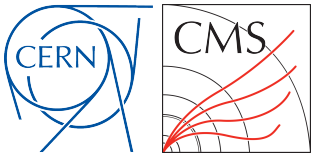


Ernst-Abbe-Hochschule Jena
Fachbereich Elektro- und Informationstechnik
Studiengang Raumfahrtelektronik

Masterarbeit

New Peak Finding Algorithm for the BCM1F Detector of the CMS Experiment at CERN

Alexander Jonas Rüde

CERN-THESIS-2018-021
19/03/2018

Betreuung:
Prof. Dr. Burkart Voss (EAH Jena)
Dr. Anne Dabrowski (CERN)

28. Februar 2018

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Diese Arbeit lag in gleicher oder ähnlicher Weise noch keiner Prüfungsbehörde vor und wurde bisher noch nicht veröffentlicht.

Ort, Datum, Unterschrift

Abstract

The Compact Muon Solenoid experiment (CMS) is located at the most powerful particle collider in the world, the Large Hadron Collider (LHC). It detects the resulting products of the proton beams that have been accelerated and collide in the center of the experiment. For the operation of a particle collider, a prompt measurement result of the so-called luminosity is of high priority, as it quantifies the ability of the collider to produce a certain number of specific particle interactions. The BCM1F detector is one of the instruments that are installed along the beam pipe to provide an online measurement of the luminosity. For reliable results, the electrical pulses that are produced by particle hits at the sensors have to be detected and characterised by their relative arrival time. In the currently operating back end system, the pulses are detected using a single threshold discriminator. This approach shows hit counting inefficiencies, especially for closely occurring, consecutive pulses. A similar pulse detection implementation for the new BCM1F back end system shows even higher inefficiency. This thesis explores the peak finding capability of an algorithm that finds pulse peaks by using the derivative of the signal. The different approaches are compared and the advantages of the derivative based algorithm are explained.

The first chapter introduces the reader to the Large Hadron Collider and the Compact Muon Solenoid experiment, as well as LHC specific terminology and quantities. Chapter 2 sets the purpose of this thesis into a context and explains the motivation of this work. The concept and details of luminosity, as well as the luminosity measurement at the CMS experiment are explained in chapter 3. The architecture and working principle of the BCM1F detector are explained in detail in chapter 4. The building blocks of BCM1F from the front end to the back end are characterised in chapter 5, to create a software model of the detector signal. In chapter 6 the discriminator based approach of the operating system and the current peak finder implementation are introduced and their inefficiencies are discussed. Chapter 7 then presents the new derivative based peak finding algorithm and its implementation in VHDL. The performance is evaluated and compared to the discriminator based approaches in chapter 8. The conclusion about the implemented derivative based peak finding algorithm is drawn in chapter 9, followed by an outlook and the perspective of the peak finder application and possible improvements in chapter 10.

Zusammenfassung

Das Compact Muon Solenoid Experiment (CMS) befindet sich am leistungsfähigsten Teilchenbeschleuniger der Welt, dem Large Hadron Collider (LHC). Es detektiert die Kollisions-produkte der beiden Protonenstrahlen die beschleunigt werden und im Zentrum des Experiments kollidieren. Für den Betrieb eines Teilchenbeschleunigers ist ein schnelles Messergebnis der sogenannten Luminosität von hoher Priorität, da diese die Fähigkeit des Teilchenbeschleunigers zur Erzeugung einer bestimmten Anzahl von spezifischen Teilchenwechselwirkungen quantifiziert. Der BCM1F Detektor ist eines der Instrumente die entlang des Strahlrohrs installiert sind, um eine Online-Messung der Luminosität zu ermöglichen. Für zuverlässige Ergebnisse müssen die elektrischen Impulse, die durch eintreffende Partikel an den Sensoren erzeugt werden, erfasst und durch ihre relative Ankunftszeit charakterisiert werden. Im zur Zeit betriebenen Backend System werden die Pulse durch einen einfachen Schwellwertdiskriminator erkannt. Diese Herangehensweise weist Ineffizienz bei der Erkennung der Pulse auf, besonders bei kurz nacheinander auftretenden Pulsen. Eine ähnliche Herangehensweise zur Pulserkennung für das neue BCM1F Backend System weist sogar größere Ineffizienz auf. Diese Arbeit untersucht die Fähigkeit eines Algorithmus zum detektieren der Pulsspitzen, der auf der Verwendung der Ableitung des Signals basiert. Die verschiedenen Herangehensweisen zur Pulserkennung werden verglichen und die Vorteile des Ableitungsbasierten Algorithmus werden erklärt.

Das erste Kapitel dient der Einleitung in den Large Hadron Collider und das Compact Muon Solenoid Experiment sowie in LHC-spezifische Terminologien und Größen. Kapitel 2 stellt den Zweck dieser Arbeit in einen größeren Zusammenhang und erklärt die Motivation dieser Arbeit. Das Konzept und die Details der Luminosität sowie die Luminositätsmessung am CMS-Experiment werden im Kapitel 3 erläutert. Die Architektur und Funktionsweise des BCM1F Detektors wird ausführlich in Kapitel 4 erklärt. Unter Berücksichtigung der Detektorbausteine vom Frontend bis zum Backend wird in Kapitel 5 der BCM1F Detektor charakterisiert und ein Softwaremodell des Detektorsignals erstellt. In Kapitel 6 werden die Schwellwertbasierten Herangehensweisen vorgestellt und ihre Schwächen diskutiert. Der neue, Ableitungsbasierte Algorithmus wird in Kapitel 7 im Detail erklärt und die VHDL-Implementierung wird erläutert. In Kapitel 8 wird der neue Algorithmus durch simulierte Daten getestet und mit den Schwellwertbasierten Implementierungen verglichen. Eine Schlussfolgerung über den neu entwickelten, implementierten Algorithmus wird in Kapitel 9 gezogen, gefolgt von einem Ausblick und perspektivischer Anwendung sowie möglichen Verbesserungen in Kapitel 10.

Contents

List of Figures

List of Tables

Glossary

AC Alternating Current.

ADC Analog-to-Digital Converter.

AMC Advanced Mezzanine Card.

AOH Analog-Opto-Hybrid.

ASIC Application Specific Integrated Circuit.

BCM1F Fast Beam Conditions Monitor.

BRIL Beam Radiation Instrumentation and Luminosity.

CMS Compact Muon Solenoid.

CPLD Configurable Programmable Logic Device.

CVD Chemical Vapor Deposition.

DC Direct Current.

DDR Double Data Rate.

FIFO First-in first-out.

FMC FPGA Mezzanine Card.

FPGA Field Programmable Gate Array.

FWHM Full Width Half Maximum.

GLIB Gigabit Link Interface Board.

HPC High Pin Count.

IC Integrated Circuit.

IDDR Input Double-Data-Rate.

IP Interaction Point.

LHC Large Hadron Collider.

LLD Linear Laser Driver.

LSB Least Significant Bit.

LVDS Low Voltage Differential Signaling.

MIB Machine Induced Background.

MMC Module Management Controller.

PAM Pulse Amplitude Modulation.

PCB Printed Circuit Board.

pCVD poly-crystalline Chemical Vapor Deposition.

PLL Phase Locked Loop.

PMF Probability Mass Function.

RF Radio Frequency.

RMS Root Mean Square.

sCVD single-crystalline Chemical Vapor Deposition.

SNR Signal-to-Noise Ratio.

SNRD Smooth Noise-Robust Differentiator.

VHDL Very High Speed Integrated Circuit Hardware Description Language.

List of Symbols

L	Luminosity
σ_p	Production cross-section
σ_{vis}	Visible cross-section
μ	Per-bunch occupancy
σ_{noise}	Standard deviation of noise distribution

1. Introduction

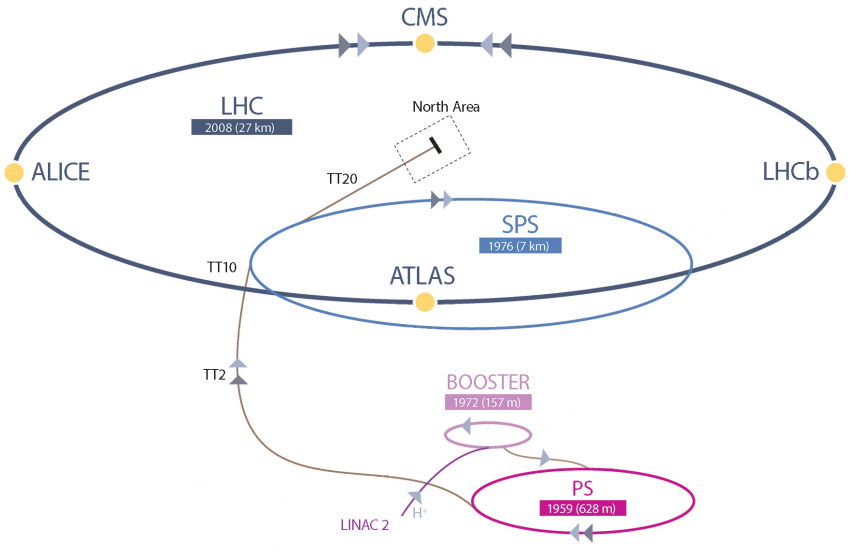


Figure 1.1: LHC acceleration and injection chain at CERN [1]

