentanen Datenqualität lässt sich noch keine Aussage über die Effizienz des untersuchten ROCs treffen. Um dies zu ermöglichen, müssen die anstehenden Strahltests qualitativ bessere Daten und höhere Statistik bereitstellen.

Um den verwendeten Protonenstrahl für Hochraten-Tests zu qualifizieren, wurde die Teilchenrate gemessen. Hierzu wurden die Triggerdaten, die Detektoren am Strahlplatz sowie ein *Gafchromic*-Film verwendet und die Raten für jede einzelne Teleskoplage berechnet. Wie in Abbildung 1 gezeigt, konnten Raten von bis zu 250 MHz auf der Sensorfläche bzw. eine Flussdichte von 400 MHz/cm^2 erreicht werden. Dies entspricht einem Durchgang von ungefähr sechs Teilchen pro Taktzyklus von 25 ns und erfüllt die Voraussetzungen für den Hochraten-Test. Das eingesetzte CMS-Strahlteleskop war imstande, Teilchen zu verfolgen, auch wenn die Gesamtzahl der Spuren unter den Erwartungen

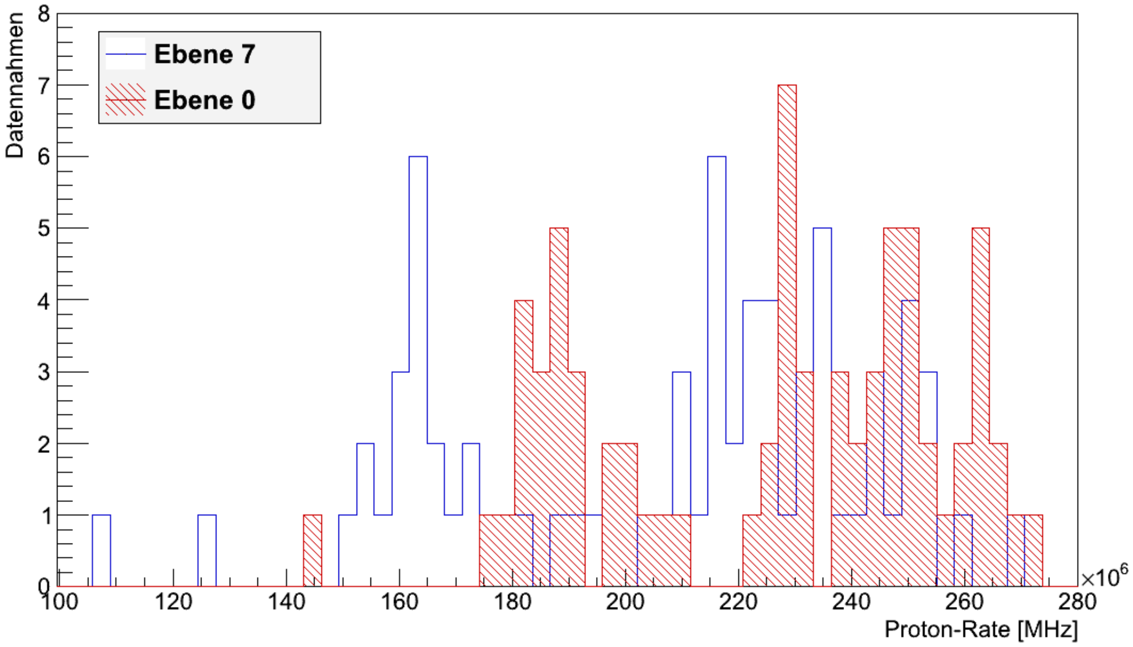
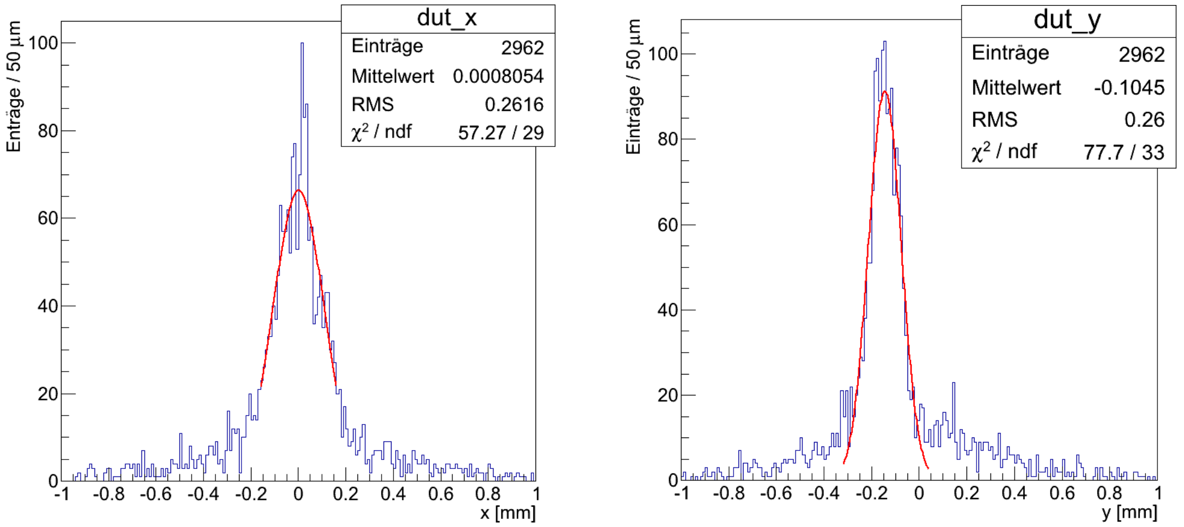


Abbildung 1: Zahl der Datennahmen mit einer bestimmten Protonenrate auf den Detektor-Ebenen. Insgesamt sind 73 Datennahmen dargestellt. Zwei verschiedene Positionen des Strahlteleskops können über die beiden Maxima bei 190 MHz und 250 MHz unterschieden werden. Die hohe Rate von ca. 250 MHz bestätigt die Eignung des Teststrahls für Hochraten-Messungen. Weiterhin erkennt man die leichte Neigung des Teleskops in Bezug zum Strahl durch die verschiedenen Raten in den Ebenen 0 und 7.

blieb. Dies lag vor allem daran, dass keine Triggersignale von den Szintillatoren verwendet werden konnten. Die Fluenz durch das Strahlteleskop während des Juli-Strahltests konnte mit Hilfe der Messplatzinstrumentierung auf 4×10^{13} Protonen geschätzt werden. Aufgrund der Strahlungsschäden im Sensormaterial und dem entsprechenden Anstieg des Leckstroms müssen für den Oktober-Strahltest neue Sensoren verwendet werden. Ansonsten würden sich die Sensoren bei einer Fluenz von etwa 8×10^{13} durch den gestiegenen Leckstrom und die fehlende Kühlung stark aufheizen und einen weiteren Anstieg des Stroms verursachen (*Thermal Runaway*).

Die EUTelescope-Software konnte ihre Leistungsfähigkeit bereits bei der relativ geringen Menge an aufgenommenen Daten beweisen. Teilchenspuren konnten rekonstruiert und visualisiert werden. Ortsauflösungen auf dem DUT von $\sigma_x \approx 0.10\,\text{mm}$ und $\sigma_y \approx 0.07\,\text{mm}$ wurden erreicht, wie in Abbildung 2 dargestellt. Dies entspricht nicht ganz den theoretischen Auflösungen für binäre Auslese, die sich über $\sigma = p/\sqrt{12}$ ergeben. Hierbei ist p die Pixelgröße (*Pitch*) in der jeweiligen Richtung. Die Auflösung kann somit für die Pixelgröße des CMS Pixeldetektors zu $\sigma_x = 0.043\,\text{mm}$ und $\sigma_y = 0.029\,\text{mm}$ bestimmt werden. Die Hauptgründe für die niedrigere Auflösung liegen bei den Problemen mit der Nutzung externer Trigger-Signale und der daraus resultierenden relativ geringen Statistik von lediglich 3000 Teilchenspuren in den ca. 10 000 Ereignissen einer Datennahme. Dies limitiert die Präzision der Ausrichtung der Teleskoplagen und damit die Ortsauflösung auf den einzelnen Sensoren. Trotzdem konnten einzelne Ereignisse mit bis zu sechs korrekt rekonstruierten Spuren gefunden werden.

Im zweiten, für Ende Oktober 2012 geplanten Strahltest wird ausschließlich das Xilinx-Testboard für die Datennahme eingesetzt werden, da die entsprechende Funktionalität für die Programmierung der ROCs in die Xilinx-Firmware implementiert wird. Das Altera-Testboard wird somit nicht mehr benötigt. Durch die wegfallende Synchronisation sollte sich die Datenqualität erheblich verbessern



(a) DUT-Residuum in x

(b) DUT-Residuum in y

Abbildung 2: Residuen der DUT-Teleskoplage, berechnet aus Daten des Juli-Strahltests RUN000157. (a) zeigt das Residuum in x mit einem einer angepassten Gauß-Verteilung und einer Standardabweichung von $\sigma_x = 0.106 \pm 0.005\,\text{mm}$. (b) zeigt das Residuum in y mit $\sigma_y = 0.069 \pm 0.002\,\text{mm}$. Das Residuum in y ist nicht ganz zentriert, was von einer nicht korrigierten Verschiebung des DUT herrührt. Beide Standardabweichungen stimmen ungefähr mit 2/3 der Pixel-Abmessungen in x und y überein, sind jedoch noch größer als von der Detektorauflösung erwartet.

und die Zahl der Bitfehler im Datenstrom deutlich zurückgehen. Des Weiteren werden eine Reihe von Verbesserungen am Versuchsaufbau in den Oktober-Strahltest Eingang finden. Beispielsweise wird der Leckstrom der Sensoren kontinuierlich beobachtet und aufgezeichnet werden. Die Auslese des Teleskops wird über optische Fasern mithilfe des IPBus-Protokolls [M$^+$11] durchgeführt werden, was höhere Übertragungsgeschwindigkeiten der Daten zulässt. Zudem wird erstmals das Strahlteleskop mit geneigten Ebenen zum Einsatz kommen. Die Ein-Pixel-Cluster die in den Sensorlagen des geraden Teleskops erzeugt werden, füllen zwar den Zwischenspeicher für Zeitstempel, nicht jedoch den eigentlichen Speicher für Pixeltreffer, der über mehr Speicherzellen verfügt. In den geneigten Sensoren werden Mehr-Pixel-Cluster erzeugt, die die beiden Zwischenspeicher im richtigen Verhältnis füllen und so eine Untersuchung der Effizienz zulassen. Das Verhalten der ROCs unter verschiedenen Bedingungen wie Strahlintensität oder zeitlicher Koordination der Auslese- und Triggersignale wird untersucht und Effizienzmessungen werden vorgenommen werden. Der Oktober-Strahltest wird die letzte Gelegenheit für einen Strahltest am CERN sein, bevor die Beschleuniger für den LS1 heruntergefahren werden. Weitere Hochraten-Tests des *PSI46dig* und des TBM08 werden vermutlich an einem im Bau befindlichen Strahlplatz am Fermi National Accelerator Laboratory (FNAL) in den USA durchgeführt werden. Beide Bauteile müssen Ende 2013 in Produktion gehen, damit die Serienproduktion der Detektormodule für den neuen CMS Pixeldetektor im ersten Quartal 2014 beginnen kann.

Das EUTelescope Analyse-Framework und die Prozessoren werden unabhängig von den Hochraten-Tests weiterentwickelt und verbessert. Weitere Arbeitsgruppen in der CMS-Kollaboration haben bereits Interesse geäußert, die für CMS-Sensoren adaptierte Software einsetzen zu wollen, wie beispielsweise Strahltests am CERN und DESY. Zudem wird der CMSPixelDecoder momentan in das C++ Framework für die IPBus-Kommunikation und -Auslese der Pixel-ROCs implementiert und für das neue Datenformat angepasst. Dies ermöglicht es, mit dem CMSPixelDecoder auch Datenströme zu dekodieren, die mit Hilfe des IPBus-Protokolls und dem Xilinx-Testboard aufgenommen wurden. Er wird somit in den Hochraten-Tests als zentrale Dekodiereinheit eingesetzt, sowohl in der Kalibration und Programmierung der Chips als auch in der späteren Analyse der Daten. Darüber hinaus werden kleine Änderungen am CMSPixelReader es ermöglichen, native *PSI46*-Daten direkt in den LCIO-Datenstrom anderer Strahlteleskope, wie beispielsweise den EUDET-Teleskopen [Cor09], zu integrieren.

Die vorliegende Arbeit gibt eine Einführung sowohl in das für den Hochraten-Test des CMS-Pixeldetektors eingesetzte EUTelescope Framework und seine möglichen Konfigurationen als auch in die verwendeten Parametersätze für das *PSI46dig*-Strahlteleskop. Alle für eine Verarbeitung von nativen *PSI46*-Detektordaten nötigen Prozessoren wurden im Rahmen der Arbeit entwickelt, getestet und werden für weitere Anwendungen gepflegt. Dies ermöglicht den Einsatz der EUTelescope-Software mit allen Vorzügen für kommende Tests des CMS-Pixeldetektors.

Contents

List of Figures

Anything one man can imagine,
other men can make real.
Jules Verne

1. Introduction

In 1964 P. W. Higgs et al. proposed their theory for electroweak symmetry breaking in the Standard Model [Hig64a, Hig64b, EB64, GHK64]. Nowadays, almost 50 years later the first hints regarding the existence of the elusive *Higgs boson* have been found. On July 4, 2012 the CMS and ATLAS collaborations who run the two general purpose experiments at the Large Hadron Collider (LHC) [EB08], announced their discovery of a new boson with an energy of about 125 GeV [C+12, A+12]. This finding is the success of many years of planning and construction toward the LHC and its experiments - even though it is not decided yet whether the discovered boson is the Standard Model Higgs boson or another particle opening up a completely new range of possible explanations. The so-called *Standard Model* of particle physics tries to provide a consistent description of matter and interactions with a set of elementary particles and three fundamental forces mediated by gauge bosons [GGS99]. The compliance of the Standard Model with nature can be measured to an astonishing accuracy but it still leaves some questions unanswered. Several experiments have been conducted during the last decades which produced results that do not comply with the predictions made by the Standard Model and open new questions. The main goal of the LHC and its experiments is the quest for answers to these questions.

The LHC is the largest and most powerful particle accelerator currently in operation. A design collision energy of 14 TeV and instantaneous luminosity in the order of $\mathcal{L} = 1 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [B+04] provide access to new fields of physics and enable the investigation of questions which arose during the last decades of research in particle physics. The luminosity is the number of collisions per time interval and beam cross section area and the key parameter to measure the performance of colliders. The Compact Muon Solenoid (CMS) [C+08] is one of the large-scale experiments at LHC which have been designed and built for the requirements defined by the physics objectives. It is a complex detector consisting of several subsystems, each with a different key task in the overall detection of particles emerging from the collision point. Some subdetectors measure particle energy, others the path or momentum. One of these parts is the CMS silicon pixel detector, a hybrid silicon detector with 2D position resolution [CMS98]. It is the innermost part of CMS with a distance of only a few millimeters to the nominal interaction point. The pixel detector measures the positions of charged particles penetrating the detector planes and reconstructs their tracks and primary vertices. Due to the fine granularity a precise measurement of the spatial resolution of the impact parameter and secondary decay vertices can be performed. Furthermore the pixel detector provides crucial information for the track seeds used by the CMS tracking algorithm to reconstruct the particle tracks through the detector [K+10].

During the coming years of operation the collision energy and instantaneous luminosity of the LHC will gradually be increased to deliver more events to the experiments. With more data at higher collision energies and therefore different production cross sections for e.g. the Higgs boson, the experiments will be able to measure the properties of the newly found particle in more detail. After the consolidation of the accelerator in the Long Shutdown 1 (LS1) in 2013/2014 the LHC will deliver double the design luminosity of $\mathcal{L} = 2 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ at a center-of-mass energy of up to 13 TeV [Zim09] instead of the current operation energy of 8 TeV. However, a higher luminosity means also more particles and tracks within the same event (*pileup*). Currently CMS has to cope with about 35 pileups per bunch crossing but at double luminosity up to 100 pileups are expected [CMS12]. To keep pace with these constantly rising demands, parts of the CMS detector will be upgraded or exchanged. The pixel detector will be completely replaced in the so-called Phase I Upgrade in the end of 2016. Not only the silicon sensor material needs to be exchanged due to radiation damage but also the Read Out Chip (ROC) which processes the signals [ML09]. With the higher instantaneous luminosity provided by the LHC the number of particles emerging from the interaction region rises. The current ROC is not capable of handling this increased occupancy without considerable data loss and inefficiencies. The so-called *PSI46dig* ROC is the replacement chip which has been designed for higher occupancies and data rates.

The operational capabilities and efficiencies of this new *PSI46dig* ROC have to be tested thoroughly. A convenient way of analyzing new detector components is provided by *beam tests*. The detector under investigation, or Device Under Test (DUT), is placed in a particle beam with precisely defined properties. The beam test tries to mimic the real operation conditions and gives insights into the behavior and efficiency of the detector. Usually so-called *beam telescopes* are used to provide particle track measurements the DUT can be compared to. For the new CMS pixel ROC especially the behavior at high occupancies is of importance and a beam test providing a high particle rate has been set up at the *H4IRRAD* beam test area [BC11] at the Super Proton Synchrotron (SPS). The scope of this thesis is to give an overview of this high-rate beam test for the new CMS pixel ROC and describe the data analysis chain for the detector raw data and the underlying software framework in detail. The EUTelescope software framework [B$^+$07b] has originally been developed for the EUDET telescopes [Cor09] but provides a variety of ready-implemented algorithms for the analysis of beam telescope data. This framework has been used and extended in order to fit the needs of beam tests with a telescope exclusively equipped with CMS single ROC pixel detectors. The concepts and processing algorithms such as clustering or track fitting are discussed and compared, and aspects of different approaches are studied. The objective is to provide an insight into the current development status and configuration and to lay the foundations for future improvements and refinements.

Following this introduction, Chapter 2 describes the concept of particle detection and therefore their interaction with matter. This covers the different interaction effects as well as the basic principles of semiconductors, doping technologies, and *pn*-junctions. Finally the building blocks and working principles of 2D position sensitive silicon tracking detectors are described. The LHC and its accelerator complex is presented in Chapter 3. Furthermore a description of the experimental beam lines of the SPS accelerator and the *H4IRRAD* area for the high-rate beam test is given. The first part of Chapter 4 introduces the CMS experiment, its physics objectives, and the detector with its subdetector systems. The second part focuses on the components and geometry of the CMS pixel detector and the planned changes for the Phase I Upgrade. Chapter 5 provides an overview of pattern recognition approaches and algorithms for tracking detectors. Clustering algorithms for connecting pixel hits from the same particle passage are described as well as the concepts of energy weighting and detector alignment, both with an emphasis on tracking detectors with beam telescope geometry. The chapter closes with the description of different track fitting approaches and effects to be taken into account. The high-rate beam test performed for the Phase I Upgrade of the CMS pixel detector is introduced in Chapter 6 with a description of the beam telescope itself and its periphery. The proton beam is characterized and measurements of the particle flux and beam spot are presented. Chapter 7 focuses on the data analysis of the beam test, describing the software frameworks and the algorithms which have been prepared and used. An introduction to the EUTelescope framework, its data processors, and its analysis strategies is given. Finally, preliminary results from the first beam test in July 2012 are presented. The experiences and results of the July high-rate beam test are summarized in Chapter 8 together with an outlook on the forthcoming high-rate beam test in the end of October and further development of the analysis framework.

The story so far:
In the beginning the Universe was created. This has made a lot of people very angry and been widely regarded as a bad move.

Douglas Adams

2. Semiconductor Detectors

After the invention of the transistor by Shockley, Bardeen, and Brattain and their Nobel Prize in 1956, the interest in the new semiconductor circuit technology increased rapidly. Also in high energy physics with its permanent demand for higher granularity and more precise position and timing resolution, first attempts in particle detectors with the usage of silicon as detector material were made. Nowadays, silicon is a widely used material in tracking devices, allowing a lightweight, radiation hard design with modern lithography methods developed for the IT chip industry. Up to now the main concern beside the costs of a large area silicon device were technical limitations. This changed with the availability of affordable and well-known integrated circuit technology on large silicon wafers.

Semiconductors are not only used in tracking devices but serve also as excellent spectrometers due to their superb energy resolution as the small bandgap allows the production of a large number of primary charge carriers. On average a minimum of about $100\,\mathrm{e}^-/\mu\mathrm{m}$ are produced by the passage of a charged particle. This has implications for tracking devices: the large amount of charge carriers leads to thin active material layers (typically $300\,\mu\mathrm{m} - 500\,\mu\mathrm{m}$ are sufficient for a distinct signal) enabling fast signal propagation and processing. In addition to fast charge collection this means less material in the tracker detector.

In this chapter the basic principles of particle detection with semiconductor detectors are described. After a short summary of the physics of particle interaction with matter and the concept of semiconductors and *pn*-junctions, and the working principle of silicon pixel tracking devices are presented.

2.1 Interaction of Particles with Matter

In order to detect the passage of particles through the detector they need to interact with the detector material and deposit a traceable amount of energy. This is characterized either by the *stopping power* $\mathrm{d}E/\mathrm{d}x$ of a material for heavy charged particles or by the *radiation length* X_0 for high-energy electrons and photons. The stopping power is defined as the amount of energy $\mathrm{d}E$ a particle loses by traversing the distance $\mathrm{d}x$ of the absorber material and is usually given in $\mathrm{MeVcm}^2/\mathrm{g}$. The radiation length is defined as both the length over which an electron loses all but $1/e$ of its energy and $7/9$ of the mean free path of a high-energy photon undergoing pair production and is usually given in $\mathrm{g/cm}^2$. The material budget is defined as the number of radiation lengths x/X_0 a particle has to cross. For tracking devices the radiation length and thus the material budget is a crucial parameter. First, by limiting the material budget multiple scattering can be reduced. Second, the energy of the particle should not be derogated too much to enable a precise energy measurement in the following calorimeters (cf. Section 4.3.2). Tracking detectors themselves as well as their support structures and cooling infrastructure are therefore designed with a low material budget.

Silicon tracking devices are not able to detect neutral particles such as neutrons since they do not interact via electromagnetic processes and therefore do not create electron-hole pairs in the active sensor material.

2.1.1 The Bethe Equation for Heavy Charged Particles

The Bethe equation describes the energy loss of heavy charged particles with matter due to electromagnetic interaction. The energy loss depends on the characteristics of the absorber material as well as on the properties of the particle such as mass, momentum, and charge. A particle is classified as *heavy* if its mass is significantly higher than the electron rest mass m_e. The Bethe equation for heavy charged particles is given by [B$^+$12b]:

$$-\left\langle \frac{\mathrm{d}E}{\mathrm{d}x} \right\rangle = Kz^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 T_{\max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right], \tag{2.1}$$

where $K = 4\pi N_A r_e^2 m_e c^2$ is a constant combining the Avogadro constant N_A, the electron radius r_e, its rest mass m_e, and the speed of light c. The properties of the absorber material are described by its atomic mass A and atomic number Z, and the mean excitation energy I. The charge of the incident particle is denominated as ze, and $\beta\gamma = p/Mc$ is the relativistic factor for the momentum with

$$\beta = \frac{v}{c} \qquad \text{and} \qquad \gamma = \frac{1}{\sqrt{1-\beta^2}}. \tag{2.2}$$

The term $\delta\left(\beta\gamma\right)$ contains density effect corrections to the ionization energy loss. $T_{\max}$ is the maximum 4-momentum transfer to an electron. In a rough low-energy approximation $T_{\max}$ can be written as e.g. given in [B$^+$12b]:

$$T_{\max} = 2m_e c^2 \beta^2 \gamma^2 \qquad \text{for } 2\gamma m_e/M \ll 1, \tag{2.3}$$

with M being the mass of the incident particle. Figure 2.1 shows an exemplary plot of the stopping power for a positively charged muon in copper as a function of $\beta\gamma = p/Mc$. At lower energies around the energies of atomic electrons as well as at higher energies above $\beta\gamma \approx 500$ other effects dominate the energy loss and the Bethe equation cannot be applied. The radiative losses at very high energies are only relevant for muons and pions. Particles with $\beta\gamma \approx 2-3$ are denominated as *minimum ionizing particles* since their mean energy loss rate is always around the minimum in the Bethe region.

Usually the fluctuations in energy loss for detectors of moderate thickness are described by the *Landau-Vavilov theory* and the energy deposited by a particle crossing the detector material follows a Landau distribution. However, for thin silicon tracking devices this theory does not provide correct predictions. The measured distributions are significantly wider than the distributions expected from the Landau-Vavilov theory for the same parameters. An example for the straggling function of 500 MeV pions in silicon of different thickness is shown in Figure 2.2. Additions to the theory have been made to correctly describe the energy loss in thin silicon detectors [Bic88].

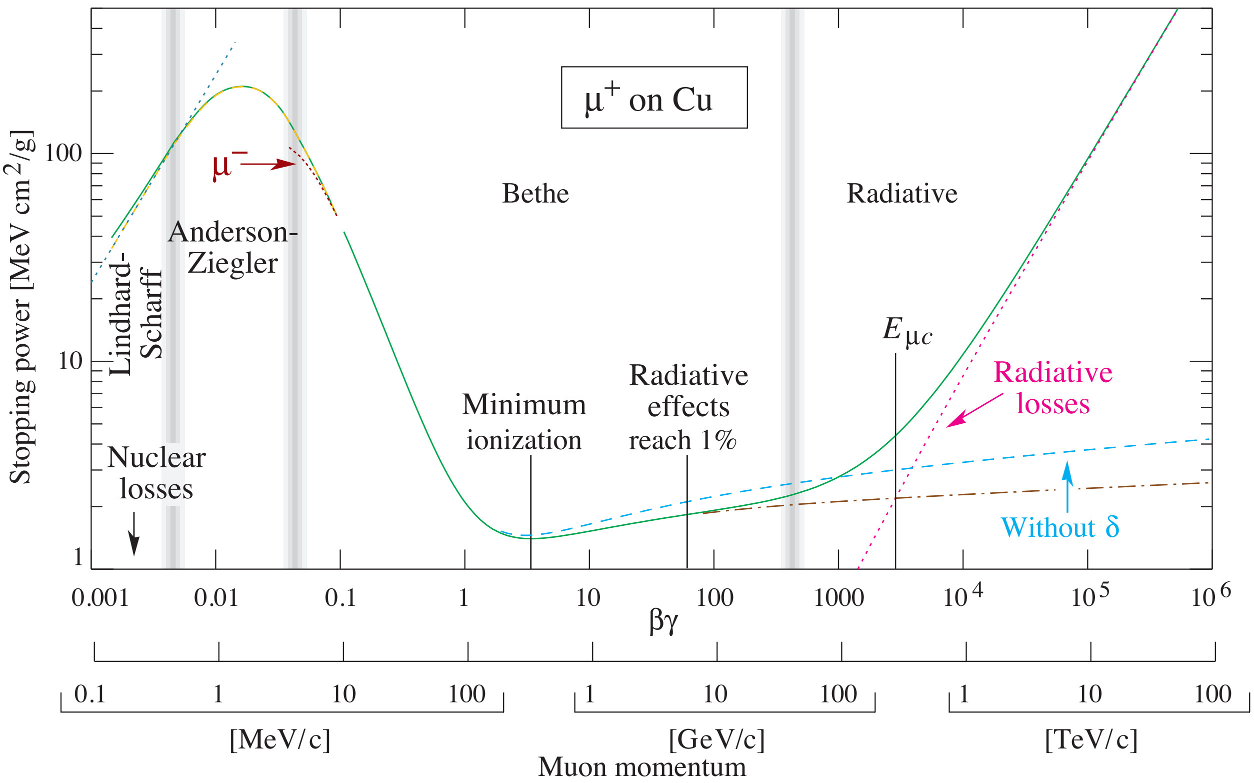


Figure 2.1: Mean stopping power $\langle \mathrm{d}E/\mathrm{d}x \rangle$ for a positively charged muon in copper according to the Bethe equation in units of $\beta\gamma$. Particles with $\beta\gamma \approx 2 - 3$ are denominated as *minimum ionizing particles*. They are characterized by experiencing the lowest stopping power within the Bethe region [B+12b].

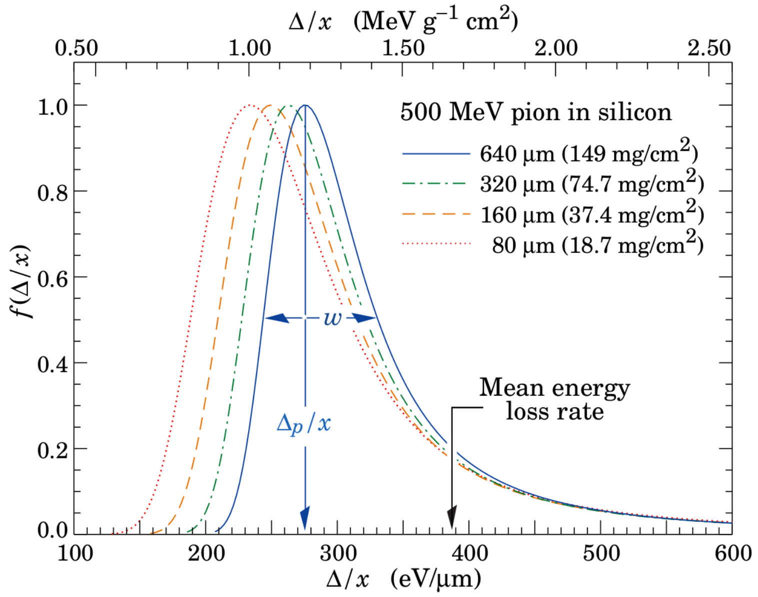


Figure 2.2: Straggling function of 500 MeV pions in silicon detectors with different thicknesses as a function of energy loss per distance Δ/x, normalized at the most probable value Δ_p/x. w is the full width half max value of the distribution [B+12b].

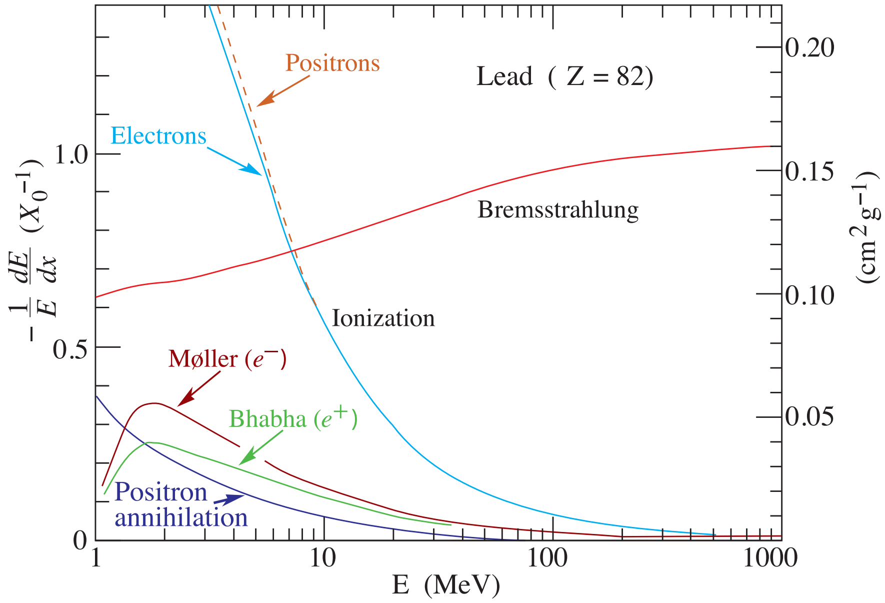


Figure 2.3: Energy loss per radiation length for light charged particles such as electrons and positrons in lead as a function of energy. The dominating process above ≈ 10 MeV is *bremsstrahlung*. At very low energies the contributions from scattering processes, ionization, and e^+ annihilations play a role [B+12b].

2.1.2 Light Charged Particles

The situation is somewhat different for lightweight charged particles such as electrons or positrons. At low energies the particles undergo scattering and ionization processes but at higher energies the major part of the energy loss is due to the *bremsstrahlung* process as indicated in Figure 2.3 for lead. Bremsstrahlung is the radiation emitted by a particle when deflected in the electric field of another particle, mostly atomic nuclei, in the absorber. It has a continuous spectrum with the intensity and mean frequency of the radiation depending on the energy loss of the particle.

Furthermore so-called *delta electrons* or *knock-on electrons* can be produced by other high-energy charged particles knocking bound electrons out of their shell. They have sufficient energy to ionize further atoms. Especially in tracking devices these delta electrons can lead to a larger volume over which the created charge carriers are spread (cf. Section 2.3.2).

2.1.3 Photon Interaction

In general, photons cannot be used for tracking purposes since they are either absorbed in the material at their interaction point, heavily scattered, or do not interact at all. However, other applications make it interesting to investigate the photon interaction with matter, for instance the calibration of the energy scale of silicon detectors in tracking devices using the distinct peaks of mono-energetic X-rays as described in [Hoß12] for the CMS pixel detector. Calibrating a position sensitive device is of importance if energy weighting is used for improving the position resolution (cf. Sections 2.3.3 and 5.1.2).

Photons interact with matter via three effects; the different interaction cross sections depend mainly on the energy of the incident photon and the mass number Z of the absorber. Figure 2.4 shows the cross sections for the participating interaction processes as a function of the energy of the incident photon for lead as absorber ($Z = 82$).

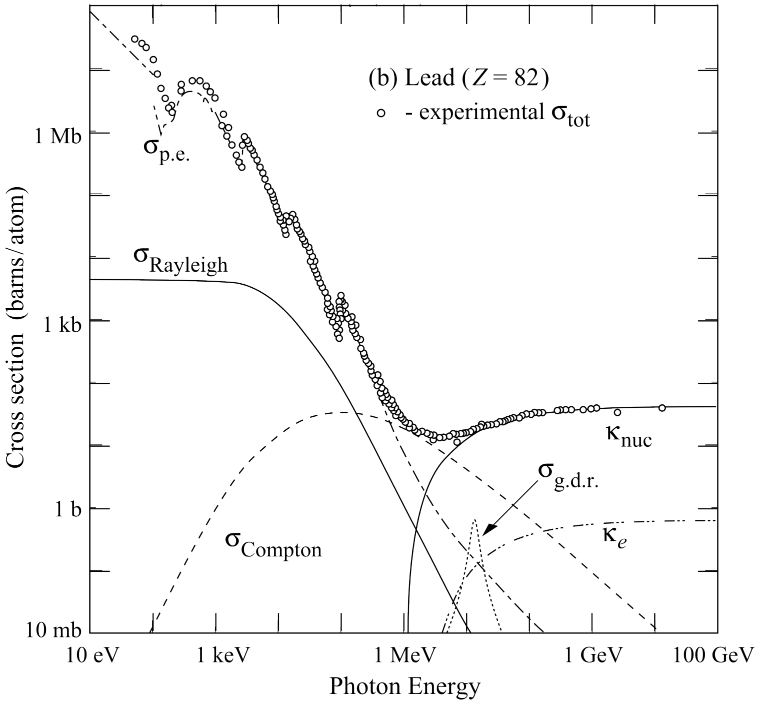


Figure 2.4: Total interaction cross section for photons in lead as a function of the photon energy. Also shown are the cross sections of the different participating processes: coherent (σ_{Rayleigh}) and incoherent (σ_{Compton}) scattering, photoeletric effect ($\sigma_{\text{p.e.}}$), and pair production (κ_{nuc}, κ_e) [B+12b].

Compton scattering

Compton scattering describes the inelastic collision of the incident photon with an electron in one of the outer shells. The photon is not absorbed but shifted in wavelength and the remaining energy is transferred to the electron which leaves its shell. The Compton effect is the dominant effect at energies around $E_\gamma \approx 1\,\text{MeV}$. The photon's wavelength shift is given by:

$$\Delta\lambda = \lambda' - \lambda = \frac{h}{m_e c}\left(1 - \cos\theta\right), \tag{2.4}$$

where λ is the initial and λ' the shifted wavelength of the photon. The electron rest mass is denominated as m_e, while h is the Planck constant, and c the speed of light. The energy transferred during the interaction depends on the impact parameter of the collision and results in different scattering angles of the reflected photon. The maximum energy transfer takes place at an angle of 180° leading to the Compton edge, a drop in the spectrum at the highest transferred energy possible for this process. Between 0° and 180° all reflection angles are possible and form the Compton continuum.

Photo ionization

In the photo ionization process the photon is completely absorbed by an inner shell electron which is removed from its shell. All energy is transferred to this electron, partly as ionization, partly as kinetic energy. The vacant position in the inner shell is filled by a cascade of outer electrons resulting in the emission of low energy photons and Auger electrons.

Photo ionization deposits all energy of the incident photon in the absorber material and therefore produces a distinct photo peak in the spectrum.

Pair production

Pair production describes the production of an e^+e^- pair in the presence of a nucleus. This is only possible if the energy of the incident photon is higher than two times the rest mass of the electron:

$$E = 2 \cdot m_e = 1022\,\text{keV}.$$

Pair production becomes dominant at higher energies, depending on the atomic number Z of the absorber. Above $\approx 10\,\text{MeV}$ almost all other effects can be neglected (cf. Figure 2.4).

The incident photon creates an e^+e^- pair and the electron is absorbed in the material while the positron annihilates with another electron emitting two photons with an energy of 511 keV each. Just as in the photo ionization case all energy is deposited in the material and should lead to a distinct peak. However, in pair production also the so-called *escape peaks* can be observed in the spectrum. They originate from the annihilation photons leaving the material without further interaction. One can distinguish the *single escape peak* with only one photon leaving, and the *double escape peak* where both photons escape the (detector-) material. These peaks can be found at discrete energies corresponding to the rest mass of the annihilated electron-positron pair:

$$E_{\text{s.esc}} = E_\gamma - 511\,\text{keV} \qquad \text{and} \qquad E_{\text{d.esc}} = E_\gamma - 2 \cdot 511\,\text{keV}.$$

2.2 Semiconductors

Solid matter can be divided into three groups according to their electrical properties: Insulators, conductors and semiconductors. Usually the *energy band model* is used to describe their different behavior. In systems with single atoms the discrete energy levels of the shell electrons can be described by solving the *Schrödinger equation* for the given potential. However, when dealing with solid matter and crystal lattices many atoms are involved, with interfering electron shells forming a common potential. The discrete energy levels split up into various sublevels which cannot be assigned to a single atom anymore and form the *energy bands*. A detailed derivation of the model can be found e.g. in [Kit05].

For the electrical properties of a material only two of these bands are relevant. The *valence band* represents the highest energy band completely filled with bound electrons at $T = 0\,\text{K}$. The next higher energy band is called *conduction band*. Not only electrons but also *holes* contribute to the conductivity of a material. Holes are positively charged vacancies in the valence band. Electrons in the band are moving in an electric field by filling the vacancy and creating a new one. This process is usually treated as movement of the holes themselves. The mobility of holes is slightly smaller than the one of electrons.

Since only electrons and holes in partly filled bands with available energy levels or states can move freely within the crystal and therefore contribute to the conductivity of the material, one has to distinguish between the different situations shown in Figure 2.5. At 0 K all electrons are in their ground states determined by the *Pauli exclusion principle* and the energy bands are filled up to a certain level. If the valence and conduction bands overlap or the conduction band is partly filled, the material is a conductor. With rising temperature the motion of the atoms in the crystal lattice increases and the conduction band electrons are slowed down resulting in a decreasing conductivity.

If the valence band is completely filled while the conduction band is empty the material is an insulator since no free charge carriers are available for conduction. A semiconductor at $T = 0\,\text{K}$ behaves just like an insulator, the difference is the size of the separating gap between the valence band and the conduction band. The *energy gap* E_G is the defining property concerning the electrical

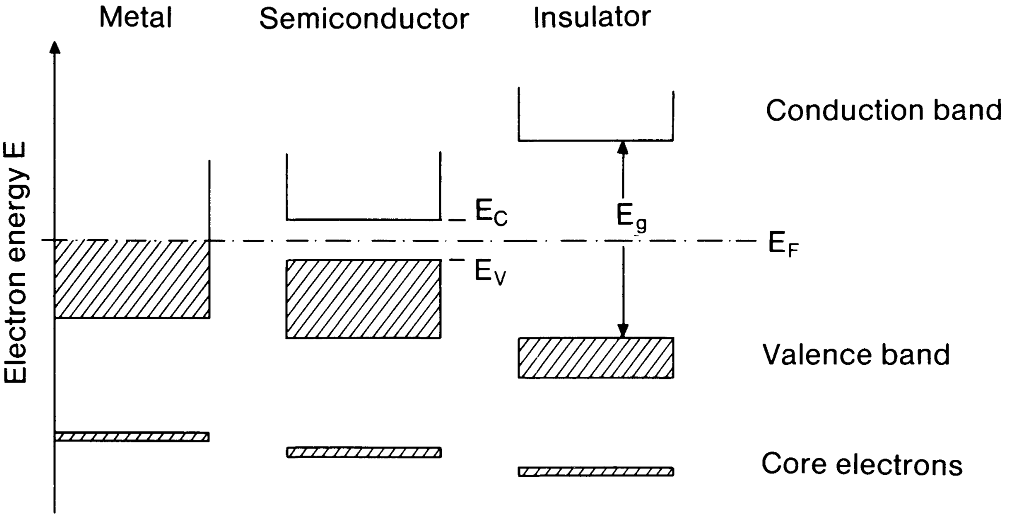


Figure 2.5: Schematic band structures for different materials. For conductors the valence band and conduction band either overlap or are only partly filled. For insulators the two bands are separated by a large energy gap $E_G = E_C - E_V$, while the gap for semiconductors is small [IL10].

behavior. For insulators the gap is rather large ($E_G > 3\,\mathrm{eV}$) and even with increasing temperature no electrons can be lifted into the conduction band. In semiconducting materials or compounds the energy gap is significantly smaller ($E_G \approx 0.20\,\mathrm{eV} - 2.00\,\mathrm{eV}$ [Kit05]) and with rising temperature more and more electrons can be excited into a state in the conduction band. Therefore, in contrast to conductors, the conductivity increases with rising temperature.

In most cases either silicon (Si) or germanium (Ge) are used due to their availability and properties. High-purity Ge is mostly used in spectrometers due to its small energy gap of $E_G = 0.74\,\mathrm{eV}$ at $T = 0\,\mathrm{K}$ which results in a high energy resolution [Kit05]. However, Ge is not particularly suitable for tracking devices since its conductivity at room temperature ($T = 300\,\mathrm{K}$) is already quite high and for proper operation it has to be cooled with liquid nitrogen (LN_2). Si can be obtained at a lower price compared to Ge and operated at room temperature due to its larger energy gap of $E_G = 1.17\,\mathrm{eV}$ at $T = 0\,\mathrm{K}$ and $E_G = 1.124\,\mathrm{eV}$ at $T = 300\,\mathrm{K}$ [Har08].

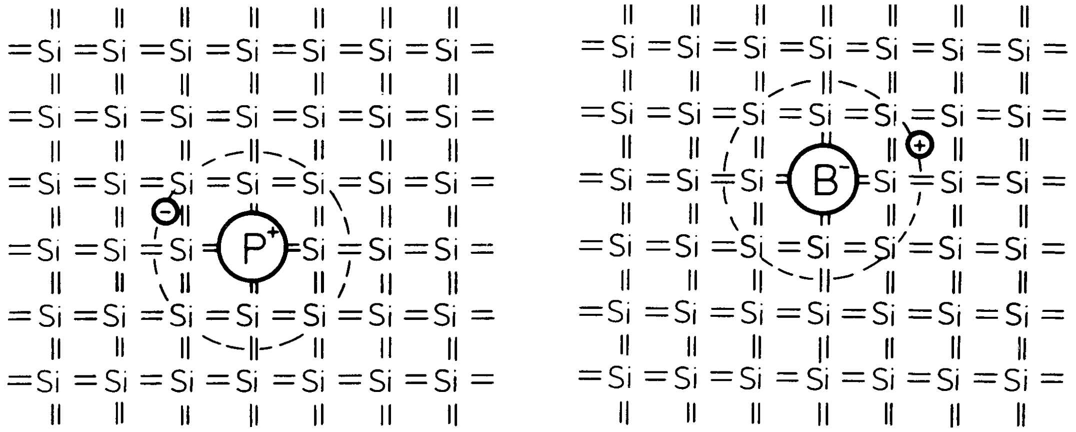
The occupation of energy states within the regarded material for a given temperature can be determined using the *Fermi-Dirac distribution*. It gives the probability of finding an electron with the given energy E:

$$f(E) = \frac{1}{1 + e^{(E-E_F)/k_B T}}, \tag{2.5}$$

where E_F is the Fermi energy or *Fermi level* and $k_B T$ is the Boltzmann constant and the temperature. At $T = 0\,\mathrm{K}$ all energy levels up to E_F are filled while at higher temperatures some electrons can be found above the Fermi level. If two regions with different Fermi levels are connected, charge carrier will drift between them to form a constant Fermi level across the interface (see Section 2.2.2).

For *intrinsic* (pure) semiconductors without impurities the charge carrier concentrations in the valence band p (holes) and the conduction band n (electrons) are equal to the total intrinsic charge carrier concentration n_i:

$$n_i = n = p.$$



(a) n-doped silicon

(b) p-doped silicon

Figure 2.6: Schematics of a doped silicon crystal. (a) If doped with a donor such as phosphorus, an additional quasi-free electron is introduced as negative charge carrier (n-type). (b) Boron as acceptor only provides three valence electrons and therefore introduces a vacancy (hole) into the lattice (p-type) [IL10].

2.2.1 Semiconductor Doping

The properties of an intrinsic semiconductor can be altered by introducing impurities into its crystal lattice. This procedure is referred to as semiconductor *doping*. Doped semiconductors are usually called *extrinsic semiconductors* since the electrical properties and the conductivity are not governed by the intrinsic charge carrier concentration but the concentration of the impurities introduced (the so-called *dopants*). According to the dominant charge carrier type the semiconductor is either called *p-type* semiconductor for holes or *n-type* semiconductor for electrons.

Silicon is a group IV element and therefore has four valence electrons. Introducing atoms with more or less valence electrons into the lattice changes the conductivity by adding additional charge carriers as shown in Figure 2.6. Adding a group III atom such as boron (B) or aluminum (Al) creates an electron vacancy (hole) at that position due to the missing fourth valence electron. Therefore group III dopants are referred to as *acceptors* (*p-type*). Group V atoms such as phosphorus (P) or arsenic (As) with five valence electrons provide an additional weakly bound electron to the intrinsic semiconductor and are thus called *donors* (*n-type*).

This process can again be described using the energy band model. The dopants create additional energy states within the energy gap of the semiconductor as shown in Figure 2.7 for the silicon energy gap. Donors introduce additional energy states slightly below the conduction band. The additional (quasi-free) electron can easily be elevated into the conduction band, contributing to the overall conduction. Acceptors, in turn, add energy states close to the valence band so electrons can be lifted into the acceptor state with minimal energy, creating a free hole in the band. The Fermi level of the semiconductor material is shifted accordingly either towards the valence or the conduction band. The positions of the introduced energy states depend on the properties of the dopant atoms.

The choice of the dopant depends mostly on the layout of the final device and the doping process itself since the different elements differ in properties such as their diffusion depth into Si. Furthermore, various techniques for the doping process itself exist, e.g. diffusion or ion implantation which are descibed in e.g. [Sch05].

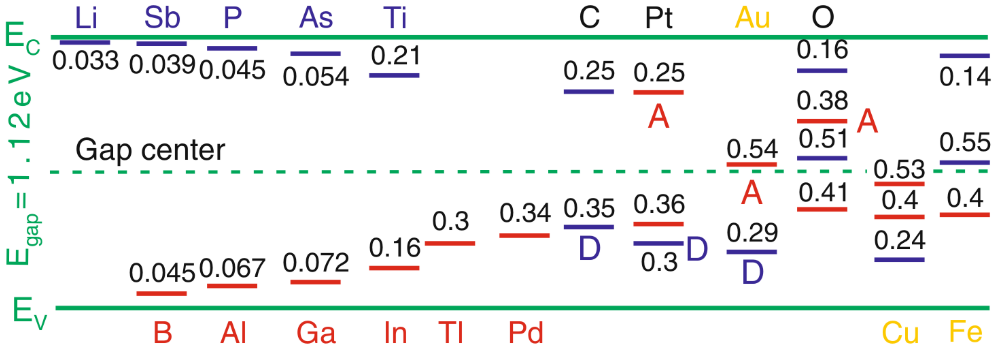


Figure 2.7: Dopant energy levels in the silicon energy gap. Usually dopants with levels close to the energy of the valence band (E_V) or conduction band (E_C) are chosen such as B or P. The impurities in the upper part are donors while those in the lower part are acceptors; otherwise marked with D (donors) or A (acceptors) [Har08].

2.2.2 The *pn*-Junction

The *pn*-junction is the key building block of almost every semiconductor device. It enables both the design of complex electronic circuits and the improvement of the electrical behavior for charge collection which is essential for semiconductor tracking devices. At room temperature a 300 µm thick intrinsic silicon substrate as used in tracking detectors contains about 10^9 free charge carriers, but since a charged particle introduces only about 2×10^4 charge carriers in the same material the signal would be lost in this thermal noise [Har08]. In order to reduce the number of free charge carriers the sensors are designed as so-called *pn*-junctions in reverse bias mode.

A *pn*-junction is a combination of two semiconductors with opposite doping. As stated above the separate *p*- and *n*-type silicon have different Fermi levels E_F due to the different acceptor and donor levels as shown in Figure 2.8 (a). As soon as they are joined the free electrons from the *n*-side of the junction will diffuse towards the lower Fermi level at the *p*-side. The electron current stops when a common Fermi level across the *pn*-junction has been established. This corresponds to the state outlined in Figure 2.8 (b). Since the free electrons and holes recombine at the interface, a depleted region (or *space charge region*) without free charge carriers is formed around the contact. The lateral dimension of the depleted region depends on the Fermi levels of the two separate substrates as well as on applied bias voltages. Two bias configurations are possible:

Forward bias connects the *p*-side with the anode and the *n*-side with the cathode. With a voltage applied the holes in the *p*- and the electrons in the *n*-side are pushed towards the junction narrowing the depleted region. With rising voltage the *pn*-junction breaks down as soon as the external voltage exceeds the internal electric field and the depleted region dissolves.

Reverse bias is the configuration used in semiconductor detectors. With the anode at the *n*-side and the cathode at the *p*-side the width of the depleted region grows with the applied bias voltage. The voltage at which the full bulk material is depleted is called the *depletion voltage*. Since the depleted region is the sensitive sensor volume, semiconductor sensors are always operated at their depletion voltage or slightly above. Charged particles passing through the depleted region create charge carriers (electron-hole pairs). Due to the applied voltage the two charge carriers are drawn towards the electrodes without recombining.

With increasing width of the depleted region the capacitance of the device grows until it is fully depleted. This allows the precise measurement of the depletion voltage by recording *C*-*V* curves which saturate when reaching the desired operation voltage with fully depleted sensor material.

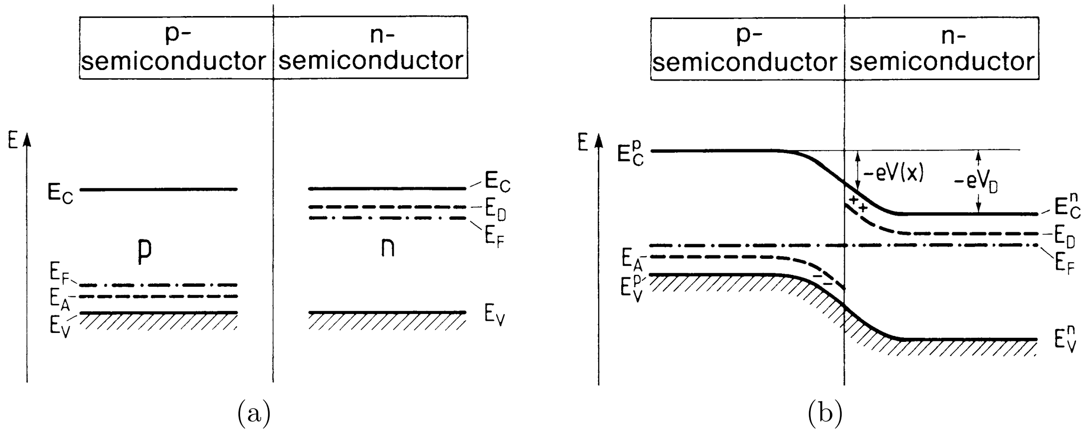


Figure 2.8: Band energies (E_V, E_C), donor and acceptor levels (E_D, E_A), and Fermi level E_F in p- and n-type semiconductors before and after joining. (a) Situation before joining the two substrates with different Fermi and band energy levels. (b) Joined junction in equilibrium with a common Fermi level and the altered conduction and valence band energies [IL10].

Even if a reverse bias voltage is applied to the pn-junction, a small current through the device known as *leakage current* can be measured. It originates mostly from remaining intrinsic charge carriers and thus increases with temperature, applied bias voltage, and radiation damage in the bulk material of the sensor. Monitoring this increase is of particular importance in semiconductor detectors since the leakage current has considerable impact on the power consumption of the device.

2.2.3 Fluence and Radiation Damage

In particle detectors, silicon is constantly exposed to a particle flux that alters the structure of the material. The total radiation a detector was exposed to is called *fluence*, the particle flux through an area integrated over time. To be able to compare the fluence of different particle types, the numbers are usually given according to the Non-Ionizing Energy Loss (NIEL) scaling hypothesis. This states that any particle flux can be reduced to an equivalent of 1 MeV neutron fluence that produces the same damage in the sensor material and is usually given in neutron equivalent per area $\mathrm{n_{eq}/cm^2}$. The NIEL fluence can then be calculated by

$$\Phi_{n_{\mathrm{eq}}} = \kappa \cdot \Phi, \tag{2.6}$$

where Φ is the original particle fluence and κ the *hardness factor* containing all particle and energy specific information. For example the hardness factor for the 23 GeV protons provided by the CERN Proton Synchrotron (PS) accelerator is $\kappa = 0.62$ [M$^+$02]. One distinguishes mainly between two types of radiation damage:

- Surface damage origins mostly from holes (positive charge carriers) that are trapped in the surface layers. Mainly the electronics suffers from this in the irradiation environment of the Large Hadron Collider (LHC) while the effects on the sensor are limited.
- The bulk damage that affects the material volume is more important for silicon sensors. The crystal lattice is modified by NIEL processes such as displacements or additional atoms forming additional energy levels in the energy gap of the semiconductor. This directly affects both the leakage current through the device and the trapping of produced charge carriers resulting in a degradation of the signal.