The Large Hadron Collider (LHC) [2], located at the European Laboratory for Nuclear Research (CERN) in Geneva, is the largest and most powerful particle accelerator in the world. It consists of a 27 km long ring of superconducting magnets and is capable of colliding two proton beams at a center of mass energy of 14 TeV. To achieve this energy, the bunched proton beams are first accelerated in several stages, prior to injection into the LHC accelerator ring. The accelerators used in the LHC injection chain are shown in figure 1.1.
The standard model of particle physics [3], which explains how the basic building blocks of matter interact, is studied in the four main experiments at the LHC. The ATLAS [4] and CMS [5] experiments are general-purpose detectors to investigate a broad range of physics phenomena. Both detectors were designed independently and use different approaches to observe the same phenomena, which allows cross-confirmation of new discoveries. The ALICE [6] and LHCb [7] experiments focus on specific phenomena and have very specialised detector systems.
The CMS (Compact Muon Solenoid) consists of several sub-detectors to accurately track and determine the products of the particle collisions that happen in the center

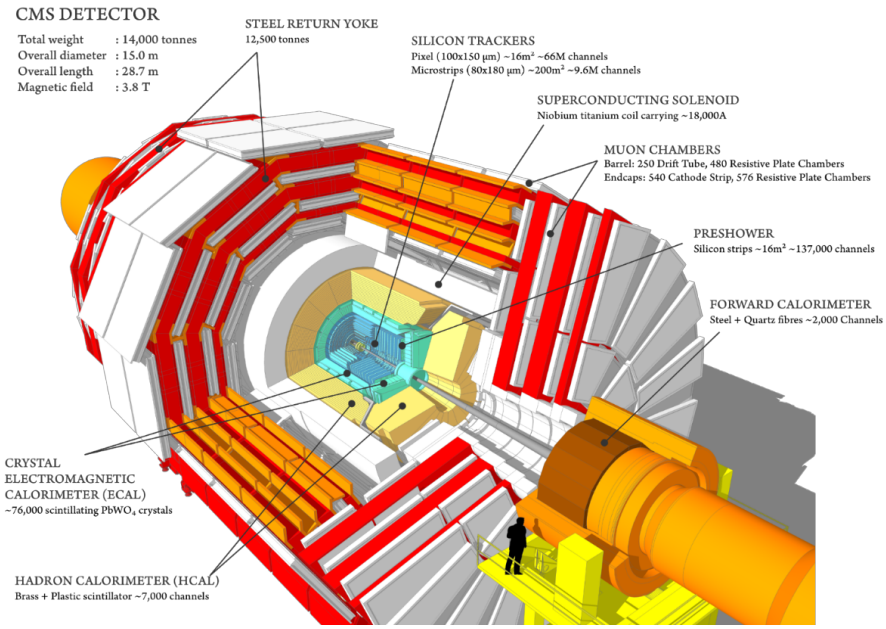


Figure 1.2: A cutaway view of the CMS detector [8]

of the detector. It is made of several concentric layers and incorporates one of the most powerful solenoid magnets ever built, capable of producing a magnetic field of up to 4 Tesla. Figure 1.2 visualises and briefly describes the different components and sub-detectors of CMS.
Together with the ATLAS collaboration, the CMS collaboration announced the discovery of the Higgs Boson in 2012 [9], one of the most important discoveries in fundamental physics.

The LHC is colliding two counter-rotating proton beams, each made of several bunches of about 100 billion protons. Each bunch is circulating in the LHC with a revolution frequency of 11.245 kHz. The circulating bunches are arranged in a certain bunch pattern, following the so-called LHC filling scheme [10]. Usually, bunches are grouped in bunch trains, where the bunches are spaced at approximately 25ns. Both beams can consist of up to a maximum of 2808 bunches. In the interaction point (IP) of each experiment, the bunches of the two counter-rotating beams collide with the so-called bunch crossing frequency of 40.08 MHz.
In every bunch crossing there are potentially more than 60 so-called inelastic proton-proton collision that result in the desired scattering and the creation of collision products. The detectors within CMS then track the trajectory of these products and determine their properties. The collected data is further processed and stored. Eventually, the data sets are evaluated for studies about the standard model and also to potentially reveal phenomena beyond it.

2. Motivation

Particle colliders are used to produce rare physical interactions, that result from the collision of the particles. The beam energy and the luminosity are the most important quantities to evaluate the performance of particle colliders [11]. The luminosity is the characteristic quantity to measure the ability of a collider to produce a certain number of specific particle interactions. Its precise measurement is indispensable for the accelerator operation and for the analysis of the physics data.

At CMS there are several detectors for measuring the luminosity (see chapter 3.3). One of these luminometers is the Fast Beam Conditions Monitor (BCM1F), which detects particles that hit one of its diamond sensors, as described in chapter 4. The luminosity of each bunch can be calculated if the number of particle hits per bunch crossing is known (see 3.2). Thus, the accuracy of the luminosity measurement depends on the capability of the detector to determine every sensor hit of a particle. Implications in the current BCM1F detector system and noise sources along the signal transmission chain add certain difficulty to correctly identify each hit. Identifying a particle in the detector signal is the task of the peak finder module, located in the back end electronics of the system. The currently implemented approaches have proven principle functionality, but lack sufficient pulse detection ability in case of closely spaced consecutive hits, especially for wide pulses. This behaviour leads to hit counting inefficiency of the detector, referring to missed identification of sensors hits, which in return leads to errors in the luminosity determination.
The inefficiency, originating from the pulse identification implementations, creates the need for a new algorithm, that is capable of reliably identifying every sensor hit. This becomes even more important with respect to the upcoming upgrades of the LHC [12], which will further increase the luminosity.

A new algorithm that addresses the challenges of correct pulse identification by using the derivative of the input signal was developed and is being presented in the scope of this work. The purpose of this development is to further increase the accuracy of the luminosity measurement by increasing the peak finding capability of the detector.

3. Luminosity Measurement

3.1 The Concept of Luminosity

The luminosity is one of the key parameters in particle collider experiments. As proportionality factor between the event rate and the production cross-section of that event, the luminosity L quantifies the ability of a collider to produce certain particle interactions at a certain rate:

$$\frac{dR}{dt} = L \cdot \sigma_p \tag{3.1}$$

The number of observed particle interactions (events) per time interval is called the event rate, dR/dt. The cross-section σ_p describes the probability of the occurrence of a certain process, produced by the particle interaction. Each process has its own production cross-section.
In search for processes with very low production cross-section, physicists seek to increase the lumiosity to achieve higher event rates and thus better statistical information of that process. For two colliding bunched beams, as used in the LHC, the luminosity mainly depends on the number of particles per bunch and the transverse beam size at the interaction point. Therefore, a real-time feedback of the luminosity measurement is important for accelerator diagnostics and beam optimisation.
Additionally, an accurate measurement of the luminosity is essential to determine the production cross-section of observed events from the determined event rate. As can be seen from formula 3.1, every uncertainty of the luminosity measurement will translate directly to the uncertainty of the cross-section measurement

3.2 The Zero-Counting Algorithm

Any detector capable of measuring particle hit rates in linear dependency of the luminosity can be exploited as a luminometer. A way of evaluating the luminosity per bunch crossing from the measured hit rate is the zero-counting algorithm [13]. This algorithm is only applicable for detectors with low occupancy, referring to the probability of a hit in a bunch crossing. As described in [14], the number of hits per bunch crossing follows a Poissonian distribution, with the probability p of a certain number of hits n:

$$p(n) = \frac{\mu^n e^{-\mu}}{n!} \tag{3.2}$$

The mean value of occurring hits μ, also called occupancy, in a particular bunch crossing

can be determined by counting the number of bunch crossings without interaction (zero counting), when the probability of having two or more interactions per bunch crossing becomes non-negligible [13]. Then, the probability of $zero$ hits $p(0)$ in that bunch crossing can be determined by using the measured hit rate N_{hit}/N_{bc} in a particular bunch crossing [14]:

$$\mu = -ln\left[p(0)\right] = -ln\left[1 - p(n \neq 0)\right] = -ln\left[1 - \frac{N_{hit}}{N_{bc}}\right] \tag{3.3}$$

Eventually, the per-bunch luminosity L_{bunch} can be calculated using the μ of that bunch crossing, the detector-dependent visible cross-section σ_{vis} and the collider revolution frequency f, as [14]:

$$L_{bunch} = \frac{\mu}{\sigma_{vis}} \cdot f \tag{3.4}$$

The visible cross-section is a detector-dependent quantity and refers to all the cross-sections of the processes that are observable by the detector. Calibrating the luminometer and determining the visible cross-section is achieved in the so called Van-der-Meer scan [15]. This method creates special experimental conditions, where the two beams are scanned against each other with gradually increasing displacement. By measuring the hit rate as a function of the beam displacement in two planes, it is possible to calculate the visible cross-section.

3.3 Luminosity Measurement at CMS

The luminosity measurement at CMS is performed online and offline. The online, bunch-by-bunch measurement leads to a prompt result that is fed back to the experiment and the accelerator. At CMS several sub-detectors for measuring luminosity and machine induced background are developed and operated under the responsibility of the Beam Radiation Instrumentation and Luminosity (BRIL) project. An overview of some of the BRIL beam instrumentation detectors is depicted in figure 3.1, indicating their location in the CMS detector.
The Pixel Luminosity Telescope (PLT) [16] is an online luminometer, based on sensors and readout electronics similar to the ones used by the CMS Pixel Tracker. The Fast Beam Conditions Monitor (BCM1F) [17] is capable of measuring the luminosity, as well as the the machine induced background (MIB). MIB refers to non-luminosity products in the CMS detector that are created primarily by the the beam or beam-halo, interacting with residual gas in the collimators or the vacuum chambers. Additionally, the Hadron Forward (HF) Calorimeter [18] is exploited for luminosity measurement, using a dedicated readout.
To ensure a reliable luminosity measurement the detectors are operated independently, allowing a permanent cross check and fast trace back of possible detector malfunctioning. The common time interval for luminosity measurement at CMS is the so called *lumi nibble*, corresponding to 4096 orbits. The data taking and averaging period for one published online luminosity measurement is four lumi nibbles, resulting in a published value approximately every 1.457 seconds.

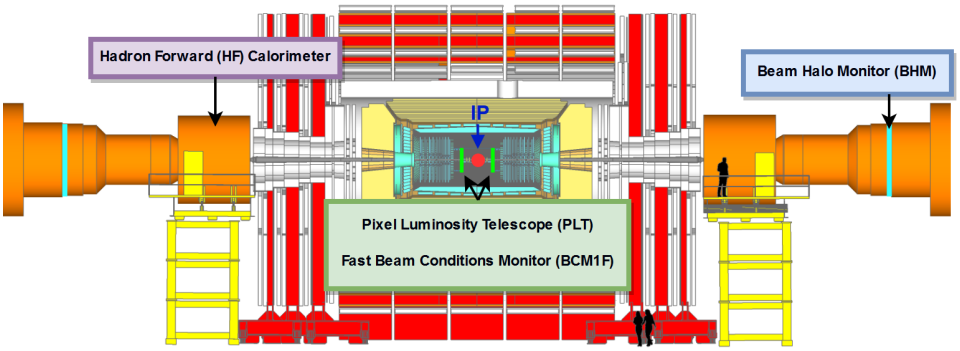


Figure 3.1: Overview of some of the BRIL sub-systems [8]

The offline calculation is based on the pixel cluster counting algorithm [19] of the recorded Silicon Pixel Detector data. Including applied linearity and efficiency corrections, it offers higher accuracy than the online measurement and is therefore used for every physics measurement. Nevertheless, its very high computational intensity due to the large data quantities renders the pixel cluster counting not suitable for fast results. For the Phase 2 upgrade of the CMS tracker [20], the feasibility of using the Inner Tracker Pixel Endcap for real-time luminosity measurement is currently investigated.

4. The BCM1F Detector

4.1 System Architecture

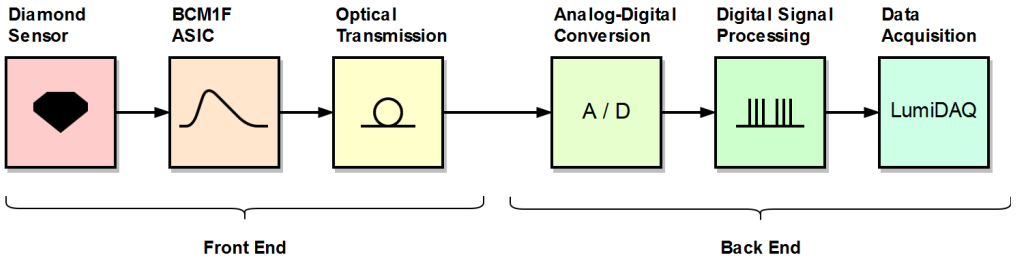


Figure 4.1: BCM1F MicroTCA Architecture block diagram

The Fast Beam Conditions Monitor (BCM1F) is an online detector system that measures the luminosity and machine induced background in the CMS experiment. A block diagram of the BCM1F readout system is provided in figure 4.1. The detector front end was initially designed for single-crystal Chemical Vapor Deposition (sCVD) diamond sensors [17] and can accommodate up to 48 independent channels. The sensors are read out by a custom analog front end ASIC, which amplifies and shapes the sensor signal. These signals are transmitted via an optoelectronic transmission chain to a dedicated back end electronics system, built in MicroTCA form factor. Here, the incoming signals are sampled, digitised and further processed in an FPGA and subsequently read out by the BRIL data acquisition software.
The BCM1F back end electronics system in fact comprises of two independent systems, one legacy VME-based and a newer MicroTCA system. The MicroTCA system will replace the VME system in the near future. It is always referred to in the scope of this work when mentioning the BCM1F back end electronics, unless otherwise stated.

4.2 Front End

The BCM1F front end electronics consist of the diamond sensors, read out ASICs and the optoelectronic transmission path. Per each detector side, the sensors and front end ASICs are mounted on two C-shaped, parallel flex-rigid PCBs perpendicular to the beam pipe, held by carbon-fibre carriages. These C-shapes are positioned around the beam pipe, thus placing the sensors radially at 7.2 cm away from the beam pipe, at a distance of $\pm$ 1.83 m from the interaction point (IP) [21]. This distance between the sensors and the IP is considered to be optimal to separate background and collision products

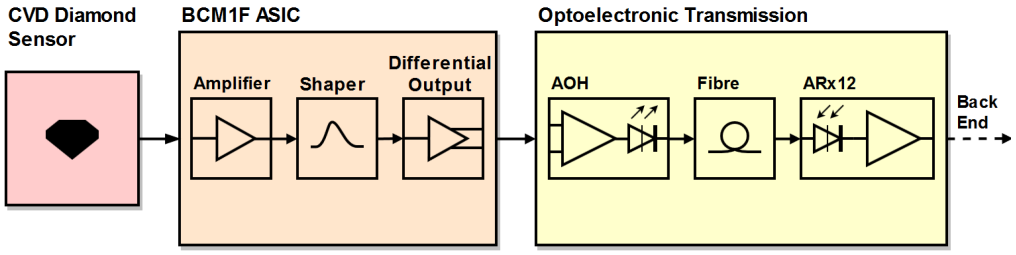


Figure 4.2: BCM1F Front End Electronics Block Diagram

according to their arrival time [22]. Additionally, the hit rates have been proven to be linear with luminosity [22], making this detector suitable for measuring luminosity. A block diagram of the front end electronics system is shown in figure 4.2.

4.2.1 Sensors

The advantage of diamond over silicon as sensor material is the unique radiation robustness of diamond, allowing a reliable particle detection in high radiation environments [23]. Diamond sensors have a band gap almost five times higher than silicon, leading to a very low dark current which does not increase with irradiation, in contrast to silicon [24]. This characteristic allows to omit active cooling of the sensors, which is a big advantage for the highly space-restricted BCM1F installation area. Additionally, diamond sensors have a very high response speed due to the high charge carrier mobility, making them suitable for particle arrival time measurements [24]. BCM1F incorporates single-crystal and poly-crystal CVD diamond sensors.

4.2.2 Front End ASIC

The custom developed BCM1F front end ASIC [25] is implemented in the IBM 130 nm CMOS RF process, which offers sufficient radiation hardness. It includes a fast transimpedance preamplifier with active feedback, a shaping stage and a high-performance differential output buffer. The ASIC was designed to achieve a quasi-Gaussian output response with a peaking time and pulse FWHM of less than 10 ns for up to 5 pF sensor capacitance, to allow the separation of at least two particle hits within one 25 ns bunch crossing cycle. For a sensor capacitance of 2 pF, the FWHM of the front end ASIC has been measured to be 9.2 ns in average, following the specification. The front end response to a 3 fC sensor input charge for different bias currents of the input transistors, measured with a sensor capacitance of 2 pF are depicted in figure 4.3. As can be seen, the pulse has a quasi-Gaussian shape with a fast rising edge and a slightly slower falling tail.

4.2.3 Optoelectronic Transmission

The analog signal is transmitted on an optical path from the inner CMS detector to the counting room, using the Analog-Opto-Hybrids (AOH) [26]. The ASIC output drives the voltage inputs of the AOH, where the signal is converted into a proportional current, to which an adjustable offset is added. The bias and gain of the AOH are adjustable through an I2C interface of the custom developed linear laser driver (LLD) [27]. A

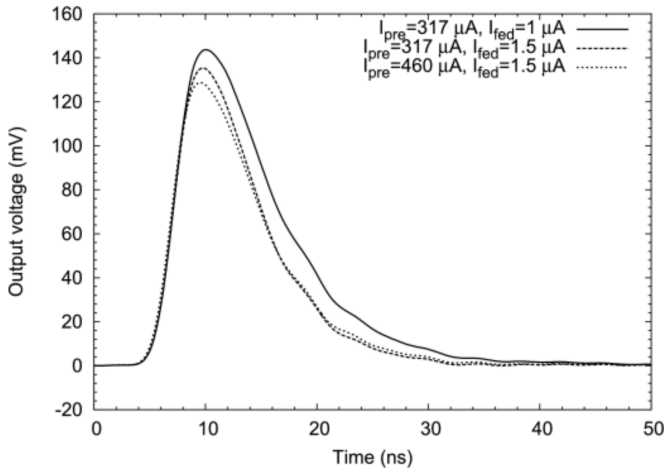


Figure 4.3: Output pulse of BCM1F ASIC for different biasing conditions and a sensor capacitance of 2 pF [25]

semiconductor laser diode located on the AOH PCB converts the current into infrared light at a wavelength of 1310 nm which is sent via an 80 m long singlemode fibre in pulse-amplitude-modulated format (PAM). In the counting room, located in an underground cavern close to the detector, the optical signal is received and converted into an electrical signal with an 8 bit equivalent dynamic range by the ARx12 optical receiver [28]. After electrical-to-optical conversion, transmitting the optical signal over singlemode fibre and reconverting the optical into an electrical signal in the ARx12 receiver, the FWHM of a test pulse was measured as 11.5 ns with an amplitude loss of 4.4 dB [29]. The AOH laser diodes suffer from effects of radiation damage, causing the bias setting to shift. Thus, a regular bias scan is required for correct adjustment of the AOH bias value.

4.3 Back End

The BCM1F back end electronic system is located in the CMS underground service cavern, protected from the hostile radiation environment of the CMS detector. It consists of high speed analog-to-digital converter FPGA Mezzanine Cards (FMC), mounted on Advanced Mezzanine Card (AMC) carrier boards, where the digitised data are processed in an FPGA. For the 48 read out channels a total of 12 FMC + AMC modules are mounted in two MicroTCA crates, each controlled by a MicroTCA Hub Module (MCH). Every module is synchronised with the LHC orbit and bunch clocks. The processed data are published to the Eventing Bus of the luminosity data acquisition software (LumiDAQ). The LumiDAQ software is responsible for producing and storing all online luminosity and beam background information.