A detailed study of different radiation damages, silicon defects, and impacts on operation of silicon detectors can be found in [Mol99].

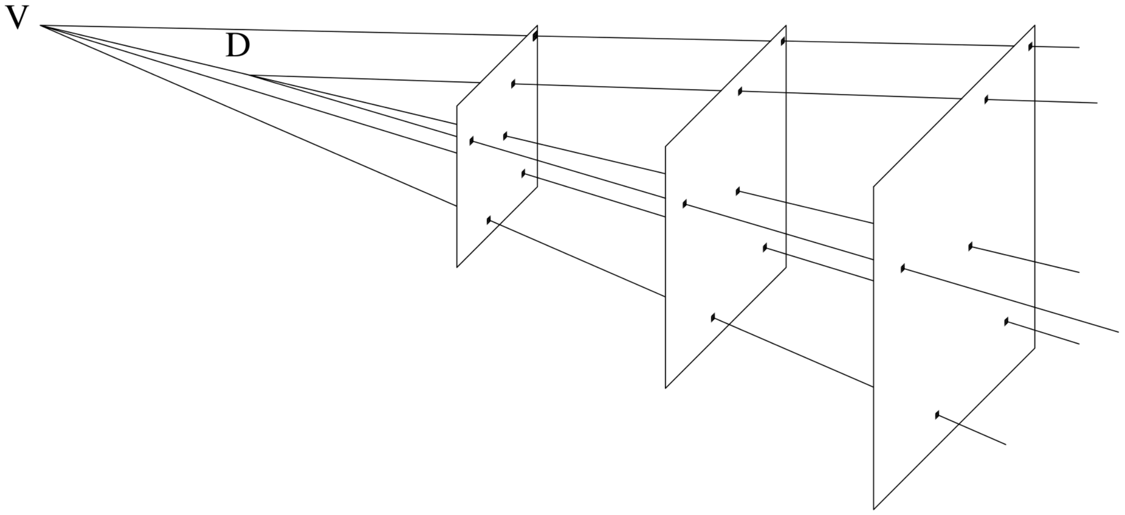


Figure 2.9: Sketch of several particle tracks emerging from a primary vertex (V) and a secondary decay vertex (D) from a short-lived particle, measured by three consecutive pixel detector planes. Pixel detectors are especially useful to isolate (D) and (V) vertices and to record many tracks without ghost hits [R+06].

2.3 2D Position Sensitive Silicon Detectors

A 2D position sensitive silicon detector or *pixel detector* is a detector type used in particle physics to measure the impact point of a particle in the sensor plane in two dimensions. With more than one sensor plane the track of a particle traversing the detector can be measured by fitting the single hits. Different algorithms for this purpose are presented in Chapter 5. The main duty of pixel detectors in collider physics is to provide a precise spatial resolution near the collision point to discriminate single particle vertices. This is especially important in cases where either a high track multiplicity is given (e.g. in heavy ion collisions or at high pileup) or studies of very short-lived particles are performed, such as *b-tagging* where the decay vertex of a hadron with b-quark has to be identified. Secondary decay vertices with only a few millimeters distance from the interaction point have to be resolved as sketched in Figure 2.9.

Pixel detectors usually consist of two main parts, the active sensor and the front-end chip for readout and data processing. The 2D resolution is achieved by segmented electrode implants in the active silicon sensor. The segmented electrode forms one part of the *pn*-junction while the sensor bulk material provides the oppositely doped material. The readout chip is usually able to buffer the recorded information for a certain time until a trigger decision has been made. With an additional memory for timestamps this allows a nanosecond timing resolution for bunch crossing selection as well as the possibility of selecting interesting events to be stored on hard-disk. Both aspects are crucial for large-scale experiments with event rates of up to 40 MHz.

The advantage of pixel detectors over silicon strip detectors is the track measurement without ambiguous detector readout that leads to ghost hits, since a true 2-dimensional spatial resolution is provided. The same spatial resolution could also be achieved with two micro strip detectors rotated by a stereo angle, but in this case for N particle hits additionally $N^2 - N$ ghost strip coincidences would be produced [R+06]. For the track density in the inner region of today's high energy particle physics experiments this would lead to a complicated or even impossible track finding.

However, the development of front-end electronics for pixel detectors with their small feature size is complex since each pixel has to be connected individually to its readout channel. The design

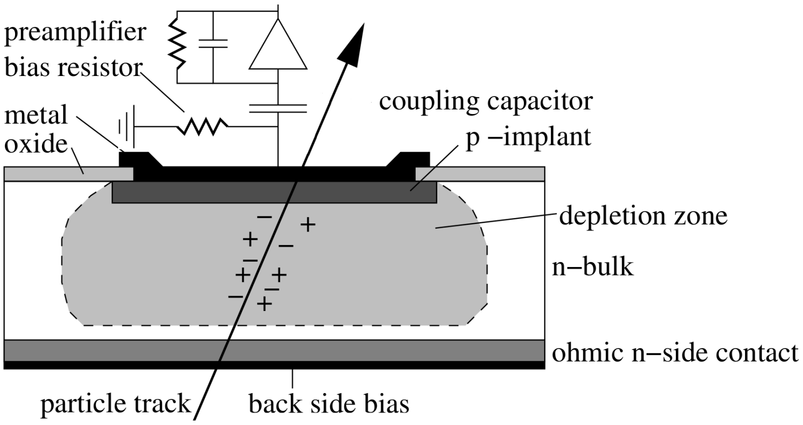


Figure 2.10: Working principle of a pixel of a semiconductor pixel detector. The traversing particle creates primary charge carriers in the depleted sensor material. They are collected by means of the bias voltage applied over the junction. The measured signal is then processed by the front-end chip electronics such as the preamplifier [R$^+$06].

of 1D resolution silicon strip detectors is easier since the electronics can be placed at each side of the sensor with larger feature sizes. Therefore in the outer tracking region preference is given to strip detectors in order to reduce the costs, the total number of readout channels, and thus the complexity and amount of data produced for every event.

2.3.1 Working Principle

The building block of a silicon pixel detector is a *pn*-junction as described in Section 2.2.2, separately for each pixel. Today mostly *planar sensors* are used such as the example sketched in Figure 2.10. The segmented implant electrodes reside at the surface of the sensor while the bulk material forms the rest of the sensor material. Usually pixel detectors are build as so-called *hybrid pixel detectors* where the front-end chip fits exactly the sensor area and is directly connected to it. The two electrodes of the *pn*-junction are connected to the readout electronics as demonstrated in Figure 2.11 where each pixel implant is connected to the corresponding readout channel via bump bonds.

The energy deposited in the sensor material by a traversing charged particle is used to create electron-hole pairs by exciting electrons from the valence into the conduction band, thus leaving a hole in the valence band. A reverse bias voltage is applied to the sensor in order to expand the depleted region over the full bulk material and suppress noise from thermal excitation. Figure 2.10 shows a simplified schematic of a single pixel cell with *n*-doped bulk material where the electrons are collected towards the *p* implant. However, other designs have been developed that have advantages over this rather simple approach, e.g. the Compact Muon Solenoid (CMS) pixel sensor *n*-in-*n* concept presented in Section 4.4.2. In planar pixel sensors the induced charge carriers have to cross the whole thickness of the sensor bulk in order to reach the implant to be read out. To improve this, other geometries are under investigation, such as *3D sensors* where the pixel implants reach through the whole bulk material and promise faster charge collection. First results of 3D pixel sensor measurements can be found e.g. in [G$^+$11].

The basic functionality provided by the front-end chip is shown in Figure 2.12. Every single pixel cell has its own processing electronics channel and usually some registers to tune the behavior of the different parts. After the charge collection in the sensor the signal is passed to the preamplifier and the pulse shaper. If the detector is equipped with a zero-suppression mechanism, the signal is then compared against an adjustable threshold. Only if the signal in the pixel cell exceeds the threshold, the pulse is counted as valid hit and fed to the Analog-Digital Converter (ADC) and the data bus.

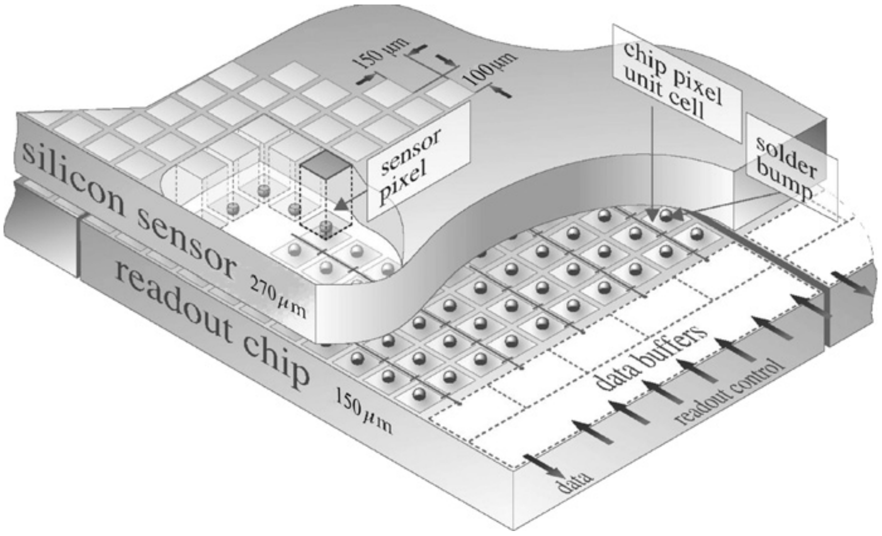


Figure 2.11: Structure of a hybrid pixel detector. The sensor is congruent to the front-end chips and each pixel implant is connected to the corresponding readout channel of the chip via bump bonds. The bias voltage is applied between the backside metallization and the readout channel on the chip to collect the charge carriers [C$^+$09].

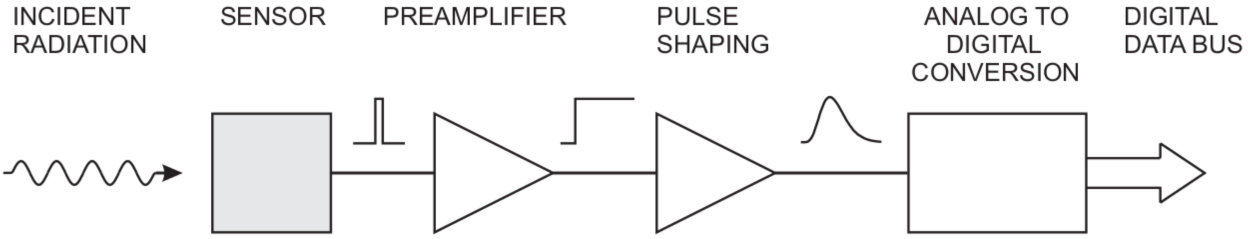
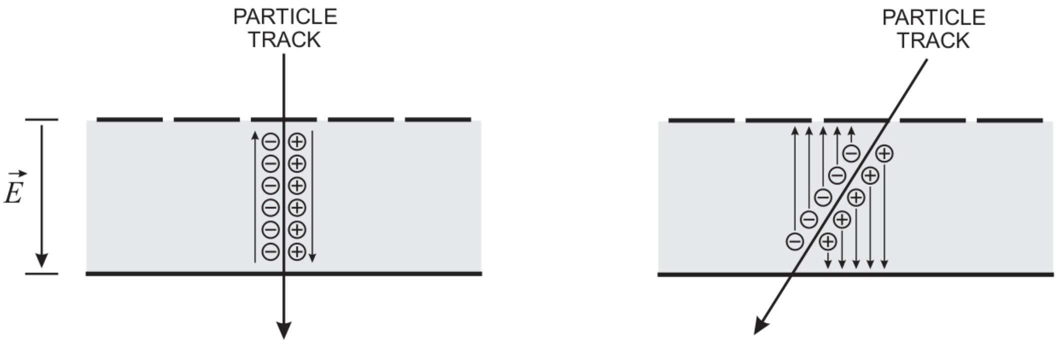


Figure 2.12: Basic working principle of a semiconductor detector pixel cell. The energy deposited in the sensor by the incident particle is amplified before feeding it into the pulse shaper. If the detector has zero-suppression capabilities the signal is then compared against the threshold and finally converted for the readout [Spi05].



(a) no charge sharing (b) charge sharing among two cells

Figure 2.13: Charge sharing between pixel cells indicated by the segmented electrode. (a) If the particle traverses the detector perpendicular to the surface all charge is deposited in one electrode segment. (b) If the particle incidents at an angle the charge is shared among all involved electrode segments [Spi05].

2.3.2 Charge Sharing

Charge sharing is a process that has to be taken into account when designing a tracking detector. If the incident particle hits the edge between two pixel cells or incidents not perpendicular to the surface and crosses several pixels, the produced charge carriers are shared among these pixels (see Figure 2.13). Since magnetic fields and the curvature of particle tracks are used to determine the particle's momentum, the particles are deflected and incident in an angle between the track and the surface normal of the sensor which originates from the *Lorentz Force*:

$$\boldsymbol{F}_L = q\left(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}\right), \tag{2.7}$$

where $\boldsymbol{v}$ is the particle's velocity, q its charge, $\boldsymbol{E}$ the electric and $\boldsymbol{B}$ the magnetic field.

Not only the particle itself but also the charge carriers produced inside the silicon bulk material are deflected by the Lorentz force resulting in an additional drift. This charge carrier drift is described by the *Lorentz angle*. In the CMS tracking detector the Lorentz angle is calculated and measured to be about 25° [CMS10]. This has been taken into account when designing the shape of the pixel detector cells (cf. Section 4.4.2). Furthermore delta electrons can lead to a wider spreading of the charge carriers.

However, charge sharing is not necessarily bad but can be used to improve the hit resolution of the tracking device. The real hit position of the incident particle can be interpolated e.g. by using the center-of-gravity method (see Section 5.1.2). Often sensor geometries are designed in such a way that the charge is shared among two pixels enabling interpolation while charge sharing over more than two cells is suppressed. The signal and Signal-to-Noise Ratio (SNR) decreases if the limited amount of charge carriers is shared among too many readout entities. In this case it gets more and more difficult to distinguish a real hit from sensor noise while the spatial resolution cannot be improved further.

Depending on the SNR it might be necessary to concentrate all charge on one pixel since the charge collected by one pixel in case of charge sharing might be too low to get a proper hit signal. Avoiding charge sharing with an appropriate sensor geometry in turn limits the spatial resolution as shown in the following section.

2.3.3 Spatial Resolution

The spatial resolution of a pixel detector is mostly determined by the pixel pitch, which means the length and width of a single pixel cell. Furthermore the selected readout mode of the detector as well as the reconstruction method and the amount of charge sharing have an influence on the resolution. One can distinguish two types of detector readout modes:

- The *binary readout* only has a single threshold to decide between noise and signal. The only information stored is whether the pixel has detected a signal or not, thus leading to a binary map of the detector showing pixel hits.
- In the *charge sensitive readout* mode a binned or analog value of the recorded pulse height is stored. This allows an interpolation between several pixels with different pulse height levels to determine a more precise position of the particle's penetration point.

In general, the spatial resolution is defined as the average difference between the true particle impact position x_t and the measured or reconstructed impact position x. Assuming a uniform density of particles, the resolution in binary readout mode with only one single pixel hit can be calculated for one dimension as (see e.g. [R$^+$06]):

$$\begin{aligned}\sigma_{\mathrm{pos}}^2 &= \frac{\int_{-p/2}^{p/2} (x_t - x)^2 \, D(x_t) \, \mathrm{d}x}{\int_{-p/2}^{p/2} D(x_t) \, \mathrm{d}x} = \frac{p^2}{12} \\ \sigma_{\mathrm{pos}} &= \frac{p}{\sqrt{12}},\end{aligned} \tag{2.8}$$

where p is the pixel pitch and $D(x) = 1$ the particle density.

If charge sharing among adjacent pixels is taken into account the spatial resolution is improved. Here one has to distinguish between two cases. If still only one pixel fires when the particle passes through (a so-called one-pixel cluster), the resolution is given by:

$$\sigma_{\mathrm{pos}} = \frac{(p - s)}{\sqrt{12}}, \tag{2.9}$$

where s is the distance over which the charge carriers are spread. For a two-pixel cluster the spatial resolution is further improved to:

$$\sigma_{\mathrm{pos}} = \frac{s}{\sqrt{12}}. \tag{2.10}$$

Hence the optimal resolution for a binary readout pixel detector is achieved if $s = p/2$.

The *charge sensitive readout* mode allows a further reduction of the spatial resolution by using the additional pulse height information. Different methods for the interpolation between the pixels of a cluster have been developed. An overview on different energy weighting algorithms is given in Section 5.1.2.

One Ring to find them,
One Ring to bring them all...
J. R. R. Tolkien

3. The Large Hadron Collider

The Large Hadron Collider (LHC) is today's largest and most powerful particle accelerator. The 27 km long ring is located at the European Organization for Nuclear Research (CERN) in Geneva, Switzerland, and resides 100 m below ground in the tunnel which has been used for the Large Electron-Positron Collider (LEP) before. The LHC has been designed to provide proton-proton collisions (and ion-ion collisions with lead nuclei) at a center-of-mass energy of 14 TeV and instantaneous luminosity of $\mathcal{L} = 1 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ in order to allow the study of new physics phenomena, the search for explanations of long-standing problems such as the mechanism of electroweak symmetry breaking, and the confirmation and refinement of the Standard Model of particle physics [GGS99] at yet unreached energy and intensity [B$^+$04].

The performance of particle accelerators is usually measured in units of instantaneous and integrated luminosity. The *instantaneous luminosity* $\mathcal{L}$ is the proportionality between the expected event rate $\dot{N}$ and the cross section σ, and is given in $\mathrm{cm}^{-2}\mathrm{s}^{-1}$. For a collider ring with two colliding beams and a bunch structure such as the LHC the luminosity is given by:

$$\mathcal{L} = \frac{n \cdot N_1 \cdot N_2 \cdot f}{A}, \tag{3.1}$$

where n is the number of bunches in the ring, f the collision rate, and $N_{1,2}$ the numbers of particles in the colliding bunches. The cross-sectional area of the bunches at the nominal interaction point is denominated as A. The current maximum instantaneous luminosity of the LHC is about $\mathcal{L} = 7.5 \times 10^{33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [CER12a]. The *integrated luminosity* is the instantaneous luminosity integrated over time:

$$\int \mathcal{L}\,\mathrm{d}t. \tag{3.2}$$

It can be used as measure for the delivered number of collisions and therefore for the amount of data recorded. The integrated luminosity is usually given in fb^{-1}, where $1\,\mathrm{b} = 10^{-24}\,\mathrm{cm}^2$.

The protons for the LHC are provided by the CERN accelerator complex which is briefly described in the next section. Once fed into the LHC the proton beams are kept on the desired orbit by 1232 dipole bending magnets and focused by 858 quadrupole magnets. The geometry of the magnets is rather complex since they have to provide very strong magnetic fields in opposite directions for each of the two proton beams. The particles are accelerated with Radio Frequency (RF) cavities from the injection energy of about 450 GeV up to the desired collision energy. After reaching this state the two beams are brought to collision by the final focusing magnets at the four Interaction Points (IPs) where the large experiments are installed. Due to initial problems with the dipole magnets the LHC is currently running at 8 TeV center-of-mass energy [CER08]. After the Long Shutdown I scheduled for the beginning of 2013 the feasible operation energy will be around 6.5 TeV per beam, i.e. 13 TeV center-of-mass energy. Due to limitations of the dipole magnet fields the originally planned energy of 14 TeV will not be reached in the near future.

The beams have a bunch structure to be able to accelerate the particles with the RF cavities. The LHC is designed to hold a maximum of 2808 bunches per beam with a distance of about 7 m at a bunch crossing time of 25 ns and 10^{11} protons per bunch [B+04]. Currently the bunch crossing time is 50 ns instead of 25 ns as planned since problems with beam-beam interaction and the injector chain have been encountered. This in turn leads to fewer bunches in the beams (currently 1374 bunches [CER12a]) with increased proton intensities to reach a comparable luminosity. The drawback of this approach for the experiments is the higher *pileup* they have to withstand. Pileups are simultaneous particle collisions within the same bunch crossing which have to be separated by the particle detectors. At the current peak luminosity the detectors experience up to 35 pileups and after the LHC luminosity upgrade simulations predict up to 100 pileups (for double the design luminosity of $\mathcal{L} = 2 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ with 50 ns bunch crossing time) depending on the bunch crossing time chosen [CMS12].

The latest LHC performance and operation statistics such as peak and integrated luminosity, the beam intensities, or the center-of-mass energy can be obtained from the LHC Performance and Statistics website [CER12a]. A summary of the operational parameters and functions of the LHC can be found in [EB08] while an extensive technical description is provided by the technical design report [B+04].

3.1 The Accelerator Complex

The particles to be collided have to be pre-accelerated by several smaller accelerators in order to be able to inject them into the LHC since the RF cavities are limited to a specific working range. For the LHC cavities this working range lies between 450 GeV and 8 TeV. Figure 3.1 gives an overview of the different accelerators and some of the associated experiments.

At the beginning hydrogen atoms are extracted from high-purity hydrogen gas and their shell electron is stripped off. The LINAC2, a linear RF-pulsed accelerator with an output energy of 50 MeV and a maximal current of 150 mA injects them into the next stage in the accelerator complex, the PSB or BOOSTER. The LINAC3 is used to accelerate the Pb ions for the lead-lead physics runs of the LHC.

The PS receives the protons from the PSB and accelerates them up to an energy of 25 GeV. It supplies particles to both the subsequent SPS accelerator towards the LHC and the various beam lines and smaller experiments.

The Super Proton Synchrotron (SPS) is the last pre-accelerator before the protons are fed into the LHC. It serves several users with only one of them being the LHC. For example the CERN Neutrinos to Gran Sasso (CNGS) beam facility [Gsc06] receives a secondary particle beam from the SPS as well as the beam lines and experiments located in the Experimental Hall North 1 (EHN1). The primary proton beam from the SPS accelerator has been used for the high-rate beam test described in Chapter 6. The SPS extraction point *TT10* and the *North Area* beam lines are shown in Figure 3.1. A detailed description of the beam line configuration and the SPS beam structure can be found in Section 3.3. With an energy of about 450 GeV the protons are finally injected into the LHC, where the final acceleration to the desired collision energy takes place.

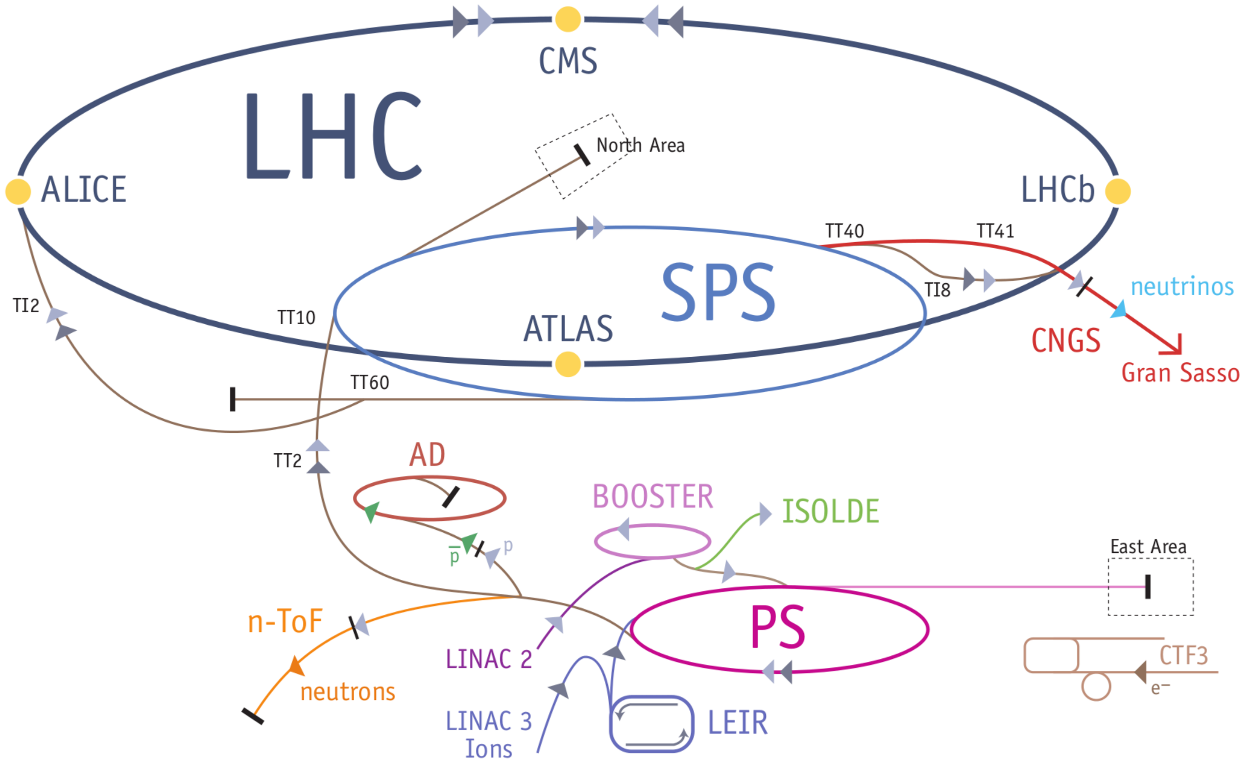


Figure 3.1: Sketch of the LHC and the CERN accelerator complex. The acceleration process starts with the extraction of hydrogen atoms. The shell electrons are stripped off and the leftover protons are accelerated in the LINAC2. The BOOSTER (or *PSB*) passes the pre-accelerated protons to the PS where they are accelerated to an energy of 25 GeV and then passed to the SPS. The SPS accelerates the protons up to 450 GeV and injects them into the LHC. In the LHC the acceleration to the final proton energy of currently 3.5 TeV takes place [Lef08].

3.2 Experiments at the LHC

Four main experiments operate at the different interaction points of the LHC. Each experiment resides in a cavern along the LHC tunnel where the two beams are brought to collision.

The Compact Muon Solenoid (CMS) is one of the two general-purpose experiments and is described in detail in Chapter 4. The second general-purpose detector is A Toroidal LHC ApparatuS (ATLAS), an experiment with the same objectives but different approaches. Although CMS and ATLAS are build for the same purpose and try to measure the same physics, the design and construction of the two detectors has been done independently and different techniques have been developed for the various subdetector systems. Furthermore different methods and algorithms are used for analysis of the data. This gives the possibility to double check results and possible discoveries with an independent experiment. A detailed description of the ATLAS experiment can be found in [A$^+$08a].

The Large Hadron Collider beauty Experiment (LHCb) is, as the name suggests, an experiment dedicated to b-meson physics. Its geometry is very different from the two barrel-shaped 4π detectors ATLAS and CMS and concentrates all subdetector systems around the beam pipe in the very forward region of the interaction point (a so-called *single-arm spectrometer*) [A$^+$08c].

A Large Ion Collider Experiment (ALICE) is a barrel detector with a high solid angle coverage, just like CMS and ATLAS, but concentrates on the measurement of lead-lead ion collisions. The ALICE collaboration tries to get new insights into the structure of the early universe by studying matter at very high densities, the so-called *quark-gluon plasma.* Further information about the physics objectives and the detector can be found in [A$^+$08b].

Several small detectors for specific tasks have been placed inside the experimental caverns and the LHC tunnel beside the four main experiments, such as the *Total Elastic and Diffractive Cross Section Measurement (TOTEM)* experiment [TOT08]. Furthermore various detectors for LHC operation and machine protection are installed for e.g. beam loss monitoring or radiation protection.

3.3 EHN1 and the *H4IRRAD* Beam Line

Several particle extraction points have been set up in the pre-accelerator chain of the LHC and both the PS and the SPS provide proton beams for additional users. This enables the conduction of other experiments during operation of the LHC.

The SPS Experimental Hall North 1 (EHN1) houses four of the beam lines fed by the SPS and provides space for various experiments such as NA61/SHINE [Abg08] and several test beam areas. Parts from CMS and ATLAS as well as electronics for the LHC have been tested in these beams before commissioning in order to qualify the radiation hardness or high-rate capabilities of the electronics. The EHN1 beams are extracted from the SPS accelerator and split to supply two primary targets (T2 for the experimental beam lines H2 and H4, T4 for the beam lines H6 and H8). An overview of the various North Area beam lines is given in Figure 3.2.

Each beam line can be operated in different modes, according to the user's request. Bypassing the target allows the primary proton beam from SPS to be delivered to the beam lines. Secondary beams of various particles can be produced using the targets, or a tertiary beam of converted or decayed secondary particle can be provided. The various configuration possibilities are described in [Eft03]. For the high-rate beam test described in Chapter 6 the high-intensity primary proton beam for the *H4* beam line has been used. This is only possible in the so-called *H4IRRAD* configuration where the beam is absorbed by large lead blocks directly after the experiment setup (beam dump). Usually this configuration is used together with a copper target to generate a high irradiation environment to simulate the conditions in the LHC tunnel [BC11].

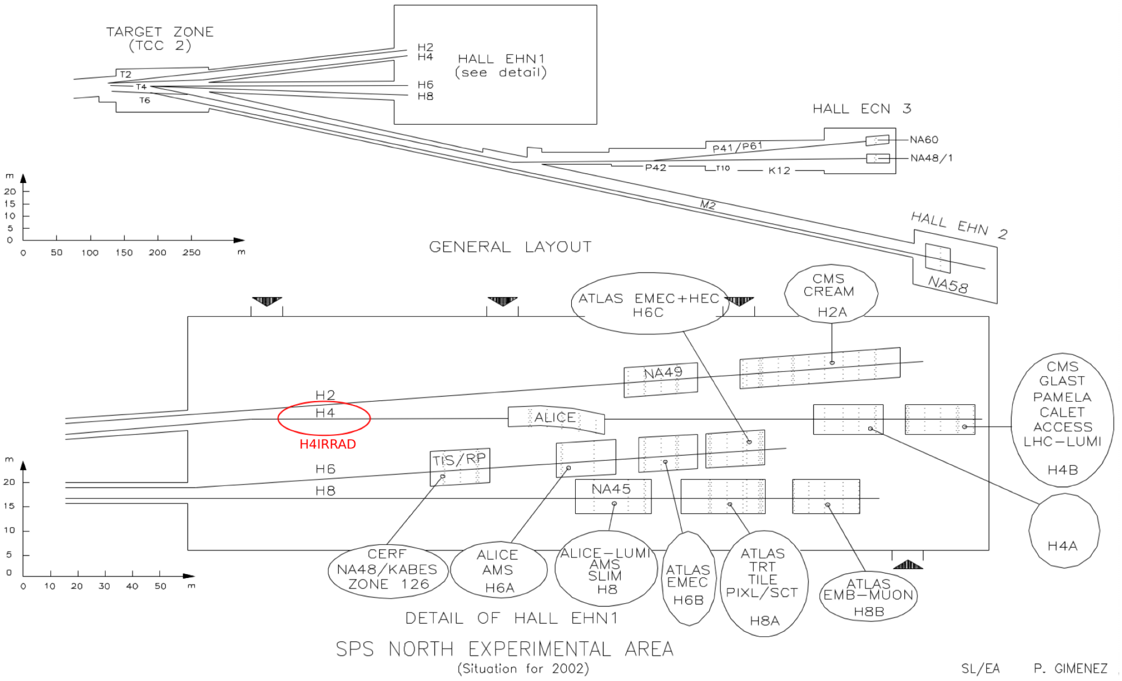
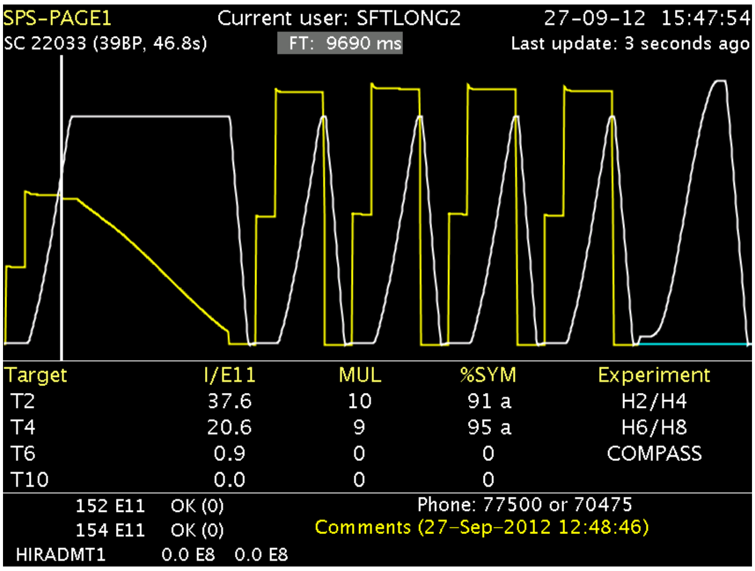


Figure 3.2: Overview of the T2 target zone and the EHN1 beam lines. The SPS proton beam is split at the T2 target into the H2 and H4 beam line. The *H4IRRAD* experimental area is located at the highlighted position near the beam entrance into the hall [Eft03].


```
SPS-PAGE1            Current user: SFTLONG2          27-09-12  15:47:54
SC 22033 (39BP, 46.8s)       FT:  9690 ms       Last update: 3 seconds ago

Target            I/E11        MUL        %SYM        Experiment
 T2                37.6         10        91 a          H2/H4
 T4                20.6          9        95 a          H6/H8
 T6                 0.9          0         0            COMPASS
 T10                0.0          0         0
        152 E11   OK (0)                   Phone: 77500 or 70475
        154 E11   OK (0)          Comments (27-Sep-2012 12:48:46)
 HIRADMT1        0.0 E8    0.0 E8
```

Figure 3.3: Example of the SPS spill structure as a function of time. The yellow curve represents the current particle intensity in the SPS machine, the white curve is the magnetic pulse with the four stages flat bottom (injection), ramp up (acceleration), flat top (particle extraction), and ramp down (beam dump). The beam intensity I to the targets is given in 10^{11} protons, the total length of the SPS Super Cycle is about 47 s [CER12c].

The SPS operates in so-called Super Cycles (SCs). Usually one SC has a length of about 47 s during which the provided beam is delivered to different users such as the targets for the experimental beam lines as well as more complex beam facilities such as CNGS. The extraction period for a single user is the so-called *spill*. For the T2 target the spill duration is about 10 s and up to 10^{13} protons per spill are delivered. The current proton beam intensity as well as the approximate number of delivered protons to a target can be obtained from the SPS Page-1 (cf. Figure 3.3). Detailed instructions on how to read the SPS Page-1 can be found in [CER12b].

After extraction the beam is transported to the test beam areas by bending magnets and its intensity and position are measured by various beam monitoring detectors. All information can be obtained by the beam line users via the TIMBER database or CESAR control systems. TIMBER is a front-end for the global *CERN Accelerator Logging Service (CALS)* and allows the logging and plotting of various beam line parameters such as the proton intensity measured at different locations by scintillators or ionization chambers. The information is stored in the database and can be accessed at any time from within the CERN network from TIMBER [CER], either by direct access for a limited time after recording or by issuing a request. CESAR stands for *CERN Experimental areas SoftwAre Renovation* and is a control system for the SPS experimental areas. With CESAR it is possible for users to directly control the beam e.g. by moving collimators in and out, or to monitor the system magnets in-time. A brief description of the aims and architecture of CESAR is given in [B+03].

A detailed drawing of the *H4IRRAD* can be found in Appendix C. A characterization of the SPS spill structure, intensity, and beam spot of the test beam has been performed as part of the high-rate beam test and is presented in Sections 6.3 and 6.4. The commissioning and operation of *H4IRRAD* is described in [B+12c], and an up-to-date operation manual for the H4 beam line is provided online at [CER10].

All the forces in the world are not so
powerful as an idea whose time has come.
Victor Hugo

4. The Compact Muon Solenoid

The Compact Muon Solenoid (CMS) detector is one of the two general purpose experiments at the Large Hadron Collider (LHC) and targeted on the discovery of new physics at the TeV energy scale. Its name origins from the muon systems (cf. Section 4.3.4) and the unique 3.8 T superconducting solenoid magnet (cf. Section 4.3.3) enclosing the central tracking region and the calorimeters. Figure 4.1 shows an overview of the whole detector and its components. CMS is a barrel-shaped detector covering almost the full 4π solid angle. This is especially important for measurements relying on the precise determination of missing transverse energy E_T^{miss} which is needed for the discrimination of certain event types.

With its total weight of about 12 500 tons and a diameter of more than 14 m CMS is one of the largest and most complex particle physics detectors ever built. The construction and planning of the CMS detector followed special demands on its performance, which can be derived from the physics objectives described in Section 4.1. [C$^+$08] summarizes them as follows:

- A good charge-sensitive muon identification with precise momentum resolution. As its name implies, the Compact Muon Solenoid is designed with special care concerning muon measurements.

- Efficient triggering for event selection and a good τ and b-jet tagging. This implies the construction of a high-resolution pixel detector for a precise vertex reconstruction near the interaction region. The CMS pixel detector is described in detail in Section 4.4.

- Good resolution for missing transverse energy E_T^{miss} and dijet-masses. For this, hermetic geometric coverage for jets and fine granularity is needed, which is both provided by the hadron calorimeter.

- Good electromagnetic energy resolution, diphoton and dielectron mass resolution and efficient photon and lepton isolation, especially at the high luminosities provided by the LHC.

The first part of this chapter summarizes the physics objectives of the CMS collaboration before describing briefly the subsystems of the detector. More detailed descriptions of all parts of the detector can be found in [CMS06], while [CMS07] summarizes the physics goals and performance studies. In the second part the CMS pixel detector with its geometry and components is described. Section 4.5 gives an overview of the pixel detector Phase I Upgrade and the scheduled changes to the detector components.

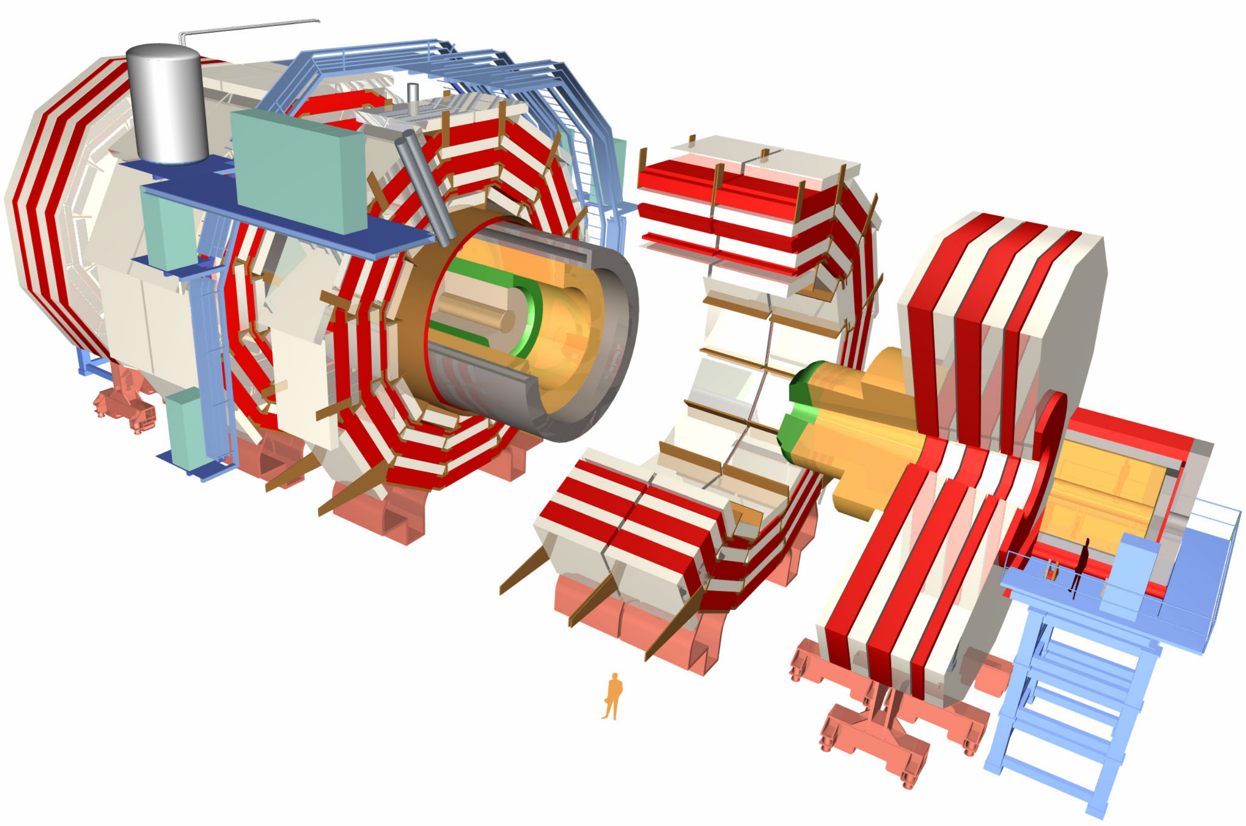


Figure 4.1: Sketch of the CMS experiment showing the main parts of the detector: The iron return yoke (red) of the solenoid magnet (grey) with the muon stations (white), the electromagnetic and hadronic calorimeters (orange and green), and the tracker detector in the center (light grey). CMS can be separated into the wheels and endcap parts as shown in the sketch. This structure allows an easy access during maintenance. The whole detector is resting on air pads which ease the displacement of the large and heavy parts [Hoo11].

4.1 Physics Objectives

Several main objectives have been identified before starting the design and construction of the CMS detector. The dimensioning and demands on the detector parts have been lead by the physics requirements defined by these goals. Beside the search for a Higgs boson, precision measurements of Standard Model parameters and the search for physics beyond the Standard Model such as Supersymmetry (SUSY) have been planned as stated in [CMS07]. In the following, all particle masses are given in the natural units eV instead of the formal notation of eV/c^2 for the sake of simplicity.

The primary objective of the Compact Muon Solenoid detector is to elucidate the nature of the electroweak symmetry breaking mechanism. Since the publication of a possible solution in 1964 by Higgs [Hig64a, Hig64b], Englert and Brout [EB64], and Guralnik, Hagen and Kibble [GHK64] several attempts have been made to shed light on the existence of the *Higgs boson*.

The Standard Model of particle physics describes the fundamental particles and their interactions via gauge bosons [GGS99]. Its consistency with nature has been measured with astonishing accuracy, but it still leaves some questions unanswered. One of them is related to the nature of electroweak symmetry breaking. The electroweak theory describes four gauge bosons which mediate the interaction processes, the charged $W^{\pm}$ bosons, the Z^0 boson, and the photon γ. However, according to the theory all of them should be massless. This is correct for the photon, but the other bosons have a non-negligible mass of 80 GeV ($W^{\pm}$) and 91 GeV (Z^0). The theory formulated by P. W. Higgs et al. tries to solve this by introducing four Higgs fields with a specifically shaped potential (the so-called *Mexican Hat* potential) leading to spontaneous symmetry breaking. With three degrees of freedom being absorbed by the massive bosons there is one degree of freedom left resulting in the massive Higgs boson.

The Higgs boson can be produced via several processes such as gluon-gluon fusion, vector boson fusion, or Higgsstrahlung. It is a short-lived particle with various decay channels with cross sections depending on its expected mass m_H. The dominant decay channels for a Standard Model Higgs boson are [C$^+$12]:

$$\begin{array}{rll} & 110\,\mathrm{GeV} < m_H < 160\,\mathrm{GeV} & H^0 \to WW \\ & 110\,\mathrm{GeV} < m_H < 135\,\mathrm{GeV} & H^0 \to b\bar{b} \to \text{jets} \\ \text{diphoton channel} & 110\,\mathrm{GeV} < m_H < 150\,\mathrm{GeV} & H^0 \to \gamma\gamma \\ \text{golden channel} & 110\,\mathrm{GeV} < m_H < 160\,\mathrm{GeV} & H^0 \to ZZ\,(ZZ^*) \to 4\ell^{\pm} \end{array} \tag{4.1}$$

Especially the *golden channel* is very promising since the four leptons (in particular four muons) produce a very distinct signal with almost no background. Beside a single Standard Model Higgs boson, other theories such as the Minimal Supersymmetric Standard Model (MSSM) [WZ74] predict up to five Higgs bosons with different masses.

Several experiments have been conducted to measure the existence of this elusive particle. The results of these analyses set the mass bounds for today's Higgs searches. The Large Electron-Positron Collider (LEP) has excluded a Higgs boson up to a mass of 114.4 GeV [ALE03], and the Tevatron experiments exclude the two regions $100\,\mathrm{GeV} < m_H < 103\,\mathrm{GeV}$ and $147\,\mathrm{GeV} < m_H < 180\,\mathrm{GeV}$ after having analyzed data from an integrated luminosity of about $10.0\,\mathrm{fb}^{-1}$ [TEV12]. Other constraints, especially for the upper mass bound of about 1 TeV, come from theory [Esp97].

The latest results on the Higgs search of the CMS and ATLAS collaborations have been presented at the *ICHEP2012* conference in Melbourne, Australia and Geneva, Switzerland and have now been published in [C$^+$12, A$^+$12]. Both experiments have independently found an excess of events around an energy of 125 GeV with a confidence level of 5σ indicating the existence of a new neutral boson. Whether these results are compatible with a Standard Model Higgs boson, a supersymmetric Higgs boson, or something else still needs to be investigated by recording and analyzing more data. The analysis of the Higgs decay channels that have either not been analyzed yet or that do not show any excess around the measured boson mass so far is of particular interest. New results can be expected for the *HCP2012*, November 2012 in Kyoto, or the spring conferences 2013.

With the excellent detector precision and the seven-fold increase of the LHC luminosity over its predecessors the parameters of the Standard Model can be measured with unreached accuracy. Among others, cross section measurements of the $W^{\pm}$ and Z boson production, the *Drell-Yan* process [DY70] and top-quark studies are scheduled [CMS07]. Since the top-quark production rate at the LHC is expected to be in the Hz range, precise studies of this quark with its couplings can be performed. This requires very efficient b-jet tagging to identify top-quark events, one of the requirements of the CMS detector.

Even though the Standard Model serves as a very precise model, several observations have been made during the last decades that show the limitations of the model, e.g. neutrino oscillations and the resulting neutrino mass [CGL09], or the existence of dark matter in the universe [Moo94]. Some of these mysteries could be solved by introducing another symmetry, called Supersymmetry (SUSY), and assigning a new partner to each of the Standard Model particles. These *squarks*, *sleptons* and *gluinos* are expected to decay in cascades that always end in the Lightest Supersymmetric Particle (LSP), the lightest and stable SUSY particle. Since it is anticipated that the LSP interacts very weakly with Standard Model particles it would very likely escape the detector and therefore introduce a significant amount of missing transverse energy E_T^{miss} into the measurement. The LSP would be a possible candidate for the dark matter in the universe.

Beside other possible discoveries of physics beyond the Standard Model such as flavor changing neutral currents or lepton flavor violation, there are hopes for first hints of a Grand Unified Theory (GUT) which combines all known forces and simplifies the description of interactions at very high energies.

4.2 The CMS Coordinate System

The collider coordinate system adapted by CMS and the other experiments at LHC is used for the description of events and positions inside the CMS detector and has its origin at the nominal collision point (Interaction Point).

The y-axis points vertically upwards while x points radially inwards toward the center of the LHC ring. Hence, the z-axis is orientated along the beam towards the Jura mountains in the west. Often spherical coordinates are better suited to describe the kinematics in particle collisions, so the polar angle ϕ is defined from the x-axis in the x-y-plane and the radial component is called r. The polar angle θ is measured from the z-axis. The pseudorapidity, which is used to quantify the angle of a particle relative to the beam axis, is defined by

$$\eta = -\ln \tan \theta/2. \tag{4.2}$$

The momentum or energy transverse to the beam direction is an important parameter for the characterization of events. These quantities are denominated as p_T and E_T and calculated from the corresponding x and y values. Missing energy in this transverse x-y-plane is denoted as E_T^{miss} [CMS06].

4.3 CMS Subdetector Systems

The CMS detector can be divided into several sub-detector systems, each with a distinct function providing crucial information for the overall measurement of particle origin, path, energy, momentum, and type.

Figure 4.1 shows an exploded view of the entire CMS detector. Shown are the different wheels and endcaps the detector system consists of. This structure ensures that CMS can be decomposed into several parts for maintenance during the shutdown periods. The individual parts can be moved using air pads and the floor of the CMS cavern is slightly tilted to ease the displacement. The inner part of CMS is enclosed by the solenoid magnet and contains the inner and outer tracker detectors and the calorimeters. Outside the magnet the muon systems are installed into the magnet return yoke.

4.3.1 Tracking Detectors

The demands on the tracking detectors of CMS are very high due to the high luminosity and short bunch crossing time provided by the LHC. Several new technologies have been developed in order to meet the requirements and provide reliable tracking of particles.

At current instantaneous luminosity provided by the LHC more than 35 *pp*-collisions occur each bunch crossing (pileups) and around 1000 particles emerging from the interaction region [CMS12]. This illustrates the need for tracking systems featuring high granularity as well as fast response. Fast signal processing requires on-detector electronics including their cooling infrastructure, which in turn leads to a higher material budget. On the other hand the material budget needs to be kept low to limit multiple scattering which is crucial for high precision track measurements. Beside these instantaneous demands the strip tracker detector (in particular the inner parts) faces another challenge. It has to survive several years of operation in a harsh radiation environment without any maintenance or replacement. The pixel detector was designed to be replaceable during operation. This option will be used for the Phase I Upgrade.

To meet these requirements the CMS collaboration has built a radiation hard silicon detector with an active sensor area of about $200\,\mathrm{m}^2$. With this detector CMS is the first experiment using an all-silicon tracker. The tracker detector comprises three layers of pixel detectors and another ten layers of micro strip detectors. The CMS pixel detector features a high granularity for precise primary vertex resolution and is described in detail in Section 4.4. The micro strip detector aims to provide a precise tracking with many track seeds while keeping the cost and the total number of readout channels low.

The strip tracker can be divided into several parts: The *inner tracker* and *outer tracker* systems and the endcap detectors. As shown in Figure 4.2 these are again subdivided into groups of strip modules that differ in strip length and pitch. For the Tracker Inner Barrel (TIB) and Tracker Inner Disk (TID) parts ($20\,\mathrm{cm} < r < 55\,\mathrm{cm}$), strips with a length of 10 cm and 80 µm pitch are used. In the outer region ($55\,\mathrm{cm} < r < 110\,\mathrm{cm}$) with the Tracker Outer Barrel (TOB) and Tracker End Cap (TEC) detectors the strips are longer (20 cm) to limit the number of readout channels. These sensors come in two parts from different wafers and the strips are connected via wire bonds. The modules indicated with double lines are back-to-back modules with two strip sensors rotated by a stereo angle of 100 mrad. This allows the measurement of 2D coordinates with the 1D strip modules even though still some ambiguities are left, which have to be excluded by other constraints. A more detailed description of the tracker geometry can be obtained from [C$^+$08].

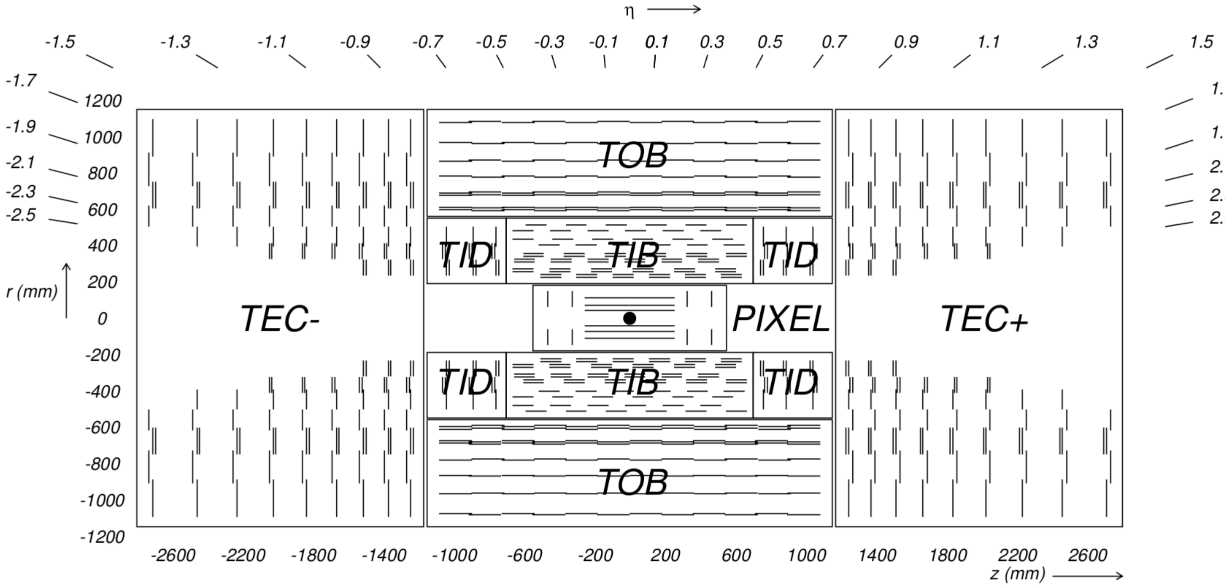


Figure 4.2: Schematic cross section of the CMS tracker with simple detector modules (single lines) and back-to-back modules (double lines) for 2D position resolution [C+08].

4.3.2 Calorimeters

Calorimeters are used to measure the energy of particles by providing enough material to stop the particles and absorb them completely. This assures both a precise energy measurement and a subsequent muon tracking with muons being the only charged particles crossing the calorimeters without high energy loss. The energy resolution of calorimeters is given by:

$$\left(\frac{\sigma}{E}\right)^2 = \left(\frac{S}{\sqrt{E}}\right)^2 + \left(\frac{N}{E}\right)^2 + C^2, \tag{4.3}$$

where S is the stochastic, N the noise, and C the constant contribution to the signal. This implies that the energy resolution depends on the incident particle and improves with higher energies. The CMS detector includes two calorimeters, the Electromagnetic Calorimeter (ECAL) for particles with electromagnetic interactions and the Hadron Calorimeter (HCAL) for particles interacting mainly hadronically. Again, the design of the detectors was driven by the demands from the LHC operation: radiation hardness, fast readout, and fine granularity.

The ECAL consists of several thousands of high-density lead tungstate ($PbWO_4$) crystals with a radiation length of only $X_0 = 0.89\,\text{cm}$. These anorganic scintillators both provide a good energy and timing resolution and ensure a thickness of $d > 25X_0$. The incident particles start an electromagnetic shower that excites the crystal lattice and produces visible light. The number of photons created is proportional to the energy deposited in the absorber. This signal is read out with photo diodes. The fine granularity with the large number of separated crystal blocks assures a spatial resolution enabling the unambiguous allocation of deposited energies to the corresponding particle tracks from the tracking detector. The ECAL covers particles with a pseudorapidity up to $|\eta| < 3.0$.

The HCAL is a so-called sampling calorimeter. It encloses the ECAL and also covers the solid angle up to $|\eta| < 3.0$. A sampling calorimeter consists of alternating layers of absorber (brass for the CMS HCAL) to create showers and sensitive scintillating material for the actual signal production. Again, photo diodes are used for the detector readout. The thickness of the HCAL varies with $|\eta|$ between $7 - 11$ interaction lengths λ_I, and is completed by a tail-catcher (HO) and forward calorimeter for a coverage up to $|\eta| < 5.0$ with Cherenkov detectors.

4.3.3 The Superconducting Solenoid

The superconducting solenoid magnet is a central part of the detector concept realized in CMS. With its inner diameter of 6 m it comprises the central tracking detectors as well as the two calorimeters. With a homogeneous 3.8 T magnetic field the particle tracks are bent inside the tracking detectors allowing the precise measurement of the particle momentum by determining the curvature of the particle track.

Due to the relatively low cold mass of about 220 t but a total stored energy of 2.6 GJ the magnet is mechanically deformed during the energizing procedure. Therefore the magnet is made of reinforced NbTi alloy, a metallic low-temperature superconductor with a transition temperature of 9.2 K [C+08]. The 10 000 t yoke of the magnet returns the magnetic field and houses the four muon stations described in the next section.

4.3.4 Muon Systems

The muon systems are designed with special care concerning costs, robustness, and efficiency. Having a good muon identification and measurement is crucial not least because the *golden decay channel* (see Equation 4.1) of the Standard Model Higgs boson into muons

$$H \to ZZ^* \to 4\mu$$

generates a clear and undistorted signal. The measurements from the tracking detectors and the muon systems are combined in a *global momentum fit* to obtain an even better momentum resolution especially for low-momentum muons.

With its 25 000 m^2 of sensitive planes in four layers (*stations*) the muon detector had to be easy to construct and operate, and also low in cost. Different types of gaseous detectors have been chosen for the barrel and the endcap regions. The barrel muon stations are equipped with standard drift chambers with rectangular cells since the muon rate is low and the magnetic field is quite uniform. The situation is a bit more complicated at the endcaps where one has to cope with higher muon rates, non-uniform magnetic field, and non-negligible neutron background. Here Cathode Strip Chambers (CSCs) are used to provide fast signal collection and radiation hardness. The fast signal processing is very useful for the CMS trigger system that heavily relies on fast muon identification (see next section).

4.3.5 The Trigger System

With a design collision rate of 40 MHz (25 ns bunch crossing time) a fast decision system is needed in order to reduce the readout rate and filter only interesting events. This is the duty of the CMS trigger system. It consists of two main parts, the Level 1 Trigger (L1) and the High Level Trigger (HLT). Both together are designed to reduce the data rate down to about 100 Hz which can be stored on herd disk for analysis.

The L1 is a hardware-implemented decision system based on information from the calorimeters and the muon systems. A basic pattern recognition and grading algorithm is applied to certain low-granularity data blocks from the subdetector systems in order to detect possibly interesting events. The fast signal processing is ensured by the integration of Field Programmable Gate Arrays (FPGAs) and hard-wired Application Specific Integrated Circuits (ASICs), since the allowed time delay (latency) of the L1 Trigger is only 3.2 µs.

If an event of interest is detected, a Level 1 Accept (L1A) signal is issued and all data from the subdetector systems is read out and passed to the HLT. The HLT is a software-implemented algorithm running on a farm of conventional CPU cores. This makes it easy to adapt the selection criteria to the current needs and the latest findings. Since the HLT has access to the full event data, complex algorithms and very specific searches for event patterns can be deployed. An elaborate description of the trigger system can be found in [C$^+$08].

For the future Phase II Upgrade of the CMS detector efforts are being made to integrate parts of the tracker detector into the trigger system forming the so-called L1 track trigger. This will be done by using local trigger modules either with two layers of strip detectors (so-called *2S* modules) or with one pixel layer and one strip detector (*PS* modules), both linked to the same readout chip. This structure will allow the discrimination of high-p_T and low-p_T particles. A brief description of the track trigger plans can be found in [CMS11].

4.4 The CMS Pixel Detector

The CMS pixel detector is the innermost part of the tracker detector. It provides precise track seeds and is responsible for the good impact parameter resolution required for a high precision secondary vertex reconstruction. It consists of hybrid silicon pixel detectors with a carefully chosen pixel cell size of 100 µm × 150 µm (see 4.4.2) and about 66 million readout channels in total.

The pixel detector has an efficient per-pixel zero-suppression mechanism and several stages of readout buffers in order to reduce the amount of data to be read out. The sensor signals are transmitted as analog pulse height values enabling position resolution improvements by taking into account the charge sharing between pixels.

The pixel sensor and its Read Out Chip (ROC) are designed to withstand high particle fluences by choosing a radiation hard design. This is of particular importance since the pixel detector has to be operated for several years without any access for service or maintenance. Its innermost layer is exposed to a fluence of about $3 \times 10^{14}\,\mathrm{n_{eq}/cm^2}$ per year. For a detailed study of radiation exposure of the CMS pixel detector see [B$^+$02].

The pixel detector consists of modules that hold the sensor, the ROCs, and the Token Bit Manager (TBM), each connected by the High Density Interconnect (HDI), a flexible printed board. The functionality of the different parts is described in the following sections.

4.4.1 Geometry

The current pixel detector is designed to cover the pseudorapidity range $|\eta| \leq 2.5$ with three space points per particle track. This is achieved by dividing the detector in two parts as shown in Figure 4.3. The Barrel Pixel Detector (BPix) is the cylindrical center part of the detector while the end-cap Forward Pixel Detector (FPix) covers the forward sensitive area. An overview of both detector geometries can be found in [Erd10].

The three BPix layers are located at radii of 4.4 cm, 7.3 cm and 10.2 cm from the nominal interaction point. The detector modules are mounted on two half shell support structures which allow the insertion of the detector with the beam pipe installed. However, this requires special half modules with only eight ROCs for the edges of the shells to maintain full angular coverage. At both ends the support structure is connected to the support tubes via the end flanges. The support tubes hold readout and control electronics such as the analog and digital opto-hybrids for the optical fiber link to the pixel Front End Driver (FED) systems. The FPix disks are placed on each side of the barrel at $z = \pm 34.5$ cm and $z = \pm 46.5$ cm. Each disk is built up of so-called detector blades which hold the pixel sensor modules and cover the area from $r = 6$ cm till $r = 15$ cm. The blades are slightly tilted and overlap each other to guarantee a hermetic coverage in ϕ. Several sensors had to be designed for the FPix blades due to their different geometrical requirements depending on the radial position.

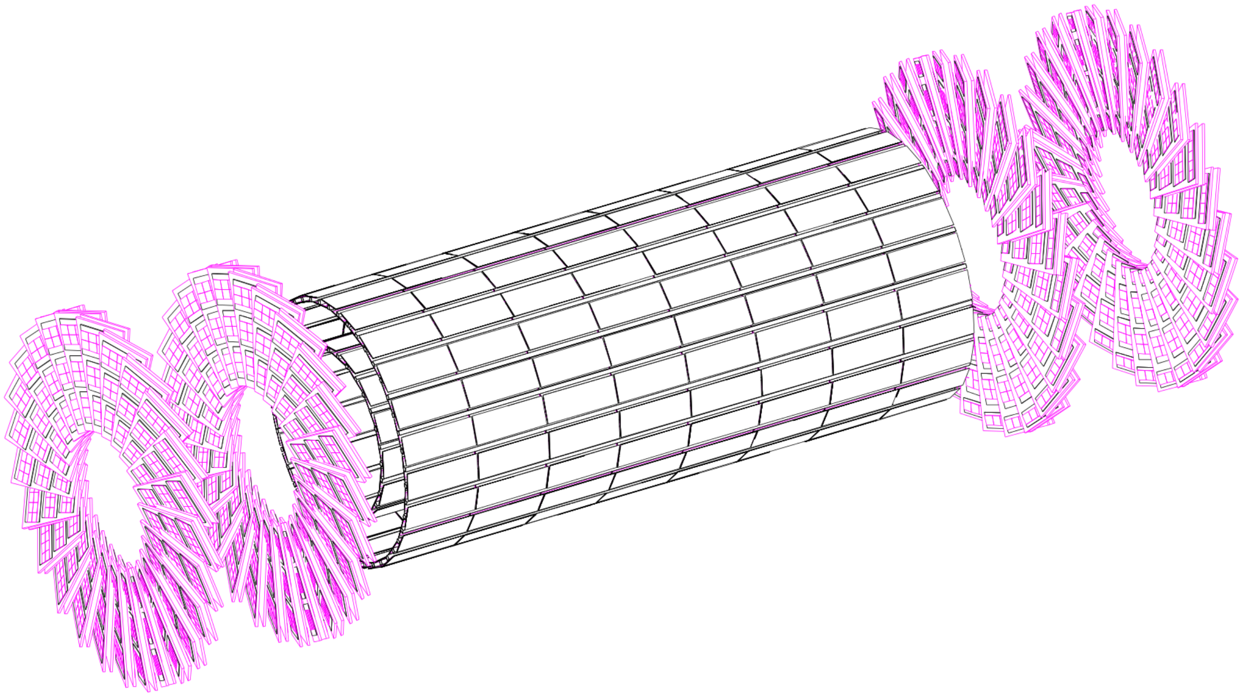


Figure 4.3: Sketch of the CMS pixel detector showing the three BPix layers in the barrel part (in black) of the detector and the two FPix disks for each of the endcaps (in magenta) [CMS98].

Several changes to the geometry are already scheduled for the CMS Phase I Upgrade. This includes an additional BPix layer and FPix disk as well as new ultra lightweight support structures and an advanced cooling system. A detailed description of the upgrade plans for the CMS pixel detector is given in Section 4.5.

4.4.2 The CMS Pixel Sensor

The CMS pixel detector uses silicon as its sensitive detector material and adopts the n-in-n implant concept where the sensor consists of a high-resistivity n-substrate and a high dose n^{++}-implant serving as electrode segment for the charge collection. The single pixel cells are isolated from each other by p-doped material, either called *p-stop* or *p-spray* depending on the doping technology used in the wafer processing. The n-in-n sensor concept has some advantages in comparison to other concepts, for instance the collection of electrons with their larger mobility instead of holes. As a consequence the Lorentz angle (cf. Section 2.3.2) is comparatively large which had an influence on the design of the pixel cells (see below).

Two different sensor designs for the BPix and FPix have been developed, sensors with moderated p-spray for the barrel part and sensors with p-stop for the forward region. Their slightly different layouts are presented in Figure 4.4. For the p-stop concept high-dose boron implants with very small spatial extend are introduced between the pixels. The adjustment of the dose is uncritical while the correct alignment of the implants is very important. With the (moderated) p-spray technology a lower boron dose implant with greater spatial extend is introduced into the bulk material. The dopant concentration varies between the pixel n-implants and assures a good High Voltage (HV) stability. All sensors have a thickness of 285 µm and are designed to deplete at comparable low bias voltages which reduces the leakage current and therefore the power consumption of the detector [Erd10].

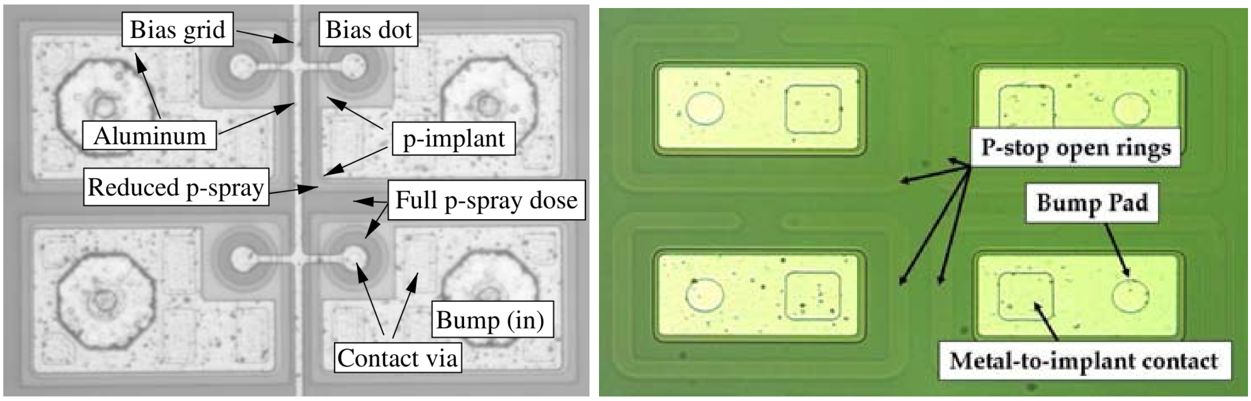


Figure 4.4: Pixel cell layout of the CMS pixel detector. Pixel cells with moderated *p*-spray are used in the BPix (left), while the FPix detector uses pixel cells with open *p*-stops (right) [Erd10].

The single pixel cells have a size of 100 µm × 150 µm. This pixel shape is determined by the best possible resolution, the cost and the overall feasibility of the detector. The best spatial resolution can be achieved by sharing the produced charge carriers between two pixel cells as described in Section 2.3.3. Thus the ideal size of a pixel cell perpendicular to magnetic field ($r\phi$) can be determined by calculating the Lorentz angle as described in [Erd10]. With a Lorentz angle of about 25° and a sensor bulk thickness of 285 µm the charge spreads over a maximal length of about 150 µm [CMS10]. Furthermore charge charing due to delta electrons has to be taken into account. Choosing a pixel size of 100 µm ensures charge sharing among only two pixels while the other dimension is fixed by the required area for the underlying Pixel Unit Cell (PUC) on the ROC that holds the per-pixel readout logic. For the CMS pixel detector this results in a length of 150 µm in z direction. With 16 ROCs and 52 × 80 pixels each, this results in an active sensor size of 64.50 mm (z) × 16.05 mm ($r\phi$) and 16 × 4160 pixels per module for the barrel sensors. For the FPix five different sensor sizes have been produced to meet the requirements of the disk geometry while the ROC is the same as for the BPix modules.

The connection between the sensor pixel cell and the Pixel Unit Cell on the ROC is established via Indium solder bump bonds. A small amount of indium is placed on the bump pad of the pixel cell, reflown to a ball shape by heating the sensor and then bonded to the ROC via flip-chip process. An elaborate description of the indium bump process developed for the CMS pixel detector can be found in [B$^+$06].

4.4.3 The Analog *PSI46v2* Readout Chip

The *PSI46v2* Read Out Chip (ROC) holds all front-end electronics and logic needed for an efficient pixel detector operation. It takes care of the processing and buffering of pixel hits and the pulse height signals and coordinates the readout and data transfer. An overview of the basic functionality, registers and block diagrams of the ROC and its components can be found in the *PSI46* external specification [Gab05].

The ROCs are fabricated in a standard radiation hard 0.25 µm CMOS technology to withstand the high particle flux in the LHC environment. Additionally the layout of the Read Out Chip is optimized towards a high radiation tolerance to withstand high particle fluences, for instance by introducing guard rings and usage of enclosed transistors. A description of the optimization techniques used in the development of the *PSI46v2* ROC is given in [S$^+$00].

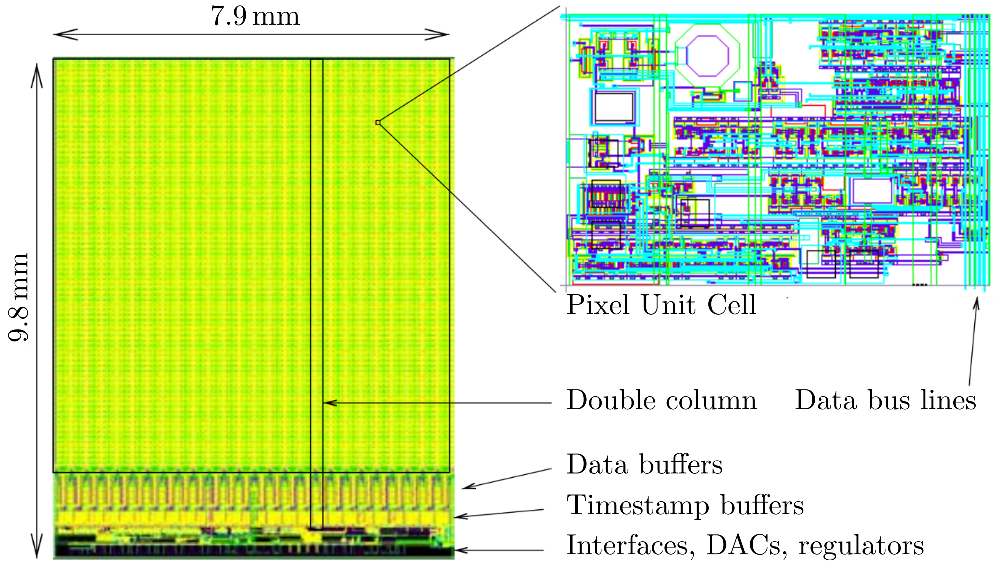


Figure 4.5: Overview of the *PSI46v2* Read Out Chip showing the different chip regions. The largest area is the active pixel matrix containing the PUCs. The active area is followed by the double column readout buffers and the readout logic. An enlarged view of the PUC is shown at the right. Modified from [Erd10].

The *PSI46v2* ROC can be divided into three main regions as shown in Figure 4.5. The largest part hosts the Pixel Unit Cells (PUCs) which exactly fit the geometry of the corresponding sensor pixels and are connected to them via Indium bumps. The PUC contains the readout logic for a single pixel: preamplifier, shaper, comparator, the programmable trim and mask bits for muting single pixels, and the comparator threshold and calibration Digital-Analog Converter (DAC) settings to tune the per-pixel individual readout and gain calibration. The bump connects the PUC to the sensor pixel and is located on the bump pad at the far end from the data bus to reduce crosstalk as shown in Figure 4.5. The drawing shows a cell in the left part of a Double-Column Interface (DCI) (see below), the corresponding right cell is mirror-symmetrical. Beside processing the signal from the sensor the ROC is also able to introduce calibration signals via the *Vcal* DAC. This DAC signal is used as reference for chip calibration and comparison of signal heights from different pixels. A detailed description of the calibration process, temperature effects, and rate dependency of the analog ROC is given in [Hoß12].

After a particle traversed the sensor material, the charge is collected via the applied bias voltage. The signal is then processed by the pre-amplifier and shaper before it reaches the comparator. The comparator uses the *Vthreshold* setting to compare the signal voltage against and decide whether it is a valid hit or just sensor noise. If the signal passes the comparator it is stored in the sample-and-hold circuit until the readout token is passed to the PUC logic. A simplified block diagram of the PUC is shown in Figure 4.6. The pixels are grouped and read out by 26 DCIs, each connecting 160 single pixels. The ROC internal readout token is passed from one pixel cell to the next, reading out the hits and storing them in the DCI readout buffers, the second region on the ROC. Additionally the timestamp of the current bunch crossing is recorded in the timestamp buffers. This allows the precise determination of the particle collision time.

The third region holds the global ROC readout periphery such as the token controller and the clock distribution. If an external readout token arrives the token controller passes a DCI readout token to the double column buffers and reads them out one by one. During this time the DCI is dead

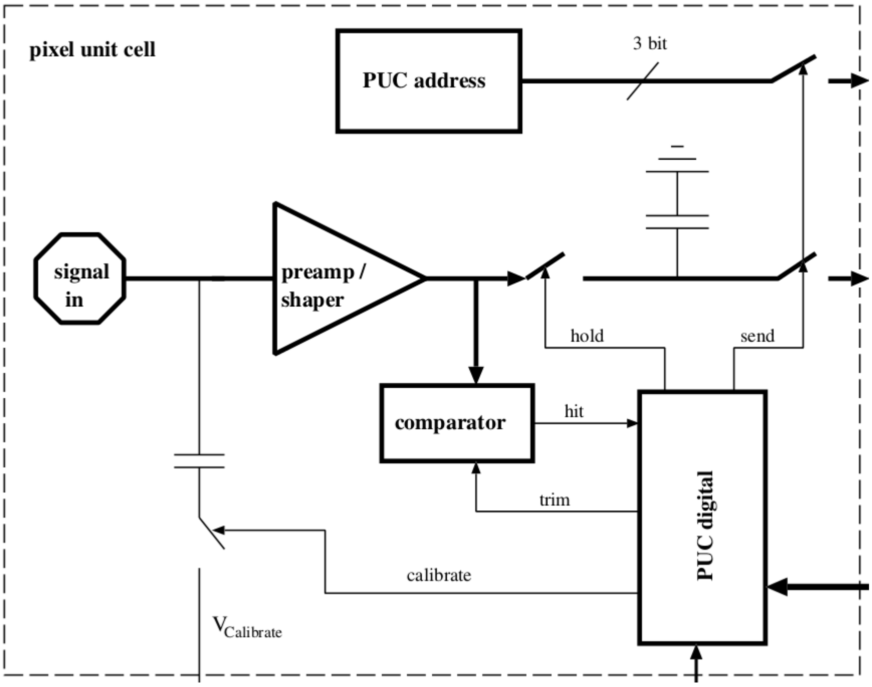


Figure 4.6: Simplified block diagram of *PSI46v2* Pixel Unit Cell showing the signal path from the bump pad (*signal in*) through the pre-amplifier and shaper to the comparator and sample-and-hold circuit. Calibration signals can be injected via the calibration capacitor with the charging voltage $V_{\mathrm{Calibrate}}$ [Gab05].

and not responsive to new hits. Currently all pixel hit information is transferred from the ROC to the TBM via analog level encoding (see section 7.5 for a detailed description). A new version of the ROC with digital data encoding (*PSI46dig*) will be used for the CMS Phase I Upgrade and is described in Section 4.5.

4.4.4 TBM and Module Structure

The sensor and the ROCs are grouped together in *modules* providing the electrical infrastructure for the readout. One barrel pixel module hosts 16 ROCs, the TBM, and the HDI. These components enable the efficient collection and transfer of the data recorded by the ROCs to the HLT facilities outside of the CMS detector where the next event selection step takes place. The whole module is glued to silicon nitride (Si_3N_4) base strips to allow solid mounting onto the support structure.

The HDI provides wiring for each of the ROCs such as the power distribution, the clock, and the signal routing. It is glued directly onto the active sensor and holds the wire bonding pads for the TBM and the ROC connections.

The TBM has been designed to collect the data from the ROCs within a given time, to transmit it while assuring the integrity, and to provide precise clocks for bunch crossing identification. Furthermore it is in charge for distributing the CMS trigger signals. When it receives the L1A signal (as described in Section 4.3.5) it distributes a readout token to the ROCs in order to read them out one by one. The data is then collected and transmitted it to the Analog Opto-Hybrids (AOHs) on the support tubes of the pixel detector. Those pass the signal to the Data Acquisition (DAQ) systems outside the CMS detector. The latest analog TBM is referred to as the so-called TBM06. With the Phase I Upgrade the list of features of the TBM will become even longer. The changes are described in detail in Section 4.5.3.

4.5 The CMS Pixel Detector Phase I Upgrade

After the LHC Phase I optimization [Zim09] with a peak luminosity exceeding $1 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ the current pixel detector will not be able to deliver the same data quality as it does nowadays due to readout inefficiencies and radiation damage. Therefore it will be replaced with a new detector with optimized readout chips and further improvements in material budget and cooling. All dependent variables have been carefully checked and a new readout system has been designed to cope with the trigger rates expected at double design luminosity of $\mathcal{L} = 2 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [ML09].

The Phase I Upgrade leaves the active sensor material and layout untouched, most changes apply to the readout chip and the mechanical periphery such as the support structure, the cooling and the front-end electronics. An extensive description of the planned upgrades for the CMS detector through 2020 can be found in the CMS Collaboration Upgrade Technical Proposal [CMS11]. The Phase I Upgrade of the pixel detector is described in the recently published CMS Technical Design Report for the Pixel Detector Upgrade [CMS12].

4.5.1 Motivation for the Detector Upgrade

In order to keep pace with the progress of machine development at the LHC which will deliver higher energetic particles at higher luminosities from year to year, the CMS collaboration refined the detector concept of the pixel detector to withstand higher particle fluxes, events with higher pileup and higher data rates.

A fourth layer in the barrel region and an additional endcap disc will close the gap between the pixel detector and the inner strip tracker and improve the track fitting and seed resolution. The beampipe radius will be reduced from an outer diameter of 59.6 mm to 45.0 mm which allows the innermost barrel layer to move closer to the interaction point to a radius of 30.0 mm [CMS12]. With the additional layer and reduced radii the upgraded BPix detector will consist of 1216 modules and about 80 million readout channels in total instead of 48 million channels in the current barrel detector. Furthermore the fourth barrel layer allows the improvement of the seed efficiency by using a *3-of-4* triplet for track seeding instead of the current *3-of-3* pixel triplet. This has an impact on the tracking efficiency especially in the forward regions around $\eta \approx 2 - 2.5$ as shown by [Tri09].

At higher luminosities the pileup will rise. At today's luminosity CMS has to cope with about 25 pileup events per bunch crossing (every 50 ns), while in Phase I the pileup depends on the pending decision whether the LHC will provide a 25 ns bunch spacing (pileup 50) or stay at 50 ns (pileup 100, cf. [CMS12]). The fourth layer and reduced radii enable the CMS pixel detector to distinguish between more pileup collisions in both cases.

The FPix detector is extended by the addition of another sensor disk on each side to support the aim of a full four-hit coverage. Since the production of the FPix modules has been quite complicated due to the different sensor sizes, the blades of the forward disks are being redesigned for the upgrade to allow the usage of a single module layout instead of five different ones. This will reduce the required amount of work for the production, qualification, and calibration significantly.

With the new pixel detector in place significantly more modules have to be read out. Furthermore a larger amount of data per module will be recorded at higher luminosities and has to be transmitted and buffered. The number of optical fibers for data transfer that go through the other parts of CMS to the Data Acquisition systems will probably stay the same even tough some discussions are going on about inserting additional fibers. The higher overall data rate implies a change in the readout of the detector modules in both cases.

4.5.2 The Data Loss Mechanism

The data loss mechanism of the current *PSI46v2* ROC has been studied in detail to improve the performance for the Phase I Upgrade by [Käs08, Mei11]. The pixel detector readout is zero-suppressed which means that each Pixel Unit Cell is equipped with a comparator. Only if the measured voltage level lies above an adjustable threshold the pixel is accounted as hit and the ADC value is buffered for readout. This technique allows to drastically reduce the amount of data that has to be transmitted and analyzed afterwards. As consequence the readout is data driven and not only dependent on trigger latency but also on the measured track rate. Without zero-suppression the amount of data per trigger would be always the same (one value per pixel) independent of the number of particles crossing the detector.

In the *PSI46v2* chip each DCI has 32 data buffers and 12 time stamp buffers. When a hit is recorded the hit information itself (pixel ID and pulse height) is stored in the data buffer while the corresponding time (bunch crossing) is stored in the timestamp buffer. Since the average cluster size is larger than one pixel, several pixel hits are be linked to the same timestamp and more data buffers than timestamp buffers are required. At higher luminosities the lack of data buffer cells to store hit information between two readout triggers will introduce a significant data loss. Currently the total estimated data loss is about 3.8 % for the innermost layer at $\mathcal{L} = 1 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, including buffer overflow, DCI dead time, and transmission inefficiencies. At double luminosity of $\mathcal{L} = 2 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ (Phase I) the current chip without any modifications would be subject to a data loss of about 16 % in the inner barrel layer.

When a double columns receives a readout token the data acquisition is stopped in order not to overwrite valid hit information. While waiting for the readout the corresponding double column is dead and does not respond to new pixel hits. With increasing data rates and readout time for each DCI the dead time caused by this readout cycle will grow. This dead time can be avoided by introducing an additional global ROC data buffer. This splits up the buffer readout into two stages: The ROC internal DCI readout to the new global data buffer and the external readout triggered by the L1A signal. The internal readout cycle is much faster since the data only has to be transmitted to the local buffer and therefore reduces the DCI dead time.

Another problem arising with higher data rates is the transfer of the hit information itself. The current analog data transfer is driven by the 40 MHz LHC clock and a levels-encoded analog signal for the pixel ID and the header information (cf. Section 7.2.2) which results in a total nominal bandwidth of about 100 MBit/s. With the current luminosity and a 100 kHz Level 1 Trigger rate the bandwidth load is about 50 %. Doubling the data volume would therefore exceed the available bandwidth.

4.5.3 Changes to Readout Chip and Module

The ROC for the CMS Phase I Upgrade is an evolution of the present *PSI46v2* ROC. The pixel structure and readout scheme stay the same while most of the changes apply to the data and timestamp buffers and the readout link. All changes to the readout architecture and the token chains are described in [Mei11].

The new *PSI46dig* ROC contains 80 data buffers and 24 time stamp buffers. To fit the extended buffers into the same area as on the *PSI46v2* ROC the memory cell length has been reduced from 36 µm to 23 µm. In order to unburden the DCI buffer a global ROC First In First Out (FIFO) readout buffer is created. As soon as a L1A signal arrives, an internal token is passed to the double columns and the data is stored in the global ROC buffer. This improves the readout speed since buffer filling is a fast operation and the new TBM parallelizes the readout of several ROCs by entangling the readout in data streams (see below).

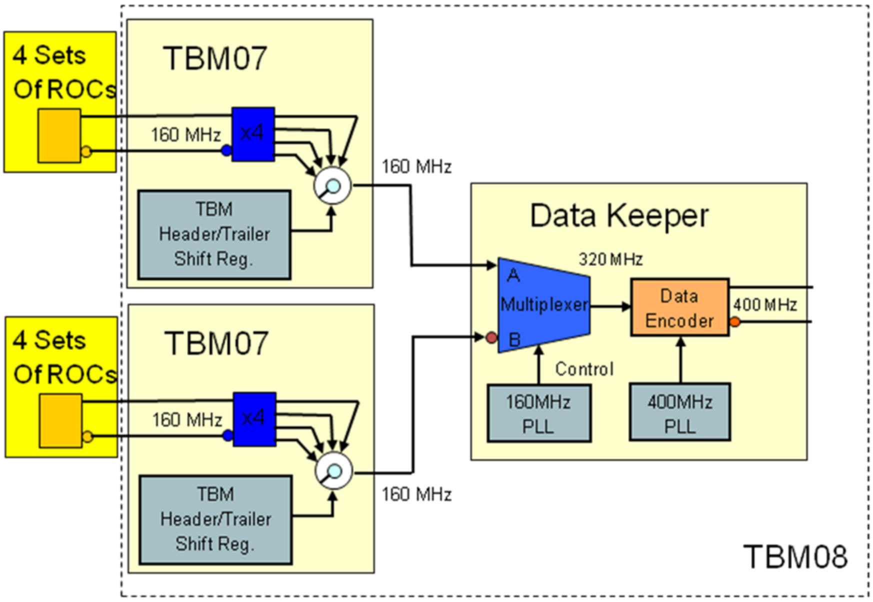


Figure 4.7: Sketch of the new TBM08 with two TBM cores and the DataKeeper for the 320 MHz channel multiplexing and the 4B5B encoding. Each of the TBM07 cores will controll and read out four ROCs [CMS12].

Since the total number of fibers and therefore the data interface to the FEDs have to stay the same a new data readout scheme has to be provided with a faster clock and a bandwidth capable of the high data rates. A Phase-locked loop (PLL) is added to the ROC to supply the logic with the needed high-frequency clock.

To cope with the higher data rates at double luminosity and the increased number of detector readout channels the analog level-encoded readout technique is dropped in favor of a new readout infrastructure. The data is transmitted from the ROC to the TBM via a fully digital 160 MBit/s interface. Several versions of the TBM are being developed for different positions in the detector. An updated version of the current TBM, the TBM07 will be used for the outer FPix only. The TBM08 comprises two TBM07 cores with digital data interfaces and the so called *DataKeeper* and will be used for the outer barrel layers and the inner FPix of the new pixel detector. It manages two buses (A/B) with 8 ROCs each and the *DataKeeper* multiplexes the two data streams bitwise into one single 320 MBit/s uplink stream as shown in Figure 4.7. Channel B is sent inverted to allow an easy separation of the streams by the receiver. For the innermost layer even this configuration might not be enough for all the accumulated data so a different layout has been chosen. Two slightly modified TBMs (TBM09, with changed token routing) will reside on one module, both operating two buses with 4 ROCs each, passing a data stream of 2 × 320 Mbit/s to the AOHs on the pixel detector support tubes.

The data stream is encoded by the *DataKeeper* using a standard Non Return to Zero Invert (NRZI) encoding. NRZI ensures that switching the polarity of the signal cables does not invert the data bit sequence. Instead of transmitting pattern itself (e.g. 1111) every bit corresponds either to a transition of the physical level (1) or to no transition of the signal (0). A separate clock signal is needed and uniform patterns like 0000 could lead to problems since it contains no signal transition. Therefore a 4-to-5 bit (4B5B) encoding scheme will be adopted for the data stream. This increases the required band width from 320 MBit/s to 400 MBit/s but provides guaranteed signal transitions by assigning a predefined word of 5 bit containing at least two signal transitions to every 4 bit block. By this the transmission integrity of the data can be improved.

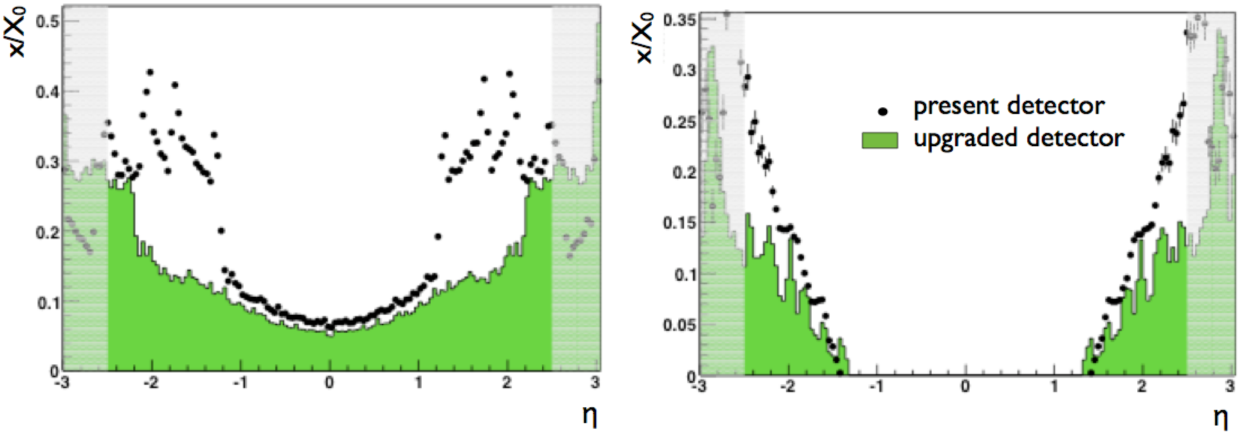


Figure 4.8: Material budget for the current and upgraded CMS pixel detector. Shown is the fraction of radiation length as a function of the pseudorapidity η. The left plot shows the BPix detector, the right plot the FPix. The volume up to a pseudorapidity of $|\eta| < 2.5$ is used for track reconstruction and has a reduced material budget [Fav10].

A micro-twisted pair cable with a standardized length of 0.95 m will be used for both powering and data transmission. The new simplified HDI design needs fewer Surface Mounted Devices (SMDs) such as high voltage capacitors and introduces less material into the central tracking volume. This can be achieved by the development of new DC-DC converters mounted on the detector support tubes outside the central tracking volume. The DC-DC converters hold some of the components formerly placed on the HDI and ensure the power supply of all additional detector modules with the limited number of power cables from the service cavern.

4.5.4 Geometry and Cooling

In order to integrate well into the CMS detector, the design of the upgraded pixel detector has to fit all existing cabling, dimensions, and connections. A new ultra lightweight support structure made of carbon fiber ladders is prepared on which the modules will be mounted. This reduces the material budget in the tracking volume significantly as shown in Figure 4.8. A new cooling agent has been chosen; instead of C_6F_{14} a two-phase CO_2 cooling system will be used. This allows smaller tube radii and lower cooling temperatures. The new modules are to be operated at a temperature of −20 °C.

In addition to the performance upgrades some simplifications will be made such as a modified design of the half shells. The new layout does not require half modules, so only one type of modules has to be produced and tested. Furthermore all modules will be equipped with the 0.95 m micro-twisted pair cable instead of separated Kapton data cable and power line with individual lengths. This lead to problems in the past since modules had to be assembled for a specific place inside the detector due to their different cable lengths.

The front-end electronics of the barrel pixel detector, such as AOHs and DC-DC converters are being moved further away from the sensitive tracking towards the supply tubes of the support structure to reduce the material inside the central tracking volume. As one can see in Figure 4.8 the material budget is significantly reduced by the upgrade, although the total number of detector modules is increased. Especially in the regions $1 \leq |\eta| \leq 2$ for the BPix and $2 \leq |\eta| \leq 2.5$ for the FPix detector profit from the changes.

4.5.5 Barrel Module Production

The current BPix detector has been completely produced, tested and calibrated by the Paul-Scherrer-Institut (PSI) in Villigen, Switzerland but the production of the new pixel detector with an additional layer is split up among several institutes. The innermost and the second layer will be produced at PSI while the production of the third layer is shared among Institutes from in Italy, Taiwan, Finland, and CERN.

The fourth layer will be assembled and tested in Germany with four different institutes participating. It consists of 512 modules, but approximately 700 modules will be produced for a safety margin concerning defective modules and to have spare parts for later replacements. Half of them will be assembled and tested at Karlsruhe Institute of Technology (KIT) and then calibrated at Rheinisch-Westfälische Technische Hochschule Aachen (RWTH). The other half will be produced and calibrated in Hamburg by the University of Hamburg and Deutsches Elektronen-Synchrotron (DESY). A brief summary of the assembly procedure steps of a CMS pixel detector module is given in [K$^+$06].

In science one tries to tell people, in such a way
as to be understood by everyone, something that
no one ever knew before.
But in poetry, it's the exact opposite.

P. A. M. Dirac

5. Pattern Recognition in Tracking Detectors

Pattern recognition is one of the crucial tasks in particle physics. Particles interact with detector material and produce (mostly electronic) signals that can be stored. This raw detector response consisting of abstract information has to be translated back into the physics properties of the single particles and the overall event.

In the 1960s pattern recognition was manual work at bubble chambers like the neutrino experiment *Gargamelle* [H^{+}73], although it was somewhat cumbersome and time consuming even with the relatively small amount of data compared to modern particle physics experiments. Bubble chambers were containers filled with liquids under pressure, mostly hydrogen, krypton, or xenon. Charged particles passing through the liquid left tiny bubbles revealing their path. These paths were photographed and analyzed manually by drawing the particle tracks, measuring the radii and searching for missing momenta or energies.

With today's particle physics experiments with almost a billion events per second this is obviously not possible anymore. Instead large computer clusters are used to analyze the data produced by the detectors. Also the algorithms evolved together with the more and more complex detector systems now forming several levels of pattern recognition.

Due to the large amount of data and the high event rate pattern recognition is a time-critical task, especially if the information is used for trigger decisions. For example at the CMS experiment at the LHC the allowed trigger latency is 3.2 µs (cf. Section 4.3.5). Within this time all necessary information has to be collected, processed and the trigger decision communicated. In offline data analysis the situation is slightly more relaxed, but still the turn-around time for a new algorithm has to be low in order to be competitive with other experiments. Event displays can be used to cross-check the accordance of the raw data with the reconstructed pattern from the algorithms for selected events since the human brain provides an excellent pattern recognition.

This chapter concentrates on the pattern recognition algorithms used in pixel tracking devices. Due to the existence of many algorithms for different geometries and special purposes only a brief overview is given, and only the algorithms used for the high-rate test beam (see Chapter 6) are described in detail. An elaborate evaluation of different pattern recognition algorithms can be found in [Man04].

5.1 Clusters, Hits, and Tracks

The detector data from tracking devices has to be processed in several steps, starting with combination of local responses and then applying a global pattern recognition.

The first step is the *clustering* of the detector data. Depending on the geometry and the granularity of the tracker, several entities of the detector respond to the passage of a single particle through the device. Clustering algorithms try to group this information together, forming a *cluster*. A cluster is therefore a set of pixel hits belonging to the same particle track. Pixel detectors are often designed to provide multi-pixel clusters for each particle track to improve the spatial resolution (cf. Section 2.3.3).

After grouping the pixels the clusters have to be translated into *hit positions*. The exact penetration point of the particle into the detector surface can be obtained using different methods described in Section 5.1.2, depending on the information available.

These hits in the individual detector planes can then be connected with particle tracks. These trajectories aim to provide the full information or the particle passage through the detector even though only a set of space points has really been measured. The shape of the tracks depends heavily on the situation in the conducted experiment such as magnetic field or expected momenta and energies of the particles to be tracked.

As discussed before (cf. Section 2.3.3) the information gathered by a pixel detector can either be *binary* with the only information being the hit itself, or *charge sensitive* with additional information about the energy deposited in the corresponding pixel. Beside this the front-end electronics can differ in the format of the data they are delivering. Two detector data formats have to be distinguished:

- The **full-frame** format is delivered as a full set of data for each channel of the detector. For a pixel detector this would e.g. be a full-size matrix with an entry for every pixel cell.
- The **zero-suppressed** or **sparse** data format contains only pixel cells that actually received a hit. Each data sequence from a pixel hit includes the address (e.g. the pixel ID or x and y coordinates) of the pixel and possibly its ADC value if the detector comes with charge sensitive readout.

The sparse data format can be an advantage and reduce the amount of data to be transferred enormously, especially for pixel detectors in which only some pixel hits among thousands of pixels are expected. The drawback is the more complex front-end electronics with a comparator which has to be integrated into the readout chips.

5.1.1 Clustering Algorithms

Different clustering algorithms can be applied to the detector raw data such as fixed frame, bricked or simple sparse clustering. The choice of the algorithm depends on the requirements, the detector, and the experimental setup. Most clustering algorithms can be applied to both sparse data (as for the CMS pixel detector, cf. 4.4.3) or to full-frame detector data without zero-suppression. If full-frame data is used, usually additional information such as the sensor noise is needed to manually set a threshold and distinguish between hit information and sensor noise.

All algorithms described here apply primarily to pixel detectors, although some of them can be used for other tracking devices as well. The implementation of the different algorithms in the EUTelescope software framework (cf. Section 7.1) is described in [B$^+$07b].

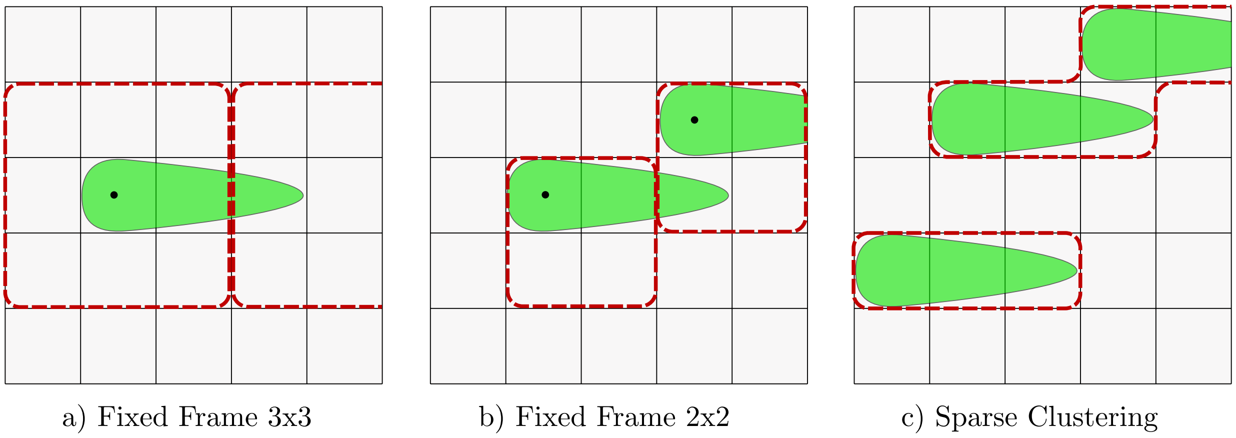


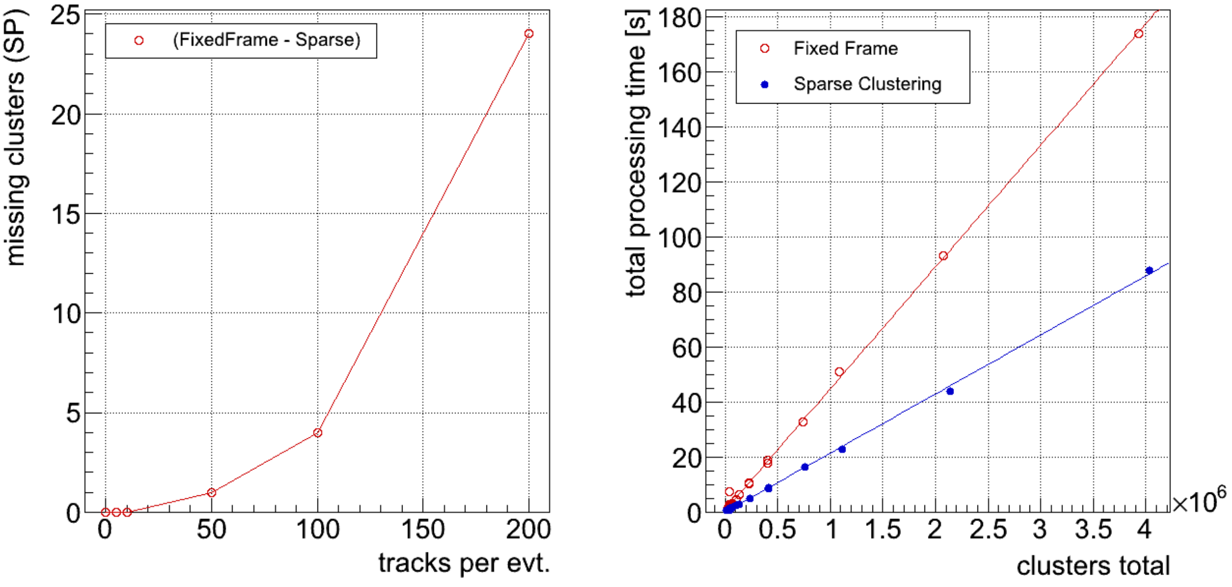
Figure 5.1: Cluster merging and cluster splitting. Charge carriers are marked in green, the clusters found by the algorithms in red. The black dots mark the cluster seeds. The fixed frame algorithm has difficulties with cluster splitting in case of charge sharing (a), whereas cluster merging is not a problem and tracks from different particles can be separated (b). Sparse clustering has difficulties with cluster merging in case of high occupancies but detects the correct shape of the clusters (c).

Fixed Frame Clustering

The *fixed frame clustering* is a fast and easy algorithm. In a first iteration it searches for cluster seeds in the pixel matrix which are defined as pixels with a distinct signal or a given Signal-to-Noise Ratio (SNR). These pixels are then sorted in descending order according to their signal and cluster candidates are formed by selecting a seed candidate and adding all adjacent pixels to the cluster. Pixels already belonging to a cluster are removed from the seed candidate vector in order to avoid ambiguities. The cluster size of $n \times m$ pixels has to be defined beforehand, knowing the expected cluster shape of the experiment.

Fixed frame clustering is a very fast algorithm and robust against cluster merging since the cluster size is predefined (see below). However one is limited to a single cluster shape and thus very similar particles (momentum, angle of incidence). For particles with different charge sharing behavior the clusters found by the algorithm might differ from the real clusters produced by the particles. The fixed frame algorithm can produce unexpected results, especially if expecting mostly 1- and 2-pixel clusters. It could either connect independent clusters or split single clusters depending on the chosen cluster size. Figure 5.1 (a) shows an example of a particle incidence with strong charge sharing among several pixel cells. With a fixed cluster size this pattern can easily generate several unconnected clusters. On the other hand when expecting high occupancies the fixed frame clustering is robust against cluster merging. Particles passages activating several adjacent pixels can still be isolated due to the fixed cluster size, resulting in two cluster seeds as sketched in Figure 5.1 (b), assuming a correct choice of the cluster size.

The *minimum cluster SNR* is a useful parameter often implemented in this algorithm. It represents the minimum overall SNR required for group of pixels in order to be accepted as a cluster. This allows the suppression of pixel groups with low signals. The fixed frame algorithm can be used for both sparse and full-frame data. When using sparse data the resulting cluster does of course not necessarily contain $n \times m$ pixels, but only the ones that received a hit.



(a) Missing clusters

(b) Processing speed

Figure 5.2: Comparison of clustering algorithm performances. (a) shows the difference of found clusters between sparse and fixed frame clustering for different numbers of tracks through a simulated 80×52 pixel detector plane. Only above approx. 50 tracks a significant amount of cluster merging is visible. (b) shows the total processing time for the two algorithms, sparse clustering is faster by a factor of two.

Sparse Clustering

In contrast to the previously described algorithm *sparse clustering* is not based on a fixed cluster size but allows every possible shape of connected pixels. The approach of the algorithm is quite simple and even faster than fixed frame clustering: for every pixel above a given SNR threshold the algorithm looks for other pixels within the maximum distance and connects them to a cluster. Again, all pixels assigned to a cluster are removed from the seed candidate list to avoid ambiguities. The maximum distance is a user-defined parameter and usually set to something close to one pixel pitch, so only adjacent pixels are connected. The output of the algorithm consists of clusters with a specific shape and often this is a very useful information, e.g. when studying the behavior of tilted sensors in a particle beam. On the other hand this behavior can lead to cluster merging in cases of high occupancies as shown in Figure 5.1 (c). Monte Carlo simulations with different sensor occupancies and number of particle tracks per trigger have been prepared [Lu12] to compare the performance of the two clustering algorithms using EUTelescope framework processors (cf. Section 7.1). Figure 5.2 (a) shows the difference in clusters found per event in a single pixel detector plane between the sparse and the fixed frame algorithm. As one can see the algorithms are comparable with one another up to a rate of 50 tracks per event, while the sparse algorithm is faster due to the missing sorting loop as presented in Figure 5.2 (b).

Other ideas for advanced clustering are currently under developments, e.g. the CMS *PSI46dig* Column Drain Cluster (CDC) algorithm. This approach will implement parts of the clustering processes directly into the front-end chips. After receiving a hit, each of the Double-Column Interfaces (DCIs) (cf. Section 4.4.3) performs a dynamic 4-by-4 pixel cluster search and the whole cluster is transmitted to the periphery. This would allow a more efficient data transfer and could be used as first clustering information. More information on CDC can be found in [CMS12].

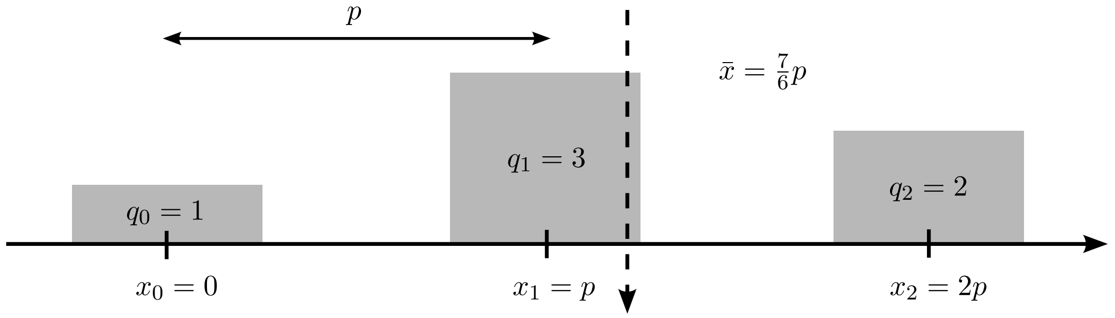


Figure 5.3: Example of center-of-gravity energy weighting between three adjacent pixels. The position $\bar{x}$ of the particle passage is calculated using the three charge values, the pixel positions, and the linear function from Equation 5.1. p is the pitch, the distance between two pixel centers.

5.1.2 Cluster Centers and Energy Weighting

Once pixels are grouped to clusters the cluster center and thus the penetration point of the particle has to be determined. This can be done by either just calculating the geometric center of the cluster, or by applying an energy weighting algorithm to the cluster. Energy weighting requires charge sensitive detector readout since the signal pulse heights of the single pixels are used to find the cluster center.

Center-of-Gravity Method

The center-of-gravity method calculates the cluster center by weighting the signals of all involved pixels. The deposited charge in each pixel is assumed to be proportional to the position of the particle traversal, so a linear weighting function can be used for calculating the cluster center:

$$\bar{x} = \frac{\sum_i q_i x_i}{\sum_i q_i}, \tag{5.1}$$

where $\bar{x}$ is the position of the particle track, q_i the charge deposited in pixel i and x_i the relative position of the corresponding pixel. Figure 5.3 shows an example of three pixels with the charges q_i. The cluster center is calculated to be:

$$\bar{x} = \frac{1 \cdot 0p + 3 \cdot p + 2 \cdot 2p}{6} = \frac{7}{6} \cdot p,$$

with p being the pitch of the pixels. This allows a very simple interpolation between two or more pixels from the same cluster.

Eta Function

Assuming a uniform charge sharing distribution might be too simple in some cases. The η function energy weighting method takes the specific shape of the charge cloud into account. Instead of a linear weighting function as applied in the center-of-gravity method, a more complex function is used reflecting the actual charge distribution among the pixel cells. This η function is determined by precise measurements of the charge distribution on the sensor with known point of impact, for example by inducing charge carriers at different positions between the pixels using a laser.