4.3.1 FMC125 Analog-Digital-Converter

The FMC125, produced by 4DSP LLC (now Abaco Systems), is a four channel ADC FMC card, following the ANSI/VITA 57.1 FMC standard. It is hosting an E2V EV8AQ160 8-bit quad ADC with a sampling frequency up to 1.25 GHz and DDR LVDS outputs connected to the High Pin Count (HPC) connector. An on-board clock tree allows clock

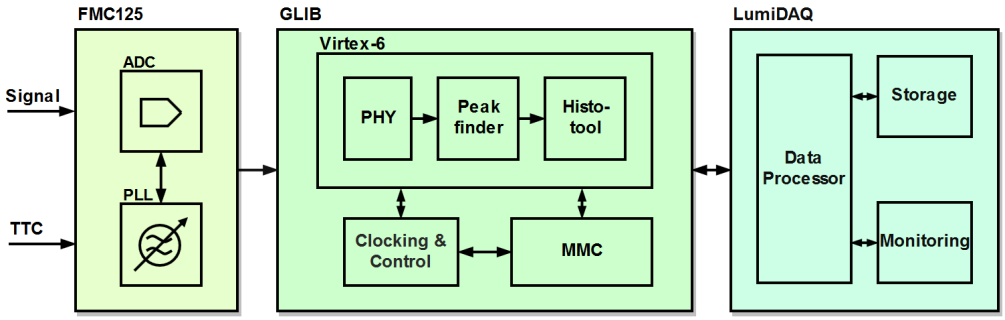


Figure 4.4: BCM1F Back End Electronics Block Diagram

signal generation and flexible synchronisation possibilities. The LHC bunch clock and orbit clock are connected at the front panel to the reference clock input and trigger input, respectively. The FMC125 boards have been modified, to allow the on-board PLL to lock to the bunch clock. Additionally, AC coupling of the ADC inputs has been implemented, using an RF-transformer with a bandwidth of 0.15 MHz to 350 MHz. The ADC clock is derived from the 40.08 MHz LHC bunch clock, using the AD9517-1 clock generator IC with integrated PLL for integer multiplication. The maximum number of samples per bunch crossing (Spbx) with a sampling frequency up to 1.25 GHz is

$$Spbx = \frac{1.25\ GHz}{40.08\ MHz} = 31.19 \tag{4.1}$$

As the PLL only allows for integer multiplication and the use of an even number eases the digital signal processing, 30 samples per second will be used, producing a sampling frequency of:

$$f_s = 30 \cdot 40.08\ MHz = 1.2024\ GHz \tag{4.2}$$

Each of the four ADC channels outputs the digitised data on a 16-bit Double Data Rate LVDS bus at half the sampling frequency.

4.3.2 GLIB Carrier Card

The Gigabit Link Interface Board (GLIB) [31] is a double width AMC, developed and produced by the CERN PH-ESE-BE Department. The card hosts a Xilinx Virtex-6 FPGA (XC6VLX130T), receiving the ADC data via the HPC FMC connector. A generic core for implementing the basic AMC functionalities is provided by the GLIB developers. Core and user code are controlled via 32 bit registers of the GLIB, modeled in XML files. The register resources can be accessed via the IPbus [32] protocol, using the provided classes for translating basic read and write operations. Clock distribution circuitry, a Gigabit Ethernet interface and a CPLD for configuration purposes are included on the GLIB AMC board. The Module Management Controller (MMC) is implemented on an Atmel AtMega128 microcontroller. It is used as communication interface of the AMC with the MCH, using the Intelligent Platform Management Interface (IPMI) standard [33]. More information about the MicroTCA standard, describing FMC, AMC and MMC specifications can be found in the PICMG MTCA.0 specification [34].
The Virtex-6 receives the ADC data with half the sampling frequency, two samples in

parallel for each channel. The samples are received in input double-data-rate (IDDR) registers that output four samples at approximately 300 MHz. The data are streamed into a FIFO that transmits the 30 samples of each bunch crossing with 40.08 MHz to the peak finder, to maintain the structure of all samples within one bunch crossing.

4.3.3 Peak Finding and Histogramming

Peak finding and histogramming is performed within the Virtex-6 FPGA. The task of the peak finder is to detect the presence of pulses in the digitised signal, that originate from particle hits. If one or more pulses are detected, the peak finder has to determine the relative timing of the peaks within that bunch crossing. This is important to distinguish hits that count for the luminosity measurement from MIB-hits. Additionally, the amplitude of the peaks is determined for monitoring the radiation damage of the sensors. The acquired information are passed on to the histogram module.
The histogram module counts the occurrence of pulses in every bunch crossing per orbit over an integration period of a lumi nibble (4096 orbits). The information of arrival time is stored in the so-called occupancy histogram, which provides the per-bunch-crossing hit count and is used for the luminosity calculation.

4.3.4 Data Acquisition

The BRIL DAQ software, also called LumiDAQ, is responsible for collecting the data from the BRIL sub-detectors. It is based on the XDAQ middleware platform [35] for the development of distributed data acquisition systems. Three components perform the BRIL data acquisition for each detector, using the publisher-subscriber concept. The data source performs the hardware configuration and readout of the back end system and publishes the histograms to the central Eventing Bus. The data processor is subscribed to one or more topics and gets notified when new data of this topic becomes available on the Eventing Bus. The processor performs local data aggregation and the results are published back to the Eventing Bus for use by other processors or storage. The central service provides data quality monitoring, storage service and data exchange with the LHC.

5. BCM1F Signal Chain Model

A good understanding of the signal chain is necessary for correct interpretation of the signals and for choosing adequate signal processing. Detailed models of the transmission path characteristics also are inevitable for creating accurate simulations that resemble the real signals as much as possible, within determined boundaries. Below, the known properties and characteristics of the signal transmission chain from the ASIC output to the ADC input are considered and modelled into a Python script.

5.1 Amplitude Frequency Response

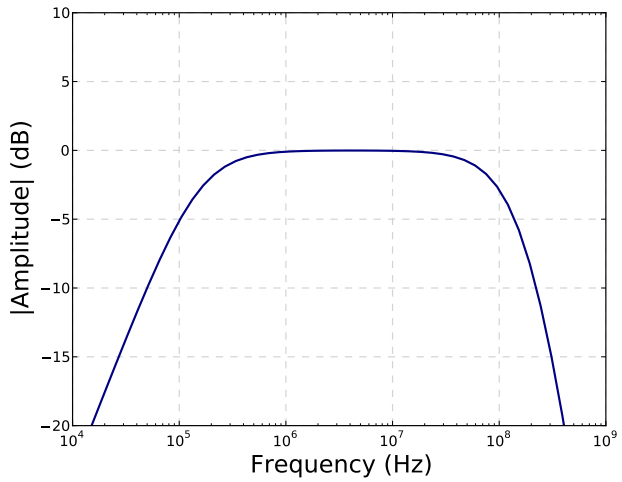
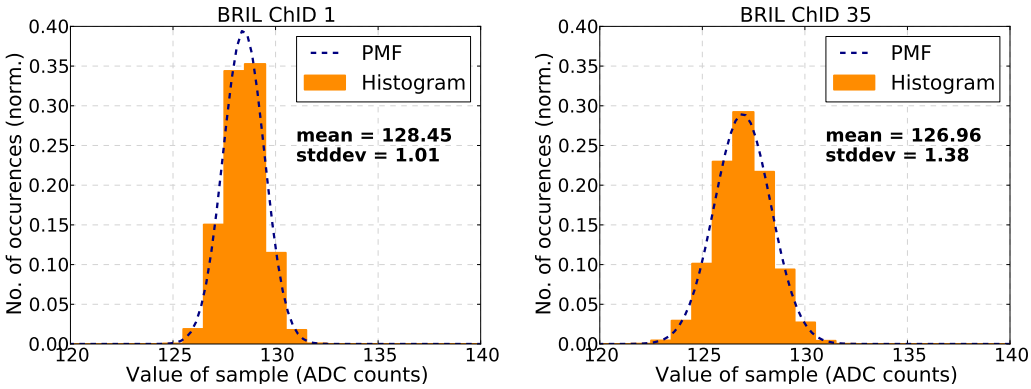


Figure 5.1: Amplitude frequency response of the BCM1F transmission path

The output signal from the ASIC is converted from an electrical into an optical signal in the AOH and transmitted via optical fibre to the ARx12 optoelectronic receiver, described in 4.2.3. As stated in [36], the optoelectronic data link has a second order low-pass-characteristic with a -3dB cut-off frequency at 110 MHz, mostly limited by the receiver bandwidth.
The receiver output is fed into the ADC input via a RF transformer. This customisation of the FMC125 board has been executed by the manufacturer. No detailed information about the RF transformer transmission behaviour are given, except the 3-dB insertion loss frequencies of 0.15 MHz and 350 MHz. Using the corner frequencies, the amplitude frequency response of the RF transformer is modelled as first order bandpass filter.