After having specified the impact position of the particle, its coordinates are usually translated from the local reference frame on the pixel sensor to a 3D space point in the global frame of reference of the tracking detector. With this space point further pattern recognition can be performed.

5.2 Detector Alignment

Alignment described the procedure of correcting mechanical displacements of the detector in recorded data and is an important task for high-precision track measurements. The mechanical precision of the detector is limited by the manufacturing process and additional displacements of the detector during installation cannot be avoided. Therefore the exact positions of the detector planes are not known and hit positions are shifted by the displacements. In order not to lose spatial resolution, the sensor planes are aligned. The recorded data can then be corrected by moving the fitted hit positions by the alignment parameters determined during this procedure. The aim of alignment is the reduction of the χ^2 value of the global track fit to improve the position and vertex resolution. This implies the minimization of the hit position residuals in the sensor planes of the detector. A residual is e.g. the mean distance between a measured and expected hit position from a fitted particle track in the given detector plane. A more general definition of residuals is given in the next section.

The so-called *track-based alignment* approach is a widely used alignment method. In large tracking detectors like the CMS tracker cosmic muons can be used to align the different layers. They create a straight path through the whole tracking volume which allows the calculation of the shifts and rotations of the single detector layers. However, in beam telescope detectors which just consist of parallel detector planes another approach is taken: the recorded data from the measurements are used to fit preliminary tracks. These tracks and the corresponding residuals are then used to move the telescope layers with respect to each other.

In general, alignment involves a large set of parameters and complex algorithms have been developed to solve the mathematical problems. For beam telescopes with their comparatively simple geometry the misalignment can be described by a sets of six alignment parameters for each sensor plane. The shifts Δx, Δy and Δz along the axes x, y and z, and the three rotations θ_x, θ_x and θ_z around these axes are needed. For planar and rigid sensors the hit positions can then be described by:

$$x = (\cos\theta_y \cos\theta_z) \cdot x_{\text{meas}} + (\cos\theta_x \sin\theta_z - \sin\theta_x \sin\theta_y \cos\theta_z) + \Delta x \tag{5.2}$$

$$y = (-\cos\theta_y \sin\theta_z) \cdot y_{\text{meas}} + (\cos\theta_x \cos\theta_z + \sin\theta_x \sin\theta_y \sin\theta_z) + \Delta y, \tag{5.3}$$

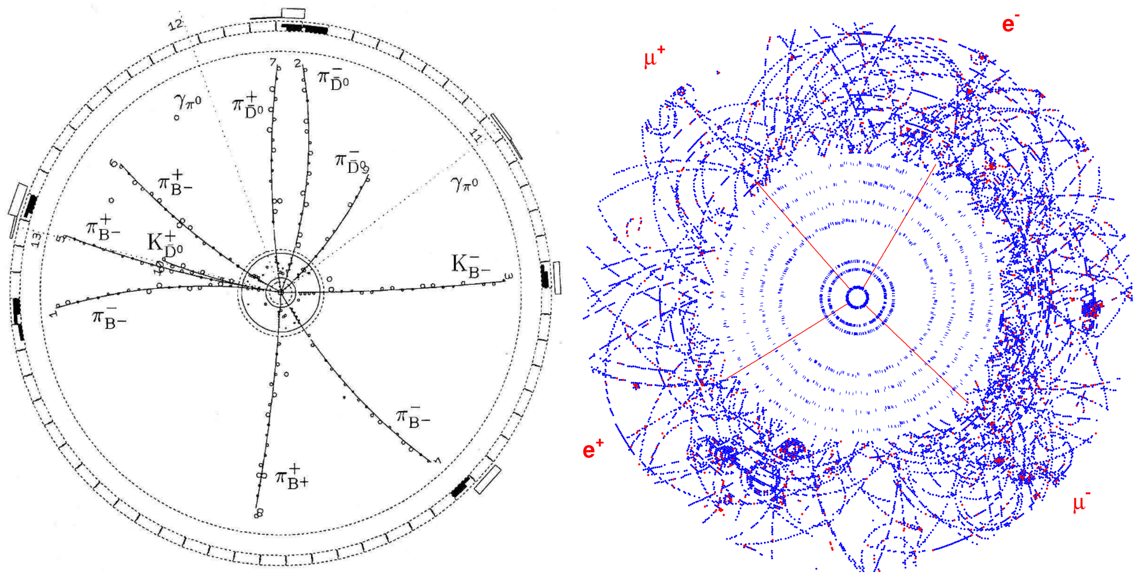
where x_{meas}, y_{meas} are the measured hit positions in the sensor plane [B$^+$07b].

The track-based alignment algorithm used in the CMS detector is called *Millepede II* and a description can be obtained from [Blo07]. The algorithm implements a very efficient matrix equation solver and is thus able to calculate the alignment constants for large sets of parameters in reasonable time. An implementation for the EUTelescope software framework exists and the algorithm can therefore be easily applied to beam telescope alignment problems. An overview of different approaches, alignment algorithms, and mathematical methods such as outlier rejection and solution of large matrix equations can be found in [Blo06b, Blo06c].

Weak Modes

Some degrees of freedom are undefined or only weakly defined when using track-based alignment algorithms which rely on the minimization of the track residuals. These so-called *weak modes* occur especially in simple, symmetrical geometries such as beam telescopes with parallel sensor planes. A typical weak mode for a beam telescope is e.g. a constant shift of all sensor planes in either x or y direction. The global χ^2 value of the fit stays the same while the telescope layers are shifted. In beam telescopes, this weak mode can be avoided e.g. by fixing the space coordinates of two telescope planes in the alignment algorithm.

In general, weak modes can be excluded by introducing additional constraints such as the fixed detector planes in the example given above. Other, more sophisticated methods are available to make sure the overall translation or misalignment is minimal [Blo06b].



(a) ARGUS experiment (b) ATLAS experiment

Figure 5.4: Comparison of event complexity in today's high-energy particle physics experiments. (a) shows an event from an e^+e^- collision in the ARGUS experiment at DESY with two B mesons, one of them being a candidate for $B^- \to K^- 4\pi^\pm$ [A+95]. (b) shows a simulation of a possible Higgs decay for LHC pp collisions in the ATLAS detector, $H \to ZZ^* \to \mu^+\mu^-e^+e^-$ ($m_H = 130\,\mathrm{GeV}$) [ATL97].

5.3 Global and Local Methods of Track Fitting

The last step of the identification of particle trajectories is the actual track fitting procedure. In high-occupancy experiments like the LHC detectors tracking becomes more and more difficult and the success of the physics reconstruction heavily relies on efficient track finding algorithms. These tightened requirements in today's experiments are demonstrated in Figure 5.4, comparing an event displays from the ARGUS experiment and a simulated event for the ATLAS detector.

The output of the track finding process consists of a set of parameters for a function describing the path of the particle through the detector. Usually a track is described by five parameters, depending on the experimental setup. Without a magnetic field in the tracking detector as often found in beam telescope setups or forward-only geometry detectors such as the VErtex LOcator (VELO) of LHCb [Aff08] the track can be described by:

- the transverse coordinates x_0, y_0 at a reference plane perpendicular to the beam,
- the track slopes $\tan\theta_x$ and $\tan\theta_y$ in the xz and yz planes, and
- the signed inverse particle momentum Q/p, where Q is the charge,

assuming coordinates similar to the CMS coordinate system described in Section 4.2. The reference plane is usually placed at the nominal interaction point.

However, these coordinates are not suited for experiments with cylindrical geometries where the magnetic field comprises the whole tracking volume like in the CMS detector. In a homogeneous field from a solenoid magnet particle tracks can be described best as a helix parallel to the magnetic field. Usually the so-called Point of Closest Approach (PCA), the minimal distance of the particle to the z-axis, is used as reference for the track. Suitable parameters in polar coordinates for this type of track are:

- the transverse impact parameter d_{xy} and the coordinate z_0 at the PCA,
- the azimuthal angle ϕ_0 of the trajectory at the PCA,
- the track slope $\cot\theta$, with the polar angle θ, and
- the signed inverse particle momentum Q/p.

For the performance of a tracking detector some quantities are of special interest in order to evaluate the performance and efficiency. Especially the *tracking efficiency* and the *track fake rate* are good measures of the performance of tracking algorithms. The efficiency is calculated by comparing the result from the track fitting procedure against a set of reference tracks, usually provided by a Monte Carlo simulation.

The tracking efficiency ϵ_{reco} is defined as the fraction of reconstructed tracks which can be identified with a true track from the simulation $N_{\text{true}}^{\text{reco}}$ divided by the total number of true tracks N_{true}:

$$\epsilon_{\text{reco}} = \frac{N_{\text{true}}^{\text{reco}}}{N_{\text{true}}}. \tag{5.4}$$

Therefore the tracking efficiency gives the fraction of particles whose trajectories could be correctly reconstructed.

The track fake rate ϵ_{ghost} or ghost rate is defined as the number of reconstructed tracks not matched to any true tracks, normalized by the total number of reconstructed tracks:

$$\epsilon_{\text{ghost}} = \frac{N_{\text{non-true}}^{\text{reco}}}{N_{\text{reco}}}. \tag{5.5}$$

The tracks produced by the pattern recognition algorithm that do not belong to any true particle in the simulation are reflected in this rate.

Parameter resolution is the achieved resolution of a given track parameter. To determine the parameter resolution either the residuals or the *pulls* are calculated for a reconstructed parameter X_i and the true parameter value. A special case of residuals was already presented for the alignment process in Section 5.2. In general, a residual can be calculated for any parameter used in the track fitting procedure and is given by:

$$R\left(X_i\right) = X_i^{\text{reco}} - X_i^{\text{true}}. \tag{5.6}$$

One distinguishes between two different types of these residuals:

Biased residuals are calculated using a track fit with all measurements available. The hit information from all sensor planes are used to perform a track fit and the difference between the measured and the fitted value for the parameter under consideration is calculated.

Unbiased residuals do not depend on the measurement from the plane under consideration. The hit information from this sensor plane is excluded from the track fit and does therefore not contribute to the global fit.

The parameter resolution is then the width of the residual distribution. The *pull* of a given parameter is the normalized parameter residual, taking into account its covariance matrix C:

$$P\left(X_i\right) = \frac{X_i^{\text{reco}} - X_i^{\text{true}}}{\sqrt{C_{ii}}}. \tag{5.7}$$

The methods in track finding can roughly be divided into two classes. Global methods use all information available to perform a global fit of all parameters. This approach does not depend on any predictions or seeds and the result is independent from the sequence the data is processed in. The drawback of these methods is the required computing effort. Treating all measurements with a large number of parameters at the same time means large matrix transformations which can be inefficient in terms of speed.

Local methods or *track following* methods take a different approach. Starting from a track seed the fit is calculated stepwise. A seed is an initial and minimal set of hits giving a first track candidate from which the track following procedure can start extrapolating. The trajectory model has to provide a so-called method of transport to allow extrapolations of the track to the next measurement (in tracking detectors a hit in the next sensor plane).

For beam tests with telescope detectors usually a simple global fit with the least squares estimation is sufficient. However, for complex tracking detectors such as the CMS often multi-step track fitting algorithms are applied. The tracking procedure of the CMS tracker is described in [K$^+$10]. It uses the pixel detector for an initial track fit and vertex reconstruction. The standard track reconstruction is then performed by the Combinatorial Track Finder (CTF) (see below) using triplet track seeds consisting of two strip tracker hits and one vertex reconstructed by the pixel detector.

In the following sections examples from both classes are briefly discussed. Fore more elaborate examples and descriptions of a number of other approaches see e.g. [Man04].

5.3.1 Least Squares Estimation

The least squares estimation is a comparatively simple global track fitting algorithm. The χ^2 value for each parameter is calculated and the fit is performed by minimizing the function:

$$\chi^2 = \sum \left(\frac{m_i - f_{\boldsymbol{\lambda}}(i)}{\sigma_i} \right)^2, \tag{5.8}$$

where $\{m_i\} = \boldsymbol{m}$ is the set of measurements with errors $\{\sigma_i\}$, and f the trajectory of the particle with the parameter set $\boldsymbol{\lambda}$. This expression can be rewritten as described in [Man04] and leads, for a linear problem, to the following solution for the parameters:

$$\boldsymbol{\lambda} = \left(F^T V^{-1} F\right)^{-1} F^T V^{-1} \boldsymbol{m}, \tag{5.9}$$

where $F = \frac{\partial f}{\partial \boldsymbol{\lambda}}$ is the derivative matrix of f and $V = \operatorname{diag}\left(\sigma_i^2\right)$ the error matrix of the measurements. The covariance matrix which is needed to calculate the parameter pulls, is then given by:

$$C_{\lambda} = \operatorname{cov}(\boldsymbol{\lambda}) = \left(F^T V^{-1} F\right)^{-1}. \tag{5.10}$$

This approach can be inefficient if the hit combination belonging to a track is not known beforehand. When having several hits per plane it is necessary to minimize the χ^2 value for every possible combination from scratch and then compare the results in terms of accuracy. Local track following methods go a different way and can avoid this problem.

5.3.2 Kalman Filter

The Kalman filter technique is a local method for track fitting. It originally was developed for real-time tracking, when the different measurements are taken at different times. The Kalman filter starts with a track seed, a first track candidate and proceeds progressively from one measurement to the next. The algorithm consists of two main steps that are repeated for every new measurement added to the fit:

- The *prediction* step extrapolates the current trajectory to find candidates for the next hit. This is very efficient since noise hits or hits belonging to other tracks can be discarded immediately. This is done by the *prediction equations*:

$$x_k^{k-1} = P_k x_{k-1} \qquad C_k^{k-1} = P_k C_{k-1} P_k^T + Q_k, \tag{5.11}$$

  with x_k being the state vector after the inclusion of k measurements m_k. The two indices $^{k-1}_{k}$ indicate the prediction from the current measurement $k-1$ to the next one k. P is the propagation or transport matrix of the track parameters and with Q_k random track perturbations can be introduced.

- The *filter* step updates the state vector. Unlike in global methods only a matrix with the dimensions of the local measurement has to be inverted, leading to a fast algorithm. This step can be used to take track perturbations like multiple scattering into account (see next section). The *filter equations* update the system state vector:

$$x_k = x_k^{k-1} + K_k \left(m_k - H_k x_k^{k-1} \right) \qquad C_k = \left(1 - K_k H_k\right) C_k^{k-1}, \tag{5.12}$$

  with H being the projection matrix of the prediction onto the measurement and $K = K\left(C, H, V\right)$ the gain matrix.

Additionally *smoothing* can be applied. The operation works recursively in the opposite direction to the filtering and is able to refine the track using the results from every single filter step. The CMS track fitting method CTF is an implementation of a Kalman filter, more details on the realization and performance of CTF can be found in [S$^+$06].

A comparison of the two approaches to the final track fit is shown in Figure 5.5. While the least square estimation performs a global track fit with all measurements resulting in the full particle trajectory with its errors, the Kalman filter technique proceeds from one measurement to the next, starting from a given track seed. The error on the track fit is reduced in each step when adding new information to the fit. The final fitting result of the Kalman filter is equivalent to the least squares method.

5.3.3 Multiple Scattering

Depending on the experimental setup *multiple scattering* might be a non-negligible effect that has to be taken care of in the track fitting procedure. Multiple scattering describes the small-angle deflection processes a charged particle undergoes when traversing material. The deflection originates mostly from Coulomb scattering on nuclei and is described by the *Molière theory*.

The mean scattering angle of a particle crossing a material with thickness x/X_0 in units of radiation length can be determined from a fit to the Molière distribution [B$^+$12b]:

$$\Delta\theta_0 = \frac{13.6\,\mathrm{MeV}}{\beta c p} z \sqrt{x/X_0} \left[1 + 0.038 \ln\left(x/X_0\right)\right], \tag{5.13}$$

where βc, p, and z are velocity, momentum and charge number of the particle. Figure 5.6 shows the schematics of multiple scattering in one plane. According to the theory, scattering through different materials should be regarded using a mean x/X_0 value for all combined scatterers instead of adding up the single contributions. An elaborate description of multiple Coulomb scattering can be found in e.g. [B$^+$12b].

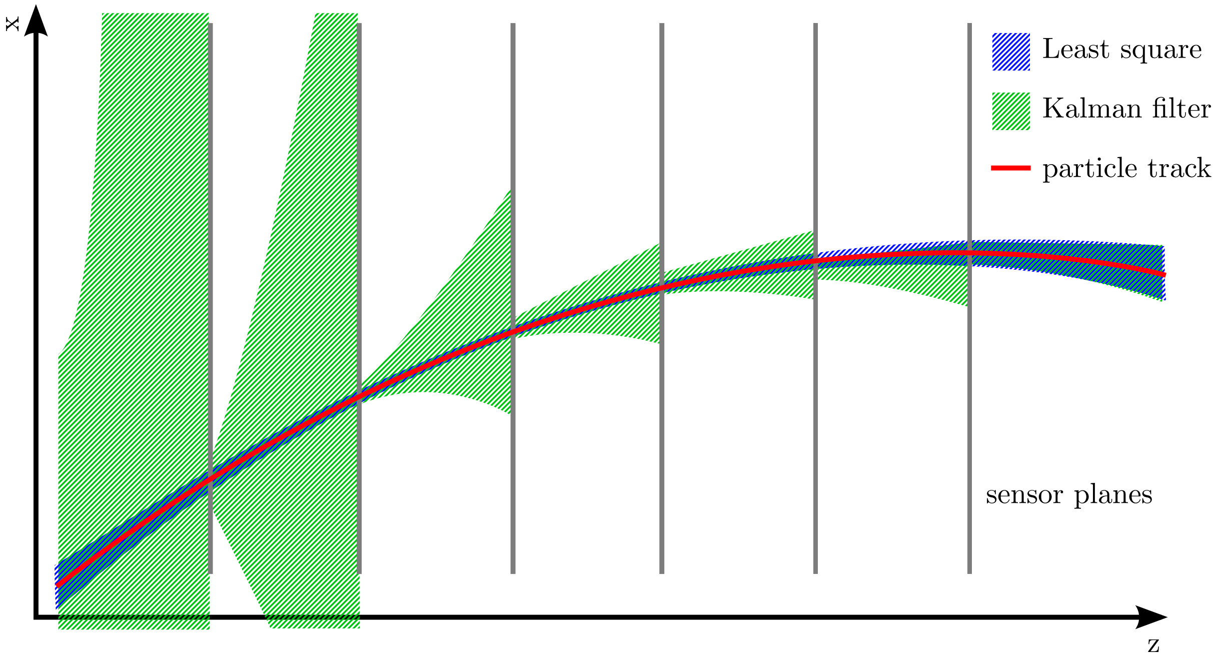


Figure 5.5: Track fit approaches to the particle track for the least square algorithm and the Kalman filter algorithm. The fit errors are shown as colored bands. While the least square method does the fitting in one optimization step with the resulting errors, the Kalman method refines its fit with every new measurement (hit in sensor plane). Both methods results in the same fitted particle track.

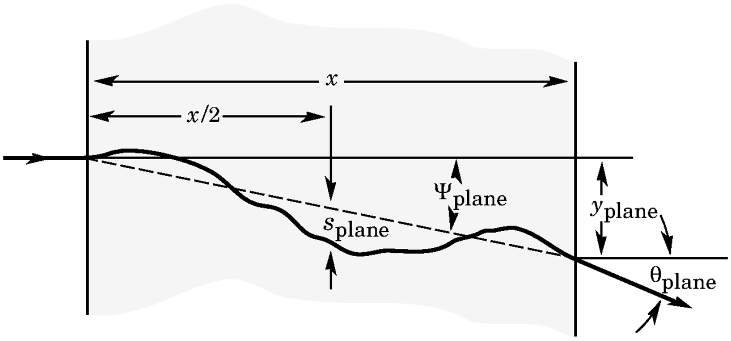


Figure 5.6: Sketch of multiple scattering describing the different quantities used for the calculation of the total Coulomb scattering angle θ_{plane} through a material with the thickness x [B$^+$12b].

Analytical Track Fitting

Analytical track finding is one approach for the least square estimation taking multiple scattering into account. The algorithm has been developed for beam telescopes and is implemented in the EUTelescope framework as described in [ZN07]. Starting from the initial approach described here the algorithm has been optimized and refined [Zar10] in order to be more efficient.

The χ^2 contribution can be calculated separately for all N detector plane. The expression consists of two terms, the first describing the normal uncertainty of the measurement, just as in the normal least square estimation (cf. Section 5.3.1):

$$\Delta\chi_i^2 = \left(\frac{m_i - f_{\boldsymbol{\lambda}}(i)}{\sigma_i}\right)^2 + \left(\frac{\theta_i - \theta_{i-1}}{\Delta\theta_0}\right)^2 \Bigg|_{i\neq 1,N} , \tag{5.14}$$

where θ_i is the track slope in detector plane i. The mean scattering angle $\Delta\theta_0$ is determined from Equation 5.13. The second term addresses the multiple scattering and is missing for plane 1 and N as it cannot be calculated due to the missing track slope parameters.

Depending on the energy of the particle considered and the material budget of the tracking detector in use, multiple scattering might not play a significant role in the track fitting procedure and could be neglected.

What we observe is not nature itself,
but nature exposed to our method of questioning.
Werner Heisenberg

6. High-rate Beam Test

Beam tests are used to perform reference measurements with detectors in precisely defined conditions. These measurements can be utilized to determine the performance of the detector and to get to know its operational behavior. Depending on the objectives beam test can be conducted in different experimental setups and environments, e.g. high-precision resolution or high-rate beam tests. The detectors are placed in a particle beam with known energy, position, and intensity and equipped with the needed infrastructure such as trigger logic or cooling devices.

Usually so-called *telescopes* are used in beam tests which consist of several consecutive detector planes. By this, beam particles traversing the detector planes can be tracked and the track information can be used to investigate the detector performance. Two types of telescopes can be distinguished. In the first all planes consist of the same type of detector and tracking and measurement is both done with the detector under investigation. The other type places the so-called Device Under Test (DUT), the detector to be characterized, between planes of well-known and understood detectors. This allows an independent measurement of the DUT performance.

A series of the high-rate beam tests has been scheduled to qualify the new Read Out Chip (ROC) *PSI46dig* for operation in the CMS pixel detector under the more difficult conditions at an instantaneous luminosity exceeding $\mathcal{L} = 2 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. Two time slots at the *H4IRRAD* beam area have been allocated in the Super Proton Synchrotron (SPS) schedule. The first beam test has been carried out between July 30 and August 2, 2012. The second test will be performed between October 24 and October 28, 2012. Probably some more beam tests in high-rate environment will be needed, especially since the new Token Bit Manager (TBM) transmitting the digital data format is not ready for submission yet but must also be tested. Those cannot be carried out at CERN since all accelerators will be shut down for the Long Shutdown 1 (LS1) starting March 2013.

The hardware setup for the beam tests at the *H4IRRAD* beam area as well as the test environment are presented in the following chapter. After motivating the beam test and the measurement program, the different telescope geometries and the mechanical support structure are presented. After that, the ROC programming and Data Acquisition (DAQ) periphery as well as the trigger logic are described. In Section 6.4 results from measurements of the beam condition are presented. In the following a coordinate system similar to the CMS system is assumed. The x and y axes are oriented horizontally and vertically, respectively; the beam points along the z axis. A general overview of the *H4IRRAD* test beam area and its characteristics can be found in Section 3.3.

6.1 Measuring the Performance of the *PSI46dig* ROC

Several changes to the ROC of the CMS pixel detector have been made for the Phase I Upgrade as described in Section 4.5. Since one of the main changes of the *PSI46* ROC affects the behavior of the chip at high particle rates and detector occupancies the high-rate beam test focuses on efficiency measurements in a high-rate particle beam. The effects of the extended and added readout buffers as well as the digital readout have to be validated and the chip layout has to be checked for possible flaws before submitting the final design for the series production of the detector upgrade. This can be done by exposing the chip to a particle flux similar to the one expected in the LHC environment at double design luminosity. The high-rate proton beam is provided by the SPS accelerator and rates up to 250 MHz can be reached (cf. Section 7.5.1). These rates translate to about 6–7 particle tracks per event through a CMS single ROC sensor and are well-suited for measuring the high-rate behavior of the ROC. In a later beam test not only the new ROC but also the first submission of the redesigned TBM will be tested to be able to operate the digital *PSI46dig* ROCs and to correctly transmit the data to the Front End Drivers (FEDs).

Due to the new readout scheme and the changed data format not only the new ROC but also new DAQ hardware and software had to be designed and prepared. The currently available test board and its Field Programmable Gate Array (FPGA) firmware has been developed for the *PSI46v2* with the analog level encoded readout (see Section 7.2.2) and is not able to read digital ROC output without modification of the firmware. Therefore a new test board and firmware is under development at Rutherford Appleton Laboratory (RAL) which also serves as prototype for the future digital CMS pixel detector FED.

The beam tests are performed with two beam telescopes with different geometries which are described in the following section. Furthermore the measurement program covers variations in trigger latency, particle rates, and other operating conditions such as Pixel Unit Cell (PUC) thresholds.

6.2 The Beam Telescopes

The *CMS pixel telescopes* are beam telescopes with eight identical detector planes. Due to the special requirements of the beam test and the demands on the tracking detectors concerning fast readout and high-rate capability no other existing beam telescopes such as the EUDET telescopes could be used. This implied the construction of telescopes consisting only of CMS pixel ROCs. The telescopes are operated in DUT mode, where seven planes provide the tracking functionality and one of the middle planes is defined as DUT. Its efficiency can be calculated by matching the particle tracks to the measurements of the DUT.

The mechanical support structure of the telescope setup is shown in Figure 6.1. Since the complete setup has to be lowered down into the *H4IRRAD* beam area from the top by a crane, the telescope is mounted on a rack crate. The crate can be roughly aligned to the beam during the lowering but the exact position of the telescope planes has to be adjusted during the beam test. To allow horizontal and vertical movement in the closed *H4IRRAD* area, the telescope in operation is mounted on two remotely controlled motor stages for translation in x and y direction. Temperature monitoring of the ROCs is performed with Pt100 temperature sensors glued the back of each ROC. A multimeter with LabView control is used for logging the temperature.

The readout is triggered by two scintillators mounted in front of the first and behind the last telescope plane. Both scintillators are cubes with an edge length of 2 mm and are equipped with two photomultiplier tubes (PMTs) each. This setup allows several levels of coincidence triggering as described in Section 6.3. Furthermore the readout electronic is able to provide random internal trigger signals with a rate of about 15 kHz which do not need additional trigger hardware. This functionality is mostly used in laboratory measurements, e.g. for ROC calibration.

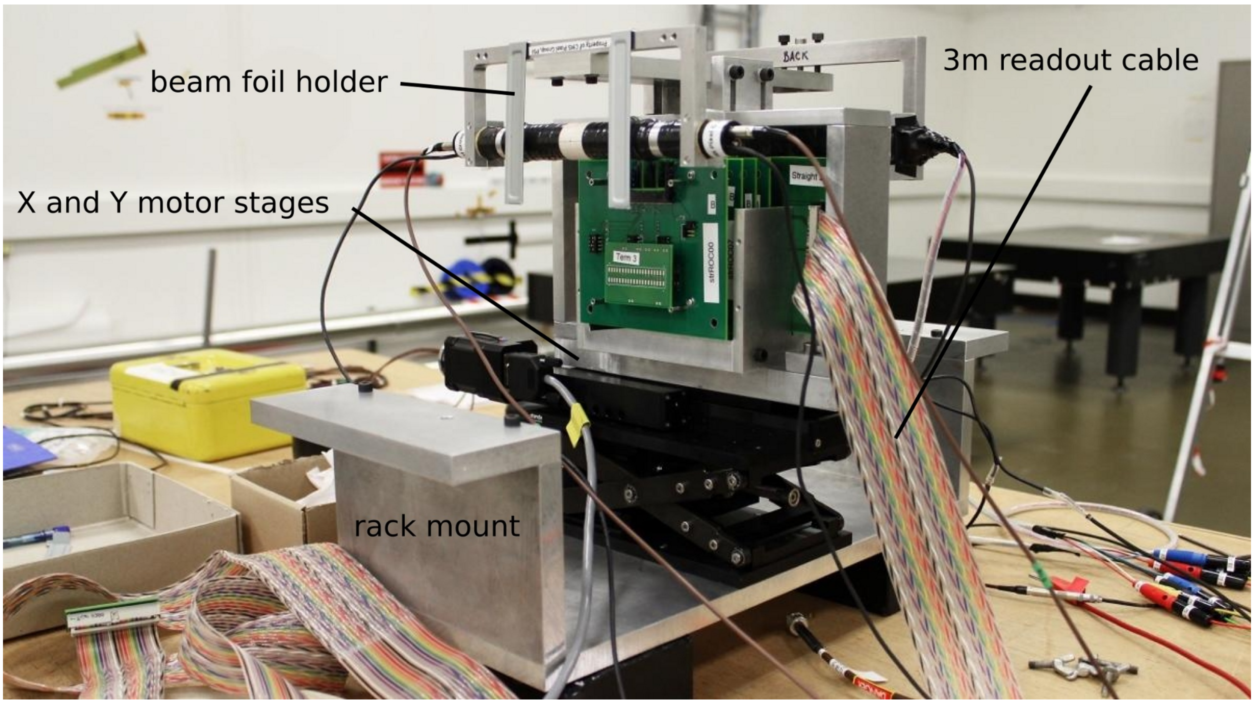


Figure 6.1: Mechanical support for the CMS pixel telescope in the laboratory. The PCB telescope planes, the 3 m readout cable, and the scintillators are already mounted. The *xy* tables allow a remote steering of telescope movement of the telescope for beam spot scanning.

6.2.1 Geometries

In order to be able to conduct various measurements under different conditions two different telescope geometries have been designed as shown in Figure 6.2. The first version holds all sensor planes perpendicular to the particle beam. This minimizes the influence of charge sharing and simplifies the evaluation due to the rather simple layout. The sensor planes of the second telescope version are tilted in *xy* and *yz* according to the Lorentz angle of 25° in the CMS tracking detector [CMS10] to simulate the real operation conditions in CMS concerning charge sharing and inclined particle incidence.

The tilted geometry is of special importance since the beam incident on perpendicular planes mostly produces single pixel clusters. The CMS pixel detector ROC is designed for larger clusters and therefore has a ratio of data buffers (with pixel hits) to timestamp buffers of more than 1:3 (cf. Section 4.5.3). Generating mostly one-pixel clusters results in an overflow of the timestamp buffers without using the full data buffer capacity. Tilting the detector planes and therefore generating larger clusters fills both the data and timestamp buffers in a ratio which is comparable to conditions during operation of the pixel detector in CMS with magnetic field.

Both telescopes consist of eight consecutive telescope planes equipped with *PSI46dig* ROCs with single chip sensors. The ROCs are glued and wire-bonded on so-called plug-in boards which are shown in Figure 6.3 (a). The plug-in boards allow an easy mounting and replacement of ROCs without disassembling the telescopes. The telescope planes are made of the PCBs shown in Figure 6.3 (b). Each of these planes holds a ROC plug-in board and establishes the connection between the readout circuit and the ROC. The Inter-Integrated Circuit (I^2C) protocol, a serial communication bus [Phi00], is used to address and program the *PSI46* ROCs independently over a common bus interface. A switch on each of the telescope PCBs defines the I^2C address of the attached ROC. The telescope PCBs are plugged together using connectors which are placed on

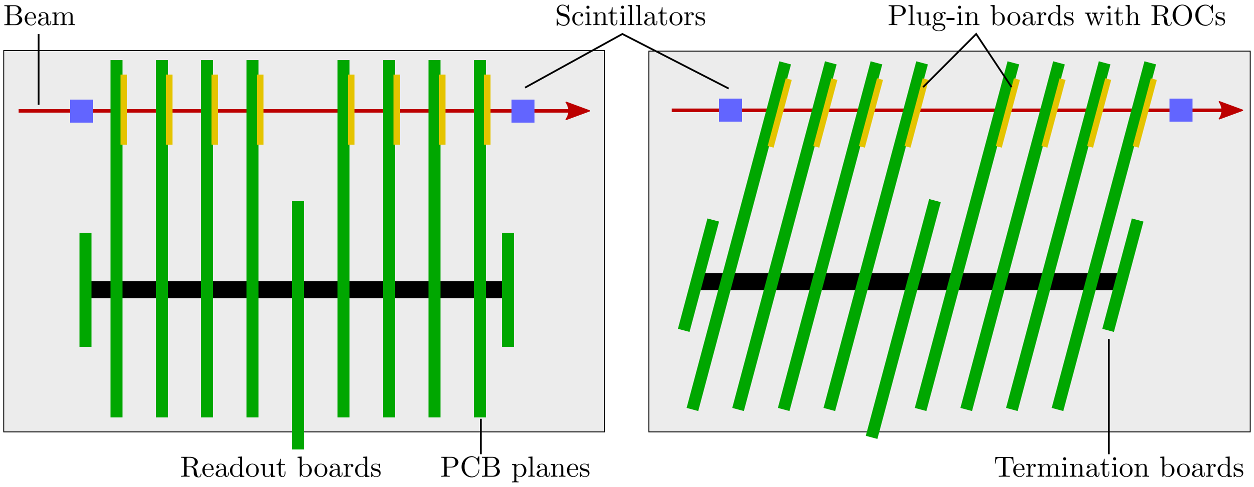
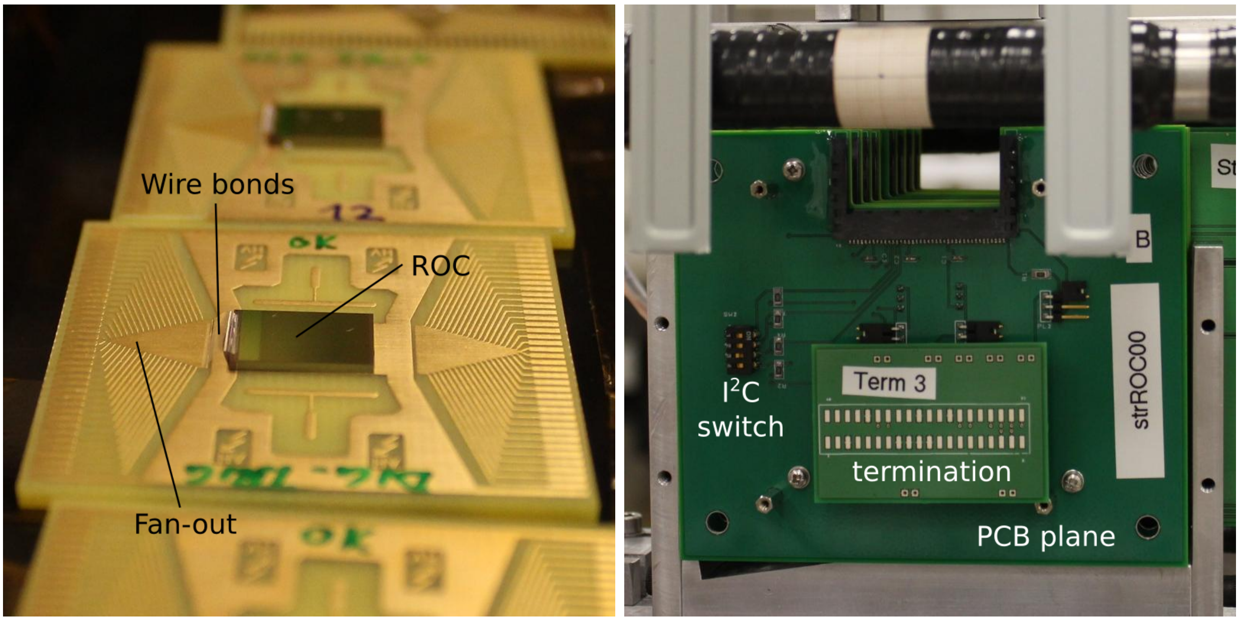


Figure 6.2: Sketch of the two different beam telescope geometries prepared for the high-rate beam test. In the first version all eight planes are perpendicular to the beam (left), while the planes of the second geometry are tilted in two directions with respect to the beam (only one shown) to generate two- and three-pixel clusters (right).



(a) ROC plug-in boards

(b) PCB plane and termination board

Figure 6.3: Components of the CMS pixel beam telescopes. (a) The ROCs are glued and bonded onto plug-in cards which provide a fan-out for the data and control lines. Temperature sensors are mounted on the backside. (b) The ROC PCBs form the telescope planes and hold the plugin-boards. The I^{2}C address for each ROC is set by a switch on the PCB. The termination board provides termination resistors and returns the token.

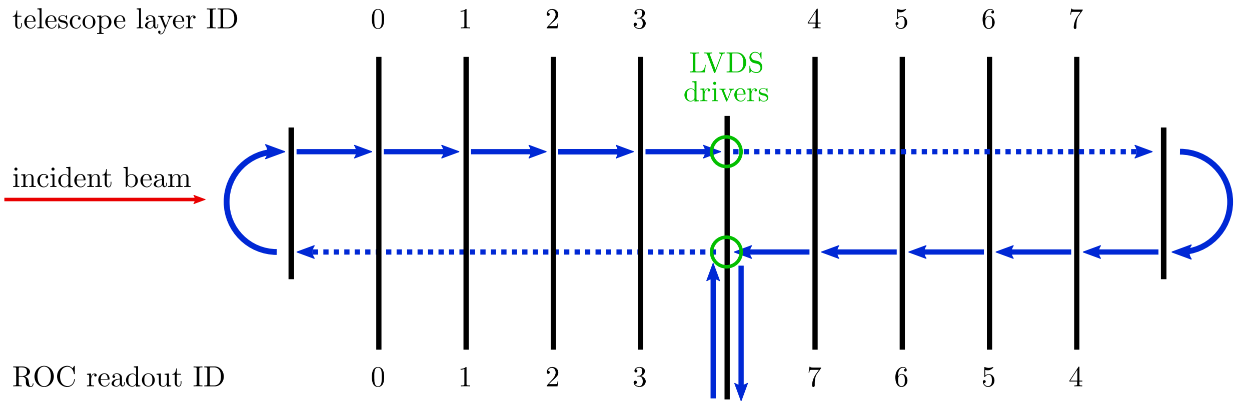


Figure 6.4: Readout token path in the beam telescope. The token is first passed to the termination boards and then routed back towards the readout PCB. The circles mark the LVDS drivers used to amplify the token signal.

both sides of the boards. This allows easy stacking and simplifies the construction of a telescope with almost any number of planes. The only limitation is the readout infrastructure and the token signal degradation between the telescope planes (see below). Spacers ensure an equal distance of 13.6 mm between the PCB planes and double this distance between the planes closest to the readout board in the middle. For the tilted geometry the connectors between the PCBs are displaced in x and y direction which generates the tilt.

Readout token and data are transmitted as Low Voltage Differential Signal (LVDS) and therefore need appropriate drivers to distribute the signal to all attached telescope planes without degradation. This is taken care of by the readout PCB placed in the middle of the telescope. It holds the LVDS drivers for both the readout token distribution and the data transmission to the test boards. In an earlier version of the telescope design the readout board was placed at the end which caused problems with the token distribution. Up to seven ROCs could be addressed correctly while the last one did not respond due to the heavily distorted LVDS signal. In the current design the token is first passed from the readout board to the termination boards on both ends of the telescope and then from ROC to ROC backwards to the readout board as shown in Figure 6.4. This doubles the distance between the two telescope layers adjacent to the readout board but ensures a correct signal transmission.

Eight is a rather large number of planes for a beam telescope, usually already three planes plus a DUT are sufficient to perform stable particle tracking. The decision for the CMS pixel telescopes was driven by several demands. First, the telescope and the DUT consist of the same detector and therefore are exposed to the same inefficiencies. By tracking particles with seven planes the overall tracking efficiency can be increased e.g. by accepting 6-of-7 or 5-of-7 patterns instead of requiring one hit in every sensor plane to form a particle track. Furthermore the safety margin to ensure stable tracking has to be rather large since access to the beam area is very limited. In case a ROC fails the tracking capabilities of the whole telescope are not affected.

6.2.2 The Telescope Readout Periphery

A 3 m long ribbon cable connects the beam telescope to the DAQ electronics. The long cable allows the placement of the readout periphery outside the high radiation area which has been identified by a simulation performed with FLUKA [B$^+$07a, F$^+$05] as shown in Figure 6.5. FLUKA is a tool for simulation of particle transport and is often used for shielding or target design calculations. Stable

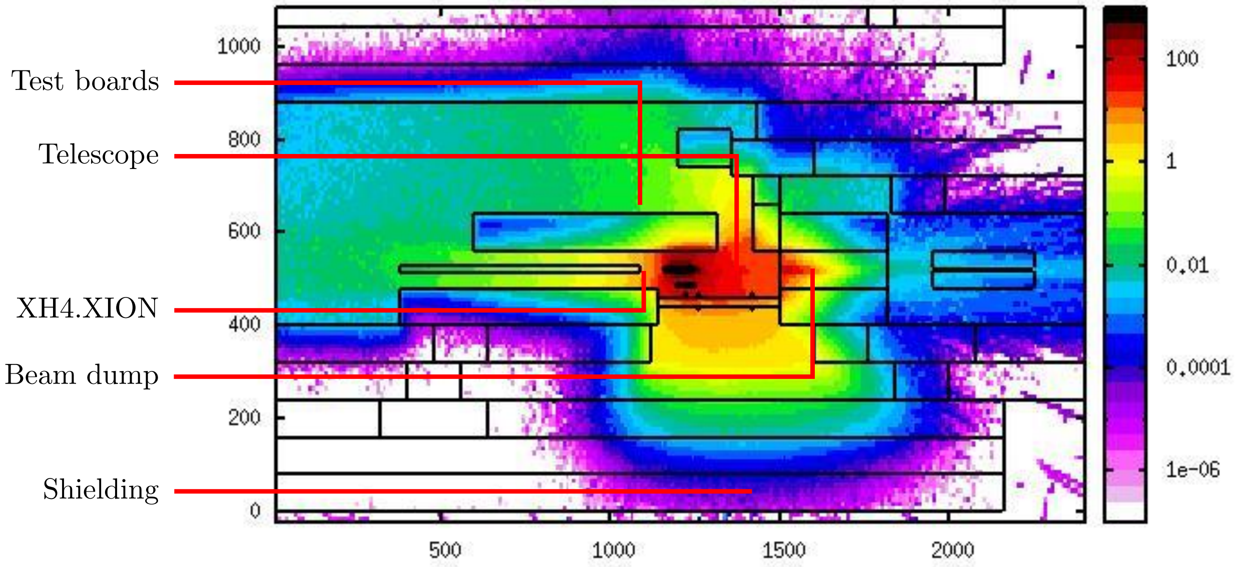


Figure 6.5: FLUKA simulation of the *H4IRRAD* beam area without a target installed. The positions of test boards and the beam telescope are marked. The dark shaded area represents the activated copper target which is only lowered and not completely removed from the area. The unit of length is cm. Modified from [Qin12].

operation of the electronics under beam test conditions in the *H4IRRAD* radiation environment has been assured by a test carried out before the actual beam test. One of the test boards together with a single ROC module has been placed at the position marked in Figure 6.5 and operated for several weeks without encountering severe problems. Only the test board stopped the DAQ a few times and had to be restarted. After each restart the DAQ worked properly again.

Due to the new digital readout of the detectors a completely new DAQ system had to be designed and tested for the high-rate beam tests. The previously used Altera test board has been developed for laboratory operation of the *PSI46v2* ROC with analog level encoded readout and is not compatible with the digital data format delivered by the *PSI46dig* ROCs. Therefore a new DAQ test board for *PSI46dig* ROCs based on a Xilinx SP605 development board [Xil12] is used for the beam tests. It features a powerful Spartan-6 FPGA and also serves as first prototype for a future CMS pixel detector FED developed by RAL.

The launch of this new test board is held in two phases. In the July beam test the firmware of the Xilinx test board has only been able to deserialize the data stream from the *PSI46dig* ROCs and the Altera test board performed the signal processing, calibration, and the actual readout. A bridge board established the connection between the two test boards as well as between the telescope and the so-called *hybrid test board setup* shown in Figure 6.6. For the October beam test the Xilinx firmware will be fully operational and capable of reading out, calibrating, and programming the ROCs. The Xilinx board will then be used as standalone readout solution without the Altera test board.

The hybrid test board setup is connected to the DAQ PCs outside the beam area by cables with a length of 50 m. While the Altera test board relies on a USB 1.0 connection via USB Ethernet Extenders, the Xilinx SP605 board provides an optical Gigabit Ethernet link with IPBus protocol [M^{+}11] via fibers. Both test board FPGAs can be reprogrammed with new firmware versions during operation using the built-in Joint Test Action Group (JTAG) USB interfaces. The software controlling the DAQ runs in an automated per-spill batch mode that checks for the current SPS

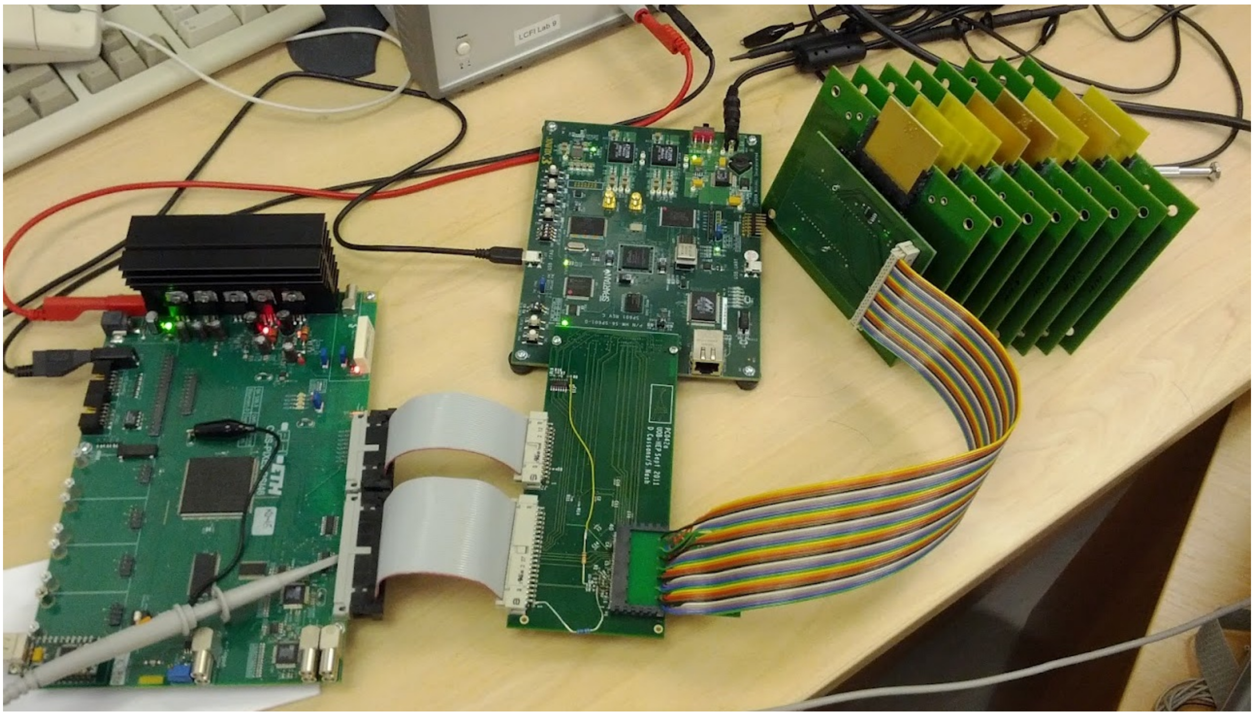


Figure 6.6: Hybrid test board setup for the July beam test with an Altera test board (left), a Xilinx development board (center, top), and the bridge board (center, bottom). The telescope (right) is an outdated revision of the tilted geometry version used for test purposes only.

spill number provided by the Trigger Server (cf. next section) and invokes a new run whenever the spill counter is increased. For the July beam test an adapted version of the *psi46takeData* software has been used while during the October beam test the DAQ will be controlled by a specifically developed software framework for the Xilinx test board readout. The acquired data is stored on the DAQ PC and mirrored to a remote server within the CERN network where the offline Data Quality Monitoring (DQM) scripts are executed (cf. Section 7.3.7). Two redundant DAQ PCs are ready for operation; both are equipped with the *psi46expert* and *psi46takeData* software packages for the DAQ as well as the FPGA programming suites *Quartus II* (for Altera Cyclone) and *ISE* (for Xilinx Spartan).

The hybrid test board setup created several problems during the July beam test. The amount of data recorded has been greatly reduced by the readout speed of the Altera test board as well as the data format which has been used. The Altera board first stores all data from a DAQ run in the internal memory with a size of 64 MB. Afterwards the data is transferred to the DAQ PC using the USB 1.0 interface. With the available bandwidth of this connection the download of the run data takes about 8 min which means that only every 8th SPS spill can be recorded (cf. Section 6.3). A USB 2.0 module has been prepared and successfully tested with high bandwidth data transfer in the laboratory but could not be operated at that speed during the beam test, probably due to under powering of the deployed USB hub. Furthermore, the intermediate First In First Out (FIFO) buffer of the Altera board has a fixed size of 16 bit but the digital data stream comes in packages of four bits. This means that e.g. one ROC header sequence is stretched to `0x7000 0xf000 0x8000` instead of `0x7f8` (cf. Section 7.2.3) and effectively three fourth of the available memory are spent storing useless data. The additional zero bits have to be removed in the decoding of the raw data as described in Section 7.3.1.

The clocks of two test boards had to be synchronized in order not to introduce bit errors in the data stream which appeared without the synchronization. But even with synchronized clocks severe data quality issues could be found indicating that this issue has not been solved completely (cf. Section 7.5.2). Due to firmware problems concerning token signal delay settings and trigger signal routing only the internal random trigger of the Altera board could be used in the July beam test.

Several enhancements will be available in the October beam test with the Xilinx Spartan-6 firmware being ready for standalone operation. The high bandwidth of the Gigabit Ethernet will allow a fast readout of the recorded data. Therefore every SPS spill can be recorded and stored. Moreover the new interface allows on-the-fly event readout during the run which facilitates additional features such as online DQM. The usage of external trigger signals will be possible and the Xilinx firmware will be capable of a full emulation of the CMS trigger rules [Var02]. It is assumed that most of the bit errors encountered in the data stream originate from the hybrid test board setup and will not be present in a standalone Xilinx readout.

6.3 Spill Structure and Trigger Logic

The spill structure of the SPS beam has been analyzed using the telescope scintillators. The beam has a spill structure due to the SPS particle distribution scheme described in detail in Section 3.3. Beside the general information on the SPS Page-1 (cf. Figure 3.3) the accelerator provides several additional signals that can be used for the detection of spills. The Warning Warning Extraction (WWE) signal arrives 1000 ms before the extraction to the respective user starts. The next signal, Warning Extraction (WE), is provided 1 ms before the imminent beam and End of Extraction (EE) marks the end of the extraction period. These signals can be used by the trigger logic at the different experiments.

The CMS pixel telescope setup uses two scintillators for triggering. One of the 2 mm × 2 mm scintillators is placed in front of the first telescope plane, the other behind the last plane. With this setup it is possible to trigger on particles traversing all telescope planes and thereby leaving distinct patterns for tracking. Both scintillators are equipped with two PMTs, one on each side. This enables coincidence triggering within one scintillator which can be used to suppress noise. The PMTs are denominated as A0 and A1 for the scintillator in front of the first telescope plane, and B0, B1 for the scintillator behind the last plane as shown in Figure 6.7. The trigger signals are processed by the Trigger Logic Unit (TLU) in which different coincidence settings as well as signal delays can be configured. The following coincidence configurations have been used during the July beam test for various measurements:

$(\mathrm{A0} + \mathrm{A1})$	front scintillator only
$(\mathrm{B0} + \mathrm{B1})$	back scintillator only
$(\mathrm{A0} + \mathrm{A1}) + (\mathrm{B0} + \mathrm{B1})$	fourfold coincidence

Furthermore the TLU processes the SPS signals and transmits all collected information to the *Trigger Server* PC. The Trigger Server provides the current spill number to be fetched over Ethernet and logs the PMT signals for every spill. This allows an easy inspection of particular spills after the beam test as well as the calculation of the overall flux. The spill number is increased whenever the SPS WWE signal is detected. The DAQ software regularly checks the spill number provided by the Trigger Server and starts a new run once the number has been increased. This approach allows the separation of single spills into data taking runs as well as the assignment of the recorded data to particular spills and their properties. The SPS WWE and EE signals are used by the TLU to veto any PMT signal outside the actual spill to suppress noise.

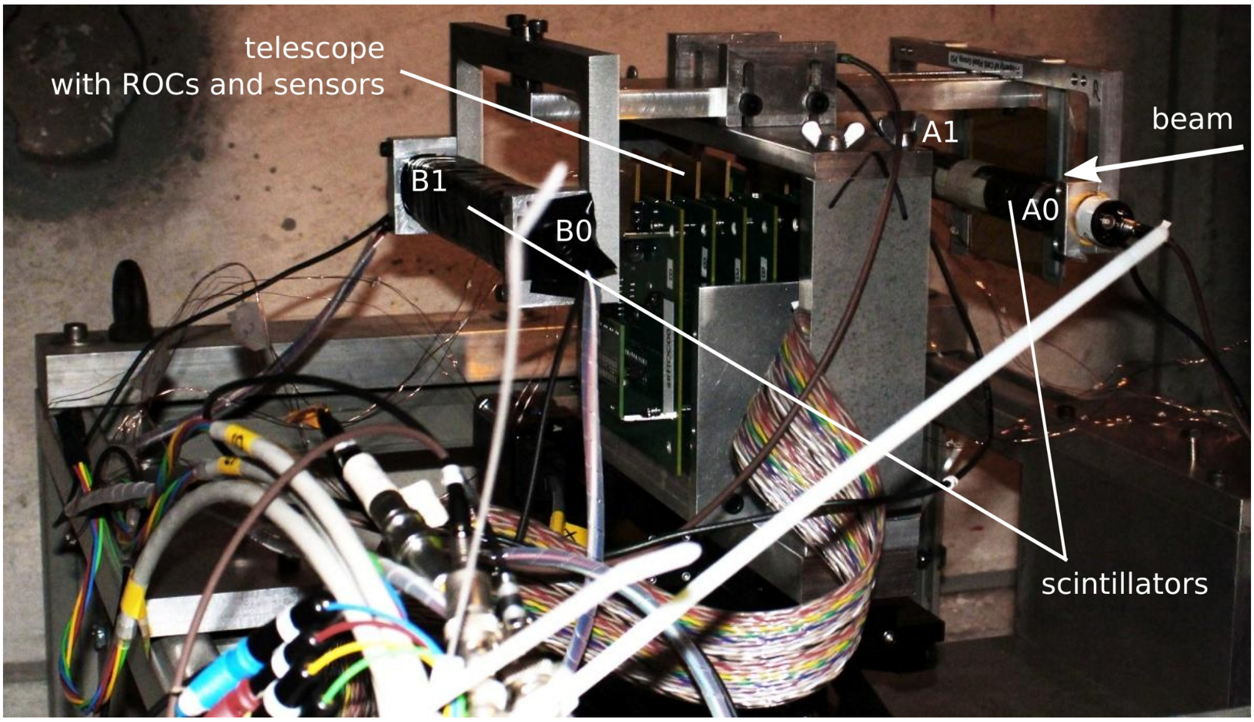


Figure 6.7: Fully equipped CMS pixel telescope installed in the test beam area. Shown is the mechanical support structure holding the telescope with the ROCs itself as well as the two trigger scintillators at both ends. The four PMTs are designated by the symbols A0,A1 for the front and B0,B1 for the back of the telescope. The beam is incident from the right side of the picture.

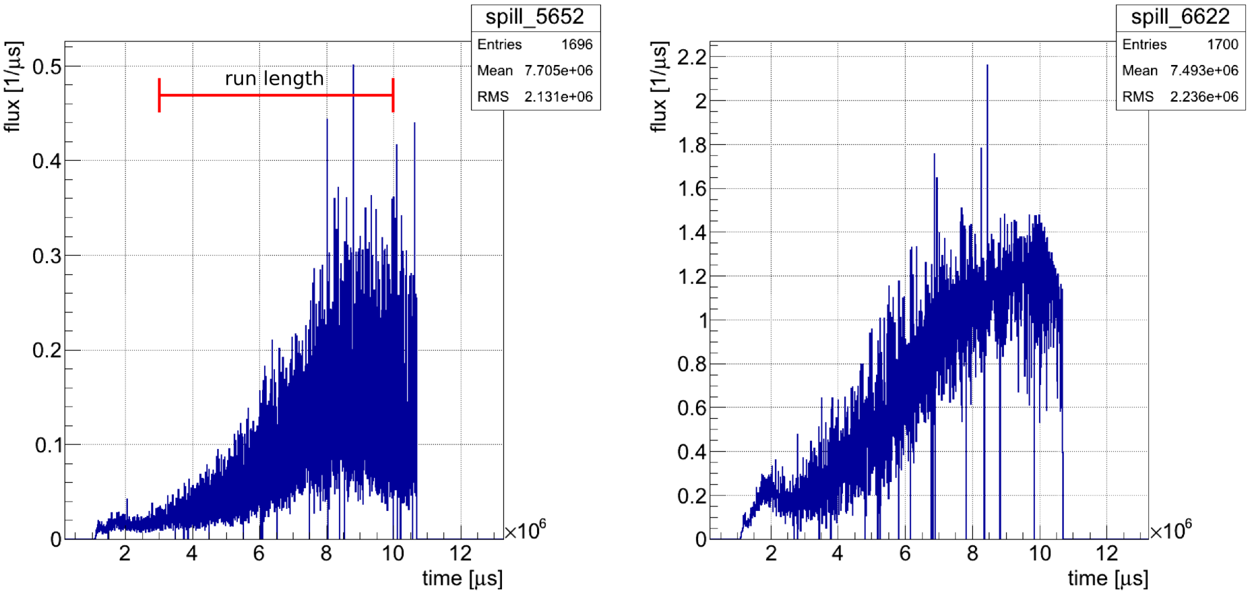


Figure 6.8: SPS beam particle flux for spills at different beam intensities as a function of time. The data has been recorded with the front scintillators and its two PMTs in coincidence. The total number of particles in spill 5652 is $N_{5652} \approx 1.99 \times 10^9$ protons and $N_{6622} \approx 2.41 \times 10^9$ protons in spill 6622. The approximate duration of one data taking run ($t_{\text{run}} = 7\,\text{s}$) is indicated in red.

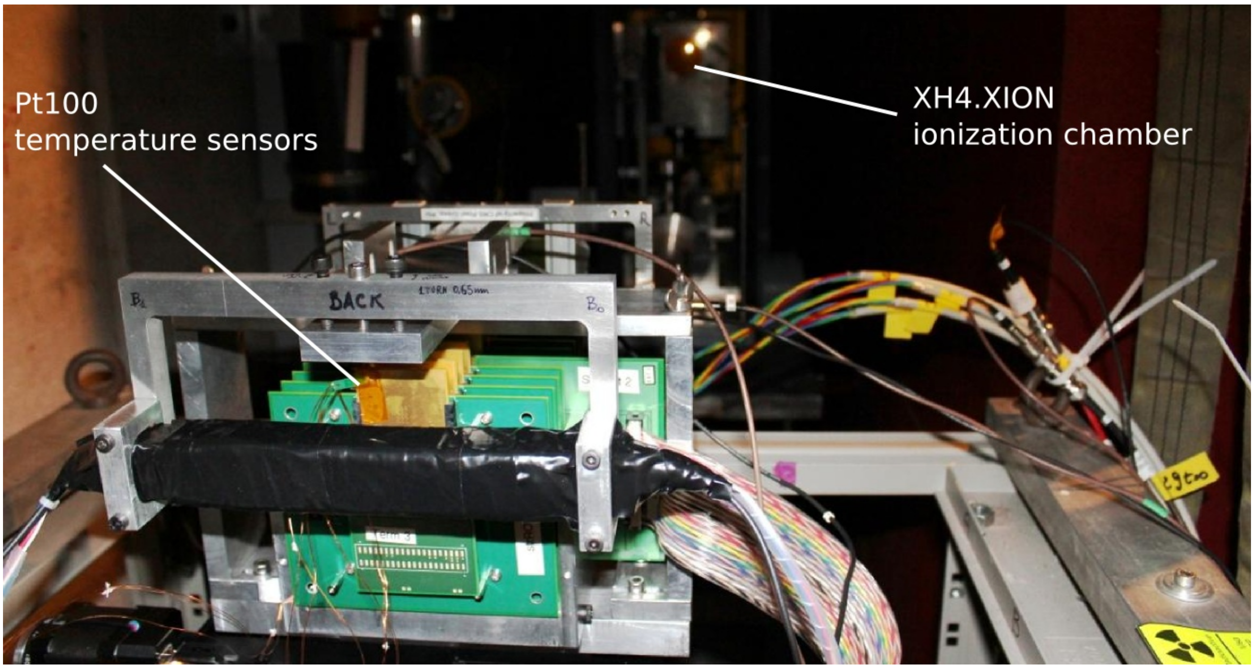


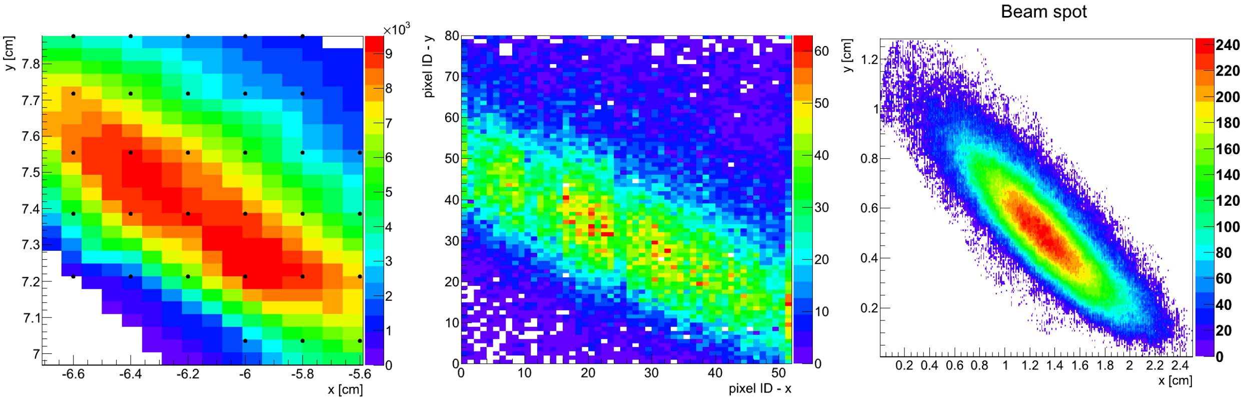
Figure 6.9: The CMS pixel telescope, looking towards the beam. The XH4.XION ionization chamber with its beryllium end window can be seen as well as the Pt100 temperature sensors on the back of the plug-in cards which hold the ROCs.

The particle flux coming from the SPS varies over the spill duration due to the increase of the magnetic field strength at the extraction magnet of the accelerator. Furthermore the shape changes for different beam intensities delivered to the users as shown in Figure 6.8. The shapes of the spill flux have been recorded using the front scintillator coincidence setting $(A0 + A1)$. The two measurements have been performed at different beam intensities which is the reason for the varying shape of the particle flux. The total number of particles is $N_{5652} \approx 1.99 \times 10^9$ protons in spill 5652 and $N_{6622} \approx 2.41 \times 10^9$ protons in spill 6622.

For the October beam test a new DAQ software will be used together with the completed Xilinx firmware. This will allow the usage of the external trigger signal provided by the TLU and a full emulation of the CMS trigger rules. Due to the higher bandwidth over IPBus Ethernet the new DAQ software will be able to record every SPS spill. Furthermore the spill number provided by the *Trigger Server* will directly be assigned to the corresponding run, which simplifies the correlation of trigger and detector data. Independently of the trigger configuration chosen a DAQ run is started by the Trigger Server detecting the WWE signal, and limited to $t_{\text{run}} = 7\,\text{s}$ which ensures that all data is taken during the spill as indicated in Figure 6.8.

6.4 Beam Monitoring and Beamspot Measurement

During the July beam test the intensity, position, and shape of the beam spot have been monitored using both the installed scintillators of the telescope and the beam line instrumentation via the CESAR and TIMBER interfaces (cf. Section 3.3). The telescope scintillators have mainly been used to evaluate the telescope position with respect to the beam. The SPS Page-1 monitor (cf. Figure 3.3) provides useful information about the current situation of the accelerator, the Super Cycle, and the T2 target. The beam line instrumentation consists of several detectors along the beam line. The most important ones for monitoring the *H4IRRAD* area are:



(a) Scintillator scan (b) Pixel sensor hit map (c) Gafchromic film [Mer12]

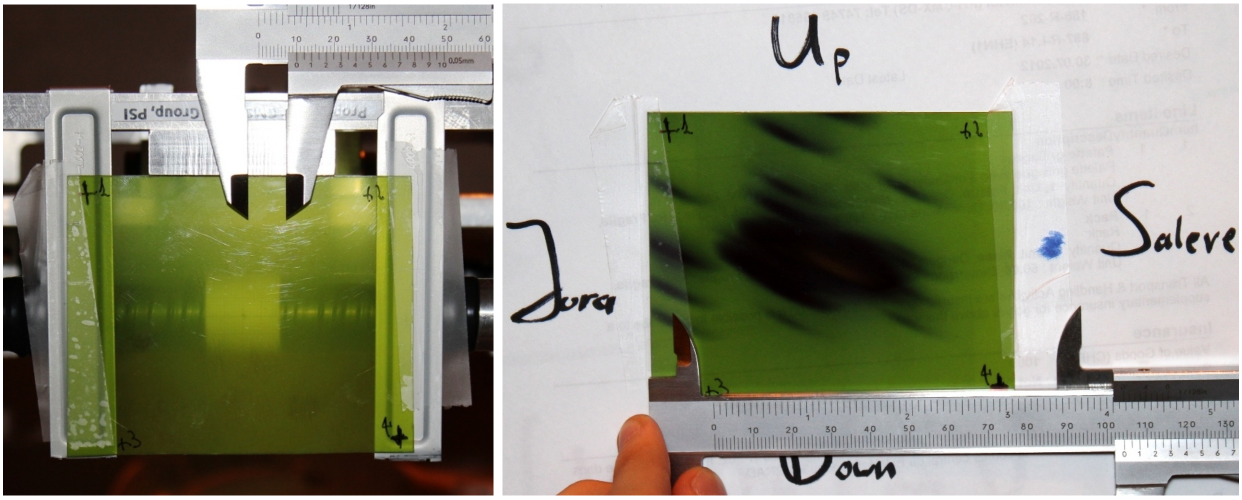
Figure 6.10: Beam spot of the SPS proton beam in *H4IRRAD*. (a) Normalized proton rate measured with the front scintillator in $(A0 + A1)$ coincidence configuration by moving the telescope with the xy motor stages. The dots mark the measuring points. (b) Measured with a telescope plane (CMS pixel sensor). The pixel pitch is $p_x = 150\,\mu\text{m}$ in x and $p_y = 100\,\mu\text{m}$ in y. (c) Beam spot obtained from the Gafchromic beam monitoring film.

XH4.XION: Ionization chamber in front of the telescope setup (distance $d \approx 1\,\text{m}$). The measurements from this detector are logged by TIMBER and have been used for the estimation of the total number of protons per spill arriving at the experimental setup. The values given by the detector have to be scaled using the calibration factor $m \approx 6300$ in order to obtain correct particle fluence (cf. Section 7.5.1).

XH4.XSCI: Scintillator detectors further upstream near the T2 target. These scintillators have been helpful in case of beam loss. A common reason for a sudden beam loss in *H4IRRAD* is a bending magnet (*XH4.BEND.315*) whose power supply trips frequently. If the scintillators still indicate incoming protons but not the ionization chamber the magnet has to be restored by the Central Control Center (CCC).

The position and shape of the beam spot has been measured using several methods. After the installation of the telescope setup in the *H4IRRAD* area, the beam position had to be found. This has been done using the front scintillator of the telescope and the xy motor stages to scan the reachable area for the highest scintillator counts with a step-size of 2 mm both in x and y direction. In total 65 scintillator counts at different positions have been recorded and each position has been measured during one full SPS spill. A map of the scan is shown in Figure 6.10 (a), the measuring points are marked with dots. The plotted proton rate has been normalized to the beam intensity of each spill recorded by the XION chamber. After the correct telescope position within the beam had been found the pixel sensors themselves could be used to record beam spot images. An exemplary hit map for the first telescope plane (ROC0) is shown in Figure 6.10 (b). The hit maps of all sensors have been plotted during the DQM process (cf. Section 7.3.7) and the position of the beam could immediately be verified for every spill. This is especially important since problems with bending magnets can easily lead to a movement of the beam spot and the telescope would be operated outside the particle beam.

A Gafchromic Film [Ash12] has been placed in the beam line in front of the telescope as shown in Figure 6.11 (a). Gafchromic film is a self-developing radiographic film used e.g. for beam imaging. After irradiation with the proton beam the beam spot areas turn black and their shape can be determined. Figure 6.11 (b) shows the irradiated film after the July beam test. The multiple beam



(a) Before irradiation

(b) After irradiation

Figure 6.11: Gafchromic film before and after the irradiation with the proton beam. (a) Front view of the telescope, non-irradiated film with one of the scintillators behind it. (b) Irradiated film. The words *Jura* and *Saleve* denote the corresponding orientation of the foil, *Saleve* is on the right side of the incident beam.

spots visible on the foil originate from the beam spot scanning procedure during which the telescope has been moved with the motor stages. The final position for data taking is marked by a large fully developed area and cannot be used to determine the beam spot. After the beam test the film has been scanned and optimized e.g. in terms of contrast, and one distinct beam spot has been extracted which is shown in Figure 6.10 (c). The image has been converted and a two-dimensional Gaussian has been fitted to the raw data to obtain a measure of the size of the beam spot (cf. Section 7.5.1).

The knowledge of the exact shape of the beam spot and its position and intensity has been used to calculate the fluence for the telescope layers as presented in Section 7.5.1.

It doesn't matter how beautiful your theory is.
It doesn't matter how smart you are.
If it doesn't agree with experiment, it's wrong.
Richard P. Feynman

7. CMS Pixel Telescope Data Analysis

In the following chapter the tools and algorithms used for the analysis of the beam test data are presented in detail, including some crucial parameters and configurations. Only the exact knowledge of these algorithms and procedures for every data processing step allows a correct and precise analysis of the beam test data. This in turn is a crucial prerequisite to be able to draw conclusions concerning the properties of the Device Under Test (DUT). Only if the behavior of the analysis software is well understood and mistakes at this level can be excluded, the actual behavior of the detector can be measured and its flaws revealed. These were the objectives when setting up the data analysis chain for the CMS pixel high-rate beam test described in Chapter 6.

After a brief description of the analysis framework chosen and its implementation in Section 7.1, the raw data format provided by the CMS pixel detector components is described (Section 7.2). Section 7.3 introduces the analysis chain and configuration of the tools used for the high-rate beam test. Finally, Monte Carlo studies for the analysis chain preparation are presented together with first results from the measurements taken at the beam test in July in Sections 7.4 and 7.5, respectively.

7.1 The EUTelescope and ILCSoft Frameworks

For the Data Quality Monitoring (DQM) and the data analysis of the beam test the already existing EUTelescope framework has been used instead of writing a new analysis chain for this particular application. This ensures the reduction of the overall workload, makes the algorithms less error-prone, and provides the option to compare the results to other experiments by using a common framework. The EUTelescope framework aims to provide full-featured data analysis tools for beam test experiments with telescope detectors. It was implemented in the context of the EUDET project supported by the European Union in its 6th Framework Programme for the European Research Area [DES10] and is embedded into the ILCsoft framework.

The EUTelescope software package was originally created as consistent data analysis chain for the EUDET telescopes [Cor09]. These telescopes consist of six planes with high-resolution MIMOSA26 pixel sensors and are constructed to provide a full-featured infrastructure for Device Under Test (DUT) measurements [H$^+$09]. This includes the telescope with the sensors, cooling, and readout electronics as well as the Data Acquisition (DAQ) tool EUDAQ and the analysis software EUTelescope. The ILCsoft framework is being developed by the ILC community and unites several tools for data processing, originally intended for application in detector development efforts towards the International Linear Collider (ILC). The core elements of the framework are the *Linear Collider*

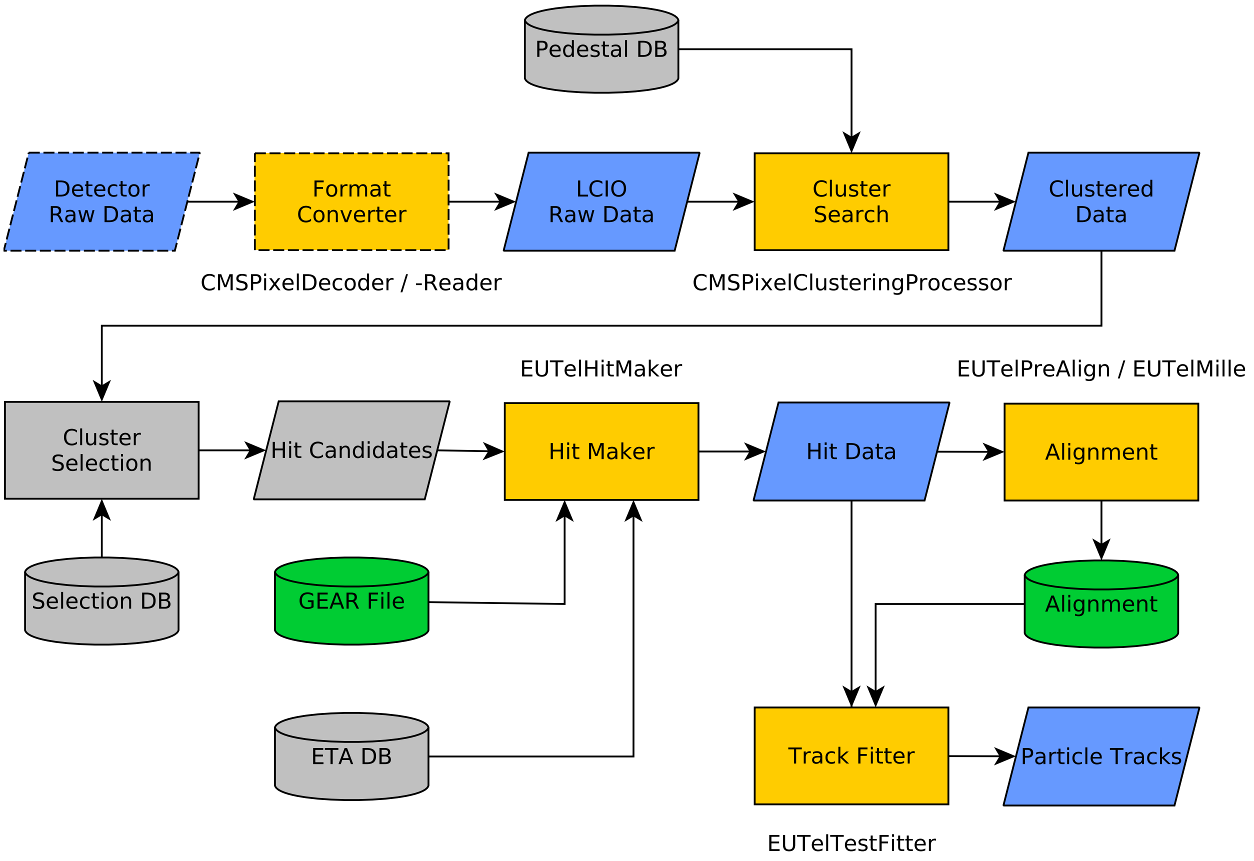


Figure 7.1: Schematic of the overall telescope data analysis strategy of the EUTelescope framework. EUTelescope provides processors for all steps except for the ones with dashed outline; these have to be implemented by the user. The names of the processors used by the CMS pixel telescope data analysis are indicated, descriptions can be found in Section 7.3. Functionalities not used for this analysis are shown grayed out. After [B$^+$07b].

I/O (LCIO) data model (cf. Section 7.1.1), the *Geometry API for Reconstruction (GEAR)* markup language (cf. Section 7.1.2) and the event processor *Modular Analysis & Reconstruction for the LINear collider (Marlin)*. The implementation of EUTelescope into the ILCsoft framework has several advantages, such as the possibility to submit large analysis jobs for Grid computing [B$^+$08]. Marlin allows the modular composition of analysis chains for various applications. Every task is implemented as an independent *processor* that is called by Marlin. The processors can expose parameters to the user which can be configured and loaded at runtime via so-called *steering files* in Extensible Markup Language (XML) format.

EUTelescope provides several processors for Marlin implementing algorithms necessary for a full track reconstruction and data analysis of beam test experiments. Figure 7.1 shows the analysis strategy of the framework which follows the pattern recognition chain presented in Chapter 5 by starting from the converted detector response and obtaining fitted particle tracks in the end. An overview of the processor range provided by EUTelescope is given in [B$^+$07b]. Most of the EUTelescope processors have been developed for the full-frame readout MIMOSA26 sensors and later adapted for zero-suppressed data. Class references for all processors can be obtained from [Hep12].

ILCsoft and EUTelescope installations are available on the DESY Andrew File System (AFS), installation instructions for a standalone version on Scientific Linux 5 machines can be found in Appendix A.1.

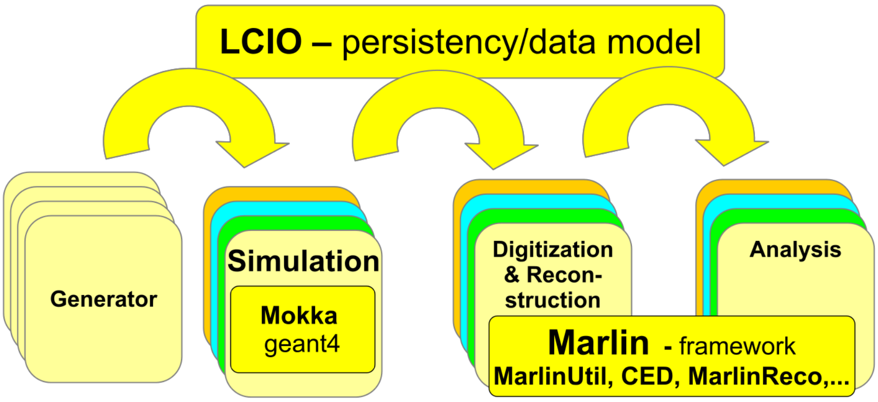


Figure 7.2: Schematic overview of the LCIO persistency data model. It provides a unique data format which can be used in all detector development steps, from simulation up to reconstruction and analysis. Modified from [GE07].

7.1.1 The LCIO Data Format

The data format used in the ILCsoft package is the *Linear Collider I/O (LCIO)* persistency framework and event data model. It provides user interfaces in different programming languages (C++, Java, Fortran) and is designed to cover all fields of detector research and development. Especially the gap between detector simulation and reconstruction should be closed with a unique data format for both sides. Figure 7.2 shows a schematic overview of the data model connecting the different modules. A detailed description of the design concept and implementation of LCIO is given in [G$^+$03].

The interface provided by the library is abstract and hides the actual storing mechanisms from the user. This allows an easy integration into other projects without having to deal with the underlying data model and prevents the user from non-trivial code changes in case the model is altered. This is of special importance since the ILC projects are still under heavy development and implementations are likely to change.

LCIO is an event-based data format. All data belonging to one trigger decision and detector readout is stored together and can be accessed via the corresponding event number. An event consists of the event header and the actual data. The header contains information on the detector, the timestamp, and the run number. The event data is stored in collections of different types. An example from a beam test LCIO file demonstrating the collections concept can be found in Appendix A.3.

Collection types for tracking detectors such as the CMS pixel telescope are denoted with *Tracker**. Each type is specific for a certain stage of the pattern recognition chain. While the *TrackerRawData* (full-frame data) and *TrackerData* (zero-suppressed data) contain raw detector data, the *TrackerPulse* collections are used to store the processed clusters. After the track reconstruction procedure the hits and tracks are stored in *TrackerHit* and *Track* collections, respectively. Each collection contains specific data entities which are all labeled with a unique hexadecimal ID. These IDs connect the otherwise independent collections and allow easy cross-references, e.g. referencing all pixel hits belonging to a particular cluster. This way a reconstructed track can still contain all information down to the original raw detector data.

Two command line tools are particularly useful for fast inspection of LCIO file contents. The structure and included collections can be obtained using the *anajob* command. The content of these collections in a single event can be extracted from the LCIO file using the *dumpevent* command. In this output the correlation of the different data entities can be observed by comparing the *corr.Data* fields and the entity IDs. Examples of outputs from both tools can be found in Appendix A.3.

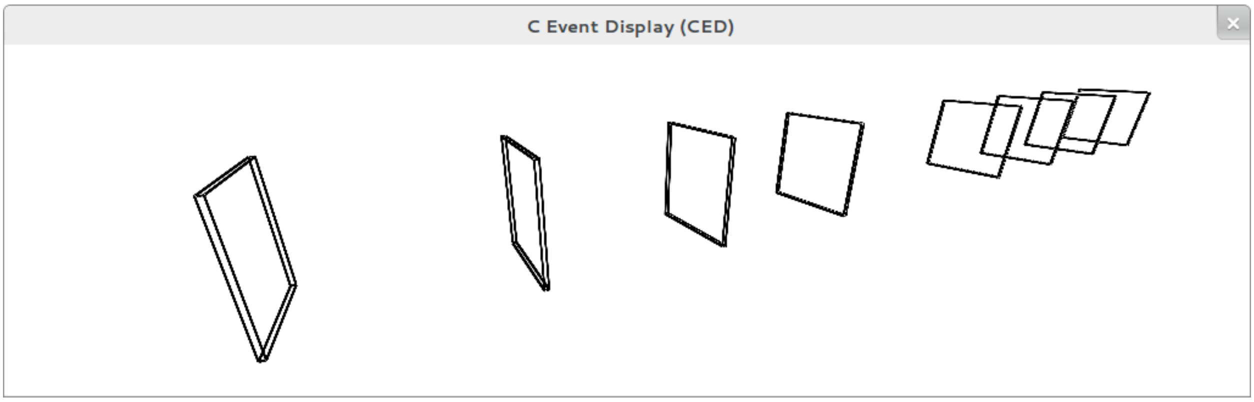


Figure 7.3: Perspective view of the GEAR representation of the straight version of the CMS pixel telescope, provided by the CED event viewer (cf. Section 7.5.5).

7.1.2 Telescope Geometry Description Using GEAR

On the software side the different telescope geometries are implemented using the *Geometry API for Reconstruction (GEAR)* markup language. GEAR is a geometry description toolkit [Gae07] which is part of the ILC reconstruction software. It implements an abstract interface for the layout description of detectors for event reconstruction. The geometry described in a GEAR file is simplified and therefore cannot be used for a detailed Monte Carlo simulation of the detector. For reconstruction on the other hand only a few parameters are enough since no precise material allocation information is needed to calculate the particle behavior. Figure 7.3 demonstrates this simplicity: shown is the geometry of the straight version of the CMS pixel telescope. The pixel segmentation is not rendered.

The integration into the EUTelescope framework allows the usage of the unchanged processor chain with the same parameters for different detector geometries, switching between them just by loading another GEAR file. The processors within the chain obtain all information needed for their algorithm directly from the GEAR framework, e.g. the pixel pitches and plane distances for the coordinate transformation from local on-plane coordinates to the global telescope frame of reference. GEAR uses XML markups for the detector description, whereof several packages for different purposes are provided. In case of silicon pixel detectors the GearType `SiPlanesParameters` can be used providing all properties needed to describe a tracking telescope.

The XML markup contains a hierarchical description of the detector parts starting with the whole detector as well as global parameters such as magnetic field. The telescope can be divided into `layers`. Every layer represents one particular telescope plane in the GEAR description. This allows e.g. the usage of several sensor types with different properties such as pixel pitch or sensor thickness in different telescope planes which is especially interesting for telescopes with exchangeable DUT. A layer consists of a `ladder` and a `sensitive` part. These elements hold the basic parameters such as position, rotation, size, pitch, and the radiation length of the sensitive detector material. Rotations of telescope planes can be obtained either by using a simple 2×2 rotation matrix for the xy plane or the Euler angles for 3-dimensional rotations.

Examples of the different parts of a GEAR detector description are given in Appendix A.2. The entire GEAR files for the two different geometries created for the CMS pixel telescopes can be found in the subfolder *simplesub/* in the project's SVN repository [EUD].

7.1.3 The RAIDA Processor

Another framework embedded in the ILCsoft suite is the *Abstract Interface for Data Analysis (AIDA)* [B$^+$12a]. AIDA provides common interfaces in either C++ or Java for different data analysis tools and tries to unify the communication with these tools. The implementation used in the EUTelescope package and therefore in this beam test analysis is the ROOT [BR97] implementation of AIDA, called RAIDA [DES12]. The main objective of RAIDA is to provide access to ROOT objects within the ILCsoft environment and allow their creation.

Usually the RAIDA processor instance is called as the first Marlin processor directly after loading the LCIO file which will be processed. Once per run the AIDA histograms of each processor using the interface are initialized. During the event processing stage information is added to the histograms. After the run has finished all histograms are written to a ROOT file and are accessible through the ROOT interface.

Almost every EUTelescope processor implements a RAIDA instance to write status histograms for the current analysis step. The *EUTelFitTuple* uses RAIDA to write the created particle tracks and hit measurements to a ROOT tree to allow an easy inspection of all events. Usually a boolean parameter like the `HistogramSwitch` is provided within every processor to turn the histogram creation on or off.

7.1.4 EUTelescope Submission Scripts and XML Templates

To be able to process several runs at once submission scripts can be used to prepare the required Marlin XML steering files from templates. Thereby the turn-around time for new analysis jobs or re-runs with altered parameters can be greatly reduced. EUTelescope ships two different Python-based submission scripts which are able to run a full analysis chain and provide a data infrastructure. For the high-rate beam test analysis the so-called *simplesub scripts* are used.

The XML steering templates contain parameter statements for the respective Marlin processors. Instead of a value for a given parameter, a python variable is placed in the statement, e.g. for the filename of the raw detector data. The submission script is called with the desired step, a configuration file, and a run number. It runs through the XML template and replaces the python variables with the run number and the parameter values provided by the configuration file. For example the command

```
python config-cmspixel.py -a clustering 298 299 300
```

would load the Marlin XML steering file for the *clustering* step in the analysis chain and process the runs 298, 299, and 300. According to the submission script settings several instances of the processor can be called and multiple runs processed in parallel. The configuration file holds all information for the analysis step such as the template filename and all parameters. In addition, the submission scripts are able to execute sequences of steps. For the high-rate beam test two different sequences have been set up:

Data Quality Monitoring: This process is called by the parameter `dqm` and contains the procedures for conversion, calibration, clustering, hit generation, and track fitting (without alignment procedure). It is used to monitor the telescope data during the beam test (cf. Section 7.3.7).

Full Analysis Chain: Invoked with `full` this process performs a full analysis of the given beam test data. It includes conversion, calibration, clustering, hit generation, prealignment, alignment, and final track fitting as well as the output of ROOT trees for further investigation.

All submission XML templates for the different processors in the CMS pixel telescope analysis chain can be found in the subfolder *simplesub/templates_cmspixel/* of the project repository [EUD].

7.2 The *PSI46* Raw Data Format

A compact binary data format has been developed for the CMS pixel detector to allow a fast transmission of the readout. The readout of a detector can either be controlled by the pixel Front End Drivers (FEDs) in the CMS cavern as for the actual operation of the CMS pixel detector, or by the test boards for simple laboratory test setups. In the following sections only the test board readout option with the CMS pixel telescope and its data format is described. Even though the responses from the Read Out Chips (ROCs) and Token Bit Managers (TBMs) are the same, additional information in the data stream might differ between the two readout options.

The data recorded during a telescope run consists of several sequences depending on the configuration of the detector. Each of the components in the signal chain adds its bit sequences to the data stream. Since the CMS pixel detector ROCs are usually operated with a TBM controlling the token signal distribution and the readout (cf. Section 4.4.4), this is the Field Programmable Gate Array (FPGA) on the test board, the TBM and the single ROCs themselves. A full event sequence therefore contains a FPGA pattern, and the TBM header and trailer patterns which enclose the actual event data. The patterns in brackets are repeated for every ROC and its pixel hits respectively:

```
FPGA_HEAD | TBM_HEAD | [ROC_HEAD [pixel hits]] | TBM_TRAILER
```

For convenience the test board firmware is also able to control a single ROC without TBM either by emulating the TBM behavior or by addressing the ROC directly. This is especially interesting for test procedures performed in the laboratory without full module. Since the new TBM08 has not been ready for submission, the high-rate beam test in July was the first experimental setup operating more than one ROC without TBM or TBM emulation. In turn this means that the data stream from the different events is not separated by the TBM headers and trailers but only by the FPGA patterns indicating the current readout state of the detector. Accordingly, the corresponding `TBM_HEAD` and `TBM_TRAILER` are missing when operating without a TBM:

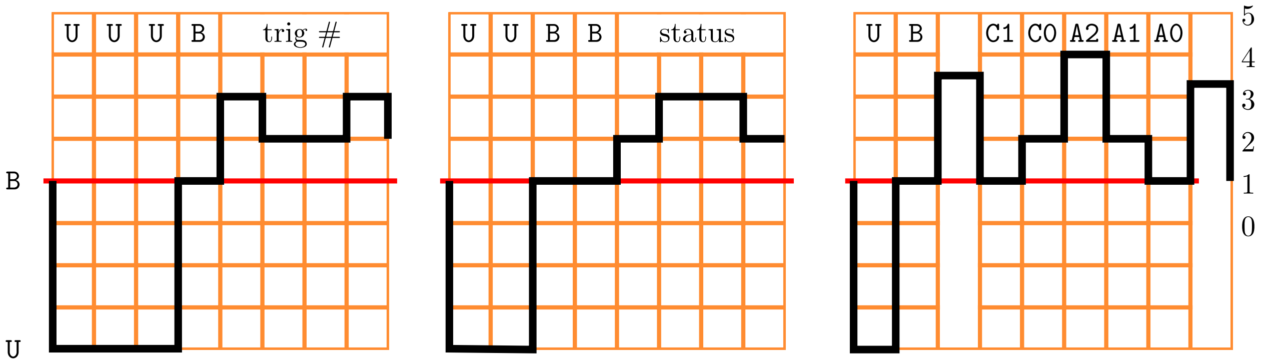
```
FPGA_HEAD | [ROC_HEAD [pixel hits]]
```

In the following sections, all numbers or patterns starting with `0x` are hexadecimal numbers. Every digit contains 4 bit encoded in the characters `0-F`. An example of raw data from a single ROC with TBM emulation by the FPGA can be found in Appendix B.1.

7.2.1 FPGA Headers

The trigger distribution and overall readout of the telescope is controlled by the test board FPGA, in the July test beam by the Altera Cyclone. The FPGA uses the Inter-Integrated Circuit (I^2C) interface to communicate with the ROCs and the commands which have been sent are written to the data stream as 16-bit *FPGA headers*. Depending on the readout state of the detector, different header patterns are written and can be decoded after downloading the data:

- Reset header `0x8008`: A reset command is issued after the readout of an event. It resets the internal data and timestamp buffers of the ROCs.
- Calibration header `0x8010`: An internal calibration pulse is sent to the ROCs.
- Trigger header `0x8004`: This header is written when a trigger signal is sent to the ROCs. A trigger header should be followed by a data header.
- Data header `0x8001`: This header marks the beginning of the event data. Everything till the next FPGA header belongs to the detector readout.
- Data header with buffer overflow marker `0x8081`: Same as above, but with the additional information that the data buffers have been full or overflowing.



(a) TBM header (b) TBM trailer (c) ROC header and hit

Figure 7.4: TBM06 and *PSI46v2* ROC header and trailer levels. `U` is the Ultrablack, `B` the Black level, 0-5 are the address levels. (a) The TBM header pattern is `UUUB` and a trigger count. (b) The TBM trailer pattern is `UUBB` and a status information. (c) The ROC header consists of `UB` and the *lastDAC* value. Afterwards the hit information is encoded and the pulse height sent as analog value.

Every FPGA header is followed by 48 status bits containing additional information about the trigger count or test board settings. Depending on the operation mode different header sequences are possible. Since the internal random trigger functionality is mainly designed for laboratory tests of the detector, a full signal cycle with calibration is performed on each trigger:

```
0x8008   0x8010        0x8004    0x8001    ...
reset    calibration   trigger   data
```

The cycle is shorter and does not contain the reset and calibration pulses when using external trigger signals. This allows e.g. *trigger stacking* which allows the readout of multiple events from the same buffer without resetting it in between:

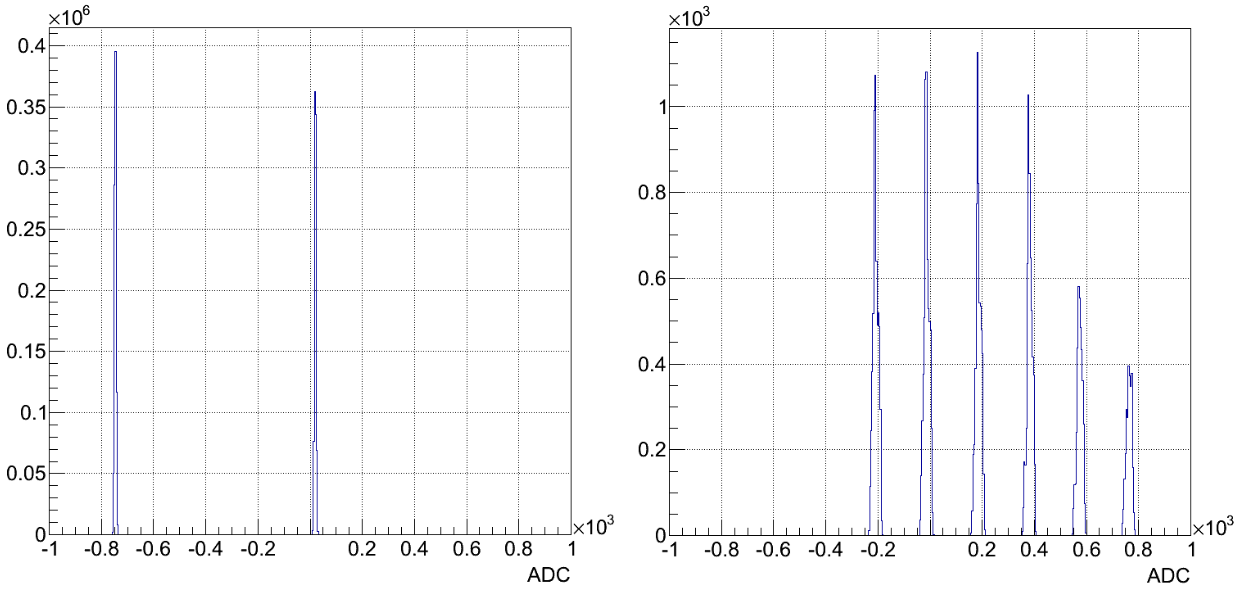
```
0x8004    0x8001    ...
trigger   data
```

The *data* FPGA headers can be used to split the raw data stream from the detector into single events. This ensures both reliable separation of the event data and straightforward trigger signal correlation. For the readout over the IPBus Ethernet interface using the Xilinx board (cf. Section 6.2.2) only the 32-bit pattern `0xFFFF 0xFFFF` followed by additional status information will be used as header separating the single events.

7.2.2 The *PSI46v2* and TBM06 Data Format

The *PSI46v2* ROC uses analog differential voltage level encoding to transmit the data with a data rate of 40 MHz. Several voltage levels are defined; the Ultrablack `U` and Black `B` levels are used to form the header patterns. These sequences are designed not to interfere with any possible detector data by providing a distinct pattern. Six address levels `0-5` are used to transmit digital data values like the pixel ID and double column address. Analog values for pulse height and *lastDAC* are directly transmitted as voltage level without level binning.

Figure 7.4 shows the different patterns for TBM and ROC. The TBM sequences start with two Ultrablack signals, followed by either `UB` (Figure 7.4 (a), marking the header) or `BB` (Figure 7.4 (b), marking the trailer). The remaining four voltage level bits are used to provide additional TBM information such as the internal trigger count or status messages. Figure 7.4 (c) shows the ROC



(a) Ultrablack and Black levels

(b) Address encoding levels

Figure 7.5: Examples of data encoding address levels in the *PSI46v2*. All levels are clearly isolated. (a) Ultrablack and Black levels for a single ROC. (b) The six address levels for the double column and pixel ID encoding.

header together with one pixel hit. The header is denominated by `UB` followed by the *lastDAC* value. The *lastDAC* is an analog voltage level returning the value of the most recently programmed Digital-Analog Converter (DAC). This allows to check the DAC programming success without additional commands and data transfers blocking the readout. After the ROC header all pixel hits from this particular ROC are transmitted. The hit contains the two-digit double column address `C0 C1` and the pixel ID within this particular double column, transmitted in three digits `A2 A1 A0`. The pulse height signal itself is transmitted as a single analog voltage level. This sequence is repeated for every pixel hit the current ROC holds.

The double column and pixel ID are always transmitted sending the Most Significant Bit (MSB) first. The translation of the double column and pixel ID sequences into x and y pixel coordinates is described in Section 7.2.4. Usually the different voltage levels of one ROC can be easily separated as shown in Figure 7.5. The spread of voltage levels between different ROCs is usually small which is important since in the CMS pixel detector up to 16 ROCs controlled by one TBM are read out at once. The address levels of a ROC or full module are stored in a file during the calibration process. This address levels file can be used to determine the level of signals during the data decoding procedure. The exact design voltage levels as well as the readout timing scheme are given in [Gab05].

7.2.3 The *PSI46dig* and TBM07 Data Format

With the higher detector occupancy and increased LHC luminosity the analog 40 MHz readout does not provide enough bandwidth and will be dropped in favor of a digital 400 MHz detector readout scheme as described in Section 4.5.3. This implies a change of the identification patterns for headers and trailers as well as the need for a new data decoder. Every header sequence of the digital format consists of 12 identification bits. For the TBM another 16 bit for status information are added as shown in Figure 7.6. The content of the status information bits is still subject to change with the upcoming submissions of the TBM07 and the combined TBM08 (cf. Section 4.5.3).

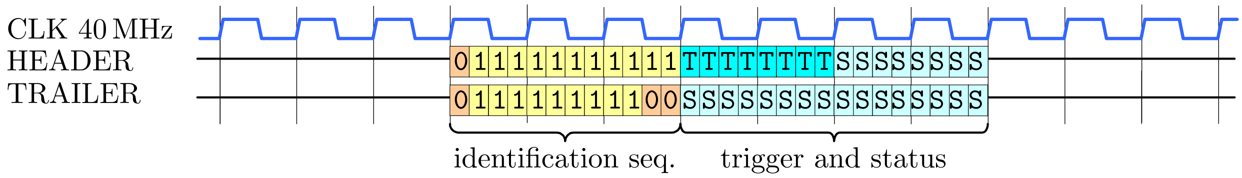


Figure 7.6: Data format of the TBM07. Both the header and trailer start with an identification sequence with nine consecutive `1`. Headers and trailers are distinguished by the following two bits, where headers are marked with `11` and trailers with `00`. Then 16 bits of trigger count `T` and status information `S` are transmitted.

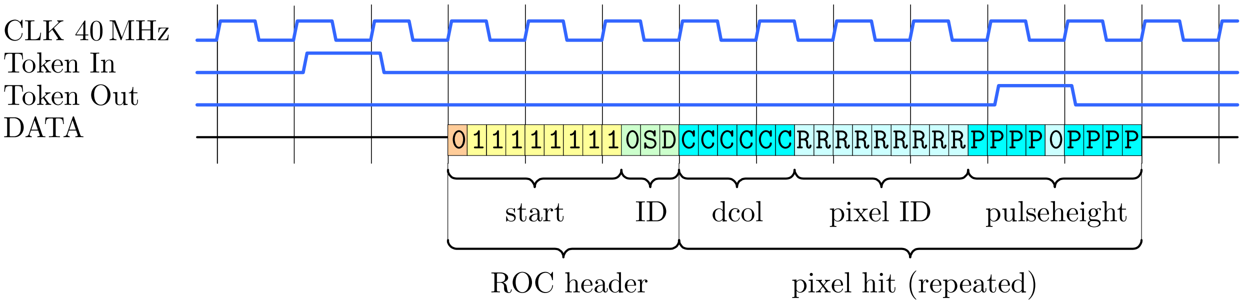


Figure 7.7: Data format of the *PSI46dig* ROC. The data stream starts one clock cycle after the *Token In* with a 12-bit ROC header which is followed by one or more pixel hits. Pixel hits consist of double column (dcol, 6 bit), pixel ID (9 bit), and pulse height (8 bit). The additional zero in the pulse height bits prohibits the occurrence of a fake ROC header. Modified from [Mei12].

The ROC header and hit patterns are shown in Figure 7.7. The header consists of eight consecutive `1` with a leading and trailing zero and two reserved status bits `S` and `D`. In hexadecimal this allows the following headers depending on the status bit settings:

S=0 D=0	S=0 D=1	S=1 D=0	S=1 D=1
0x7f8	0x7f9	0x7fa	0x7fb

The pixel hit sequence contains the 6 bit double column `CCCCCC`, the 9 bit pixel ID `RRRRRRRRR`, and the 8 bit pulse height `PPPP 0 PPPP` with a range from $0 - 255$. An additional zero bit is placed in the middle of the pulse height pattern in order to prohibit the occurrence of a fake ROC header sequence.

In the current submission of the *PSI46dig* ROC the pixel ID bits are inverted. This has to be taken into account when decoding the raw data. Due to this flaw it is possible to detect fake ROC header sequences in the middle of a pixel hit and the decoder processor has to be sensitive to the position where the pattern occurs (see Section 7.3.1 and 7.5.2).

7.2.4 Pixel ID and Double Column Encoding

The coordinates of a pixel hit can be determined from the hit information either by calculating the coordinates directly or by using Gray code tables. Gray code is a reflected binary code often used in signal transmission. It decodes values in such a way that two successive numbers only differ by one bit [Gra53].

The pixels on a CMS pixel ROC are grouped in 26 double columns as described in Section 4.4.3 and the double column ID runs from 0 to 25. From the analog voltage levels or the digital data bits which are marked in red the double column ID and the pixel ID can be constructed as follows:

```
int doublecol =          C1*6 + C0;
int pixelid   =  A2*36 + A1*6 + A0;
```

For the digital data format, C1 consists of the three Most Significant Bits (MSBs) and C0 of the Least Significant Bits (LSBs). The same applies to the pixel ID, where A2, A1, and A0 consist of three bits each. To retrieve the correct pixel column more information than just the doublecol value is needed. The pixel ID describes the location of the single Pixel Unit Cells (PUCs) within one double column and the last digit decides on the column affiliation. All even pixel IDs belong to the left part of a double column, the odd IDs to the right column:

```
int column = doublecol*2 + pixelid%2;
```

where % is the modulo operator returning the remainder of the division. The pixel row information can be obtained only from the pixel ID. The rows are numbered from 0 to 79 starting with the row closest to the double column buffers, i.e. to the global ROC readout electronics. The pixel numbering follows a zigzag pattern starting with the left column in row 0. Every pixel row can be calculated from the corresponding pixel ID:

```
int row = abs(int(pixelid/2) - 80);
```

For example the pixel hit with the bit patterns 0 4 3 4 2 (or 0b000100011100010 for digital readout) can be decoded by first calculating doublecol=4 and pixelid=134. This in turn leads to column=8 (left side of the double column, since pixelid%2=0) and row number row=13. Hence, the regarded pixel hit was recorded in the pixel with the coordinates $x = 8$ and $y = 13$. All Gray code tables relating pixel and double column IDs and their coordinates can be found in [Gab05].

7.3 The Telescope Data Analysis Chain

A EUTelescope data analysis chain has been set up and configured for the CMS pixel telescope. The chain has been tested and tuned using Monte Carlo simulations and data from laboratory measurements with single ROC modules. The main part of the preparation has been the development of a decoder algorithm for the different *PSI46* ROC flavors. Furthermore the data has to be converted to LCIO, calibrated and then fed to the subsequent Marlin processors.

Some of the existing EUTelescope processors used for the data processing have been altered in order to fit the needs of the high-rate beam test. A flowchart of the full analysis chain for the pixel telescope showing the different processors and data collections used in the analysis steps can be found in Appendix B. The EUTelescope processors used for the CMS pixel telescope analysis are described in the following sections.

7.3.1 The CMSPixelDecoder

The first step in the analysis chain is the decoding of the raw data and the translation into the LCIO format. The so-called *CMSPixelDecoder* which is accomplishing this task has been developed as part of this thesis. The decoding is done independently from the LCIO conversion and the decoder is outsourced into a standalone library to enable easy portability to other frameworks. Originally it has been developed specifically for this high-rate beam test but is able to decode a wide range of data formats produced by the different versions of the *PSI46* ROCs and various setup configurations. It is completely implemented in C++.

The processor is able to decode data streams from any number of both analog *PSI46v2* or digital *PSI46dig* ROCs. The data stream can contain TBM signatures or can be recorded without. In addition a set of parameters and flags can be used to influence the verbosity and decoding behavior for various situations. The input format of the decoder is a filestream with raw detector data. A new instance of the decoder is invoked with the constructor, either for a digital or analog ROC. For digital data the decoder instance constructor is defined as:

```
CMSPixelDecoderDigital(const char *file, int *status, unsigned int
    noOfROC, unsigned int flags, unsigned int evtSel = 0, unsigned int
    verbosity = 1);
```

where `file` is the data file containing data from `noOfROC` ROCs, and the unsigned integer value `verbosity={0-3}` sets the verbosity level:

- `verbosity=0`: Only major errors are printed to `stdout` such as a missing address levels file for analog ROCs or the number of ROCs from the GEAR file does not correspond to the number read from the address levels file.
- `verbosity=1`: Both errors and warnings are printed. Warnings are issued e.g. if an event is skipped due to invalid data length or the number of detector ROC headers was incorrect. Furthermore address decoding errors from pixel hits produce a warning.
- `verbosity=2`: This setting activates the debugging output additionally to errors and warnings. All steps which are carried out are listed with additional information. This includes notifications about empty events as well as the coordinates for every pixel hit which has been decoded.
- `verbosity=3`: With the highest debug setting the processor is rather verbose and prints the raw data and the pattern sequences for every single step, including all messages mentioned previously. The usage of this option is not recommended for productive decoding since it slows down the procedure. Usually this mode is only used for decoding algorithm debugging.

The processor statistics (see below) are printed in any case, independently from the debug setting. The `status` value contains the constructor error code in case the file could not be opened or another problem occurred. The following flags can be set to influence the decoding of the raw data by:

```
flags += <FLAG_NAME>;
```

- `FLAG_EMPTYEVENTS` can be used to tell the decoder what to return. If not set the decoder skips events (trigger signals) in the data stream that do not contain at least one pixel hit and instead returns the next event with valid hits. This behavior might not be desired in some cases where the synchronization with another detector with the same trigger signal is needed, e.g. when using the CMS pixel ROC as DUT in an EUDET telescope. If the flag is set the decoder returns every event even if it only contains header information.
- `FLAG_LAZYDECODING` sets the strictness of decoding and can be used to select different events. If this flag is not set the decoder checks every event against several requirements such as total data length, presence of TBM headers and trailers, or the correct number of ROC headers. For digital ROCs strict decoding is not possible yet, so the flag is mandatory in this case. This flag is deprecated and will probably be merged with the `evtSel` setting in the next version of the CMSPixelDecoder (see below).
- `FLAG_HAVETBM` switches between input data formats recorded with TBM or TBM emulation and plain data from the ROCs. Not setting this flag to the correct value might lead to unexpected results from the decoder since the algorithm either skips the first data bits or tries to decode a present TBM header as ROC data.
- `FLAG_IPBUS` sets the input format of the raw data. Without this flag the decoder expects data taken with an Altera Cyclone FPGA and its specific headers and data format, recorded over the USB interface (cf. Section 7.2.1), with activated flag the algorithm decodes data recorded with the Xilinx Spartan-6 FPGA over the IPBus interface.

A possibility to select events from the decoded data is provided by the evtSel variable. This parameter was introduced for the July beam test where several quality issues were discovered while decoding the data (these decoding problems are described in Section 7.5.2). evtSel=0 returns all events that could be decoded without major problems, no quality cut is set. Only the events containing the exact number of ROC headers expected by the configuration can be selected using evtSel=1. These events can again be filtered with evtSel=2 which will only return events that have no bit errors within the ROC headers. For a flawlessly working readout chain and electronics all three settings should produce the same result.