(a) pCVD diamond sensor **(b)** sCVD diamond sensor

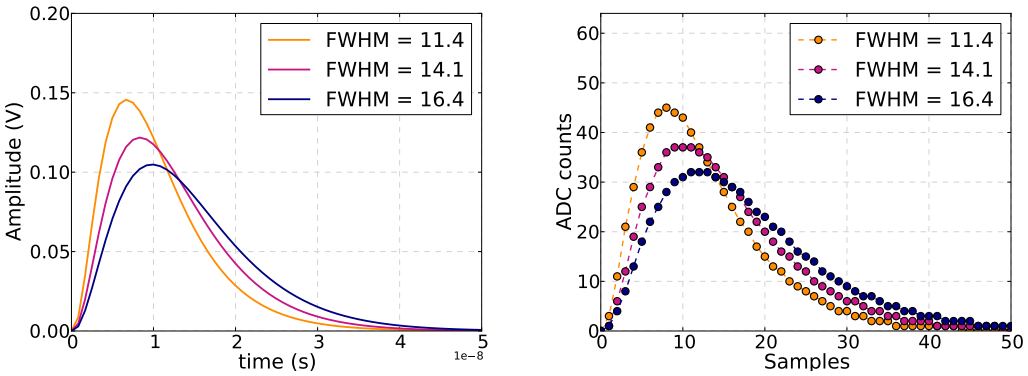
Figure 5.2: Measured baseline histogram and simulated PMF of two BCM1F channels

The high frequency cut-off of the system is mainly dominated by the second-order low-pass characteristic of the optoelectronic transmission path. The low frequency cut-off is solely determined by the RF transformer.
Using the available information about the signal transmission chain, the modelled amplitude frequency response of the system is shown in figure 5.1. The transmission path 3dB-bandwidth is ranging from 0.15 MHz to approximately 102 MHz.

5.2 Noise

To determine the properties of the noise of the digitised signal, histograms from static baseline measurements with over two million samples are created. Under the assumption of a Gaussian noise source, the means and standard deviations of the baseline signals are calculated from the histograms. The standard deviation corresponds to the RMS value of the noise [37]. The mean indicates the static baseline value of the signal. The measured noise includes all noise sources along the signal transmission chain.
Figure 5.2 shows the histograms of two BCM1F channels, normalised to the total number of occurrences. It can be seen that both channels have different mean and standard deviation values. With an AC-coupled ADC input, the mean value of the baseline should be in the middle of the ADC input range, corresponding to 128 ADC counts for an 8 bit ADC. Due to the non-calibrated offset of the ADCs, the mean differs from channel to channel, explaining the deviation of both presented channels. The difference between the RMS noise of both channels most likely originates from the unequal radiation damage of the sensor and other detector parts, as well as the different sensor characteristics. For reliable conclusions, the noise properties have to be evaluated in regular time intervals for every channel separately, as the noise is expected to increase with progressing irradiation.

In the software simulation, the noise is modelled using the Numpy random.normal() [38] routine, that takes as input arguments the calculated mean and standard deviation of the histograms. The Probability Mass Function (PMF) of the generated noise is calculated and depicted in 5.2. The PMF describes the probability of the occurrence of a certain value [39].



(a) Simulated ASIC output pulses **(b)** Simulated digitised pulses

Figure 5.3: Simulation of ASIC output pulses for different FWHM and an input charge of $Q_{in} = 3\ fC$

5.3 ASIC output signal

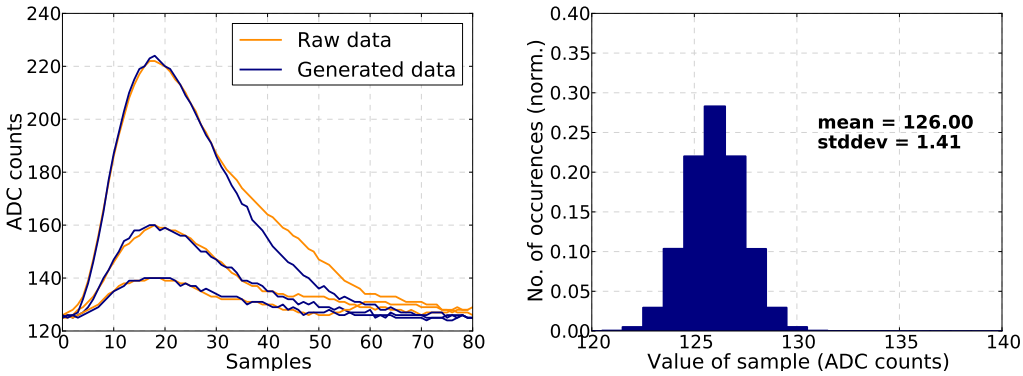
The time response of the BCM1F ASIC is given by the inverse Laplace transformation of the transfer function as provided and explained in detail in [29] and [40]. This allows the precise reconstruction of the ASIC output signal for different input charges and is implemented in a Python function. As stated in chapter 4.2.2, the FWHM of the ASIC pulses is specified to be less than 10 ns. This only applies for very narrow input charge pulses. For the diamond sensors, the output pulse FWHM has been observed to be significantly higher in various BCM1F channels. This non-ideal behaviour should be accounted for in the simulation, with a FWHM up to 15 ns.
Using the calculated time response of the ASIC, figure 5.3a shows output pulses for different FWHM and an input charge of 3 fC.

5.4 Digitisation

The digitisation is simulated by sampling the signal in time steps of the inverse sampling frequency and calculating the corresponding ADC reading for an 8-bit ADC with an input full scale range of $0.5V_{pp}$:

$$ADC\ Reading = Round\left(G \cdot Signal\ Voltage \cdot \frac{ADC\ Resolution}{Input\ Range}\right) \tag{5.1}$$

The gain G implements the amplitude loss of the signal transmission (see 4.2.3), that has not been captured by the modeled transfer function. Figure 5.3b shows the applied simulated digitisation of the simulated pulses in figure 5.3a.
The described approximation to the digitisation process does not take into account any offset, gain errors or non-linearities of the ADC. Typically, the ADC has a differential non-linearity of $\pm$ 0.18 LSB and an integral non-linearity of $\pm$ 0.4 LSB [41].



(a) Raw and generated pulses **(b)** Baseline histogram of generated noise

Figure 5.4: Comparing raw data with simulated data from the BCM1F software model

5.5 Software model

The BCM1F software model was implemented to generate simulated data for testing and comparison purposes. The ASIC output pulse simulator calculates the ASIC response for a given input charge in time steps that are much higher than the inverse sampling frequency of the ADC. The number of pulses and the FWHM of these pulses are provided as parameters. The ASIC output pulses are convoluted and fed into the transmission path model. Subsequently, digitisation for an 8 bit ADC is modelled. As the noise in section 5.2 was determined for the digitised signal with a resulting RMS noise in ADC counts, the generated noise is added at the end of the signal simulation.

Figure 5.4a shows raw data of BCM1F BRIL ChID 35, Fill 6371 and simulated data pulses that were generated by using the BCM1F software model. The input charges and pulse FWHM were adjusted to match the raw data pulses. It can be seen that the rising edges of the generated pulses resemble the rising edges of the raw data pulses. For the biggest pulse, the raw data pulse has a longer falling edge than the generated pulse. For the smaller pulses, a difference in the falling edge is not obvious. The noise of the signal was modelled using the calculated mean and standard deviation from figure 5.2b. Figure 5.4b shows the corresponding baseline histogram. The generated noise has a lower mean value and a slightly lower standard deviation than the raw data baseline, but still serves as a good approximation.
The BCM1F model is capable of creating signals that have high similarity to recorded data. By adjusting the different input parameters such as input charge, pulse FWHM and noise distribution it is possible to generate realistic pulses for the different characteristics of the BCM1F channels.

6. Discriminator Based Hit Rate Measurement

6.1 VME System Pulse Detection

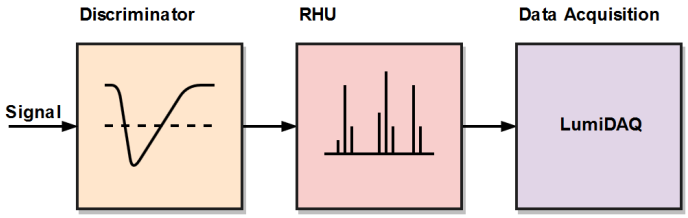


Figure 6.1: VME system block diagram

As stated in chapter 4, the BCM1F detector consists of two independent back end systems. While the MicroTCA system is currently under development, the VME back end is the operating system and therefore responsible for reading out the sensors as well as creating the occupancy histograms for the calculation of the luminosity. The signal chain is common for both back end systems from the sensors to the optical receiver. The ARx12 outputs the received signal with positive polarity to the MicroTCA system and the VME system receives the same signal with negative polarity. Hit counting within the VME system is based on a fixed-threshold discriminator and a custom developed histogramming module, as depicted in figure 6.1. The analog signal from the optical receiver is fed into the leading edge discriminator module V895 [43] from CAEN S.p.A., that creates a logical output signal if the input crosses a configurable threshold. The discriminator output is connected to the Real-time Histogramming Unit (RHU) [42], which samples the time of the threshold crossing relative to the orbit clock and builds the occupancy histogram with 6.25 ns bin width.
Therefore, the VME system does not perform any peak finding, but only counts the crossings of a static threshold. This can lead to undercount for pulses below the threshold or in case of pulse pileup as well as to overcount in case of a noise spike crossing the threshold. An additional limit of the VME system is, that after the discriminator module detected a threshold crossing, it is not able to detect another crossing for the next 12 ns, due to the discriminator implementation [43].
Figure 6.2 visualises the inefficiency of the fixed threshold discrimination for two piled-up pulses and one single pulse. With a discriminator threshold of -10 mV, the first and the last pulse are detected by the threshold crossings. The second pulse is not detected due to the pileup, that causes the signal to stay below the threshold between the first

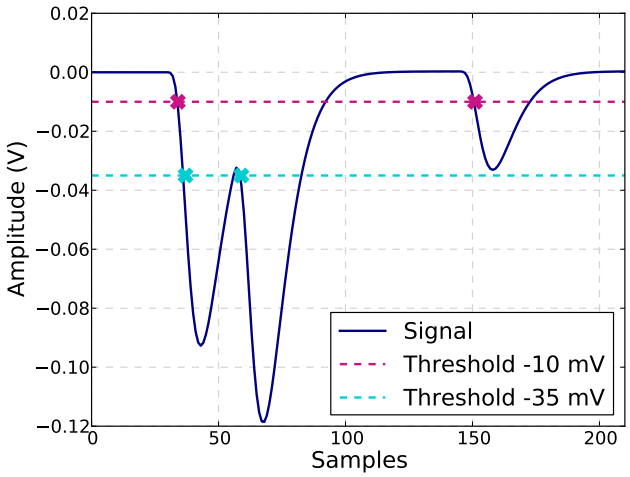


Figure 6.2: VME fixed threshold discrimination pulse counting

and the second pulse. This could be avoided by setting a lower threshold, so that the signal goes above threshold between both pulses. However, with a threshold of -35 mV the last pulse is not detected as is it too small and thus does not cross the threshold. By setting a fixed threshold, a tradeoff between finding piled-up pulses and detecting smaller pulses has to me made. This will necessarily lead to undercount.
A study of the BCM1F signals has shown that situations of pileup of multiple pulses, with the signal staying below threshold for several 100 ns occurs often, especially for the pCVD sensors [44]. In these situations, the VME system is only able to detect the threshold crossing of the first pulse and does not count the other occurring pulses.