The data processing in CMSPixelDecoder is event based which makes it fast and reliable. It does not try to run through the whole data stream searching for event headers and splitting it but only decodes the data up to the next event header and returns the result immediately. This enables both a fast decoding and on-the-fly data processing by appending newly recorded data to the stream and then calling the decoding method get_event for the next event. get_event returns the decoded data for this event:

```
vector <vector <CMS_pixelhit> > event_data;
status = readout->get_event(&event_data);
```

Every vector element of event_data represents one ROC which in turn holds all pixel hit elements. Pixel hits are described by the CMS_pixelhit data structure containing three integer values for the x and y pixel coordinates, and the pulse height ADC value.

The overall decoding strategy of the CMSPixelDecoder is shown in Figure 7.8. After cutting off the next event data sequence from the data stream the raw data has to be prepared for decoding. For the Altera test board readout of the *PSI46dig* this means for example the removal of extra zeros in the data stream (cf. Section 6.2.2). The data sequence then runs through some general checks on data length and the TBM header and trailer detection. If the data passes these tests the decoding is performed, starting at the first ROC header found in the sequence and discarding all preceding data. The following bits are again compared to the ROC pattern and if it agrees the ROC counter is increased by one. If no accordance is found the CMSPixelDecoder sends the same data to the hit decoding unit which tries to extract the pixel coordinates and the pulse height. These steps are repeated until the end of the data sequence is reached. After more checks on the number of valid pixel hits and the total noOfROC the event is returned by the method.

At any time during the decoding procedure of a data stream the decoder's statistics can be obtained by calling the print_statistics() method. It prints detailed statistics about the current status of the decoding to stdout such as the number and type of detected FPGA headers and counts for accepted, rejected, and empty events. Furthermore the total number of decoded pixel hits and failed pixel address decodings as well as the initial settings for the cut criteria are given. The statistics allow a fast assessment of the decoding process and reveals potential data or configuration problems without having to look at histograms where wrong decoding might lead to patterns which are difficult to discover. An example of statistics output from a telescope data run in the high-rate beam test is given in Appendix B.1. A reset of the decoder statistics can be issued by starting a new decoding run. This is done by creating a new instance of the decoder.

Several other projects are currently attempting to use the CMSPixelDecoder as their data decoding instance such as CMS pixel detector beam tests at CERN and DESY. Furthermore the decoder is used in the calibration and DAQ framework currently under development for the Xilinx test board IPBus readout for the October beam test. This allows coherent data decoding for both calibration and offline data analysis. The CMSPixelDecoder code can be obtained from the [EUD] in the files include/CMSPixelDecoder.h and src/CMSPixelDecoder.cc. An example implementation of the decoder can be found within the CMSPixelReader.

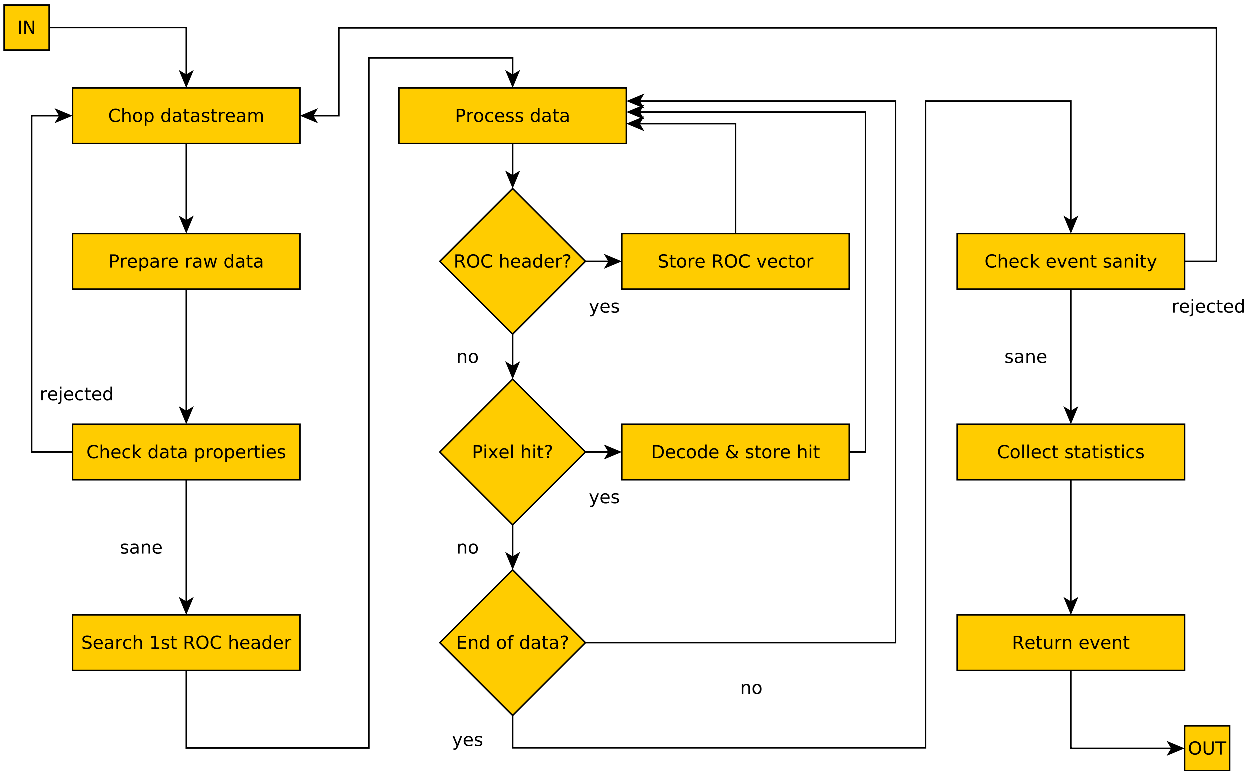
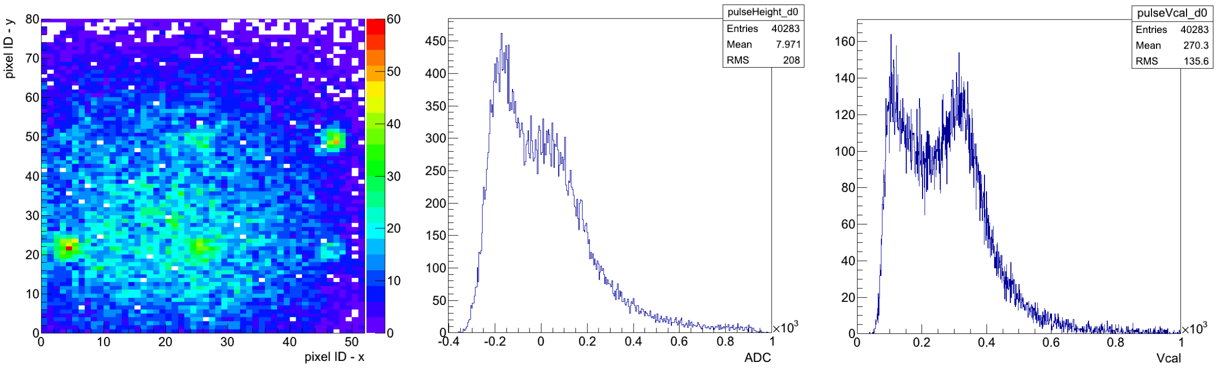


Figure 7.8: Data processing strategy of the CMSPixelDecoder. First the event is read from the data stream and prepared for decoding. After some general checks like total data length and presence of TBM headers (if any) the decoding starts with the search for the first ROC header. Then the algorithm checks for another header and if not tries to decode the data as a pixel hit. After reaching the end of the sequence the re-checked event is returned.



(a) Detector hit map

(b) Raw pulse height

(c) Gain calibrated

Figure 7.9: CMSPixelReader and CMSPixelCalibrateEvent histograms from a single ROC measurement with scintillator trigger and ^{90}Sr source. (a) shows the *hit map*, the accumulated hits on the detector. (b) is the uncalibrated distribution of pulse height values and (c) the pulse height distribution after gain calibration. The shape is now similar to a Landau distribution with a most probable value of 0.35 Vcal and additional entries at lower energies around 0.1 Vcal. These pixel hits belong to two-pixel clusters (cf. Section 7.3.4) and clustering them reveals the Landau shape as shown in Figure 7.11 (a).

7.3.2 CMSPixelReader and CMSPixelCalibrateEvent

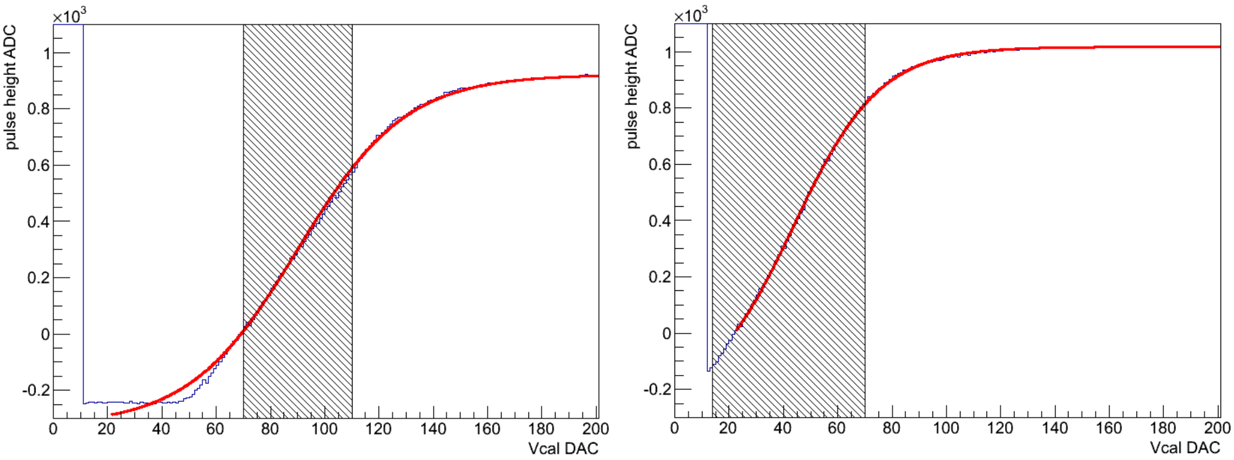
The CMSPixelReader integrates the CMSPixelDecoder into the ILCsoft framework. It provides the conversion of the data necessary for using the EUTelescope framework for data processing. The decoder instance runs over the raw data input file serving decoded per-ROC event data which is stored in LCIO collections by the CMSPixelReader processor. A RAIDA instance is used to produce ROOT histograms for each run allowing fast judgment on data quality, detector status, and its position within the beam or the radioactive source. Both detector hit map and pulse height distribution histograms in arbitrary ADC units for every ROC are created as shown in Figure 7.9 (a) and (b) for a measurement with a single ROC and ^{90}Sr source.

After the conversion and decoding of the detector data, a gain calibration has to be performed. Due to differences in the processed ROC and its resistors and capacitors the recorded pulse height from different pixels for the same amount of primary charge carriers is not necessarily the same. To get comparable values which enable e.g. energy weighting in clusters (cf. Section 5.1.2) every single pixel has to be calibrated. In the *PSI46* chips this can be done by injecting a known *Vcal* pulse into the PUC and measuring the resulting pulse height delivered by the ROC. For every pixel a curve with several of these measurements with different *Vcal* values is recorded and a hyperbolic tangent fit is performed:

$$y = p_3 + p_2 \cdot \tanh\left(p_0 \cdot x + p_1\right), \tag{7.1}$$

where x is the injected calibration pulse in $Vcal$ units, y the uncalibrated PUC response (pulse height in ADC units) and p_{0-3} the fit parameters. Some of the DAC registers of the ROC can be used to tune this calibration to extend the linear response range as shown in Figure 7.10. Especially for the *PSI46v2* this has a large impact while the pulse height calibration of the *PSI46dig* is almost independent of the corresponding DAC settings. An elaborate description of the calibration procedures and the DAC settings for gain calibration can be found in [Hoß12]. The parameters of this fit are stored in a calibration file. With this file and the inverse function

$$x = \frac{\operatorname{artanh}\left(y - p_3\right)/p_2 + p_1}{p_0}, \tag{7.2}$$



(a) Pule height calibration before

(b) and after parameter tuning

Figure 7.10: Pulse height calibration for standard and tuned DAC settings. (a) shows the linear range of the calibration of a PUC with the standard DAC settings and the fitted tanh function. (b) shows the extended linear range of the calibration curve obtained with tuned settings which enables a more precise pulse height determination.

the recorded data can be calibrated into comparable *Vcal* DAC units which can then be translated into the number of primary charge carriers created in the sensor by a linear function. In the EUTelescope analysis chain for the CMS pixel telescopes this is done by the CMSPixelCalibrateEvent processor. The raw Analog-Digital Converter (ADC) value of every single pixel hit is transformed using the parameter set from the corresponding calibration file. The calibrated detector data is written to a new LCIO collection and a histogram of the calibrated pulse height spectrum is stored as shown in Figure 7.9 (c). Additionally a so-called *NANMap* histogram is produced showing all pixels which failed to be calibrated and returned *Not a Number* due to the boundaries of the artanh function. The LCIO data written by the CMSPixelCalibrateEvent processor is ready for clustering and energy-weighted hit position determination.

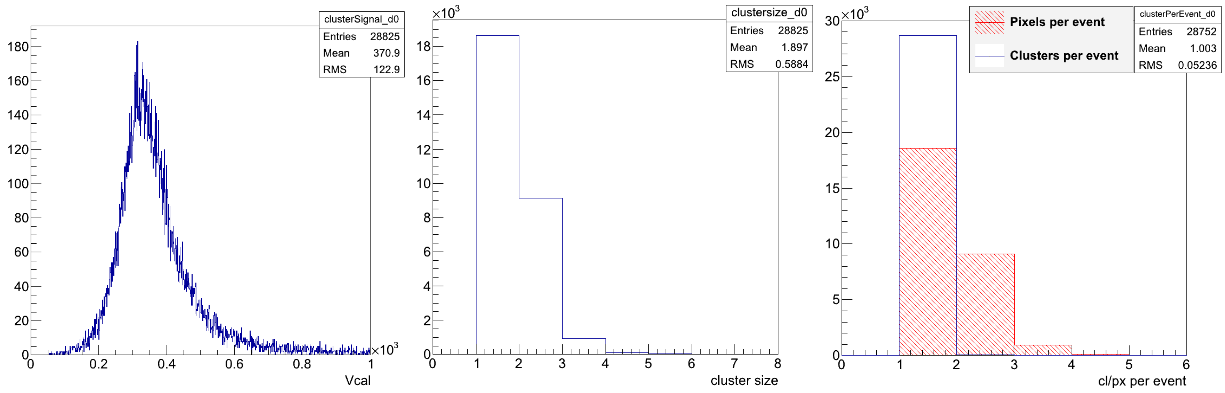
During the July beam test no gain calibration could be performed. The *psi46expert* software which is usually used for the pulse height calibration procedure does not fully support the *PSI46dig* ROCs yet. Therefore the calibration step has been skipped for the data analysis of this beam test by directly clustering the uncalibrated raw detector data. This of course has implications on the energy weighting algorithm (cf. Section 7.3.5) and limits the precision of the hit position determination.

7.3.3 Hot Pixel Handling

The detection and exclusion of *hot pixels* is an important task and has been included in the CMS pixel telescope analysis chain. A *hot channel* or *hot pixel* is a pixel that fires with too high frequency e.g. due to a wrong threshold setting. Already one single hot pixel can affect the clustering and tracking procedure since it creates additional clusters in the data set for every event. Two different behaviors of hot pixels can be distinguished:

- Continuously firing pixels with a constant signal frequency
- Hot pixel bursts where otherwise properly working pixels fire continuously only for a certain time

In general hot pixels can be detected by counting the firing frequency of every pixel during the run data processing. If the firing frequency exceeds a certain threshold the pixel is denoted as *hot*



(a) Signal distribution

(b) Cluster size

(c) Cluster/pixel per event

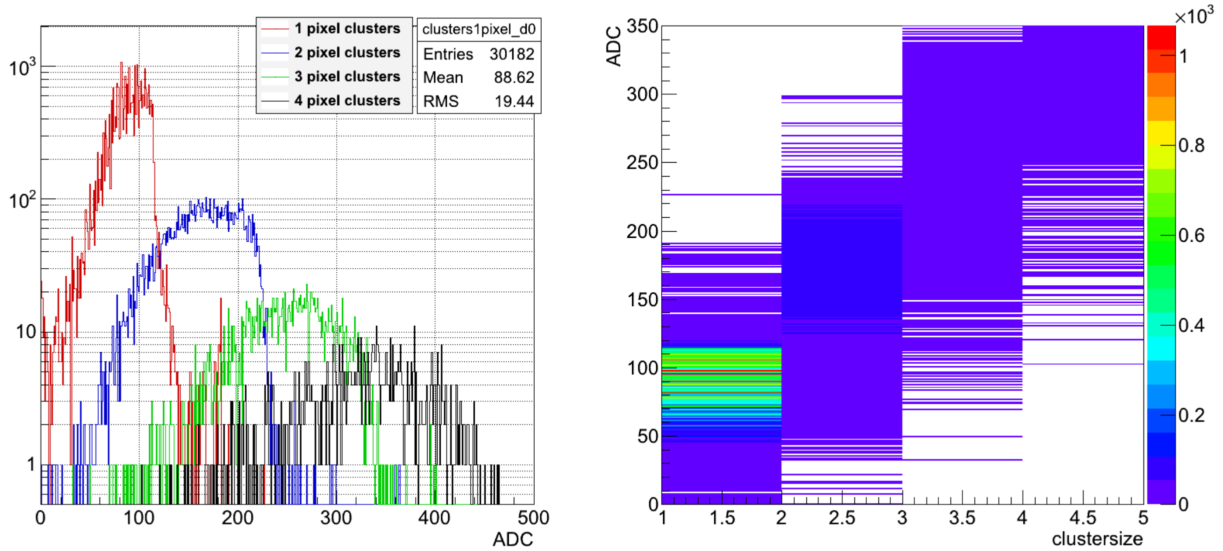
Figure 7.11: CMSPixelClusteringProcessor histograms from a ^{90}Sr source measurement. (a) shows the signal distribution from clusters with all sizes as function of their charge in $Vcal$ DAC units. The number of clusters as function of their size is shown in (b), and (c) shows the number of clusters and fired pixels per event.

pixel. In the analysis chain of this high-rate beam test, the hot pixel detection takes place during the calibration procedure. The EUTelHotPixelKiller processor collects firing pixels in a map and calculates the firing frequency by dividing by the number of regarded events (the `NoOfEventPerCycle` parameter). This frequency is compared to the processor parameter `maxAllowedFiringFreq` and the pixels exceeding this threshold are marked as hot. This procedure is repeated several times, according to the `TotalNoOfCycle` processor parameter. For the high-rate beam test four cycles with 100 events each are suitable settings. This approach is very effective for continuously firing pixels, and only a small fraction of events has to be processed. However, it is not possible to detect bursts since therefore all events would have to be considered which slows down the procedure. This has not been implemented for this beam test and no similar effects occurred. Currently it is not foreseen in the EUTelHotPixelKiller processor to allow manual hot pixel flagging but this feature might be added in the future.

The hot pixels are stored in a separate LCIO database which can be used by subsequent processors. The CMSPixelClusteringProcessor (see below) reads the database and excludes these pixels from the clustering process. The configuration of the EUTelescope simplesub submission script provides a parameter to switch the hot pixel handling off completely. The performance of EUTelHotPixelKiller has been studied with Monte Carlo simulations under various conditions as will be described in Section 7.4.1.

7.3.4 CMSPixelClusteringProcessor

The pixel hit clustering is performed by the CMSPixelClusteringProcessor. It implements an efficient sparse clustering algorithm connecting all adjacent pixel hits to one cluster (cf. Section 5.1.1). The algorithm is the same as in the clustering processor APIXClusteringProcessor developed for ATLAS pixel detector beam tests but has some additional functionality such as hot pixel handling or counting of the total number of clusters per telescope plane. Furthermore additional histograms are produced and written to the ROOT file as described below. The most important parameters of the clustering processor are the minimum total charge for a cluster to be accepted (`MinCharge`) and the minimum number of pixels `MinNumberOfPixels` that are needed to form a cluster. Both parameters can be used to reduce the influence of noise hits to the cluster finding, e.g. by excluding one-pixel clusters for tilted sensors. If the `HotPixelCollectionName` parameter is given the processor will check for a corresponding hot pixel database. All pixels listed in this collection will be excluded



(a) Cluster signal spectra

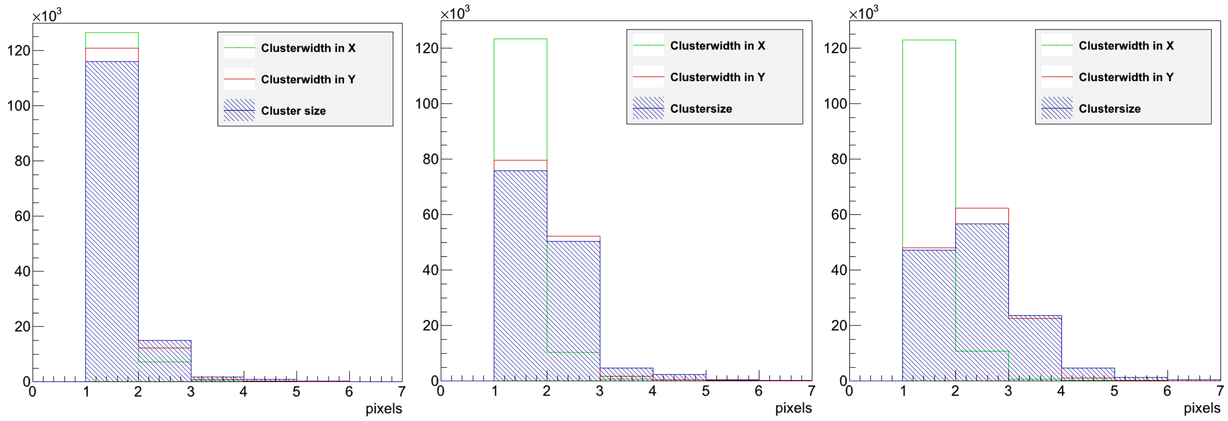
(b) Cluster size vs. charge

Figure 7.12: Histograms for different cluster sizes obtained from RUN000297 from the July beam test. The data has not been calibrated and is therefore given in ADC units. (a) shows the signal spectra from clusters up to a size of four pixels in a 1D histogram in logarithmic scale. (b) shows a 2-dimensional plot of the cluster signal as a function of the cluster size. The charge distribution for every cluster size is visible in the color coded columns.

from the cluster finding procedure beforehand. The clusters adjacent to a hot pixel are not affected by this approach even though there might be some crosstalk in the ROC or sensor which could smear out the cluster energy distribution. However, this effect hardly affects the tracking procedure, and rejecting these clusters would degrade the overall tracking efficiency.

Due to the importance of the clustering step several histograms are produced to allow an easy inspection of the clustering process and its results. For example the signal distribution for all clusters, and the cluster size distribution are shown in Figure 7.11 (a) and (b), respectively. Figure 7.11 (c) shows the total number of clusters and pixels per event. It can be seen that even though about one third of the events contains more than one pixel, almost all events include only one cluster. The plots have been produced with the same data used for Figure 7.9. Additional histograms with the charge distributions of only one-pixel clusters, two-pixel clusters and so on are written to be able to study their signals separately as shown in Figure 7.12 (a). Furthermore, a 2D histogram of the cluster signal distribution and the cluster size provides a fast overview of the cluster situation as shown in Figure 7.12 (b). These plots have been generated from July beam test data which has not been calibrated (cf. Section 7.3.2). The clusters are stored in a new *TrackerPulse* collection in the LCIO file. Every cluster entry holds the unique IDs of the associated pixel hits (cf. Section 7.1.1) as well as the total cluster charge. This information can be used by energy weighting algorithms to determine the hit position from cluster size, shape, and charge.

An alternative processor with the fixed frame clustering algorithm (cf. Section 5.1.1) has been set up to compare the results. The XML steering file calls the EUTelClusteringProcessor, and the fixed frame algorithm is handed over as parameter. This step can be invoked by calling the *simplesub* script with the `clustering_ff` argument. In the July beam test this processor has not been used due to limitations in the current implementation concerning the total number of clusters that can be found within one sensor plane.



$\alpha = 0°$ $\alpha = 15°$ $\alpha = 30°$

Figure 7.13: Cluster size and width in x and y for different tilt angles. For $\alpha = 0°$ the 3 GeV electron beam incidents perpendicular to the sensor plane. Both the increase of larger cluster sizes and the different behavior in x and y direction can be observed and shows that the sensor has been tilted along y. The histograms are scaled to the total number of clusters at $\alpha = 0°$.

Measurement Example

Measurements for a detailed study of cluster sizes and charge distribution in a CMS pixel sensor have been performed with 3 GeV electrons at DESY [Pit12]. Both analog and digital single ROCs have been operated as DUT in an EUDET telescope and have been tilted in different angles with respect to the beam. Figure 7.13 shows the cluster sizes from the CMSPixelClusteringProcessor algorithm for different tilt angles of the analog *PSI46v2* ROC in the beam. The tilt angle is measured from the sensor plane surface normal to the beam so that $\alpha = 0°$ indicates a perpendicular incidence. The growing cluster size is clearly visible for higher tilt angles, furthermore the tilt direction can be determined by considering the cluster width in x and y independently. While the cluster width in x direction stays constant, the cluster width in y grows continuously indicating a tilt along the column direction of the ROC.

7.3.5 Energy Weighting and Alignment

The hit position determination is done by the EUTelHitMaker processor deploying a center-of-gravity energy weighting algorithm (cf. Section 5.1.2). This approach is especially interesting for the tilted telescope geometry for which clusters larger than one pixel are expected. For single-pixel clusters the center-of-gravity method does not improve the position resolution. The input collection of this processor has to be of the *TrackerPulse* type containing clustered, zero-suppressed pixel hits. A *TrackerHit* collection with three-dimensional hit coordinates in the global telescope frame of reference forms the output. To perform the conversion from the local sensor plane coordinates the GEAR description of the beam telescope with the corresponding pixel pitches and plane distances is needed.

The most important parameter for EUTelHitMaker is the `CoGAlgorithm`, which defines the center-of-gravity algorithm to be used. Possible values are `FULL` which uses the full cluster size, `NPixel` for limiting the energy weighting to the N pixels with the most significant signal, and `NxMPixel` which will use a subset of the cluster with a size of $N \times M$ pixels. For all options except the full center-of-gravity algorithm additional parameters are needed which specify e.g. the submatrix size or the number of pixels to be used.

Furthermore the processor is able to perform an η function correction for the hit position (cf. Section 5.1.2). This additional correction can be enabled using the `EtaSwitch` parameter. The η functions for both the x and the y direction have to be provided in two LCIO collections which have to be loaded beforehand. For the CMS pixel telescope the full cluster center-of-gravity algorithm without η correction is used.

The alignment procedure of the telescope planes is done in two steps. The *prealignment* applies a rough shift of the telescope planes in x and y without any rotation. This can be useful to improve the overall alignment quality since the alignment algorithm which is called afterwards performs better for smaller deviations from the aligned position. The EUTelPreAlign processor provides an algorithm which calculates the prealignment shifts by comparing hit positions in the different telescope planes. This can be problematic if the data contains many random noise hits which do not belong to a particle track. This can be seen in the July beam test data prealignment presented in Section 7.5.3. The plane shifts determined by the prealignment are written to a separate LCIO database which can be specified in the processor parameters.

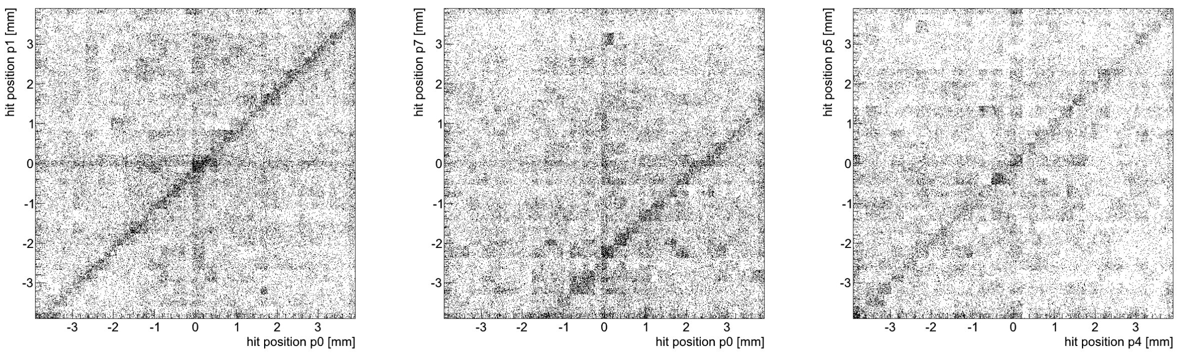
For the actual alignment the EUTelMille processor is used. EUTelMille implements the *Millepede II* algorithm (cf. Section 5.2) which performs a track-based alignment. A brief description of the implementation into the EUTelescope framework is given in [B+08]. EUTelMille is a rather complex processor with many parameters which need to be tuned. The tracks for the alignment procedure can be either generated by the processor using a simple track finding algorithm (`InputMode=0`) or read from a LCIO file created by a tracking processor (`InputMode=1`). In case of internal track finding the prealigned hits from the previous step are connected to preliminary tracks. Due to the track-based approach it is of importance to only pass candidates for real tracks to the alignment algorithm. Otherwise the alignment procedure will use randomly connected hits with a large overall χ^2 value of the track and the process will lead to randomly aligned telescope planes. EUTelMille currently provides two options to ensure this:

Residual cuts: The EUTelMille processor produces residual plots for the preliminary track fitting. These residuals can be tuned using the `DistanceMax` parameter which decides on which hits to take into account for the track fit. The residuals in both x and y for all planes can then be cut to remove outliers using the `ResidualsX/YMax/Min` floating point vectors, e.g. the lower cut-off in x direction for eight planes would be defined as:

```
<parameter name="ResidualsXMin" type="FloatVec"> 40 35 55 40 40 55 55
    40 </parameter>
```

where the eight cut-offs are ordered according to the z position of the sensors. The downside of this approach is the manual adjustment of parameters needed for every run to be aligned. The residual cuts have to be determined by hand and stored in the configuration file of the submission script. Furthermore this approach is problematic in case of high track multiplicities.

Correlation bands: Correlation bands can be used to set selection cuts for the alignment process. Only passing hits from within the correlation bands ensures that only candidates from real tracks are processed. The correlation of hit positions in different telescope planes can be visualized using the EUTelCorrelator processor. Plots for both x and y direction are created which show the correlation of hits in the respective planes as demonstrated in Figure 7.14. The plots show the hit positions in x for two planes in a 2D representation. Band structures indicate correlated hit positions while uniform distribution of data points mark uncorrelated hits in the corresponding planes. Displacements of telescope planes or a tilt of the whole telescope with respect to the beam lead to shifts of the correlation bands as indicated in Figure 7.14 (b).



(a) Planes 0, 1 (b) Planes 0, 7 (c) Planes 4, 5

Figure 7.14: Correlation of the x coordinate of hits for different combinations of telescope planes. The data has been taken from the July beam test RUN000297. A pronounced correlation band is visible among the random uncorrelated hits in all combinations. The shift of the correlation band in (b) represents the displacement of plane 7 with respect to plane 0 in x direction by $\Delta x \approx 2.5\,\mathrm{mm}$. This is not necessarily a misalignment of the telescope planes but a tilt fo the whole telescope with respect to the beam.

The selected track candidates and hit combinations are collected by EUTelMille and passed to Millepede II. The telescope planes are then shifted and rotated by the algorithm to minimize the residuals of the tracks. For telescopes with a DUT the alignment procedure is usually performed in two steps. In the first iteration the DUT is excluded from the alignment process (parameter `ExcludePlanes`) and the tracking telescope layers are aligned. The second iteration then fixes all telescope planes (with `FixedPlanes`) and aligns the DUT to the fixed telescope. However, this procedure mainly applies to stationary telescope geometries with an exchangeable DUT. With homogeneous telescopes and an analysis chain including the prealignment procedure the alignment can usually be performed in one iteration with all telescope planes. The output of the EUTelMille processor is again a collection of alignment constants which is stored in a separate alignment LCIO database.

The EUTelApplyAlignmentProcessor takes care of moving the hit positions according to the alignment. It loads the alignment LCIO database and applies the alignment constants to the hits whereby different methods can be selected which differ mainly in the order in which the rotations and shifts are applied. Furthermore the application of rotations can completely be bypassed by setting `CorrectionMethod=0`. The corrected hits are written into a new LCIO collection which can be used for track fitting.

7.3.6 Track Fitting

The final step in particle reconstruction is the track fitting procedure. Three Marlin processors with different algorithms have been studied for the CMS pixel telescope beam test. Due to the high-energy proton beam only a negligible amount of multiple scattering has been expected. Furthermore the homogeneous telescope setup with eight identical planes simplifies the fit of the particle track further since the interaction in every plane is the same in contrast to a telescope with a different DUT. In the latter case special algorithms such as General Broken Lines (GBL) [Blo06a] might be useful since they take advantage of the additional information about the different DUT and telescope detectors, e.g. in terms of material budget.

The simplest and most basic approach for track finding is the fitting of a straight line without curvature or deviations through the telescope planes. The EUTelLineFit provides such an algorithm but has not been used due to the restriction of the implementation to one track per event. With a high-rate beam and an expected track multiplicity of 6-7 (cf. Section 7.5.1) this processor would not have been capable of processing the data correctly. The successor of EUTelLineFit is the EUTelMultiLineFit processor which is able to fit multiple tracks from subsets of hits within one event. However, in the current development status of this processor only six telescope planes can be processed. This parameter has been hardcoded into the algorithm for EUDET telescope tracking applications and is non-trivial to change. Future versions might provide the possibility for tracking with any number of telescope planes.

The processor which has finally been chosen for the track fitting procedure is the EUTelTestFitter. It performs an analytical track fit as described in Section 5.3.3 taking multiple scattering into account. For high energies and momenta βcp the mean scattering angle of the particles $\Delta\theta_0$ described in Equation 5.13 becomes very small. With the difference of the track slopes $\theta_i - \theta_{i-1}$ vanishing Equation 5.14 equals a simple least squares fit without multiple scattering. The EUTelTestFitter processor provides the `Ebeam` parameter which allows the regulation of the particle beam energy in GeV.

The algorithm of EUTelTestFitter is rather simple; a description of its implementation can be found in [B$^+$07b]. The χ^2 value for track candidates formed by every hit combination within the current event is calculated, and the ones which pass the `Chi2Max` criterion are selected. However, there is a set of parameters that can be used to influence this behavior and to reduce the number of hit combinations to be taken into account. The three switches `AllowMissingHits`, `AllowSkipHits`, and `AllowAmbiguousHits` set cuts for the track candidate selection. If activated, adjustable χ^2 penalties can be added to the global χ^2 value of the track candidate for every missing or skipped hit. Correlation band information as described above can be taken into account using the `UseSlope` switch for track pre-selection. Limits on the track slope obtained from the correlation plots can be applied using `SlopeX/YLimit`. The boolean parameter `useDUT` can be used to include or exclude the DUT in the track fitting procedure. This makes it possible to create either biased or unbiased residuals for the DUT plane (cf. Section 5.3).

Two different XML templates for track fitting have been created for the beam test data analysis. Track fitting performed without the alignment constants applied to the hit positions can be invoked using the submission script with the option `tracks_noalign`. This is mainly addressed to comparison with the full procedure which applies both the prealignment and the alignment constants and can be invoked with the `tracks` option.

An instance of the EUTelDUTHistograms processor is called after the track fitting. This processor is designed for analysis of the DUT performance based on the analytic track fitting results from the EUTelTestFitter processor. Several predefined histograms are created such as the fitted and measured hit position in both x and y as well as matched and unmatched tracks from the tracking procedure. Finally the EUTelFitTuple processor writes all event data into a ROOT tree object where all events can be accessed separately by simple ROOT scripts to perform further analysis. An example of residual calculation using this ROOT tree is given in Section 7.5.4.

7.3.7 Data Quality Monitoring

A crucial part of conducting an experiment like a beam test with a limited time slot or number of repetitions is the monitoring of the data which is being recorded at the moment. This procedure is usually referred to as Data Quality Monitoring (DQM) and can involve different measures. Two different kinds of DQM can be distinguished. The *online* DQM provides information about the recorded data and the detector itself already during the data taking and allows a quick intervention. An *offline* DQM procedure is decoupled from the measurement process itself since the data is analyzed afterwards and criteria for data quality (and possible cuts) are set after the data taking has finished.

For the July high-rate beam test a *quasi-online* DQM has been set up. The hybrid test board setup with the Altera FPGA does not allow online data acquisition or monitoring but stores the readout data in the test board memory and transfers it to the DAQ PC after the run has finished (cf. Section 6.2.2). Therefore it has not been possible to perform an online DQM. The quasi-online DQM system analyzes the data directly after the transmission from the test board has finished:

- The DAQ software starts a new data taking run whenever a new SPS spill arrives (cf. Section 6.3). The run duration is $t_{\text{run}} = 7\,\text{s}$.
- The data is downloaded to the DAQ PC after the run has finished. This takes several minutes due to the small transmission bandwidth (cf. Section 6.2.2). The run data is then mirrored to a server both as backup storage and for the DQM processing itself.
- The server runs the DQM analysis process for every new run arriving. The analysis process creates histograms which are written to ROOT files.
- A script [MS12] is executed after the analysis has finished. It picks selected histograms from the files and creates a website which can then be accessed for data monitoring. The website contains histograms for e.g. the detector hit maps, the cluster size, or preliminary track residuals. All histograms are provided both as PNG and ROOT files.

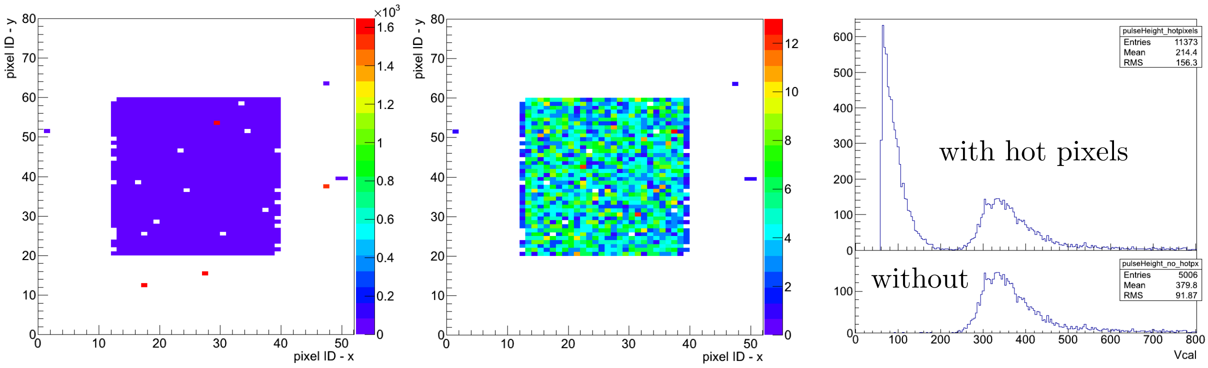
The same framework as for the data analysis is used, but in a different configuration. The submission script run option `dqm` skips some steps such as the calibration or the alignment procedure (cf. Section 7.1.4). The overall latency of this process is about 20 min with the main contribution being the data transfer from the test board memory. With the Xilinx board operational in the DAQ in the October beam test this time will be greatly reduced down to some seconds due to the Gigabit optical Ethernet connection. An example of a DQM website created during the July beam test can be found in Appendix B.1. The script for the website authoring with ROOT histograms [MS12] is contained in the project SVN repository [EUD] in the subfolder *simplesub/dqm*.

7.4 Monte Carlo Studies

Monte Carlo studies have been prepared [Lu12] in order to test the analysis chain and calibrate the different processors with their parameter settings for the CMS pixel telescope. The telescope geometry and sensors have been simulated using the *GEANT4* toolkit [A$^+$03] and samples with different track multiplicities, event numbers and telescope plane spacings as well as varying misalignments have been prepared.

Different studies have been performed on this data in order to determine the effects of parameter settings on detection efficiency of the DUT as well as the behavior of the analysis processors in extreme situations such as very high track multiplicity or many hot pixel channels. The knowledge of this helped to estimate the influence of various parameters and allowed the tuning of the processors such as the alignment process without actual telescope data from particles.

Beside general tests and parameter tunings some specific situations have been analyzed. For this purpose a dedicated converter from the Monte Carlo data into the LCIO format has been written [Tze12] with a set of options which allow e.g. the merging of events for higher track multiplicity or the simulation of limited detection efficiencies. The code of the converter can be found in the project repository [EUD].



(a) Hit map with hot pixels (b) Hit map without hot pixels (c) Pulse heights

Figure 7.15: Hit maps and pulse height distributions from a hot pixel simulation with 5000 events. (a) shows the simulation with the rectangular beam in the middle and four hot pixels appearing in the hit map. The few pixels outside the beam area originate from multiple scattering in the sensor planes. (b) shows the hit map from the same simulation after the hot pixels have been detected and removed. (c) represents the pulse height distribution of the simulation with and without the hot pixels. The left peak in the upper plot results from the hot pixels while the right peak is the actual distribution from the particle beam and present in both distributions.

7.4.1 Simulation of Hot Pixels

A parameter for the simulation of hot pixels has been introduced in the Monte Carlo data converter. With `PHotChannel` a certain percentage of hot pixels in every telescope plane can be introduced into the data converted from the *GEANT4* simulations. Usually no hot pixels should occur with a calibrated ROC and thresholds correctly set. However, hot pixels rarely appear during data taking. If more than a few pixels turn out to be hot, the threshold settings have to be checked and the ROC recalibrated. A percentage of `PHotChannel=0.2%` results in about 8 hot pixels per ROC in the telescope which already overestimates the expected situation.

Histograms from a simulation with these settings are shown in Figure 7.15. The Monte Carlo simulation contains 5000 events with one track each. This should result in a total of about 40 000 clusters in the telescope, or 5000 per plane. Since the simulation has been performed with the straight telescope geometry most of the clusters contain only a single pixel. In Figure 7.15 (a) the four random hot pixels introduced by the converter together with the particle tracks produce about 11 000 clusters within this single ROC. Not only in the hit map but also in the pulse height distribution the hot pixels create a distinct signal as shown in Figure 7.15 (c). The EUTelHotPixelKiller (cf. Section 7.3.3) identifies these hot pixels by measuring their firing frequencies. Since the Monte Carlo converter logs the IDs of the randomly selected hot pixels it is possible to compare the results from the hot pixel removal to the truth. It has been shown that up to a unrealistically high fraction of hot pixels of `PHotChannel=20%` almost no clusters are lost. All hot pixels have been identified correctly. The subsequent processor (in the CMS pixel telescope case the CMSPixelClusteringProcessor) excludes the hot pixels from further analysis steps. The hit map which is shown in Figure 7.15 (b) represents the same simulation run with hot pixels after their removal by the CMSPixelClusteringProcessor.

With the tilted telescope geometry where multi-pixel clusters are expected the impact of hot pixels to the clustering is even weaker since no complete clusters can be obscured by a continuously firing pixel. Excluding them only leads to a slightly shifted cluster center. In the July beam test no hot pixels have been encountered and the hot pixel detection mechanism has been switched off in the configuration of the submission script for analysis.

7.4.2 DUT Detection Efficiency

One of the main objectives of the high-rate beam test is the investigation of the efficiency of the new *PSI46dig* ROC at high rates. This efficiency can be measured by calculating the detection efficiency of one ROC which has been operated as Device Under Test. The comparison of the particle tracks with the hits recorded by the DUT allows conclusions regarding buffer overflows and missing hits. The parameter `Efficiency` has been introduced in the Monte Carlo data converter to enable the simulation of different detection efficiencies of the ROCs. The determination of the detection efficiency of the DUT is not quite simple because every telescope plane consists of the same detector and is therefore subject to the same inefficiencies.

In general the DUT detection efficiency can be determined by

$$\epsilon = \frac{N_{\text{tracks+DUT}}}{N_{\text{tracks}}}, \tag{7.3}$$

where N_{tracks} is the number of tracks which could be reconstructed either from Monte Carlo or recorded data, and $N_{\text{tracks+DUT}}$ is the number of tracks which can be matched to a hit on the DUT at the track penetration point. However, this depends on the requirements for a track through the tracking planes of the telescope, for example whether hits from all seven planes of only from six out of seven (*6-of-7*) are required. The exact tracking mode has not been chosen yet. The matching rule for one dimension has been arbitrarily set to

$$x_{\text{fitted}} - x_{\text{measured}} = \Delta x < 3 \cdot \frac{p}{\sqrt{12}}, \tag{7.4}$$

where x_{fitted} is the penetration point of the fitted track, x_{measured} the prealigned measured hit position and p the pixel pitch in the coordinate considered. Therefore a measured hit is counted as *matched* if the fitted position lies within three times the binary position resolution (cf. Section 2.3.3). This parameter has to be tuned according to the resolution obtained in the data.

The detector efficiency has only been studied with the Monte Carlo simulations since no valid detector data is available for analysis. Due to readout and data quality problems (cf. Section 7.5.2) the data recorded in the July beam test cannot be used to determine the ROC efficiency. Sensible settings for the tracking mode and hit matching in the DUT plane have to be determined from the data to be recorded in the October beam test. Furthermore it might be worth considering the usage of every telescope plane as DUT. Currently the DUT is arbitrarily set to layer 4 of the telescope but interchanging the DUT layer and calculating the efficiency for every telescope plane might lead to more precise results. Finally, the impact of different tracking configurations on the calculated efficiency of the DUT has to be studied in detail and a closure test with Monte Carlo data relating the input efficiencies of all ROC with the tracking efficiency has to be performed.

7.5 Data Analysis

During the July beam test 73 valid data runs with one Super Proton Synchrotron (SPS) spill (cf. Section 3.3) each have been recorded. This translates to about 800 000 events with 3×10^7 pixel hits after an event selection cut with `evtSel=2` (exclude faulty events, cf. Section 7.3.1). All data has been recorded using the internal random trigger generator on the Altera test board (cf. Section 6.2.2) and the DAQ runs have been limited to the spill duration. Furthermore several runs for trigger latency scans have been performed with external triggers. The telescope and its ROCs could be addressed, programmed and read out using the hybrid test board setup. The first high-rate beam test has mainly been used to characterize the proton beam using the telescope and its periphery such as the scintillators and the motor stages. The suitability of the beam area for a high-rate beam test of tracking detectors has been evaluated.

Still some problems need to be solved for the next beam test. Difficulties were encountered concerning the pulse height calibration of many ROCs without TBM and a new calibration software framework is currently being prepared for the October beam test. Other tests such as the *Pixel Alive* test, which activates each PUC after the other and checks for the according data bits in the detector readout, or the *Address Decoding* test to check a correct address encoding could be performed. The output of the *psi46expert* calibration suite for these calibration and tuning tests can be found in Appendix C. Furthermore the hybrid test board setup has not been optimal and several problems related with clock and signal phases occurred. Due to the resulting major problems concerning the data quality of the readout none of the planned DUT analyses have been carried out yet (see Section 7.5.2). Most of these problems might be solved by using the Xilinx SP605 test board as sole readout board with a firmware fully functional.

In the following sections the evaluation of the data taken in the July beam test is presented. This includes calculations on the particle rate and fluence (Section 7.5.1) as well as a detailed investigation of the readout bit errors which occurred (Section 7.5.2). Furthermore the telescope alignment (Section 7.5.3), DUT residuals calculations (Section 7.5.4), and event inspection with CED (Section 7.5.5) are presented.

7.5.1 Particle Rate, Fluence, and Sensor Irradiation

One of the main goals for the July beam test was the qualification of the beam area for a high-rate beam test. Since the *H4IRRAD* configuration is usually used for irradiation studies (cf. Section 3.3) the suitability for tracking detector tests had to be ensured. One of the main concerns has been the particle rate which can be reached. This has been studied in the July beam test using the information from the sensor hit maps, the triggers, and the Gafchromic film (cf. Section 6.4). The particle rate r_i incident on telescope plane i can be calculated as follows:

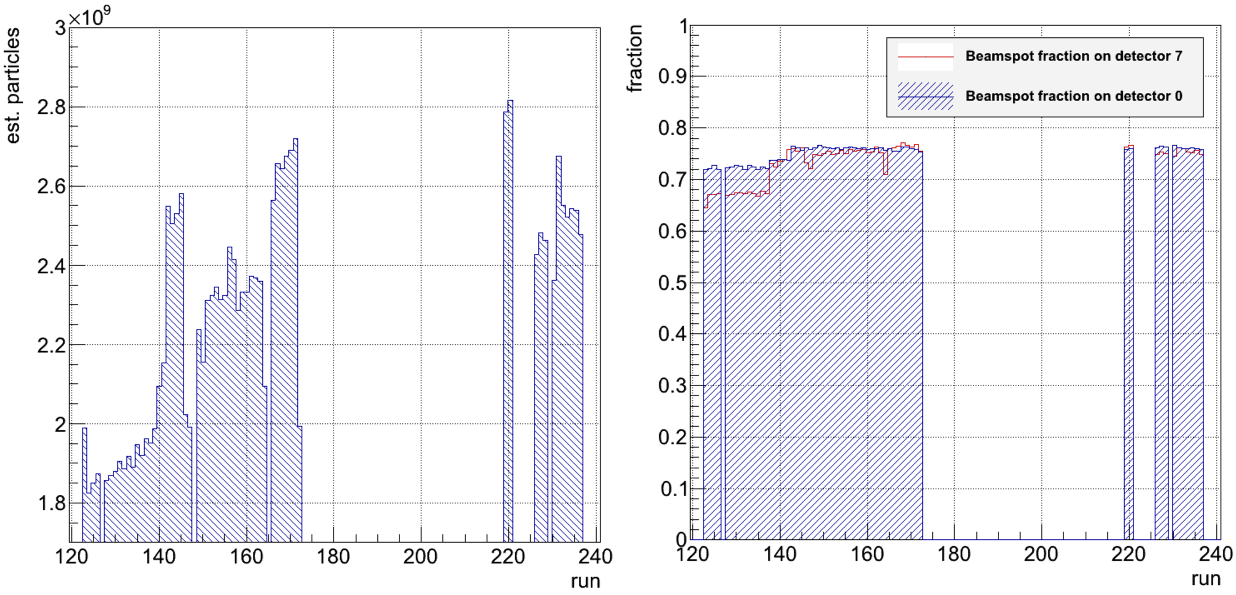
$$r_i = \frac{(N_p \cdot f_{\mathrm{sp}} \cdot f_{i,\mathrm{bs}})}{\Delta t}, \tag{7.5}$$

where N_p is the total number of protons in the spill recorded by the *H4IRRAD* monitoring detectors (XH4.XION, cf. Section 6.4). The fraction of triggers which has been recorded by the DAQ with respect to the total spill count is given by f_{sp}, while $f_{i,\mathrm{bs}}$ gives the area fraction of the beam spot visible on the hit map of detector i. The run duration $\Delta t \approx 7\,\mathrm{s}$ has been constant for all runs. The uncertainty on the particle rate is dominated by the large uncertainty on N_p measured by the ionization chamber (see below).

The total number of particles per spill can be determined using the beam line monitoring system TIMBER (cf. Section 3.3). The numbers obtained have to be calibrated using the linear function

$$N_p = m \cdot N_{\mathrm{ion}},$$

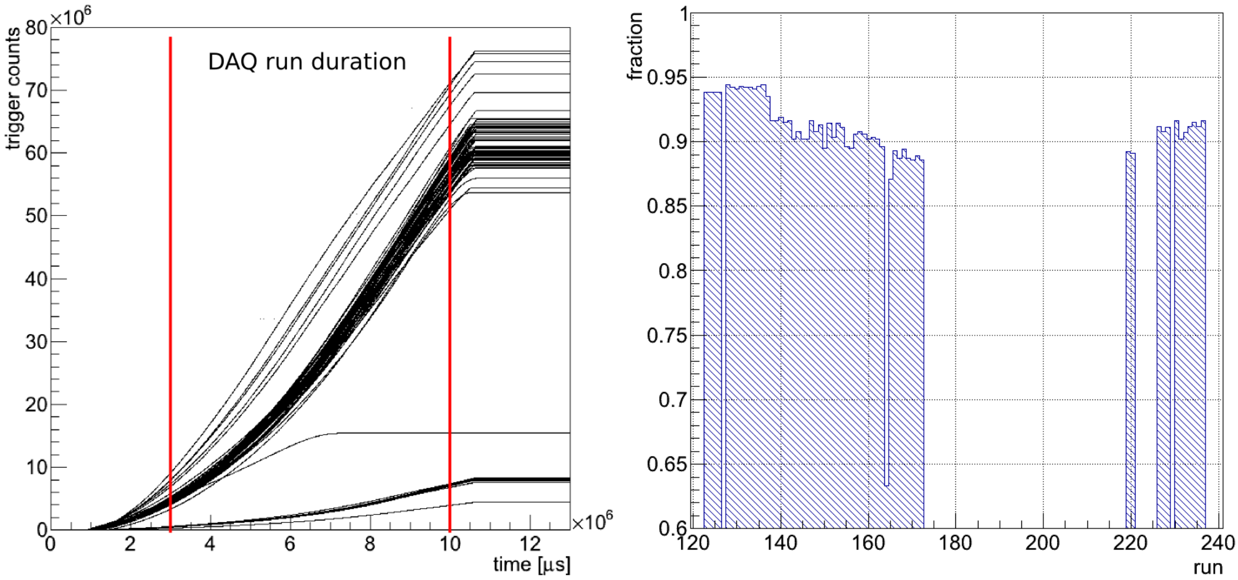
where N_{ion} is the count provided by the XH4.XION ionization chamber and $m \approx 6300$ is the calibration factor. The chamber counts have been calibrated at proton rates up to 10^8 protons per spill using two scintillators of the beam line instrumentation as described in [B+12c] and the linear fit obtained has been extrapolated to higher rates. Furthermore the calibration varies significantly over time and the factor chosen is only an average of several calibration measurements and has been suggested by the beam line operators. Therefore the total number of particles and the subsequently calculated values can only serve as rough estimates.