6.2 Discriminator Based Peak Finder

For emulating the legacy system behaviour, a discriminator based peak finder was implemented in VHDL for integration into the GLIB firmware of the BCM1F MicroTCA system. It examines one bunch crossing at a time, but also buffers the next bunch crossing, to cover the case of pulses that do not end in the examined bunch crossing. At first, the module builds an array that flags all samples of the examined and next bunch crossing, that exceed a configurable *value threshold*. By detecting a positive and negative threshold crossing, it determines the range of consecutive samples above the threshold. If in the examined bunch crossing a positive threshold crossing is detected, but not a negative one, the algorithm is able to detect the negative threshold crossing in the next bunch crossing. In case of a detected positive and negative threshold crossing, the number of samples within that range is determined. If the number of samples exceeds the *time-over-threshold*-threshold, the sample with the maximum value is determined, using a binary search. The position and value of this sample are the output of the discriminator based peak finder. The application of the *time-over-threshold*-threshold prevents overcount due to noise spikes crossing the *value threshold* for a short time. The implementation only allows to detect a maximum of one peak per bunch crossing.

The discriminator based algorithm proves functionality and reliable peak determination

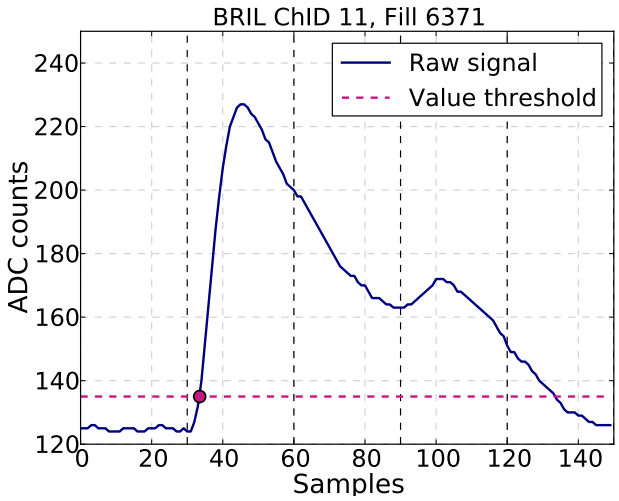


Figure 6.3: Two missed pulses due to pileup effects

for single, isolated pulses. However, it has some inherent drawbacks, that can lead to missed peak detection and thus contribute to undercount and detector inefficiency. An example is given in figure 6.3. It shows the raw data of a detector channel with sCVD diamonds, containing two pulses that are two bunch crossings apart. As the first pulse has an unusually large amplitude (likely a heavy particle) and thus a long falling edge, there is no gap between the peaks that goes below the threshold due to pileup of both pulses. With a *value-threshold* of 135 ADC counts, the discriminator based algorithm detects a positive threshold crossing before the first peak, but no negative crossing in the same or the next bunch crossing. Thus, the peak finder is not able to detect the first pulse, although the pulse is rather large in amplitude. Also the second, piled-up pulse, is not detectable by the peak finder, because it starts with samples that are already above the threshold and thus no positive threshold crossing can be detected. The only option to detect these pulses with the discriminator based algorithm is to increase the value-threshold. However, this would render the peak finder unable to find pulses with lower amplitudes.

The above characteristics, inherent to the algorithm and its implementation, lead to an undercount of the sensor pulses and corrupt the luminosity-measurement-accuracy of the detector. Additionally, the peak amplitude determination of the algorithm is directly affected by baseline shifts.

7. Derivative Based Peak Finding Algorithm

The direct proportionality between luminosity and hit rate is exploited to measure the luminosity with the BCM1F detector. Calculating the luminosity is done by applying the zero-counting algorithm as described in chapter 3. To achieve an accurate measurement result, every sensor hit per bunch crossing has to be detected and characterised. This is the task of the peak finding module.

The properties of the VME system and the discrimination based peak finder, discussed in chapter 6, are unsuitable for an accurate luminosity measurement at present and future conditions, due to the inefficiencies at pulse pileup.
To guarantee reliable hit counting, a derivative based algorithm was developed, addressing the aforementioned inefficiencies. Using the derivative of a signal, local maxima can be easily located by detecting descending zero-crossings. The differentiated signal is not influenced by any DC components and suppresses low frequency signals, such as baseline shifts. This makes the derivative a useful tool for detecting peaks in the BCM1F signal.

7.1 Smooth Noise-Robust Differentiator

Using the derivative of a signal, which per definition indicates the slope of the signal, the peaking time of a pulse is determined by detecting the descending zero-crossing of the differentiated signal. The magnitude response of an ideal differentiator is proportionally rising with the input frequency. On one hand, this behaviour is advantageous for suppressing the low frequency baseline shifts. On the other hand, this leads to an accentuation of the high frequency noise, thus decreasing the signal-to noise ratio (SNR).Therefore, the high frequency noise components need to be attenuated by applying low pass filtering techniques.

A wide spread approach for differentiating and low-pass-filtering a signal using a relatively simple computing scheme is the Savitzky-Golay Filter [45]. Due to the ripples in the magnitude response at high frequencies, this filter does not provide complete noise suppression in the higher frequency range.

A differentiator with implied low pass filtering is proposed by Pavel Holoborodko [46]. This so called smooth noise-robust differentiator (SNRD) guarantees noise suppression at higher frequencies while, at the same time, providing an efficient computational structure. In table 7.1 the smooth noise-robust differentiator for three different window sizes

N	SNRD
5	$\frac{2(f_1-f_{-1})+f_2-f_{-2}}{8h}$
7	$\frac{5(f_1-f_{-1})+4(f_2-f_{-2})+f_3-f_{-3}}{32h}$
9	$\frac{14(f_1-f_{-1})+14(f_2-f_{-2})+6(f_3-f_{-3})+f_4-f_{-4}}{128h}$

Table 7.1: Smooth noise-robust differentiator formulas for N=5,7,9 [46]

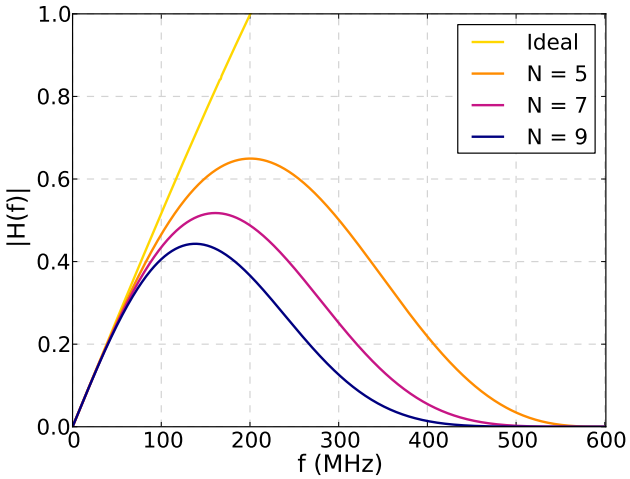


Figure 7.1: Magnitude response

N are presented. The window size, also called smooth width or filter length, defines the number (N-1) of adjacent samples, that are symmetrically located around the point for which the SNRD is calculated for. Therefore, it needs to be selected in accordance with the signal bandwidth and the desired amount of noise suppression.
The magnitude response of the smooth noise-robust differentiator is visualised in figure 7.1, using the sampling frequency of the FMC125 of $f_s = 1202.4\ MHz$. It can be seen, that for low frequencies the response behaves like the ideal differentiator (yellow). For higher frequencies the magnitude response flattens and completely suppresses frequencies towards the Nyquist-frequency. As the system bandwidth has the high cut-off at approximately 102 MHz, it is desireable to have a linear response in that region and a noise suppressing behaviour for higher frequencies.
The SNRD provides high frequency filtering and differentiation of the signal, while being efficiently implementable into a digital system and is therefore the preferred choice for the derivative based peak finding algorithm.