(a) Calibrated particle count N_p

(b) Beamspot area fraction on the sensor

Figure 7.16: (a) Particle count N_p of the XH4.XION ionization chamber per data taking run of the CMS pixel telescope. The counts obtained from TIMBER have been calibrated using the calibration factor $m \approx 6300$ [B$^+$12c]. (b) Fraction of the beamspot area visible on the hit map of the detector planes $f_{i,\mathrm{bs}}$ as function of runs. This fraction has been obtained by fitting the fixed-sigma Gaussian distribution extracted from the Gafchromic film to the hit maps of the detectors for every run.



(a) Trigger counts for (A0 + A1)

(b) Recorded spill fraction f_{sp}

Figure 7.17: (a) Total counts of the front scintillator with the two PMTs in conicidence as function of time. Shown are all spills which have been successfully recorded by the detector. The DAQ run duration is marked in red. (b) Fraction of triggers f_{sp} within the DAQ run time with respect to the total trigger count. The fraction has been calculated for all recorded runs.

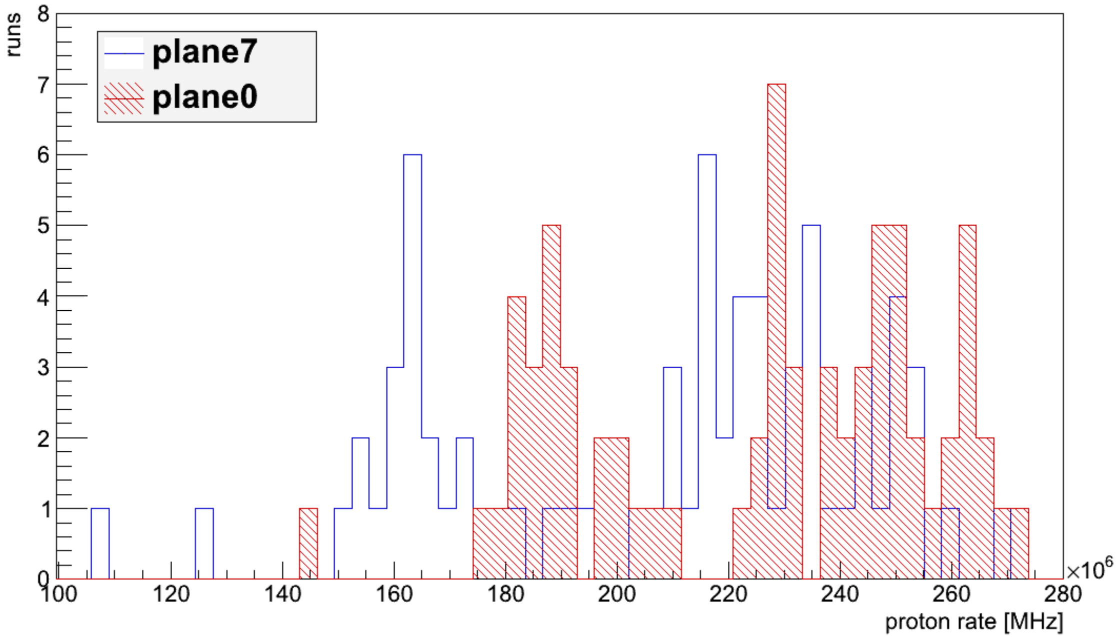


Figure 7.18: Number of runs with a certain rate of protons incident on the detector planes calculated from Equation 7.5. A total of 73 runs is shown. Two positions of the telescope can be distinguished by looking at the mean proton rate per run. The lower rate is around 190 MHz, the higher one at 250 MHz. These rates proof the suitability of the beam for high-rate measurements. From the different rates for the two planes it can be seen that the telescope has been slightly tilted with respect to the beam.

The estimated total number of particles per data taking run is shown in Figure 7.16 (a). The shape and size of the beam spot has been calculated by scanning, converting, and fitting the Gafchromic film with a 2D Gaussian distribution (cf. Section 6.4). The values for σ_x, σ_y as well as the rotation angle have been determined from the fit [Mer12]:

$$\begin{aligned} \sigma_x &= 4.617(3)\,\mathrm{mm} \\ \sigma_y &= 0.958(2)\,\mathrm{mm} \\ \theta &= -25.29(1)^\circ \end{aligned}$$

The Gaussian distribution with the fixed values for σ_x and σ_y obtained from the film has then been fitted to every sensor plane hit map produced by the DQM processor. By comparing the area on the sensor to the full beamspot the fraction of beamspot area $f_{i,\mathrm{bs}}$ covering the sensor surface has been determined for each run separately as shown in Figure 7.16 (b). The uncertainty on the beam spot fraction can be neglected compared to the uncertainty on the total number of protons.

Since a DAQ run only covers part of the SPS spills the fraction of particles within the DAQ run window f_{sp} has to be calculated for every spill. This is possible using the trigger counts recorded by the Trigger Server (cf. Section 6.3) which are shown in Figure 7.17 (a). A new DAQ run has been started about 3 s to 3.1 s after the arrival of the Warning Warning Extraction (WWE) signal from the SPS and every run lasted $t_{\mathrm{run}} = 7\,\mathrm{s}$. The deviation between a DAQ time slot starting at $t_{\mathrm{start}} = 3\,\mathrm{s}$ and at $t_{\mathrm{start}} = 3.1\,\mathrm{s}$ is only about $\Delta N_{\mathrm{trig}} \leq 0.5\,\%$ and can be neglected. The fraction of the trigger count within the DAQ window and the total trigger count gives the fraction of the spill that has been recorded as shown in Figure 7.17 (b). The data used for the calculation of the particle rate is collated in Table C.1 of the Appendix, exemplarily for planes 0 and 7.

Table 7.1: Leakage current I_{leak} and mean temperature of the eight CMS pixel telescope ROCs during the July beam test. The leakage current has been measured both without proton beam and during spills.

time	I_{leak} off-spill	I_{leak} on-spill	temperature
Jul 31, 9 am	1.6 µA	–	29 °C
Jul 31, 6 pm	3.3 µA	6.0 µA	32 °C
Aug 01, 0 am	15 µA	35 µA	32 °C
Aug 01, 1 am	20 µA	40 µA	30 °C
Aug 02, 11 am	28 µA	–	30 °C
Aug 02, 10 pm	50 µA	65 µA	32 °C
Aug 03, 2 am	54 µA	79 µA	32 °C

Figure 7.18 shows the proton rates r_i incident on the sensor planes of the CMS pixel telescope. Two positions of the telescope can be distinguished by their mean proton rates of $r_0^{\mathrm{low}} \approx 190\,\mathrm{MHz}$ and $r_0^{\mathrm{high}} \approx 250\,\mathrm{MHz}$ which are incident on the first telescope plane. The particle flux can be calculated from this taking the sensor area into account:

$$f_i = \frac{r_i^{\mathrm{high}}}{A_{\mathrm{sensor}}} \approx 400\,\mathrm{MHz/cm^2}, \tag{7.6}$$

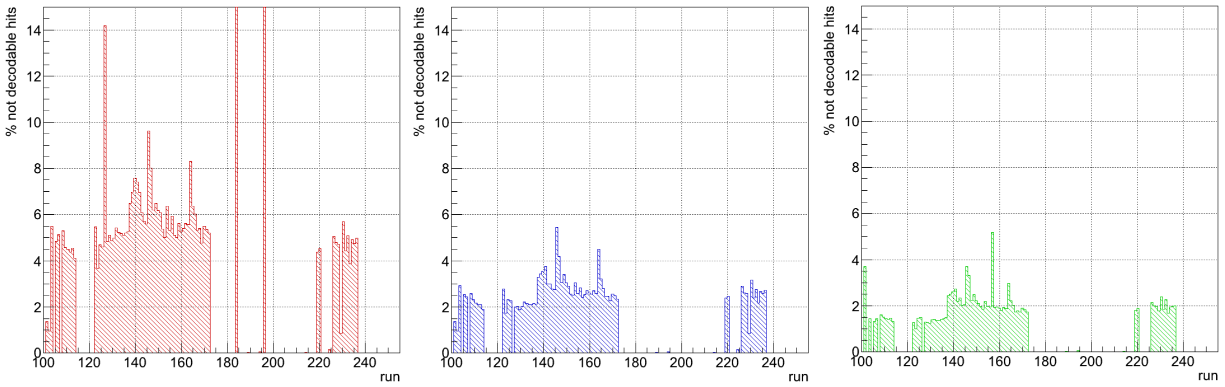
with the area of one single ROC sensor $A_{\mathrm{sensor}} = 8.0\,\mathrm{mm} \times 7.8\,\mathrm{mm} = 62.4\,\mathrm{mm}^2$ (cf. Section 4.4.2). The track multiplicity, i.e. the number of particle tracks per event can be estimated as the ratio of particle rate and timing resolution of the detector:

$$n_{\mathrm{tracks}} = \frac{\bar{r}}{\nu} = \frac{250\,\mathrm{MHz}}{40\,\mathrm{MHz}} \approx 6, \tag{7.7}$$

where $\bar{r}$ is the average particle rate for the telescope position with higher counts and ν the timing resolution of the detector. Each trigger validates one time slot of $\Delta t = 1/\nu = 25\,\mathrm{ns}$ which is then read out. This track multiplicity agrees with the estimated number calculated from simulations and is suitable to test the high-rate behavior of the *PSI46dig* ROC.

The total fluence of the CMS pixel telescope from the July beam test has been estimated to be 4×10^{13} protons with an energy of 320 GeV [Ell12]. This estimation is based on the proton counts provided by the ionization chamber within TIMBER. The sensor leakage current has not been monitored continuously but some values have been recorded during the July beam test. Before the first irradiation the leakage current of the whole telescope (all eight ROCs) was $I_{\mathrm{leak}} = 1.6\,\mu\mathrm{A}$ at a temperature of about 31 °C and with a bias voltage of 80 V applied. The measurements are listed in Table 7.1 and seem to agree with the expected rise due to the gradually increasing radiation damage in the sensor material.

Simulations show that the sensors can probably be operated up to a fluence of 8×10^{13} protons before they will be subject to thermal runaway [Ell12]. The increased leakage current heats up the sensor material and due to the lack of cooling the increasing temperature causes a further rising of the leakage current. Therefore the preparation and bonding of new sensors and ROCs is required for further measurements. Furthermore continuous logging of the sensor leakage current is planned for the October beam test.



(a) `evtSel=0` (b) `evtSel=1` (c) `evtSel=2`

Figure 7.19: Percentage of hit decoding errors in the data streams from the July beam test runs. Runs below 100 are not usable due to wrong clock phase settings in the test boards.
(a) Errors for decoding with `evtSel=0` including malformed ROC headers.
(b) Reduced error rate, headers according to specification (`evtSel=1`, cf. Section 7.2.3).
(c) Result for decoding with `evtSel=2`.

7.5.2 Decoding Errors

One of the main issues in the July beam test is the quality of the recorded data. The raw data contains many severe bit errors compromising the overall data quality. Figure 7.19 shows the percentage of pixel hits which contain bit errors and therefore cannot be decoded. Only runs which have been used for analysis are shown. Depending on the selection criteria set in the decoding process (cf. `evtSel` in Section 7.3.1) up to 6 % of all pixel hits from one run fail to be translated into x and y coordinates and a pulse height value. With such a high rate of defects in the stream it is impossible to draw any conclusions concerning efficiencies of the ROC itself. Runs 0–64 are not analyzed due to wrong input data phase settings in the Xilinx board resulting in undecodable data.

In the following section the different defects are characterized. All examples are taken from the July beam test RUN000144 if not noted otherwise but the problems occur in the same manner in all other runs as well. Again, all numbers denoted with `0x` are given in hexadecimal notation. Numbers starting with `0b` follow the binary system notation.

ROC header patterns inside hits

Due to the flaw in the *PSI46dig* chip logic mentioned in Section 7.2.3 the pixel ID is transmitted with inverted bits. This allows the occurrence of ROC header sequences within the pixel hit section of an event and complicates a reliable ROC header detection. The following example includes two pixel hits within the eight ROC headers; the second one contains the ROC header sequence `0x7f8`:

```
0x7f8 0x7f8 0x7f8 0x7f8 0x873e88 0x87f868 0x7f8 0x7f8 0x7f8 0x7f8
```

The hit containing the ROC patterns can be correctly decoded demonstrating that this is not a bit error. The ROC header is marked in red.

```
0x87f868 = 0b100001111111100001101000
```

The inverted pixel ID `000 000 011` translates to `003` in decimal notation which is the last upper pixel on the right side of the corresponding double column (cf. Section 7.2.4). However, the decoding strategy of the CMSPixelDecoder has been chosen carefully and the decoder is not sensitive to ROC headers which appear at positions within the 24 bit hit sequences (cf. Figure 7.8).

The occurrence of ROC headers at the beginning of pixel hit sequences caused by this effect can be excluded. The double column is sent correctly by the ROCs and the maximum values of the three-bit packages for the two digits of the `dcol` ID are constrained by the set of address levels from the analog *PSI46v2*. The encoding of the pixel address has not been changed and hence only six address levels with 5 being the highest number are used. A double column ID mimicking the ROC header patterns would start with `0b011111` which translates into `C1=3, C0=7`, where the `C0` value cannot be reached. However, the possibility of a random bit error creating such patterns is still present.

Unspecified ROC headers

Beside the ROC header sequences specified for the *PSI46dig* other patterns occur within the datastream of the recorded events which are not in accordance with the specification:

```
0x7f0 0x3f8 0x3f9 0x3fa 0x3fb
```

In most cases the rising edge of the signal is cut off or arrives too late resulting in a sequence `0011` starting with two zeros. This can either be due to a bit error in the data stream or an inaccuracy on the ROC and has to be investigated in the October beam test when the general readout problems are solved. A simplified example from the beam test data demonstrates the problem:

```
0x7f8 0x6236aa 0x3f80ac 0xb01262 0x3017f8 0x42d68a
0x7f8 0x7f9 0x7f8 0x7f8 0x7f8
```

The first ROC header marked in red would not be detected according to the specification and all hits between this unidentified header and the next ROC header would be lost. Furthermore the overall event then contains only 6 instead of the expected 8 ROC headers which would confuse the telescope plane allocation to the ROCs. This can be avoided by allowing the decoder to identify these headers beyond specification. However, since this deformation might be connected to general bit errors, the `evtSel=2` switch only accepts events with ROC headers within specifications and rejects events with malformed header sequences completely.

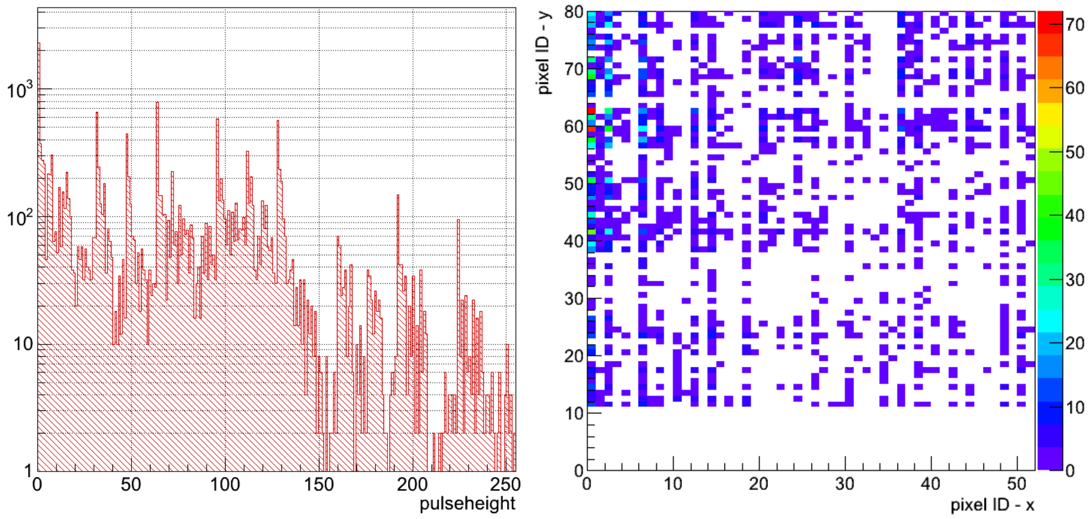
Failed Pixel Address Decoding

Some pixel hits cannot be decoded due to bit errors within the double column, the pixel ID or the zero bit which divides the pulse height to prevent too many successive ones. These bit errors occur without any regularity and cannot be corrected. Figure 7.20 (a) shows the pulse height distribution of pixel hits which failed to be decoded. This distribution indicates that the bit errors do not only occur in the pixel address but also in the pulse height; otherwise a Landau-shaped distribution would be expected. An example of a pixel hit with corrupted address is `0x51122f`, the binary notation with inverted pixel ID bits is:

```
0x51122f = 0b 011 100 101 110 110 0001 0 1111 = DCOL 22 PIX [5,6,6]
              CCC CCC RRR RRR RRR PPPP - PPPP
```

While the double column bits can be decoded and result in `DCOL 22`, the pixel ID contains a bit error which produces a pattern not listed in the specification Gray code tables [Gab05]. Since it has not been clear if the flaw with inverted pixel ID affects all pixels (cf. Section 7.2.3) it has been checked whether the address decoding could fail due to a non-inverted pixel ID sent by the ROC. The previous example can be decoded when not inverting the pixel ID bits:

```
0x51122f = 0b 011 100 010 001 001 0001 0 1111 = DCOL 22 PIX 41-L
              CCC CCC RRR RRR RRR PPPP - PPPP
```



(a) Pulseheight distribution

(b) Hit map of non-inverted IDs

Figure 7.20: Pixel hits with corrupted address, extracted from July beam test RUN000144 and processed with `evtSel=1`. (a) Pulse height distribution from all pixels with failed address decoding. (b) Hit map of telescope plane 3 with all pixel hits which can be decoded using non-inverted pixel ID.

and the decoded pixel ID represents the pixel 41 in the left column (cf. Section 7.2.4). For verification every pixel hit that failed during normal address decoding has been re-checked using a non-inverted pixel ID. For example in RUN000144 of the July beam test 18 000 of 31 000 failed pixel address decodings can be recovered using a non-inverted pixel ID and `evtSel=1`. The hit map with the recovered pixel hits shown in Figure 7.20 (b) contains arbitrary patterns instead of a beamspot shaped area which would be expected for valid data. This most likely indicates that the successful decoding with non-inverted pixel IDs works only by chance and does not contain any valid data.

Incorrect number of ROC headers per event

Some events contain fewer ROC headers than expected. Mostly this incorrect number of headers originates from a cut-off header at the beginning of the event as shown in the following example:

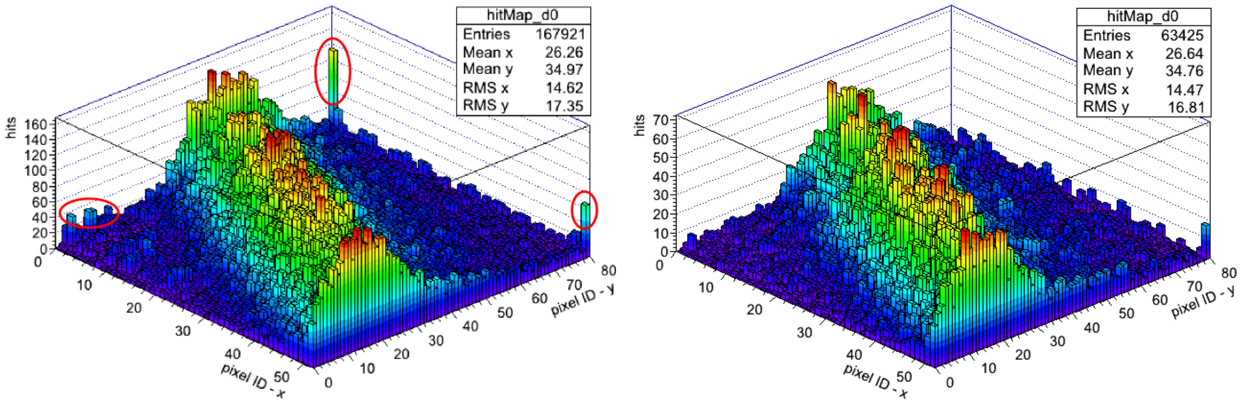
```
0x_f8  0x22fe85  0x22748a
0x7f8  0x167cc3  0x3f80f3  0x6e0527  0xace527  0xe847f0  0x8030c4
0x7f8  0x7f8  0x7f8  0x7f8
```

Even though it would be possible to reconstruct the first ROC header based on the 8 bit which are left from the sequence, these events are rejected with `evtSel=1`. This does not affect the total number of events too much, e.g. in RUN000144 only 1137 events out of 24 798 contained only seven ROC headers. With `evtSel=0` the decoder returns invalid data, because the decoding algorithm cuts off all data before the first ROC header which can be correctly detected (cf. Section 7.3.1). In this case ROC1 becomes ROC0 and the last plane is missing and filled with an empty one.

Sometimes events without a single ROC header occur. In most cases these events contain only zeros (marked in red) which are repeated several thousand times. The reason might be a missing `TOKEN_OUT` signal from the telescope and the test board is not able to detect the end of the readout:

```
[0x0000] 0x003232 0xe2e366 0xf6f6f8
```

Events without ROC headers are rejected independent of the `evtSel` setting.



(a) Decoded with `evtSel=0`

(b) Decoded with `evtSel=2`

Figure 7.21: Decoding artifacts in hit maps from `evtSel=0` decoding runs. Both histograms have been produced using RUN000298 from the July beam test data. (a) shows the hit map with the beam spot and some pronounced artifacts. These can be removed using `evtSel=2` as shown in (b).

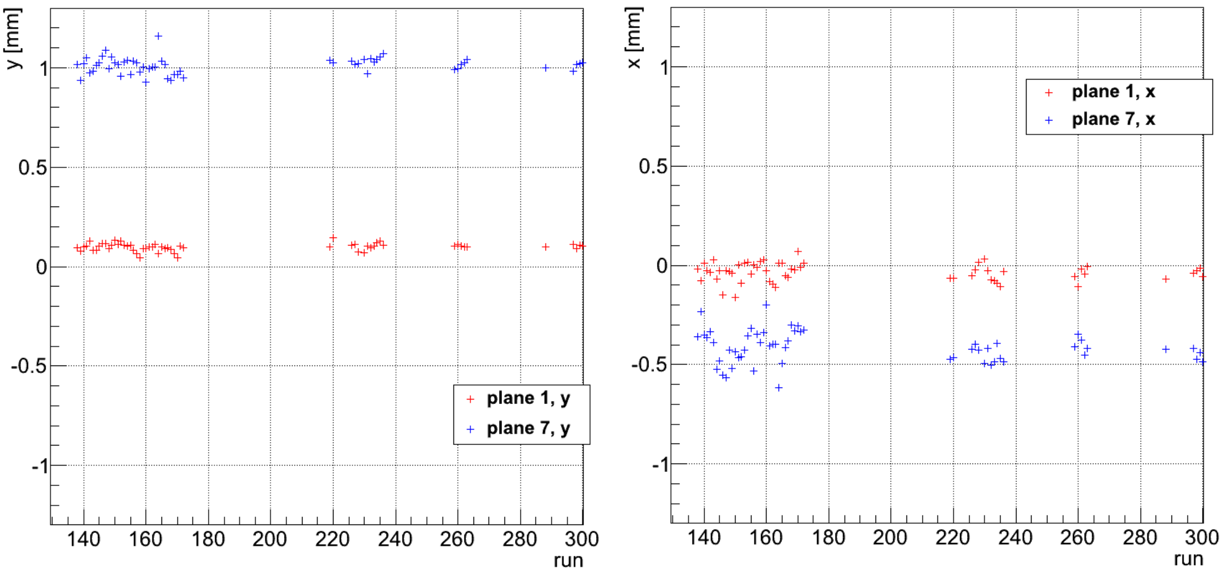
To conclude, the `evtSel` event selection criterion is very useful to select only valid pixel hits and telescope data when dealing with many readout and bit errors. Most of the decoding artifacts in hit maps which can be observed with the `evtSel=0` setting can be removed using `evtSel=1` or `evtSel=2` as shown in Figure 7.21. With this, the rate of failed pixel hit decodings can be reduced to about 2 %. Whether the very strict `evtSel=2` setting is really necessary and improves the decoding results noticeable has to be confirmed. Most of the bit errors probably originate from the hybrid test board setup. The two FPGA clocks and the phase setting for the data sampling has not been stable during the July beam test. It remains to be seen in the October beam test if the readout with the Xilinx SP605 board and the IPBus interface is more reliable and less prone to errors.

According to the latest measurements with the fully implemented Xilinx firmware and the new DAQ framework corrupted ROC headers can still be seen in the data stream from the detector, even when only addressing a single ROC. This could indicate that at least this problem originates from the ROC itself and is not part of general readout problems. This has to be cross-checked with more data taken with the new setup under beam conditions in the October beam test.

7.5.3 Telescope Alignment

The alignment of the telescope in the beam has been studied using both the prealignment processor output and the alignment process. Figure 7.22 shows the prealignment constants for the runs 137 to 301 which have been used for data analysis. In Figure 7.22 (a) the prealignment constants in y direction for planes 1 and 7 are shown. It can be observed that the prealignment is quite stable over the data taking time. The prealignment of the telescope planes in x direction is less stable and is subject to greater variation as shown in Figure 7.22 (b). The reason for this is not yet understood.

The average alignment constants for all telescope planes have been calculated and are shown in Figure 7.23. This includes both the prealignment done by matching hit positions as well as the track-based alignment approach from the EUTelMille processor (cf. Section 7.3.5). The internal simple track fit algorithm of the EUTelMille processor has been used to calculate the alignment constants. The errors indicate that the variation among different runs is quite large. Especially the track-based alignment introduces large uncertainties. Since the data in the July beam test has been taken with local random triggers, the large variations might originate from noise hits which have been connected as tracks by the internal simple track finder.



(a) Prealignment in y, plane 1,7

(a) Prealignment in x, plane 1,7

Figure 7.22: Prealignment constants for the July beam test runs which have been selected for data analysis. (a) shows the prealignment constant in y direction for plane 1 and 7 of the telescope. It is stable over all runs and has only small deviations. (b) shows the prealignment constants for plane 1 and 7 in x direction. It can be observed that the x prealignment is much more coarse than the prealignment in y.

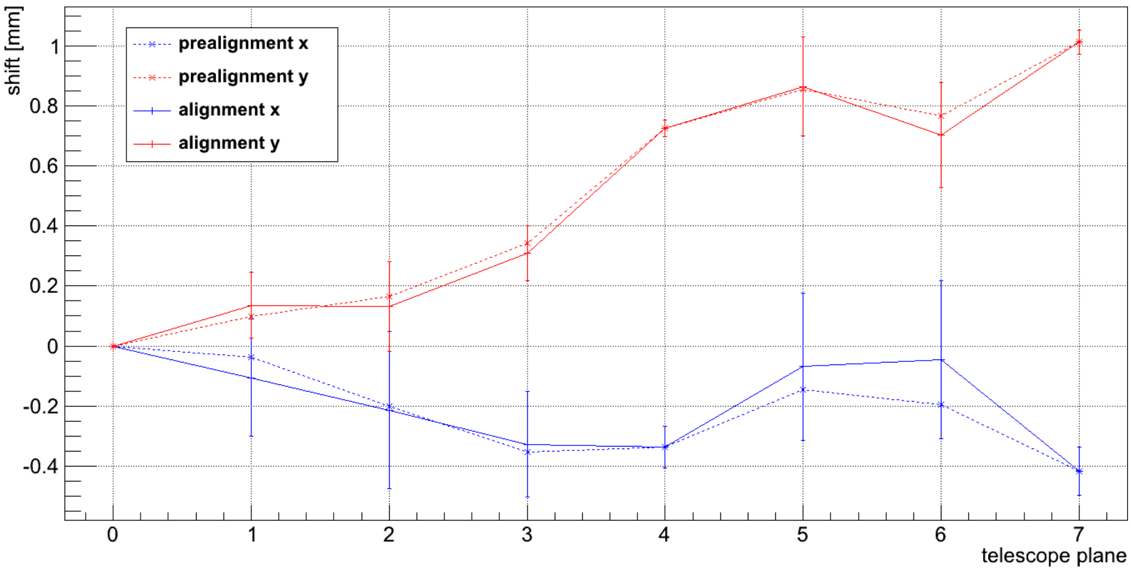
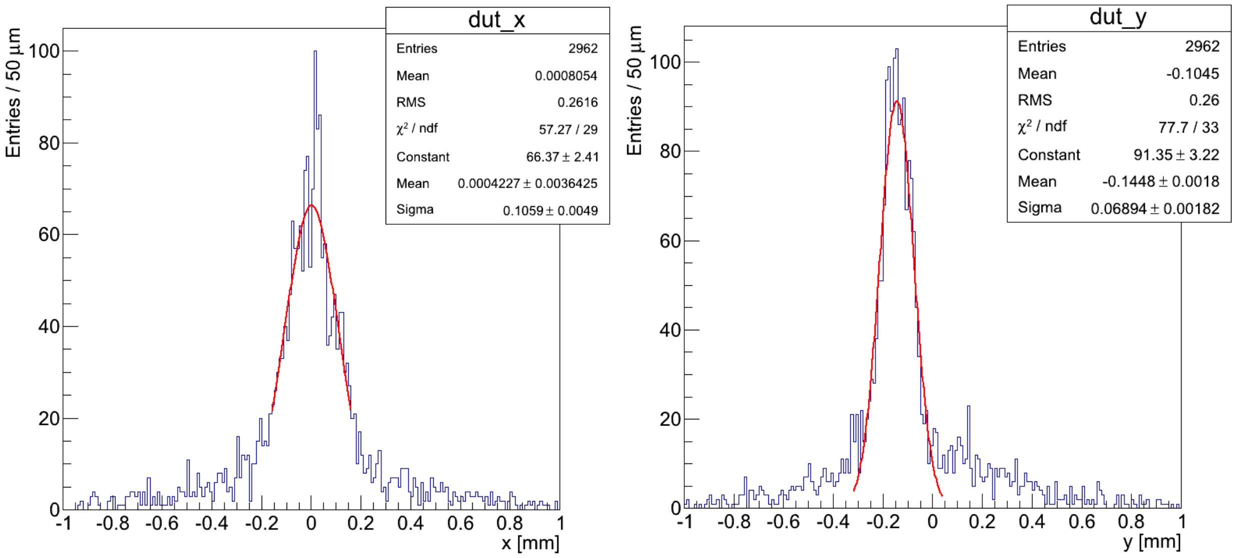


Figure 7.23: Average prealignment and alignment constants in x and y for all telescope planes. Plane 0 is fixed and plane 4 as DUT is excluded from the alignment process. The dashed line represents the prealignment corrections while the solid line contains both prealignment and alignment corrections. The errors shown are the combined RMS deviation from alignment and prealignment.



(a) DUT residual in x

(b) DUT residual in y

Figure 7.24: Unbiased residuals of the DUT telescope plane, calculated from July beam test data RUN000157. (a) shows the residual in x with a fitted Gaussian distribution and $\sigma_x = 0.106 \pm 0.005\,\text{mm}$. (b) is the residual in y with $\sigma_y = 0.069 \pm 0.002\,\text{mm}$. The residual in y is slightly off-center due to an uncorrected misalignment of the DUT. Both standard deviations are slightly larger than the resolution expected for 1-pixel clusters in x and y, respectively.

To improve this situation it is necessary to use external triggering so that every event contains valid particle tracks. Furthermore it might be useful not to rely on the simple track fit within EUTelMille but take advantage of the possibility to feed in externally fitted tracks (`InputMode=1`, cf. Section 7.3.5) and use EUTelTestFitter to perform the actual track fitting even for the alignment procedure. This is currently not possible due to a defect in the EUTelMille processor but could help to improve the precision of the overall telescope alignment and reduce the variations.

7.5.4 DUT Residuals

The unbiased residuals for the DUT telescope plane have been calculated according to Equation 5.6. Figure 7.24 shows an example from the July beam test RUN000157. The unbiased residuals for the DUT are plotted for both orientations. In x direction the residual has a standard deviation of $\sigma_x = 0.106 \pm 0.005\,\text{mm}$. This lies within the expected range with the pixel pitch in x being 150 µm (cf. Section 4.4.2). The slightly smaller pitch of 100 µm in y results in a smaller standard deviation of the residual of $\sigma_y = 0.069 \pm 0.002\,\text{mm}$. Since only single pixel clusters have been produced by the proton beam which has been incident perpendicular to the sensor surface, the hit position cannot be determined more precisely using the center-of-gravity energy weighting method. Therefore the position resolution is not expected to be smaller than the binary position resolution of $p/\sqrt{12}$ which can be calculated to $\sigma_x = 0.043\,\text{mm}$ and $\sigma_y = 0.029\,\text{mm}$. Here, p denominates the pixel pitch in x and y, respectively (cf. Section 2.3.3). This resolution is not reached due to the limited alignment precision in the data recorded during the July beam test.

With the tilted telescope geometry which will produce multi-pixel clusters and higher statistics the position resolution will improve in the analysis of the October beam test data.

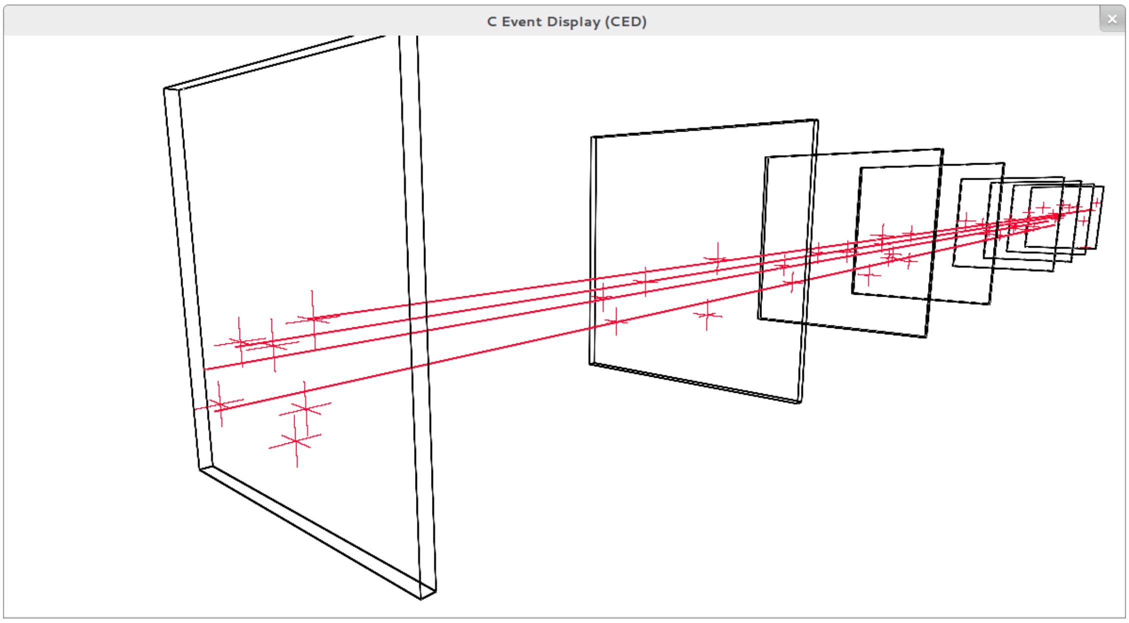


Figure 7.25: Event display provided by CED showing an event from the July beam test RUN000297 with four particle tracks through all telescope planes. Additionally some random noise hits which are not matched to any track are visible.

7.5.5 Event Inspection with CED

The C Event Display (CED) is a client - server based graphical user interface which can be used to display single events and cross-check e.g. the track fit. It uses the OpenGL graphics system [SA12] to provide 3D perspective views with fast rendering, free rotating, and zooming of the display. To display events from a LCIO file a CED server instance has to be provided which is invoked with the `glced` command. The EUTelescope Marlin processor EUTelEventViewer can then be executed in a second terminal providing the LCIO file to display. The processor will automatically detect the CED server and connect to it. The next event is displayed by hitting the return key. The user manual for CED can be obtained from [Hoe11].

Especially with a newly prepared analysis chain it can be very useful to review the outcome of the tracking process by eye. Event displays such as CED provide the possibility of a fast and easy inspection of event data. The implementation into the ILCsoft framework with the CEDViewer which directly reads LCIO data simplifies this even further. Figure 7.25 shows an example event from RUN000297 from the July beam test data after the full analysis procedure. Shown are four particle tracks through the telescope together with some noise hits. By rotating and zooming the detector view in the display the accordance of the single hits with the corresponding track can be checked.

In summary, the July beam test mainly provided the possibility to characterize the beam and *H4IRRAD*. The beam line has been qualified for high-rate proton beam tests with particle rates of up to 400 MHz/cm^2. Furthermore the EUTelescope analysis framework and its configuration for the CMS pixel telescope have shown their capability of tracking particles and first residuals of the DUT have been calculated. The position resolution is still below the expected resolution from binary readout. This mostly originates from insufficient alignment due to low statistics. The main concern is the data quality and the bit errors occurring in the data stream. They probably originate from the hybrid test board setup and not from the ROCs themselves but this has to be confirmed with the October beam test data.

The wheel is come full circle.
William Shakespeare

8. Summary and Prospects

The detectors at the Large Hadron Collider (LHC) are designed to cope with the LHC design luminosity of $\mathcal{L} = 1 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [B+04]. However, with the gradually increasing instantaneous luminosity the demands on the detectors are continuously rising and the design luminosity will probably be exceeded even before the so-called Phase I Luminosity Upgrade. At double design luminosity of $\mathcal{L} = 2 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [Zim09] the current pixel detector of the Compact Muon Solenoid (CMS) experiment would be subject to severe inefficiencies due to the high occupancy (cf. Section 4.5.2, [Käs08, Mei11]). The installed Read Out Chips (ROCs) do not provide enough buffer cells to store all pixel hits until the next Level 1 Trigger (L1) arrives which has a latency of up to $3.2\,\mu\mathrm{s}$ [C+08]. Furthermore the sensor material suffers from radiation damage which results in rising leakage current and increasing power consumption of the overall pixel detector. Therefore the CMS pixel detector Phase I Upgrade is scheduled for the end of 2016 [CMS12]. The pixel detector will be completely replaced by a new detector featuring an additional sensitive layer and new ROCs and Token Bit Managers (TBMs) which are able to cope with the higher occupancy. The current analog voltage level based data transmission will be dropped in favor of a 400 MHz full-digital readout providing the bandwidth necessary for the higher data rate to be transferred over the same number of optical links to the pixel detector Front End Drivers (FEDs) outside CMS.

A series of high-rate beam tests has been scheduled to qualify the new *PSI46dig* ROC. The main goal of the beam tests is to measure the efficiency of the chips in a high-rate environment similar to the conditions expected for operation at the LHC during Phase I. The beam tests use a 320 GeV proton beam provided by the Super Proton Synchrotron (SPS). A beam telescope consisting of eight *PSI46dig* single ROC modules of which seven planes are used to measure particle tracks. The efficiency of the so-called Device Under Test (DUT), one of the ROCs, can be determined by matching the measured tracks from the telescope planes to the hits recorded by the DUT.

Two beam tests are schedules at the *H4IRRAD* beam test area [BC11]. The first beam test has been carried out in July, 2012, while the second one will be performed at the end of October, 2012. The July beam test has mainly been a proof-of-concept for the beam area, the detector hardware, its readout periphery, and the EUTelescope data analysis chain. A completely new detector periphery had to be set up including telescope hardware as well as the software framework and data decoders for the new digital readout data format. Two different telescope geometries with eight *PSI46dig* ROCs each have been designed and built (cf. Section 6.2.1). Both telescopes are operated without TBM since the new digital TBM08 is not ready for submission yet. In the July beam test, the readout of the telescope has been performed by a hybrid test board setup consisting of the well-known Altera

Cyclone test board used for laboratory tests of the *PSI46v2* chip, and a new prototype for a future pixel FED based on commercially available Xilinx Spartan-6 SP605 development boards [Xil12]. The telescopes are using two 2 mm × 2 mm scintillators located before and behind the planes for triggering and are equipped with temperature sensors to monitor the sensor temperature during operation. The Trigger Logic Unit (TLU) is able to reliably determine the beginning and end of the SPS spills in which the particle beam is divided. The Data Acquisition (DAQ) is started according to the spill signal which allows an easy and automated DAQ without manual intervention. The data files are separated into single spills so that the detector data can be correlated with the recorded trigger signals and the beam line monitoring information.

The EUTelescope software framework, a full-featured analysis tool for beam test experiments, is used for the data analysis of the high-rate beam tests [B$^+$07b]. It is embedded into the ILCsoft framework and provides several processors implementing algorithms necessary for full particle track reconstruction and data analysis. These processors allow a fast assembly of analysis chains and only the decoding of the detector raw data and the conversion into the common Linear Collider I/O (LCIO) format has to be implemented by the respective user (cf. Section 7.1).

With the CMSPixelDecoder a unified library for fast decoding of data streams from CMS pixel ROC setups has been developed as part of this thesis (cf. Section 7.3.1). It has demonstrated its functionality and performance under different conditions by decoding data from *PSI46* ROCs in the laboratory as well as the more complex task of translating the readout of the CMS pixel telescopes. The output has been tested thoroughly and several adjustments have been made during the analysis of the July beam test data to provide maximum data integrity and measures for data quality. The decoder is able to decode the raw data format from both analog *PSI46v2* and digital *PSI46dig* ROCs, operated with or without TBM. This allows not only the usage for the high-rate beam tests but also opens a wide field of applications in laboratory measurements with CMS pixel ROCs. Several parameters have been implemented which allow the tuning of the decoding procedure and the selection of events based on different data quality cuts. Furthermore the CMSPixelDecoder provides detailed statistics about the decoding status and progress including counts for the total number of events as well as single pixel hits, failed decoding attempts, and detected ROC headers. This is especially useful for debugging and tuning detector readout and test board firmwares. Some modifications to the CMSPixelDecoder are scheduled for the near future to provide a more general interface for other applications. Furthermore with the IPBus data format [M$^+$11] as delivered by the Xilinx board a new readout scheme will be supported extending the current functionality of decoding data recorded with the Altera test board infrastructure.

Many of the preexisting EUTelescope processors could be used and only small adjustments had to be made. The data converter from the native detector data into the common LCIO format has been developed from scratch during this thesis. This conversion is done by CMSPixelReader, a EUTelescope processor taking advantage of the CMSPixelDecoder unit. Subsequently, the calibration of the zero-suppressed detector data is done by the CMSPixelCalibrateEvent processor which is able to read the pulse height calibration files produced by the commonly used *psi46expert* suite for detector calibration (cf. Section 7.3.2). The clustering step uses a custom implementation of the sparse clustering algorithm and extends its functionality and the information gathered during the process. The sparse clustering algorithm connects all adjacent pixel hits to clusters as described in Section 5.1.1. The processor is able to exclude hot pixels from the data set before performing the actual cluster search. Native EUTelescope processors accomplish all subsequent processing steps such as energy weighting, determination of the hit positions, alignment, and track fitting. This allows the quick construction of complete analysis chains and the usage of complex and well-tested algorithms such as Millepede II for alignment (cf. Section 7.3.5). Data sets from Monte Carlo simulations have been used to adjust the framework parameters and study its behavior under different conditions (cf. Section 7.4). This included preparations for the DUT efficiency measurements as well as the simulation of hot pixels in the data which have to be detected and excluded from the analysis (cf. Section 7.4.1).

Measurements of the particle rate delivered by the test beam have been performed. The particle rates on the detector surfaces of the telescope planes have been calculated using the information from the beam line monitoring detectors, a Gafchromic film, the telescope scintillators, and the hit maps of the *PSI46dig*. It has been shown that particle fluxes up to $400\,\mathrm{MHz/cm^2}$ and track multiplicities of approximately 6 can be achieved (cf. Section 7.5.1). This even exceeds the needs for a high-rate beam test mimicking the occupancy at double design luminosity of the LHC. It has been shown that the CMS pixel telescope is capable of tracking particles even though the total number of tracks was below the expectations due to local random triggering in the readout test board and data quality issues (see below). The telescope hardware itself as well as the periphery worked with some technical glitches. The ROCs could be programmed and tuned, and working points in the Digital-Analog Converter (DAC) parameter space could be found. The telescope planes have been exposed to a total fluence of about 4×10^{13} protons for the full July beam test run. Due to the irradiation of the sensors and the corresponding increase of leakage current, new sensors are needed for the October beam test. Otherwise the current sensors will be subject to thermal runaway at double fluence of around 8×10^{13} protons since no cooling can be applied with the current telescope geometry (cf. Section 7.5.1).

The EUTelescope framework has demonstrated its good performance for telescope data analysis even with the relatively small amount of data taken during the July beam test. Particle tracks from the telescope data have been successfully reconstructed and DUT resolutions of $\sigma_x \approx 0.10\,\mathrm{mm}$ and $\sigma_y \approx 0.07\,\mathrm{mm}$ have been achieved. The theoretical resolutions for a binary readout of the detector with single pixel clusters and no energy weighting can be calculated by $p/\sqrt{12}$ with pp being the pixel pitch. Assuming the pitch of the CMS pixel sensor the resolutions are expected to be $\sigma_x = 0.043\,\mathrm{mm}$ and $\sigma_y = 0.029\,\mathrm{mm}$. However, these resolutions could not be achieved due to the low track statistics in the data recorded with random triggers (cf. Section 7.5.4). Events with the track multiplicity expected have been successfully reconstructed and the C Event Display (CED) has been used to visualize these findings (cf. Section 7.5.5).

One of the major problems in the July beam test are bit errors in the raw data stream. Depending on the selected decoding strategy of CMSPixelDecoder up to 6 % of the pixel hits within one data-taking run cannot be decoded due to bit errors. The errors occur both in the ROC headers and the pixel hits. While the corrupted ROC header sequences can mostly be detected and recovered due to uniform deformation this is not possible for the pixel hits. The quality of the data to be analyzed can be improved by rejecting events without the correct number of ROC headers or with bit errors within the ROC headers. This allows to greatly reduce the pixel hit decoding error rate to about 2 % (cf. Section 7.5.2). However, with these data quality issues it is not possible to draw any conclusions regarding the efficiency of the *PSI46dig* ROC and the upcoming beam tests have to provide data with better quality and less bit errors. It is assumed that most of the bit errors encountered in the data stream originate from wrong phase settings and insufficient clock synchronization between the two test boards in the hybrid setup. The October beam test might solve most of these problems since only the Xilinx board and its fully functional firmware will be used for programming and readout of the telescopes.

Some minor shortcomings have been discovered and solved for the October beam test beside the major changes concerning the readout periphery of the telescopes. For example the leakage current of the telescope will be constantly monitored and logged. Furthermore measurements with the tilted geometry are scheduled. The single pixel clusters produced in the sensors of the straight telescope geometry do not fill up the data buffers to their full capacity but only the timestamp buffers which have fewer storage cells. The multi-pixel clusters created in the tilted telescope planes allow the detailed study of data buffer occupancies at high rates in the ratio of pixel hits to timestamps which is expected for the operation in the CMS detector. The behavior of the ROC under varying conditions such as beam intensity or trigger and ROC timings will be studied and efficiency measurements will be performed. The October beam test is the last time slot available at the SPS before the shutdown of all CERN accelerators for Long Shutdown 1 (LS1). Further high-rate tests of the *PSI46dig* ROC and the TBM08 will probably be performed at a new beam line currently under construction at Fermi National Accelerator Laboratory (FNAL). The last submission for both the CMS Phase I Upgrade ROC and TBM will be in fall 2013 since the series production of the modules is scheduled to start at the beginning of 2014.

The exact data analysis plans and efficiency calculations for the DUT have to be revised once valid beam test data has been recorded. Especially the parameters for the final track fitting mode as well as the matching rule for hits on the DUT have to be determined (cf. Section 7.4.2). A detailed study of the impact of these parameters is needed in order to be able to understand the efficiency determination of the *PSI46dig* ROC with a telescope consisting of the same detector. Furthermore, advanced methods such as calculation of the efficiency for every telescope layer as DUT could be studied.

The EUTelescope analysis framework and its processors for the CMS pixel detector will be improved and developed further independent from the high-rate beam test. Other working groups have shown interest in employing the EUTelescope framework for their analysis of CMS pixel sensors such as groups at CERN and DESY. Furthermore the CMSPixelDecoder is currently being implemented into the C++ framework for the IPBus communication with the new Xilinx board. It will serve as central decoding unit for both calibration data read from the ROCs and detector data taken during beam test runs. Another modification of the CMSPixelDecoder and CMSPixelReader will allow the native data from *PSI46* ROCs to be integrated as DUT data into the LCIO data stream from beam telescopes such as EUDET.

This thesis aims to provide an overview of both the EUTelescope framework itself and a possible configuration applicable for CMS pixel detector beam tests. Processors necessary for handling native *PSI46* detector data within a LCIO-driven framework have been implemented and tested as part of the work, and will be maintained in the future. This enables the usage of EUTelescope and its advantages for CMS pixel detector tests.

Appendix

A Examples of ILCsoft and EUTelescope Data

A.1 ILCsoft Offline Installation Instructions

These instructions are intended to provide a offline installation possibility (i.e. without Andrew File System (AFS) connection) for the ILCsoft and EUTelescope beam test analysis package. This instruction is tested on standard Scientific Linux 5 machines. For a fast and slim installation some unnecessary packages are not installed such as Qt and MarlinGUI. If the additional functionalities offered by these packages are needed they can be installed later. All steps assume the current directory is the target ILCsoft installation directory. The configuration files for the ILCinstall script with the optimized package selection can be obtained from the project repository [EUD].

ILCsoft Installation

- Get the ILCsoft installer (ILCinstall), preferably version *v01-12*.
- Enable the EPEL and ATrpms repositories as described in `http://linux.web.cern.ch/linux/updates/slc5.shtml`
- Install needed packages:

  ```
  yum install cernlib cernlib-g77 cernlib-g77-static cernlib-g77-devel
      gsl gsl-devel cmake
  ```

- Download and compile a recent version of CMake (> 2.8.7):

  ```
  wget http://www.cmake.org/files/v2.8/cmake-2.8.7.tar.gz && tar -xvf
      cmake-2.8.7.tar.gz
  cd cmake-2.8.7/ && ./configure && gmake
  ```

- Download and compile ROOT if needed:

  ```
  wget ftp://root.cern.ch/root/root_v5.32.01.source.tar.gz && tar -xvf
      root_v5.32.01.source.tar.gz
  cd root &&./configure --enable-gdml && make
  ```

- Make sure that GSL is found and ROOT built with LibMathMore and GDML.
- Make sure all ROOT libraries can be found, link there for the installer:

  ```
  ln -s /where/ever/your/root/lives root/5.32.01/
  ```

- Link the cernlib library includes to the right place:

  ```
  ln -s /usr/include/cernlib/2006 /usr/lib64/cernlib/2006-g77/include
  ```

- Make sure you have the proper ILCinstall config files in place and check for paths to be correct.

```
./ilcinstall_v01-12/ilcsoft-install -i
   ilcinstall_v01-12/release-slim.cfg
```

Solutions for possible problems arising during the installation:

- Installer cannot find the Java compiler javac. Add the following line to the configuration file:

```
ilcsoft.module("LCIO").envcmake['INSTALL_JAR']='OFF'
```

- LCCD does not compile due to CONDDB. Add the following line to the configuration file:

```
ilcsoft.module("LCCD").envcmake['LCCD_CONDDBMYSQL']='ON'
```

Running ILCsoft/Marlin/EUTelescope

Add the binaries for ILCsoft/Marlin/EUTelescope to the PATH variable:

```
source ILCPATH/v01-12-slim/Marlin/v01-01/build_env.sh
```

...and go!

The CMSPixel branch with the CMS specific processors

Currently the processors developed for the CMS pixel telescope are not yet merged back into the trunk of the EUTelescope SVN repository. To use these processors one needs to switch to the *cmspixel* branch, e.g. by:

```
cd Eutelescope/HEAD
svn switch
   https://svnsrv.desy.de/public/eutelescope/Eutelescope/branches/cmspixel
svn update
cd build && cmake && make clean && make install
```

A.2 GEAR Markup Examples of Silicon Beam Telescopes

A complete beam telescope is described by *geartype* and the *layers* describing the individual telescope planes:

```
<detectors>
  <detector name="SiPlanes" geartype="SiPlanesParameters">
    <siplanesID ID="0" />
    <siplanesType type="TelescopeWithDUT" />
    <siplanesNumber number="8" />
    <layers>
    <!-- telescope layers -->
    </layers>
  </detector>
</detectors>
```

The layers hold all information necessary for the reconstruction process. This includes the position, dimensions, and pitches as well as the material thickness and the radiation length of the material:

```
<layer>
  <ladder ID="0" positionX="0." positionY="0." positionZ="0.00"
     sizeX="7.8" sizeY="8.0" thickness="0.285" radLength="93.66" />
  <sensitive ID="0" positionX="0." positionY="0." positionZ="0.00"
     sizeX="7.8" sizeY="8.0" thickness="0.285" npixelX="52" npixelY="80"
     pitchX="0.15" pitchY="0.1" resolution="0.010" rotation1="1"
     rotation2="0" rotation3="0" rotation4="1" radLength="93.66" />
</layer>
```

Rotations can be obtained either with a rotation matrix for the XY plane of with three angles for 3D rotations:

```
 rotation1="1" rotation2="0" rotation3="0" rotation4="1"
 rotationZY="20.0" rotationZX="30.0" rotationXY="0.0"
```

The entire Geometry API for Reconstruction (GEAR) files for the CMS pixel telescope geometries can be obtained from [EUD].

A.3 LCIO Data Format Examples

Example of the structure of an event

This listing has been obtained using the *anajob* tool which can be used to get a fast overview of the content of a LCIO file:

```
anajob [filename.slcio]
```

It prints all collections for every event. The collections in one single event (3240) from run 150, taken in *H4IRRAD* with the CMS pixel telescope (straight version) are shown below:.

```
///////////////////////////////////
EVENT: 3240
RUN: 150
DETECTOR: CMSPixelTelescope
COLLECTIONS: (see below)
///////////////////////////////////

------------------------------------------------------------
COLLECTION NAME      COLLECTION TYPE    NUMBER OF ELEMENTS
============================================================
alignedhits          TrackerHit                92
cluster              TrackerPulse              92
fithits              TrackerHit                16
fittracks            Track                      2
hits                 TrackerHit                92
original_zsdata      TrackerData               92
------------------------------------------------------------
```

The **original_zsdata** contains the decoded detector data which has been transformed into clusters stored in the **cluster** collection. The **hits** and **alignedhits** collections contain the hit space points extracted from these clusters before and after alignment, respectively. Finally, the fitted particle tracks and the corresponding hits from the fit are stored in **fittracks** and **fithits**.

Example of event data in LCIO

Example of the content of one event in the LCIO data format, printed with the *dumpevent* tool. The command takes two arguments beside the filename, the run number and the event number:

```
dumpevent [filename.slcio] <run> <event>
```

In this case the event contains two collections, the TrackerData collection *original_zsdata* containing single pixel hits and the TrackerPulse collection with clusters. The CellIDEncoding of the collection determines the contained information encoding scheme. Each of the entities has its own 8-digit hexadecimal ID. The clusters refer with the *corr.Data* field to the IDs of the correlated pixel hits. The output has been altered and some information has been deleted for the sake of simplicity.

```
============================================================
 Event: 151 - run: 3051 - timestamp 1348499902000000000 - weight 1
============================================================
 date:      24.09.2012  15:18:22.000000000
 detector : CMSPixelTelescope

 collection name : cluster
--------- print out of TrackerPulse collection ---------

parameter CellIDEncoding [string]:
   sensorID:5,clusterID:12,xSeed:9,ySeed:10,xCluSize:9,yCluSize:9,

[   id   ] | charge | corr.Data  | cellid-fields
----------|--------|------------|--------------|
[00000acd] |   367.00| 0000000aee |
   sensorID:0,clusterID:1,xSeed:33,ySeed:51,xCluSize:1,yCluSize:1
    [...]
----------|--------|------------|--------------|

 collection name : original_zsdata
--------- print out of TrackerData collection ---------

parameter CellIDEncoding [string]:
   sensorID:5,clusterID:12,sparsePixelType:5,type:6,

 [   id   ] | cellid-fields
------------|----------------
 [00000aee] | sensorID:0,clusterID:1,sparsePixelType:1,type:1chargeADC :
    33.00,51.00,367.00,
    [...]
------------|----------------
```

The first line of every collection (starting with `parameter`) defines the cell encoding, i.e. the fields contained in the data field of every entry. For clusters this is the sensor ID and the cluster ID together with the coordinates of the seed pixel and the size of the cluster in x and y. For single pixel hits the sensor, cluster ID to which it belongs, and the pixel type and the chargeADC field with three values (x, y, and pulse height) are stored.

B CMS Pixel Telescope Analysis - Supplementary Information

In this section some examples of raw data from the CMS pixel ROC is given (Section B.1). Furthermore the CMS pixel telescope analysis chain flowchart is shown in Figure B.1.

B.1 CMSPixelDecoder Examples

Raw *PSI46v2* Data

Example output of raw data from an analog *PSI46v2* ROC. Test board headers are red, Ultrablack and Black levels are marked in green and blue, respectively. All test board headers are explained in Section 7.2.1.

```
8008    0    3 1249 :201289
8010    0    3 125f :201311
8004    0    3 12ca :201418
8081    0    3 12dc :201436
    TBM    -750 -750 -750    1    0  186  374    0
    ROC 0  -748   20  344
    TBM    -750 -750    1    1  374 -188  186 -188
8008    0    3 3a49 :211529
8010    0    3 3a5f :211551
8004    0    3 3aca :211658
8001    0    3 3adc :211676
    TBM    -750 -750 -750    1    0  374 -188 -188
    ROC 0  -752   16  348
            388 -196    4  200    4 -192
    TBM    -750 -750    1    1  374 -188  186 -188
8008    0    3 4449 :214089
8010    0    3 445f :214111
8004    0    3 44ca :214218
8001    0    3 44dc :214236
    TBM    -750 -750 -750    1    0  374 -188    0
    ROC 0  -756   12  356
           -196  776  380  196  584 -164
           -188  772  188 -196  200 -188
            396 -196  204 -196  396   68
            392  196   -4  196 -192  696
    TBM    -750 -750    1    1  374 -188  186 -188
8008    0    3 4e49 :216649
8010    0    3 4e5f :216671
8004    0    3 4eca :216778
```

CMSPixelDecoder Statistics

Example output from the `print_statistics()` method of the CMSPixelDecoder from a data taking run of the high-rate beam test telescope (run 000298). Even though the event selection cut `evtSel` is already set, the data quality issues described in Section 7.5.2 can still be observed by looking at the number of failed pixel address decodings.

```
[ VERBOSE ]   Processor statistics:
[ VERBOSE ]   Detected data headers: 24937
[ VERBOSE ]   Sane events: 10608
[ VERBOSE ]        with 510198 sane pixel hits in total.
[ VERBOSE ]        failed to convert 10973 pixel addresses.
[ VERBOSE ]   Empty events: 1525
[ VERBOSE ]   Rejected events: 12803
[ VERBOSE ]   Events without ROC headers: 0
[ VERBOSE ]   Evaluated events with wrong # of ROC headers: 0
[ VERBOSE ]        with 1 ROCs: 48x, 4 ROCs: 4x, 5 ROCs: 15x, 6 ROCs:
   392x, 7 ROCs: 339x,
[ VERBOSE ]   Event selection cut was: 2
[ VERBOSE ]   Dropped headers: 0 (no trigger/data/reset)
```

C High-rate Beam Test - Supplementary Information

Table C.1: List of July beam test runs with corresponding spill number. The ionization chamber counts N, the fractions of beam spot on plane 0 and 7, the fraction of DAQ trigger counts, and the proton rates for planes 0 and 7 are given [Sha12].

run	spill	N_{ion}	$f_{0,\mathrm{bs}}$	$f_{7,\mathrm{bs}}$	f_{sp}	r_0	r_7
123	5652	315 826	0.719	0.476	0.938	1.92×10^{08}	1.27×10^{08}
124	5798	289 613	0.721	0.633	0.938	1.76×10^{08}	1.55×10^{08}
125	5803	293 597	0.728	0.647	0.938	1.80×10^{08}	1.60×10^{08}
126	5809	297 408	0.720	0.639	0.938	1.81×10^{08}	1.60×10^{08}
128	5818	294 747	0.723	0.607	0.944	1.81×10^{08}	1.52×10^{08}
129	5825	296 629	0.724	0.613	0.942	1.82×10^{08}	1.54×10^{08}
130	5832	298 231	0.728	0.650	0.941	1.84×10^{08}	1.64×10^{08}
131	5839	302 310	0.726	0.652	0.943	1.86×10^{08}	1.67×10^{08}
132	5846	299 325	0.720	0.623	0.942	1.83×10^{08}	1.58×10^{08}
133	5853	304 273	0.727	0.641	0.942	1.87×10^{08}	1.65×10^{08}
134	5860	299 942	0.724	0.631	0.941	1.84×10^{08}	1.60×10^{08}
135	5867	309 135	0.720	0.621	0.943	1.89×10^{08}	1.63×10^{08}
136	5874	304 680	0.724	0.632	0.944	1.88×10^{08}	1.64×10^{08}
137	5881	311 441	0.721	0.627	0.935	1.89×10^{08}	1.64×10^{08}
138	5888	309 597	0.737	0.671	0.916	1.88×10^{08}	1.71×10^{08}
139	5896	315 370	0.738	0.688	0.916	1.92×10^{08}	1.79×10^{08}
140	5904	332 503	0.738	0.692	0.919	2.03×10^{08}	1.90×10^{08}
141	5912	341 730	0.739	0.664	0.915	2.08×10^{08}	1.87×10^{08}
142	5920	404 624	0.737	0.704	0.916	2.46×10^{08}	2.35×10^{08}
143	5930	397 598	0.765	0.719	0.902	2.47×10^{08}	2.32×10^{08}
144	5939	401 590	0.762	0.707	0.908	2.50×10^{08}	2.32×10^{08}
145	5949	409 757	0.757	0.720	0.902	2.52×10^{08}	2.39×10^{08}
146	5959	321 152	0.761	0.650	0.902	1.98×10^{08}	1.69×10^{08}
147	5967	315 959	0.758	0.671	0.916	1.97×10^{08}	1.75×10^{08}
148	5975	13	0.761	0.703	0.908	8.08×10^{03}	7.47×10^{03}
149	5984	355 233	0.766	0.694	0.913	2.24×10^{08}	2.03×10^{08}
150	5993	342 056	0.763	0.700	0.895	2.10×10^{08}	1.93×10^{08}
151	6002	366 806	0.762	0.702	0.914	2.30×10^{08}	2.12×10^{08}
152	6011	368 854	0.760	0.692	0.903	2.28×10^{08}	2.07×10^{08}
153	6021	372 070	0.761	0.702	0.914	2.33×10^{08}	2.15×10^{08}
154	6030	367 188	0.759	0.718	0.911	2.28×10^{08}	2.16×10^{08}
155	6039	368 738	0.762	0.704	0.896	2.27×10^{08}	2.09×10^{08}

Table C.1 – Continued

run	spill	N_{ion}	$f_{0,\text{bs}}$	$f_{7,\text{bs}}$	f_{sp}	r_0	r_7
156	6048	388 125	0.764	0.702	0.895	2.39×10^{08}	2.19×10^{08}
157	6057	383 129	0.762	0.725	0.906	2.38×10^{08}	2.26×10^{08}
158	6067	362 674	0.760	0.713	0.908	2.25×10^{08}	2.11×10^{08}
159	6077	370 188	0.761	0.715	0.906	2.30×10^{08}	2.16×10^{08}
160	6086	370 235	0.758	0.723	0.902	2.28×10^{08}	2.17×10^{08}
161	6096	376 399	0.760	0.711	0.903	2.32×10^{08}	2.18×10^{08}
162	6105	376 006	0.760	0.719	0.902	2.32×10^{08}	2.19×10^{08}
163	6114	374 407	0.762	0.714	0.896	2.30×10^{08}	2.16×10^{08}
164	6124	332 326	0.757	0.573	0.633	1.43×10^{08}	1.08×10^{08}
165	6127	14	0.760	0.701	0.871	8.33×10^{03}	7.69×10^{03}
166	6136	407 054	0.756	0.714	0.893	2.47×10^{08}	2.33×10^{08}
167	6145	421 700	0.754	0.719	0.887	2.54×10^{08}	2.42×10^{08}
168	6154	419 643	0.763	0.737	0.894	2.58×10^{08}	2.49×10^{08}
169	6164	424 714	0.764	0.722	0.887	2.59×10^{08}	2.45×10^{08}
170	6173	426 962	0.760	0.718	0.886	2.59×10^{08}	2.44×10^{08}
171	6183	431 747	0.759	0.724	0.889	2.62×10^{08}	2.50×10^{08}
172	6193	316 328	0.754	0.725	0.886	1.90×10^{08}	1.83×10^{08}
219	6277	442 221	0.758	0.718	0.892	2.69×10^{08}	2.55×10^{08}
220	6287	447 167	0.760	0.727	0.891	2.72×10^{08}	2.61×10^{08}
226	6314	385 068	0.762	0.700	0.912	2.41×10^{08}	2.21×10^{08}
227	6323	393 779	0.764	0.698	0.908	2.46×10^{08}	2.25×10^{08}
228	6332	390 917	0.762	0.714	0.911	2.44×10^{08}	2.29×10^{08}
230	6342	374 856	0.767	0.723	0.916	2.37×10^{08}	2.23×10^{08}
231	6350	424 705	0.761	0.725	0.902	2.62×10^{08}	2.50×10^{08}
232	6359	404 891	0.760	0.713	0.907	2.51×10^{08}	2.36×10^{08}
233	6368	400 296	0.760	0.716	0.912	2.50×10^{08}	2.35×10^{08}
234	6385	403 594	0.761	0.708	0.915	2.53×10^{08}	2.35×10^{08}
235	6395	402 984	0.760	0.698	0.912	2.51×10^{08}	2.31×10^{08}
236	6405	393 156	0.758	0.697	0.916	2.46×10^{08}	2.26×10^{08}
259	6460	463 018	0.759	0.718	0.896	2.83×10^{08}	2.68×10^{08}
260	6470	422 976	0.762	0.729	0.903	2.62×10^{08}	2.50×10^{08}
261	6479	421 389	0.765	0.715	0.902	2.62×10^{08}	2.45×10^{08}
262	6489	427 879	0.762	0.723	0.898	2.63×10^{08}	2.50×10^{08}
263	6498	425 537	0.767	0.731	0.905	2.66×10^{08}	2.53×10^{08}
287	6537	439 565	0.000	0.471	0.927	0.00	1.73×10^{08}
288	6539	432 843	0.760	0.720	0.903	2.67×10^{08}	2.53×10^{08}
297	6622	383 048	0.765	0.702	0.915	2.41×10^{08}	2.21×10^{08}
298	6631	393 419	0.757	0.687	0.909	2.44×10^{08}	2.21×10^{08}
299	6640	391 312	0.765	0.698	0.917	2.47×10^{08}	2.25×10^{08}
300	6803	365 393	0.764	0.698	0.913	2.29×10^{08}	2.10×10^{08}
301	6830	335 149	0.754	0.611	0.884	2.01×10^{08}	1.63×10^{08}

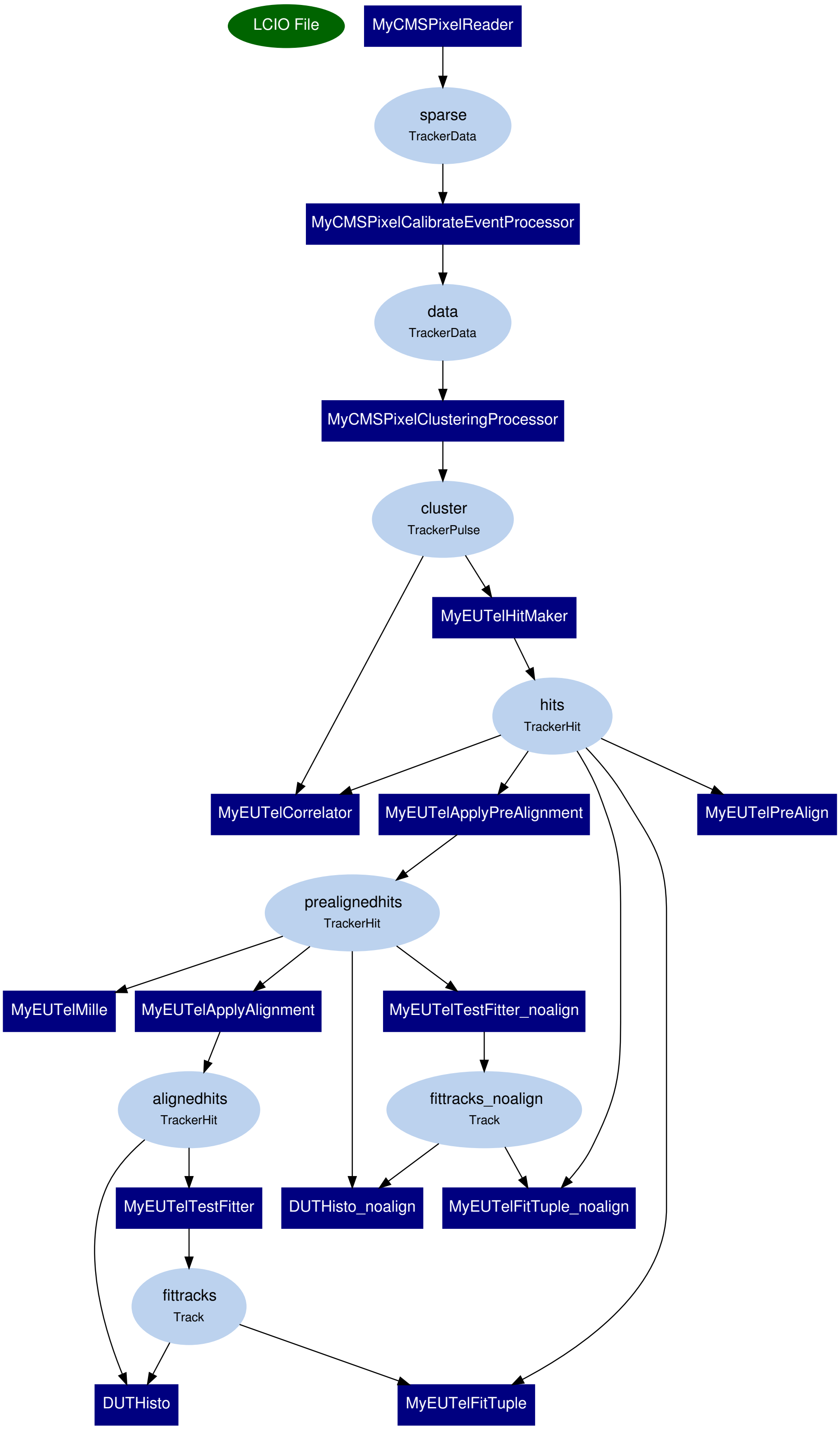


Figure B.1: Flowchart of the CMS pixel telescope analysis chain. Flowcharts of XML steering files can be created using Marlin with the commandline argument `Marlin -d`.

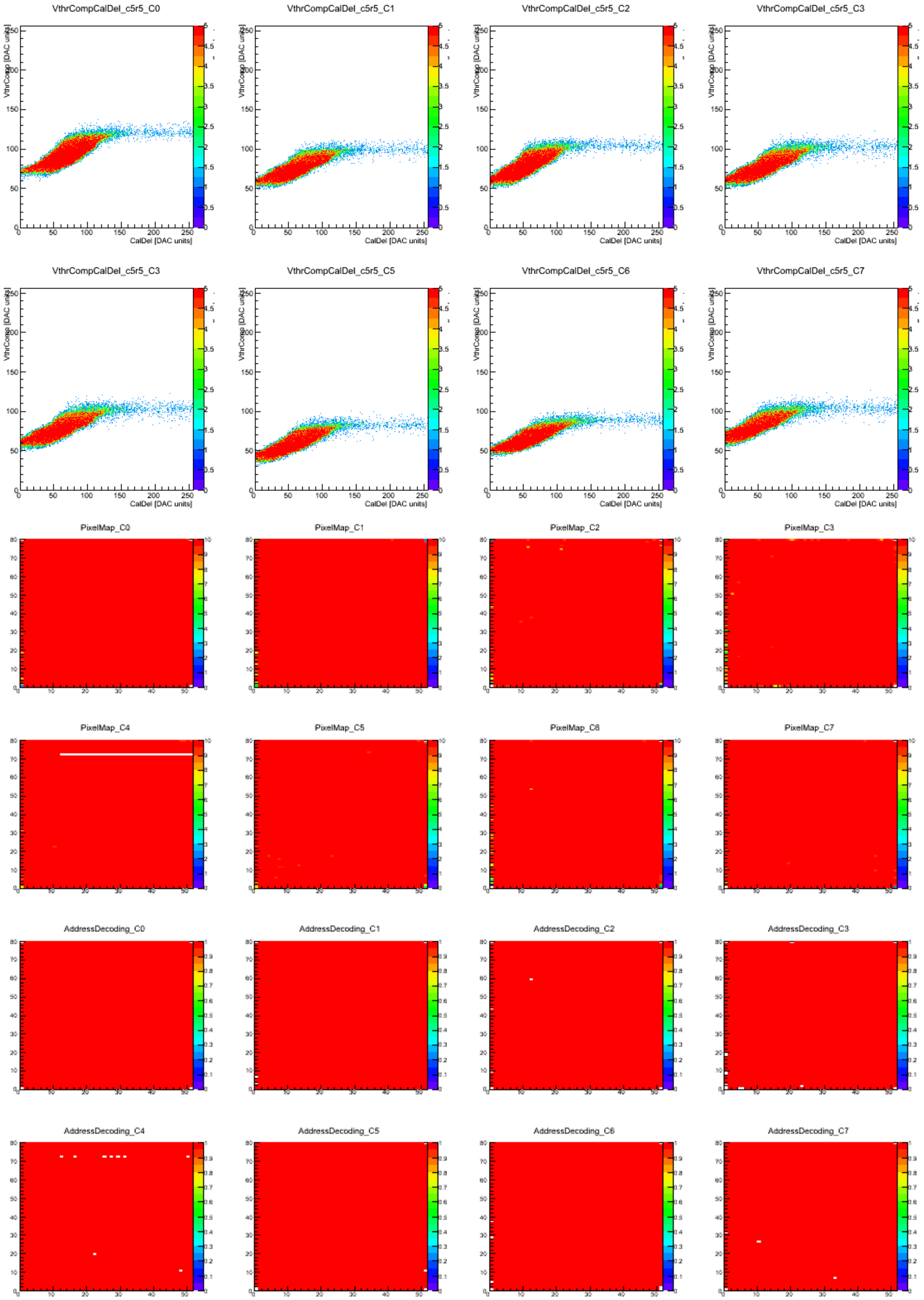


Figure C.2: Output of the standard *DacDac* test (CalDel and VthrComp), *PixAlive*, and *AdrDec* calibration tests for all CMS pixel telescope planes after parameter tuning and working point determination. All ROCs worked fine except one dead column in ROC4.

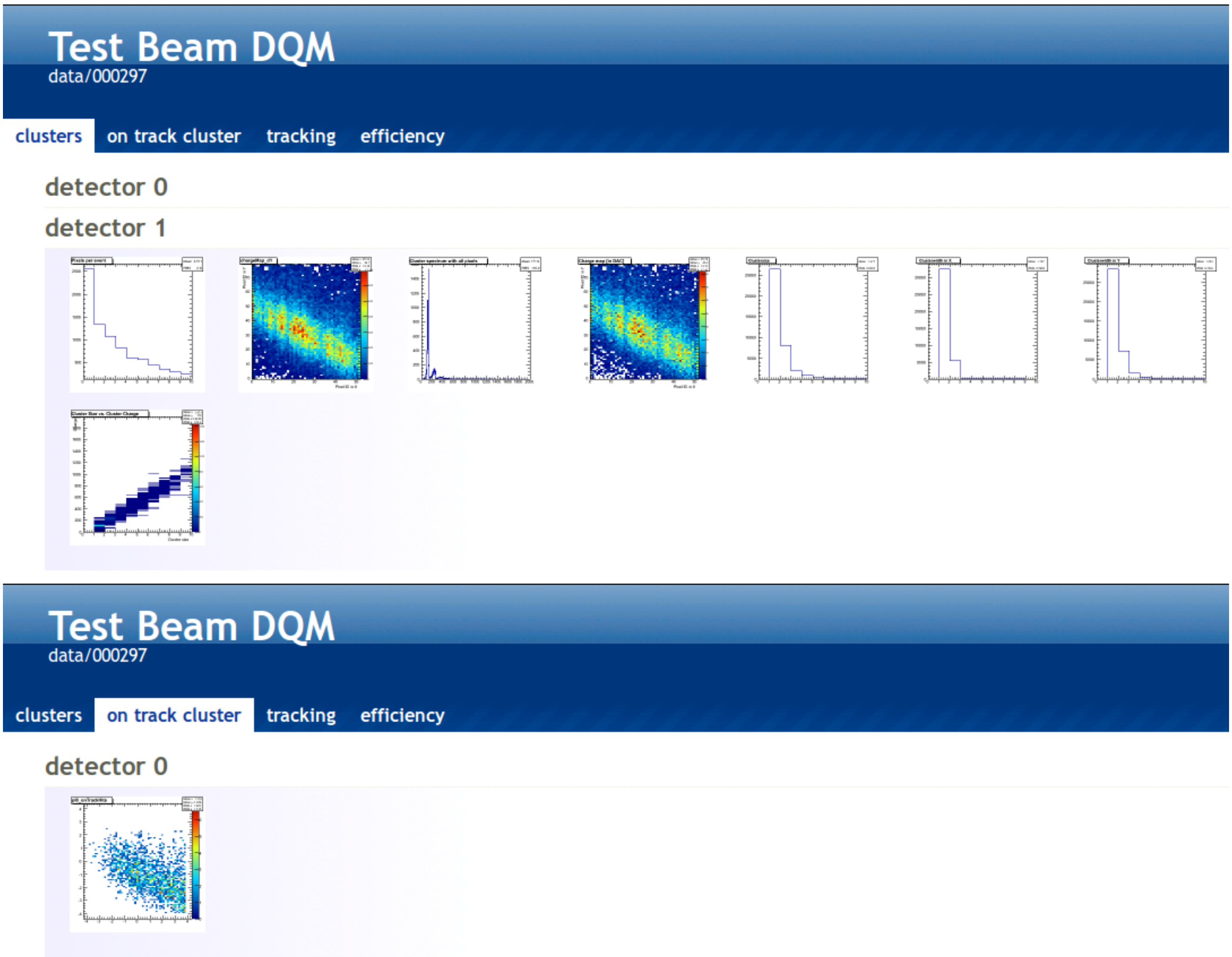


Figure C.3: Examples of the DQM website showing the results from one telescope data taking run in the July beam test (RUN000297). The histograms for every telescope plane (*detector*) can be expanded or collapsed.

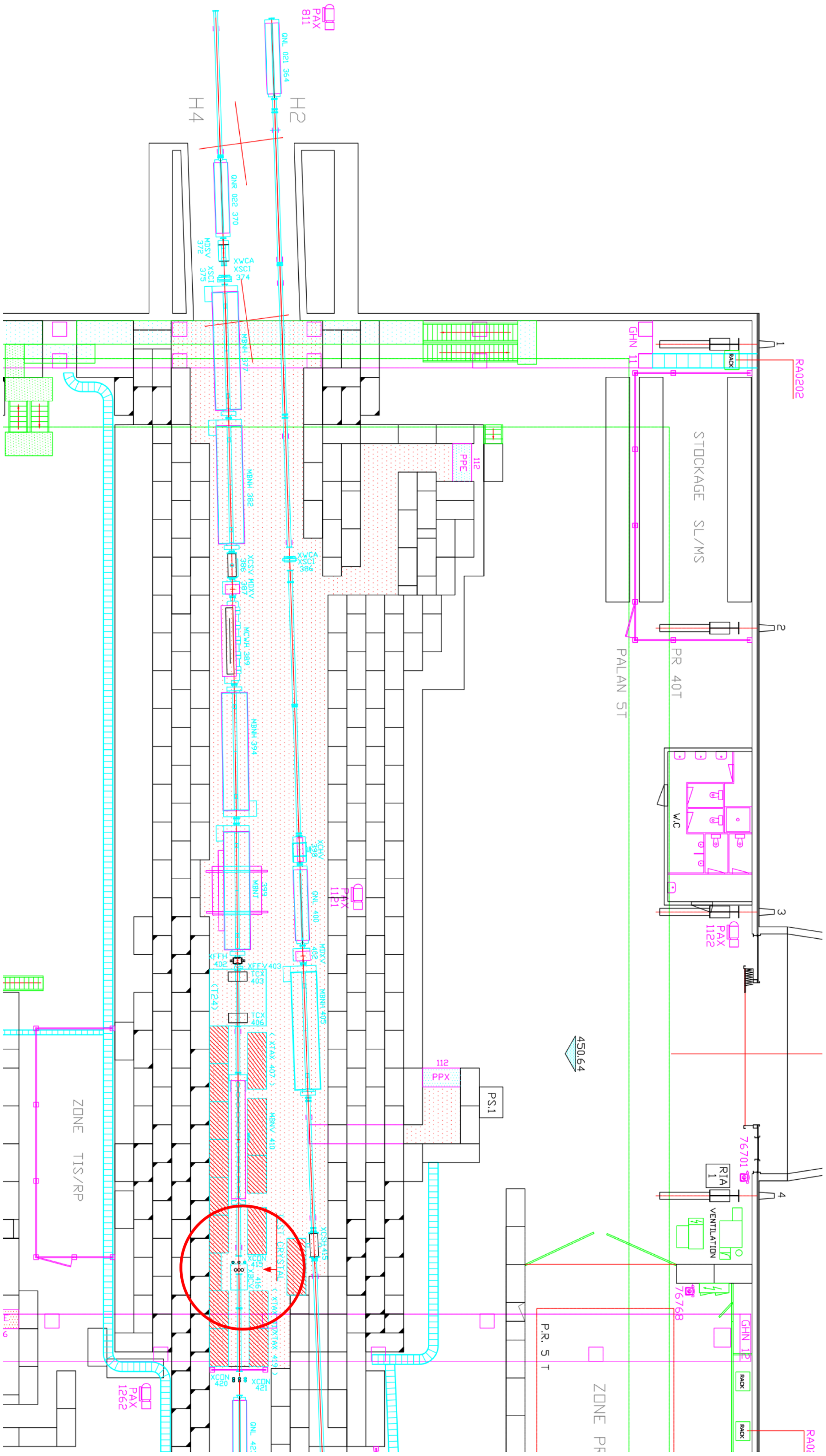


Figure C.4: Detail from the ENH1 experimental hall. The H4IRRAD area is marked by the red circle. The drawing shows the irradiation area without the beam dump installed [CER10].

List of Acronyms

ADC	Analog-Digital Converter
AFS	Andrew File System
AIDA	Abstract Interface for Data Analysis
ALICE	A Large Ion Collider Experiment
AOH	Analog Opto-Hybrid
ASIC	Application Specific Integrated Circuit
ATLAS	A Toroidal LHC ApparatuS
BPix	Barrel Pixel Detector
CALS	CERN Accelerator Logging Service
CCC	Central Control Center
CDC	Column Drain Cluster
CED	C Event Display
CERN	European Organization for Nuclear Research
CMS	Compact Muon Solenoid
CNGS	CERN Neutrinos to Gran Sasso
CSC	Cathode Strip Chamber
CTF	Combinatorial Track Finder
DAC	Digital-Analog Converter
DAQ	Data Acquisition
DCI	Double-Column Interface
DESY	Deutsches Elektronen-Synchrotron
DQM	Data Quality Monitoring
DUT	Device Under Test
ECAL	Electromagnetic Calorimeter
EE	End of Extraction (SPS signal)
EHN1	Experimental Hall North 1
FED	Front End Driver
FIFO	First In First Out
FNAL	Fermi National Accelerator Laboratory
FPGA	Field Programmable Gate Array
FPix	Forward Pixel Detector
GBL	General Broken Lines
GEAR	Geometry API for Reconstruction
GUT	Grand Unified Theory
HCAL	Hadron Calorimeter
HDI	High Density Interconnect
HLT	High Level Trigger
HV	High Voltage
I^{2}C	Inter-Integrated Circuit

ILC	International Linear Collider
IP	Interaction Point
JTAG	Joint Test Action Group
KIT	Karlsruhe Institute of Technology
L1	Level 1 Trigger
L1A	Level 1 Accept
LCIO	Linear Collider I/O
LEP	Large Electron-Positron Collider
LHC	Large Hadron Collider
LHCb	Large Hadron Collider beauty Experiment
LINAC	Linear Accelerator
LS1	Long Shutdown 1
LSB	Least Significant Bit
LSP	Lightest Supersymmetric Particle
LVDS	Low Voltage Differential Signal
Marlin	Modular Analysis & Reconstruction for the LINear collider
MIP	minimum ionizing particle
MSB	Most Significant Bit
MSSM	Minimal Supersymmetric Standard Model
NIEL	Non-Ionizing Energy Loss
NRZI	Non Return to Zero Invert
PCA	Point of Closest Approach
PCB	Printed Circuit Board
PLL	Phase-locked loop
PMT	photomultiplier tube
PS	Proton Synchrotron
PSB	PS-Booster
PSI	Paul-Scherrer-Institut
PUC	Pixel Unit Cell
RAL	Rutherford Appleton Laboratory
RF	Radio Frequency
ROC	Read Out Chip
RWTH	Rheinisch-Westfälische Technische Hochschule Aachen
SC	Super Cycle
SMD	Surface Mounted Device
SNR	Signal-to-Noise Ratio
SPS	Super Proton Synchrotron
SUSY	Supersymmetry
TBM	Token Bit Manager
TEC	Tracker End Cap
TIB	Tracker Inner Barrel
TID	Tracker Inner Disk
TLU	Trigger Logic Unit
TOB	Tracker Outer Barrel
TOTEM	Total Elastic and Diffractive Cross Section Measurement
VELO	VErtex LOcator
WE	Warning Extraction (SPS signal)
WWE	Warning Warning Extraction (SPS signal)
XML	Extensible Markup Language

Bibliography

[A$^+$95] H. Albrecht et al. Search for Rare B Decays. *Phys. Lett. B*, 353(4):554 – 562, 1995.

[A$^+$03] S. Agostinelli et al. Geant4 - A Simulation Toolkit. *Nucl. Instrum. Meth. A*, 506(3):250–303, July 2003.

[A$^+$08a] G. Aad et al. The ATLAS Experiment at the CERN Large Hadron Collider. *J. Instrum.*, 3(08):S08003, August 2008.

[A$^+$08b] K. Aamodt et al. The ALICE Experiment at the CERN Large Hadron Collider. *J. Instrum.*, 3(08):S08002, August 2008.

[A$^+$08c] J. Alves et al. The LHCb Detector at the Large Hadron Collider. *J. Instrum.*, 3(08):S08005, August 2008.

[A$^+$12] G. Aad et al. Observation of a New Particle in the Search for the Standard Model Higgs Boson with the ATLAS Detector at the LHC. *Phys. Lett. B*, 716(1):1 – 29, 2012.

[Abg08] N. Abgrall. Hadroproduction Measurements: The NA61-SHINE Experiment at CERN. *AIP Conf.Proc.*, 981:157–159, 2008.

[Aff08] A.A. Affolder. LHCb VErtex LOcator (VELO): Module Production and Performance. *Nucl. Instrum. Meth. A*, 596(1):25 – 28, 2008.

[ALE03] ALEPH, DELPHI, L3 and OPAL Collaborations. Search for the Standard Model Higgs Boson at LEP. *Phys. Lett. B*, 565(0):61–75, 2003.

[Ash12] Ashland Specialty Ingredients. Gafchromic XR-SP2 Self-Developing Radiographic Film, 2012. Retrieved September 26, 2012, from `http://www.gafchromic.com/`.

[ATL97] ATLAS Collaboration. ATLAS Inner Detector Technical Design Report Volume I. Technical Report CERN-LHCC-1997-016, CERN, 1997.

[B$^+$02] G. Bolla et al. Sensor Development for the CMS Pixel Detector. *Nucl. Instrum. Meth. A*, 485(1-2):89–99, June 2002.

[B$^+$03] V. Baggiolini et al. The CESAR Project - Using J2EE for Accelerator Controls. In *Proceedings of ICALEPCS2003*, Gyeongju, Korea, 2003.

[B$^+$04] O.S. Brüning et al. LHC Design Report. Technical report, CERN, Geneva, 2004.

[B$^+$06] C. Broennimann et al. Development of an Indium Bump Bond Process for Silicon Pixel Detectors at PSI. *Nucl. Instrum. Meth. A*, 565(1):303 – 308, 2006.

[B$^+$07a] G. Battistoni et al. The FLUKA Code: Description and Benchmarking. In *Proceedings of the Hadronic Shower Simulation Workshop*, volume 896 of *AIP Conference Series*, pages 31–49, Fermilab, 2007.

[B$^+$07b] A. Bulgheroni et al. EUTelescope: Tracking Software. *EUDET-Memo-2007-20*, 2007.

[B$^+$08] A. Bulgheroni et al. EUTelescope, the JRA1 Tracking and Reconstruction Software: A Status Report (Milestone). *EUDET-Memo-2008-48*, 2008.

[B$^+$12a] G. Barrand et al. AIDA - Abstract Interfaces for Data Analysis, 2012. Retrieved October 15, 2012, from `http://aida.freehep.org/`.

[B$^+$12b] J. Beringer et al. Review of Particle Physics. *Phys. Rev. D*, 86, 2012.

[B$^+$12c] B. Biskup et al. Commissioning and Operation of the H4IRRAD Mixed-Field Test Area. CERN-ATS-Note-2011-121 PERF, 2012.

[BC11] M. Brugger and M. Calviani. H4IRRAD - A Radiation Test Area for LHC Equipment. CERN-EDMS-No.1145469, 2011.

[Bic88] H. Bichsel. Straggling in Thin Silicon Detectors. *Rev. Mod. Phys.*, 60:663–699, July 1988.

[Blo06a] V. Blobel. A New Fast Track-Fit Algorithm based on Broken Lines. *Nucl. Instrum. Meth. A*, 566(1):14–17, October 2006.

[Blo06b] V. Blobel. Alignment Algorithms. In *Proceedings of the LHC Detector Alignment Workshop*, CERN, September 2006.

[Blo06c] V. Blobel. Software Alignment for Tracking Detectors. *Nucl. Instrum. Meth. A*, 566(1):5–13, October 2006.

[Blo07] V. Blobel. *Millepede II – Linear Least Squares Fits with a Large Number of Parameters*, 2007. Retrieved April 12, 2012, from `http://www.desy.de/~blobel/Mptwo.pdf`.

[BR97] R. Brun and F. Rademakers. ROOT - An Object Oriented Data Analysis Framework. *Nucl. Instrum. Meth. A*, 389:81–86, September 1997.

[C$^+$08] S. Chatrchyan et al. The CMS Experiment at the CERN Large Hadron Collider. *J. Instrum.*, 3(08):S08004, August 2008.

[C$^+$09] G.B. Cerati et al. Radiation Tolerance of the CMS Forward Pixel Detector. *Nucl. Instrum. Meth. A*, 600(2):408–416, March 2009.

[C$^+$12] S. Chatrchyan et al. Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC. *Phys. Lett. B*, 716(1):30–61, 2012.

[CER] CERN. TIMBER - The LHC Logging System. Retrieved July 26, 2012, from `http://www.cern.ch/timber`.

[CER08] CERN Press Release. Incident in LHC Sector 3-4. PR09.08, September 2008.

[CER10] CERN Engineering Department. *H4 Beam Line Manual*, 2010. Retrieved July 26, 2012, from `http://sba.web.cern.ch/sba/BeamsAndAreas/H4/H4manual.htm`.

[CER12a] CERN. LHC Performance and Statistics, October 2012. Retrieved September 12, 2012, from `http://lhc-statistics.web.cern.ch/LHC-Statistics/`.

[CER12b] CERN. *Manual: SPS Page-1 - How to read the SPS Page-1*, 2012. Retrieved September 27, 2012, from `http://op-webtools.web.cern.ch/op-webtools/Vistar/Doc/SPS1.pdf`.

[CER12c] CERN. SPS Page1 - OP Webtools, 2012. Retrieved September 27, 2012, from `http://op-webtools.web.cern.ch/op-webtools/Vistar/vistars.php?usr=SPS1`.

[CGL09] A.G. Cohen, S.L. Glashow, and Z. Ligeti. Disentangling Neutrino Oscillations. *Phys. Lett. B*, 678(2):191 – 196, 2009.

[CMS98] CMS Collaboration. The CMS Tracker Technical Design Report. Technical report, CERN, April 1998.

[CMS06] CMS Collaboration. CMS Physics Technical Design Report Volume I : Detector Performance and Software. Technical report, CERN, February 2006.

[CMS07] CMS Collaboration. CMS Physics Technical Design Report, Volume II: Physics Performance. Technical report, CERN, June 2007.

[CMS10] CMS Collaboration. Commissioning and Performance of the CMS Pixel Tracker with Cosmic Ray Muons. *J. Instrum.*, 5(03):T03007, 2010.

[CMS11] CMS Collaboration. Technical Proposal for the Upgrade of the CMS Detector through 2020. Technical Report CERN-LHCC-2011-006, CERN, Geneva, June 2011.

[CMS12] CMS Collaboration. CMS Technical Design Report for the Pixel Detector Upgrade. Technical Report CERN-LHCC-2012-016, CERN, 2012.

[Cor09] E. Corrin. The EUDET High Resolution Beam Telescope - The Final Digital Readout. In *2009 IEEE Nuclear Science Symposium Conference Record (NSS/MIC)*, pages 816–819. IEEE, October 2009.

[DES10] Deutsches Elektron-Synchrotron DESY. EUDET: Detector R&D Towards the International Linear Collider, 2010. Retrieved March 15, 2012, from `http://www.eudet.org/`.

[DES12] Deutsches Elektronen-Synchrotron DESY. RAIDA - A ROOT Implementation of AIDA, 2012. Retrieved October 15, 2012, from `http://ilcsoft.desy.de/portal/software_packages/raida/`.

[DY70] S.D. Drell and T.-M. Yan. Massive Lepton-Pair Production in Hadron-Hadron Collisions at High Energies. *Phys. Rev. Lett.*, 25:316–320, August 1970.

[EB64] F. Englert and R. Brout. Broken Symmetry and the Mass of Gauge Vector Mesons. *Phys. Rev. Lett.*, 13:321–323, August 1964.

[EB08] L. Evans and P. Bryant. LHC Machine. *J. Inst.*, 3(08):S08001, August 2008.

[Eft03] I. Efthymiopoulos. *The EHN1 Beams, SPS/EA Training Lecture Program*, March 2003. Retrieved September 12, 2012, from `http://sba.web.cern.ch/sba/Documentations/docs/Training/ie_spstraining_ehn1.pdf`.

[Ell12] A. Elliott-Peisert. Private communication, September 2012.

[Erd10] W. Erdmann. The CMS Pixel Detector. *Int. J. Mod. Phys. A*, 25(07):1315, 2010.

[Esp97] J.R. Espinosa. Theoretical Higgs Mass Bounds in the Standard Model and Supersymmetric Extensions. *Surveys in High Energy Physics*, 10(1-4):279–311, 1997.

[EUD] EUDET Collaboration. EUTelescope SVN Repository, Branch cmspixel. `https://svnsrv.desy.de/public/eutelescope/Eutelescope/branches/cmspixel`.

[F$^+$05] A. Ferrari et al. *FLUKA: A Multi-Particle Transport Code.* CERN, Geneva, 2005.

[Fav10] C. Favaro. A New CMS Pixel Detector for the LHC Luminosity Upgrade. In *Proceedings of 8th International Conference on Radiation Effects on Semiconductor Materials Detectors and Devices, Florence, Italy*, October 2010.

[G$^+$03] F. Gaede et al. LCIO - A Persistency Framework for Linear Collider Simulation Studies. In *Proceedings of CHEP03, La Jolla, California*, March 2003.

[G+11] P. Grenier et al. Test Beam Results of 3D Silicon Pixel Sensors for the ATLAS Upgrade. *Nucl. Instrum. Meth. A*, 638(1):33–40, May 2011.

[Gab05] K. Gabathuler. PSI46 Pixel Chip - External Specification. Technical report, Paul Scherrer Institut, 2005.

[Gae07] F. Gaede. *Gear - A Geometry Description Toolkit for ILC Reconstruction Software*, April 2007. Retrieved March 15, 2012, from `http://ilcsoft.desy.de/gear/`.

[GE07] F. Gaede and J. Engels. Marlin et al. - A Software Framework for ILC Detector R&D. *EUDET-Report-2007-11*, 2007.

[GGS99] M.K. Gaillard, P.D. Grannis, and F.J. Sciulli. The Standard Model of Particle Physics. *Rev. Mod. Phys.*, 71:S96–S111, March 1999.

[GHK64] G.S. Guralnik, C.R. Hagen, and T.W.B. Kibble. Global Conservation Laws and Massless Particles. *Phys. Rev. Lett.*, 13:585–587, November 1964.

[Gra53] F. Gray. Pulse Code Communication, United States Patent 2632058, March 1953.

[Gsc06] E. Gschwendtner. CERN Neutrinos to Gran Sasso (CNGS): First Beam. In *2006 IEEE Nuclear Science Symposium Conference Record*, pages 1489–1492. IEEE, 2006.

[H+73] F.J. Hasert et al. Observation of Neutrino-like Interactions without Muon or Electron in the Gargamelle Neutrino Experiment. *Phys. Lett. B*, 46(1):138 – 140, 1973.

[H+09] C. Hu-Guo et al. A Ten Thousand Frames per Second Readout MAPS for the EUDET Beam Telescope. In *TWEPP (Topical Workshop on Electronics for Particle Physics), Paris, France*, September 2009.

[Har08] F. Hartmann. *Evolution of Silicon Sensor Technology in Particle Physics (Springer Tracts in Modern Physics)*. Springer, December 2008.

[Hep12] EUDET / Hepforge. EUTelescope Class References, 2012. Retrieved March 15, 2012, from `http://eudaq.hepforge.org/Eutelescope/`.

[Hig64a] P.W. Higgs. Broken Symmetries and the Masses of Gauge Bosons. *Phys. Rev. Lett.*, 13:508–509, October 1964.

[Hig64b] P.W. Higgs. Broken Symmetries, Massless Particles and Gauge Fields. *Phys. Lett.*, 12(2):132–133, September 1964.

[Hoe11] H. Hoelbe. *C Event Display (CED) User Manual*, 2011. Retrieved March 15, 2012, from `http://ilcsoft.desy.de/CED/current/doc/manual.pdf`.

[Hoo11] B. Hooker. Upgrades Workshop Envisions the Future of CMS Experiments. *Fermilab Today*, November 28, 2011.

[Hoß12] J. Hoß. X-Ray Calibration of Pixel Detector Modules for the Phase I Upgrade of the CMS Experiment (IEKP-KA/2012-17). Diplomarbeit, Karlsruhe Institute of Technology (KIT), October 2012.

[IL10] H. Ibach and H. Lüth. *Solid-State Physics: An Introduction to Principles of Materials Science*. Springer, Berlin, Heidelberg, 2010.

[K+06] S. König et al. Assembly of the CMS Pixel Barrel Modules. *Nucl. Instrum. Meth. A*, 565(1):62–66, September 2006.

[K+10] V. Khachatryan et al. CMS Tracking Performance Results from early LHC Operation. *Eur. Phys. J. C*, 70(4):1165–1192, November 2010.

[Käs08] H.C. Kästli. Phase I Pixel ROC. Presented at the 2013 Pixel Replacement/Upgrade Discussion Meeting, CERN, October 2008.

[Kit05] C. Kittel. *Einführung in die Festkörperphysik.* Oldenbourg Wissenschaftsverlag, October 2005.

[Lef08] C. Lefevre. LHC: The Guide. Retrieved September 12, 2012, from `http://cdsweb.cern.ch/record/1092437`, January 2008.

[Lu12] R.-S. Lu. Private communication, April 2012.

[M+02] M. Moll et al. Relation between Microscopic Defects and Macroscopic Changes in Silicon Detector Properties after Hadron Irradiation. *Nucl. Instrum. Meth. B*, 186(1-4):100–110, January 2002.

[M+11] J. Mans et al. *IPBus: Simple IP-based μTCA Control System*, 2011. Retrieved September 19, 2012, from `https://cactus.hepforge.org/trac/wiki/IPbusIntro`.

[Man04] R. Mankel. Pattern Recognition and Event Reconstruction in Particle Physics Experiments. *Reports on Progress in Physics*, 67(4):553–622, April 2004.

[Mei11] B. Meier. CMS Pixel Detector with New Digital Readout Architecture. *J. Instrum.*, 6(01):C01011–C01011, January 2011.

[Mei12] B. Meier. ROC Submission. In *Pixel Upgrade Plenary.* CERN, February 2012.

[Mer12] S. Mersi. Private communication, August 2012.

[ML09] B. Meier and J. Lazo-Flores. Pixel Front-end Development for CMS. *Proceedings of Science (VERTEX2009)*, 028, September 2009.

[Mol99] M. Moll. *Radiation Damage in Silicon Particle Detectors - Microscopic Defects and Macroscopic Properties.* PhD thesis, Universität Hamburg, 1999.

[Moo94] B. Moore. Evidence Against Dissipation-less Dark Matter from Observations of Galaxy Haloes. *Nature*, 370:629–631, August 1994.

[MS12] S. Mersi and X. Shi. Test Beam DQM, 2012. Retrieved July 26, 2012, from `https://svnsrv.desy.de/public/eutelescope/Eutelescope/branches/cmspixel/simplesub/dqm`.

[Phi00] Philips Semiconductors. The I^{2}C Specification, Version 2.1, 2000. Retrieved October 5, 2012, from `www.classic.nxp.com/acrobat_download2/literature/9398/39340011.pdf`.

[Pit12] D. Pitzl. Private communication, April 2012.

[Qin12] G. Qin. Private communication, September 2012.

[R+06] L. Rossi et al. *Pixel Detectors: From Fundamentals to Applications (Particle Acceleration and Detection).* Springer, 2006.

[S+00] W. Snoeys et al. Layout Techniques to Enhance the Radiation Tolerance of Standard CMOS Technologies demonstrated on a Pixel Detector Readout Chip. *Nucl. Instrum. Meth. A*, 439:349–360, 2000.

[S+06] T. Speer et al. Track Reconstruction in the CMS Tracker. *Nucl. Instrum. Meth. A*, 559(1):143 – 147, 2006.

[SA12] M. Segal and K. Akeley. The OpenGL Graphics System - A Specification, Version 4.3, 2012. Retrieved October 10, 2012, from `http://www.opengl.org/registry/doc/glspec43.core.20120806.pdf`.

[Sch05] E.F. Schubert. *Doping in III-V Semiconductors*. Cambridge University Press, August 2005.

[Sha12] A. Sharma. Private communication, August 2012.

[Spi05] H. Spieler. *Semiconductor Detector Systems*. Oxford University Press, 2005.

[TEV12] TEVNPH Working Group for the CDF and D0 collaborations. Updated Combination of CDF and D0 Searches for Standard Model Higgs Boson Production with up to 10.0 fb^{-1} of Data, 2012. Retrieved August 16, 2012, from `http://arxiv.org/abs/1207.0449`.

[TOT08] TOTEM Collaboration. The TOTEM Experiment at the CERN Large Hadron Collider. *J. Instrum.*, 3(08):S08007, 2008.

[Tri09] A. Tricomi. Tracker Upgrade General Meeting - Simulation Progress Report. Presented at the CMS Tracker Week, CERN, October 2009.

[Tze12] Y.-M. Tzeng. Private communication, May 2012.

[Var02] J. Varela. CMS L1 Trigger Control System. Technical Report CMS-NOTE-2002-033, CERN, Geneva, September 2002.

[WZ74] J. Wess and B. Zumino. Supergauge Transformations in Four Dimensions. *Nucl. Phys. B*, 70(1):39 – 50, 1974.

[Xil12] Xilinx Inc. Spartan-6 FPGA SP605 Evaluation Kit, 2012. Retrieved October 2, 2012, from `http://www.xilinx.com/products/boards-and-kits/EK-S6-SP605-G.htm`.

[Zar10] A.F. Zarnecki. Analytical Track Fitting: Performance and New Options. *EUDET-Memo-2010-14*, pages 1–11, 2010.

[Zim09] F. Zimmermann. CERN Upgrade Plans for the LHC and its Injectors. *Proceedings of Science (EPS-HEP2009)*, 140, 2009.

[ZN07] A.F. Zarnecki and P. Niezurawski. EUDET Telescope Geometry and Resolution Studies. *EUDET-Report-2007-01*, pages 1–36, 2007.

Danksagung

Zu guter Letzt möchte ich mich bei all denjenigen bedanken, die mich vor und während meiner Diplomarbeit unterstützt und zu deren Gelingen beigetragen haben.

An erster Stelle gilt mein Dank Prof. Dr. Ulrich Husemann für die ausgezeichnete Betreuung sowie Themenwahl und das Ermöglichen der Mitarbeit an einem externen Projekt. Vielen Dank auch für die Unterstützung bei Dingen wie dem Besuch von Workshops, die über eine „normale“ Diplomarbeit hinausgingen. Prof. Dr. Guido Drexlin danke ich für die Übernahme des Korreferats.

Furthermore I would like to thank all the members of the high-rate beam test workgroup at CERN and RAL: Anna Elliott-Peisert, Ulysses Grundler, Kristian Harder, Andreas Kornmayer, Rong-Shyang Lu, Stefano Mersi, Xin Shi, and Yeng-Ming Tzeng. Thank you for your support. It has been a pleasure to work with you and I'm looking forward to continuing this in the future.

Also, many thanks to Igor Rubinsky and Daniel Pitz from DESY for answering questions concerning the EUTelescope software, and for fruitful discussions about the CMS pixel ROC and data decoding difficulties. We will meet again in Hamburg soon.

Dr. Thomas Weiler und Stefan Heindl möchte ich für ihre Unterstützung und die unermüdlichen Korrekturen an dieser Arbeit danken. Besonders Stefan Heindl war stets hilfsbereit zur Stelle - unabhängig davon, ob es um technische Probleme, inhaltliche Fragen oder eine Diskussion über das aktuelle (Welt-) Geschehen ging. Auch an Robert Eber geht ein Dank für seine Lektoren-Tätigkeit.

Ich danke Dr. Hans Jürgen Simonis für die Einrichtung und Betreuung unserer liberalen IT-Infrastruktur und die vielen Gespräche über das aktuelle Opernrepertoire des Badischen Staatstheaters. Frank Hartmann möchte ich für seine Unterstützung sowie für die vielen interessanten Vorträge zum aktuellen Status von CMS danken.

Für die Überbrückung müßiger Stunden mit vielen Sprüchen und Gelächter danke ich meinen Mitdiplomanten Jan Hoß, Stefan Heitz und Jan „Jam“ Müller. Sicher werden sich Möglichkeiten bieten, unterbrochene Auseinandersetzungen zu Ende zu führen.

Vielen Dank auch an alle anderen Mitglieder des IEKP Campus Nord - die Arbeit hat nicht zuletzt aufgrund der netten gemeinsamen Abende, den täglichen Kaffeerunden und dem freitäglichen Grillen Spaß gemacht. Für die Unterstützung bei organisatorischen Fragen, Dienstreise-Anträgen und Abläufen im Universitäts-Dickicht danke ich Diana Fellner-Thedens sowie Brigitte Gering.

Besonders jedoch möchte ich mich bei meinen Eltern Karin und Hans-Georg Spannagel bedanken. Sie haben mir das Studium ermöglicht, mich immer bei all meinen Vorhaben und Plänen voll unterstützt und ermutigt. Meiner Schwester Janika danke ich für ihre interessierten Fragen zu physikalischen Themen, die mich vieles nochmals reflektieren ließen sowie für die interessanten Lektüren und ausschweifenden politischen Diskurse. Ein besonderer Dank gebührt Andrea Müller für ihre unendliche Geduld mit meinen Launen, ihre Unterstützung und für die schöne gemeinsame Zeit.

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, 2. November 2012

Simon Spannagel