7.2 Algorithm Description

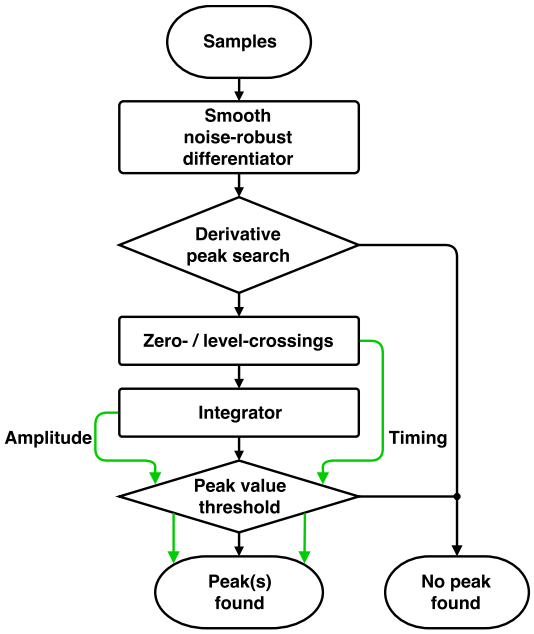
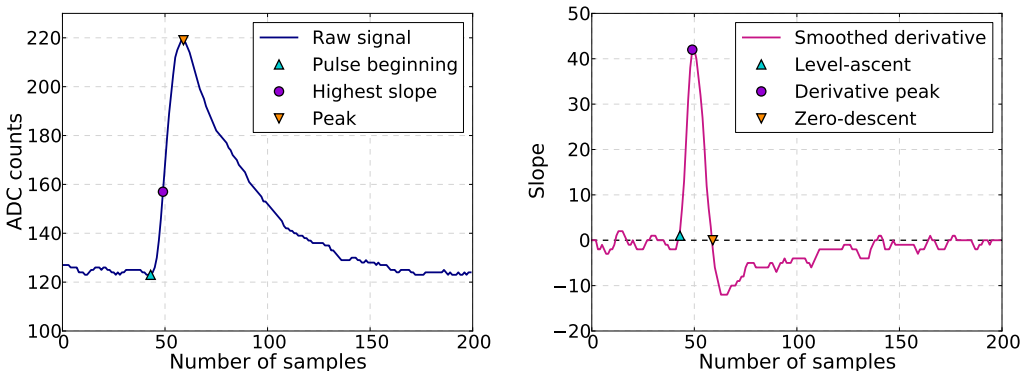


Figure 7.2: Peak finding algorithm flow chart

The derivative based peak finding algorithm is illustrated as a flow chart in figure 7.2. Below, the steps of the algorithm are explained in detail and an applied example using acquired raw data is given in figure 7.3. The algorithm is run for one bunch crossing at a time.

1. The derivative of the bunch crossing under examination is calculated using the smooth noise-robust differentiator. For the calculation of the first two and last two sample points of every bunch crossing, the last two samples of the previous and first two samples of the next bunch crossing are taken into account (for N=5). This ensures seamless differentiation of all sample points in one bunch crossing.

2. The derivative of a pulse, originating from a particle hit, itself consists of a positive pulse, followed by a negative pulse. The peak of the positive derivative pulse indicates the point of the fastest rising slope of the original pulse signal. Because of the constant rise time of the pulses, provided by the ASIC shaper stage, it is possible to discriminate against slow pulses that are not originating from a particle hit. Therefore, the algorithm detects peaks in the derivative by applying a configurable threshold. If a sample is exceeding the derivative threshold, it is counted as a derivative peak when the adjacent previous and following configurable number of samples have a lower value than the examined sample point. The algorithm also takes into account, that the peak can consist of two or more samples with the same value. In this case, the peak is determined as the first sample of the flat top. In figure 7.3, the determined peak in the derivative is superimposed to the original data, indicating the fastest slope of the rising edge, both marked as violet circles.



(a) Raw signal pulse

(b) Smoothed derivative of raw signal pulse

Figure 7.3: Applied peak finding algorithm, BCM1F BRIL ChID 1, Fill 6371

3. From the position of every located derivative peak, the algorithm is detecting an ascending level-crossing happening before the peak and a descending zero-crossing after the peak. The ascending crossing of a positive level close to zero just before the derivative peak marks the beginning of the original pulse, just as it starts to rise with a certain slope. In figure 7.3b the ascending crossing of a level of 1 ADC count is indicated with a turquoise triangle, corresponding to the pulse beginning in figure 7.3a, marked in the same way. The descending zero-crossing after the derivative peak indicates the peak position of the original pulse, as indicated by the orange triangles in figure 7.3. The index of the descending zero-crossing is stored to determine the peak position. Because a background particle hit can occur at any time within the bunch crossing, the algorithm is also able to detect zero- and level-crossings at the transition from the previous and to the following bunch crossing.

4. When the positions of the ascending and descending zero- and level-crossings are determined, the derivative will be integrated between these points. This gives the amplitude of the original signal peak at the position of the descending zero-crossing, neglecting the amplitude offset due to baseline shifts.

5. A threshold is applied to the calculated peak amplitude to discriminate against very small pulses, that are unlikely to originate from a particle hit. In case of a crossed threshold, the amplitude will be stored together with the corresponding timing value and these data are passed on to the histogram module.

7.3 Algorithm Parameters

The parameters of the peak finding algorithm, presented below, have to be understood in detail and carefully chosen to achieve the desired peak finding performance. Due to progressive radiation damage of components and differences in diamond sensor characteristics, most of the parameters for the peak finder have to be adjusted for every channel separately. Also, the planned substitution of the diamond sensors with silicon

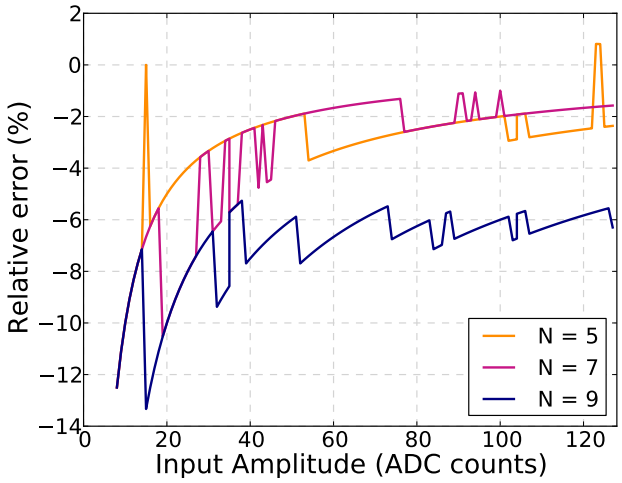


Figure 7.4: Relative error of computed amplitude for different SNRD *window size* for a pulse with FWHM = 10 ns

sensors will require adaptation of the algorithm parameters, with respect to changing sensor parameters.

7.3.1 Window Size

The *window size* of the smooth noise-robust differentiator determines the noise suppression and has to be chosen in accordance with the system bandwidth to ensure linear differentiation in the pass band. As the BCM1F system bandwidth has the high cut-off at 102 MHz, it is desireable to have a linear response in that region and a noise suppressing behaviour for higher frequencies. Nevertheless, the magnitude response for all three *window sizes* starts to flatten within the system bandwidth. This leads to an attenuation of the amplitude which is higher for bigger *window sizes*. Figure 7.4 reflects this behaviour, as it shows the relative error of the amplitude computed by the algorithm, to the given amplitude of an input pulse with FWHM = 10 ns. For achieving the best amplitude determination accuracy, the smallest window size should be chosen. As an accurate amplitude measurement is not strictly necessary for luminosity measurement, a higher window size for improved noise suppression can be advantageous. For pulses with higher FWHM, the relative error might be smaller than in the above figure, as wider pulses have a narrower frequency spectrum and thus the attenuation of the higher frequencies due to the SNRD response would have a lower impact on the amplitude.

7.3.2 Derivative Threshold

The *derivative threshold* is one requirement to detect a peak in the derivative, that is originating from a fast rising pulse, such as from a particle hit. This can be used to discriminate against slow signals, for example baseline shifts and therefore needs to be adjusted in accordance with the rise time of the pulses. In theory, the front-end ASIC should guarantee a constant rise time of about 10 ns independent from the pulse amplitude. This implies that the amplitude of the pulse derivative depends on the pulse

amplitude. Thus, the *derivative threshold* has to be set in accordance with the desired minimum detectable amplitude.

7.3.3 Peak Isolation

The *peak isolation* is a criterion to determine if a point in the derivative that crossed the *derivative threshold* is an actual peak, by ensuring its isolation. This value determines the number of preceding and succeeding data points that have to have a smaller value than the point which qualifies as a derivative peak candidate by crossing the *derivative threshold*.

7.3.4 Amplitude Threshold

The *amplitude threshold* depends on the responsiveness of the sensor and the expected minimal detectable amplitude without baseline shift. This threshold is helpful to discriminate against small but fast pulses, that crossed the *derivative threshold*. Such pulses would be falsely counted as hits when using simple thresholds in the raw data. Noise induced signals for example, which can be narrow spikes with fast edges but a small amplitude, can successfully be rejected by the *amplitude threshold*. The *amplitude threshold* should be set as the minimum amplitude that is likely to origin from a particle hit.

7.4 Implementation

The algorithm was first implemented in Python 2.7. After thorough testing and optimisation, the algorithm has proven concept with simulated data.
The target device for the peak finder implementation is a Xilinx Virtex-6 (XC6VLX130T) FPGA, located on the GLIB AMC card. Firmware implementation for the GLIB is done in VHDL-93, using the Xilinx ISE 14.7 development environment with Mentor Graphics Modelsim as simulator. The peak finder firmware has to be integrated into the user logic part of the GLIB firmware.

7.4.1 Requirements

As one GLIB card hosts four detector channels, the peak finder module needs to be instantiated four times per card. Due to the ongoing development of the BCM1F user logic, a precise specification of available resources for the peak finder is not feasible. Nevertheless, some coarse constraints for the implementation are identified. The use of Block RAM is highly restricted and should be limited to a bare minimum, as they are required for the histogram module and raw data storage. Operations that use a lot of LUTs, such as arithmetic operations, shall be implemented in the Virtex-6 DSP slices to spare available logic resources. Although the DSP slices are not used by other firmware components, a margin of 10 % of DSP slices should be left free.
Available clocks within the user logic are the bunch clock (40.08 MHz) and a 80.16 MHz clock. Histograms are integrated over a period of one lumi nibble, thus 364.2 ms. The histogram module has implemented double buffering, giving the peak finder the opportunity of a maximum latency of one lumi nibble. This sets very low constraints on

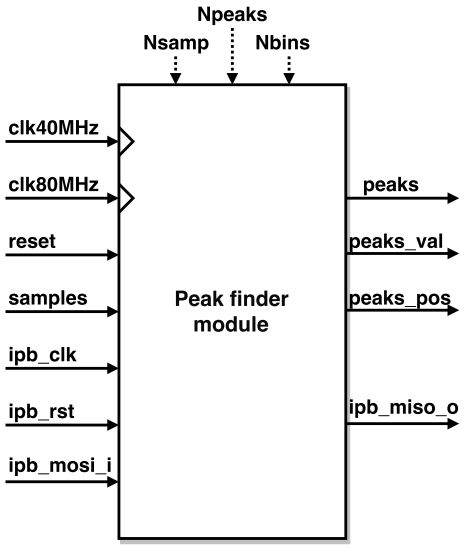


Figure 7.5: Peakfinder interface diagram of one instance

the latency of the peak finder, which then is only limited by the available registers that are necessary to buffer the amount of data.

7.4.2 Interface

A block diagram of the VHDL implementation of the peak finder module is shown in figure 7.5. The generics inputs and the ports are described below and are summarized in Table 7.2. The modules clock and sample inputs are driven by the GLIB firmware ADC-data-receiver module, the IPbus ports are driven by the IPbus core inside the GLIB system logic. The outputs of the peak finder are connected to the histogram module.

Generics

In VHDL, generics are used to parametrise a design at synthesis-time. As the ADC is intended to run at a sampling frequency of 1.2024 GHz, the number of samples per bunch crossing is normally set to 30. However, the module is capable of processing a lower even number of input samples per bunch crossing by setting *Nsamp*, to provide the possibility of using lower sampling frequencies.
The maximum number of detectable peaks within one bunch crossing can be adjusted via *Npeaks*. As the utilisation of most resources scales with this parameter, shown in figure 7.6, *Npeaks* should be set in accordance with the available FPGA resources. A setting of $Npeaks > 4$ is not feasible due to resource limits and also does not seem reasonable with respect to the probability of such an event.
The possibility of binning the detected peak position is provided for data reduction and ease of the luminosity computation in the XDAQ processor. *Nbins* sets the vector size of the output port *peaks_pos* and thus sets the maximum number of bins.

Generic	**Description**	
Nsamp	Number of input samples per bunch crossing	
Npeaks	Number of detectable peaks per bunch crossing	
Nbins	Bin size for peak position output	
Port	**Description**	**Direction**
clk40MHz	40.08 MHz clock	in
clk80MHz	80.16 MHz clock	in
reset	Synchronous reset	in
samples	*Nsamp* bytes of input samples	in
ipb_clk	IPbus clock	in
ipb_rst	IPbus reset	in
ipb_mosi_i	IPbus master-out slave-in	in
ipb_miso_o	IPbus master-in slave-out	out
peaks	Indicator of number of detected peaks	out
peaks_val	Detected peak amplitude values	out
peaks_pos	Binned position of detected peaks	out

Table 7.2: Peak finder interface port description

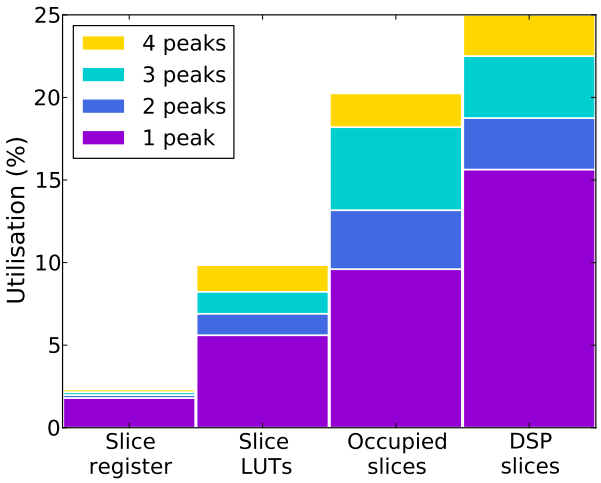


Figure 7.6: Resource utilisation for different settings of *Npeaks*, for one peak finder instance

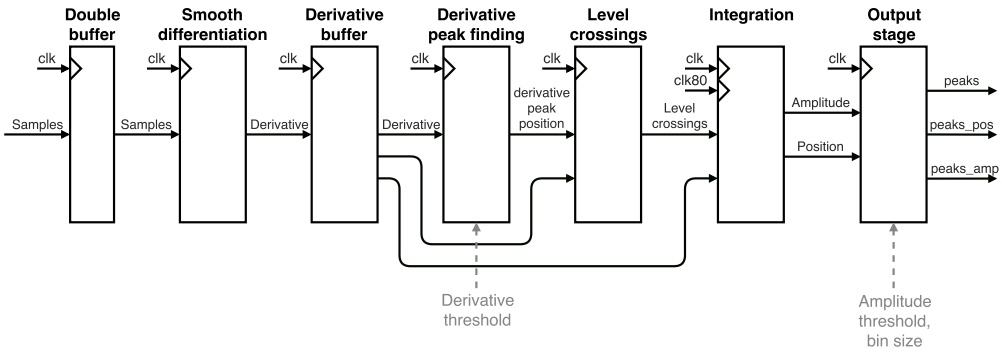


Figure 7.7: Pipeline stages of the peak finder module

Ports

The peak finder makes use of both the 40.08 MHz and 80.16 MHz clocks, provided at the *clk40MHz* and *clk80MHz* inputs.
Synchronous reset is used throughout the whole GLIB firmware and is also applied at port *reset*.
The input samples for one bunch crossing are provided via *samples*, synchronised to the bunch clock.
As interface to the histogram module, *peaks* indicates the number of detected peaks, whereas *peaks_val* provides the amplitudes and *peaks_pos* provides the (binned) position information. All three outputs have a vector size equal to *Npeaks*.
An IPbus module has been integrated to enable software configurability of the module parameters through the ports *ipb_clk*, *ipb_rst*, *ipb_mosi_i*, *ipb_miso_o*. The IPbus clock and reset signals, as well as the input for the data ports are provided and implemented in the GLIB system logic.

7.4.3 Module Design

As the latency requirements are very low, multi-stage pipelining is implemented. This eases the routing of the design and allows for comfortable timing margins at the foreseen clock frequencies. The pipeline stages, shown in figure 7.7, mostly resemble the different stages of the algorithm, however some of the pipeline stages are also pipelined internally. With the implementation of the pipeline stages, as explained below, the module has a latency of 12 bunch clock cycles. The stages are described as follows:

- The samples are fed into a double buffer register, which also provides buffering for the last samples of the previous bunch crossing for seamless derivative calculation.
- In the next stage, the smooth noise-robust differentiator is applied to the samples. The computation is implemented in DSP48E1 blocks for add- and subtract-operations, the multiplication and division are implemented using shift operations. Block diagrams for the SNRD computation for two different *window sizes* can be found in the appendix.
- The derivative is triple buffered to provide the differentiated signal to the following stages. This allows for detecting pulses that start in the previous or end in the next bunch crossing, with respect to the bunch crossing under examination.

Address	Size	Description
0	[7:0]	Derivative threshold
1	[4:0]	Peak isolation
2	[7:0]	Amplitude threshold
3	[31:0]	Binning LUT 1
4	[31:0]	Binning LUT 2
5	[31:0]	Binning LUT 3

Table 7.3: IPbus registers

- Subsequently, the implementation is looking for peaks in the derivative using comparators, taking into account the *peak isolation* parameter. The positions of detected derivative peaks within the bunch crossing are passed on to the next stage, if they are crossing the *derivative threshold*.
- The ascending level-crossing, as well as the descending zero-crossing are determined. The implementation also takes into account the previous and next bunch crossing, provided by the derivative buffer, to account for the case that level- and/or zero-crossings are happening outside the examined bunch crossing.
- The integration of the previously determined range is implemented using pipelined DSP blocks, calculating the amplitudes of detected peaks.
- If the computed amplitude crosses the *amplitude threshold*, the peak positions and amplitudes will be outputted. For the luminosity calculation it is necessary to bin the peak positions, possibly with differently sized bins. Therefore a look up table is implemeneted, to provide parameterised bin sizes.

The parameters listed in table 7.3 are configurable for every peak finder instance via software. This is done by implementing an IPbus slave, where each parameter is represented in a 32-bit IPbus register with read-write access. Modelled in XML files, the registers can be accessed by software during run time.

8. Simulation Results

8.1 Simulation Setup

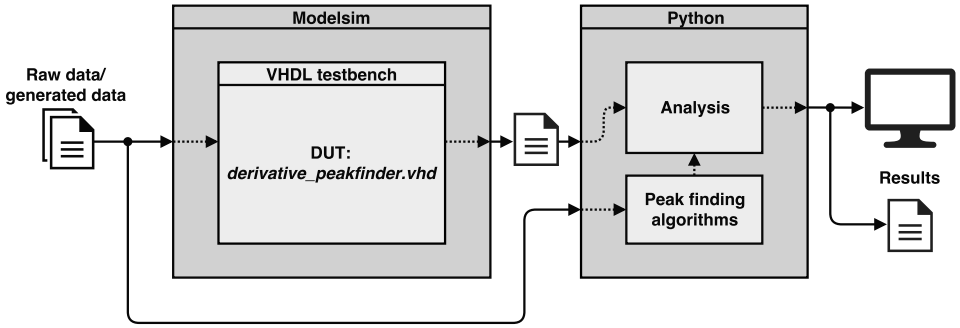
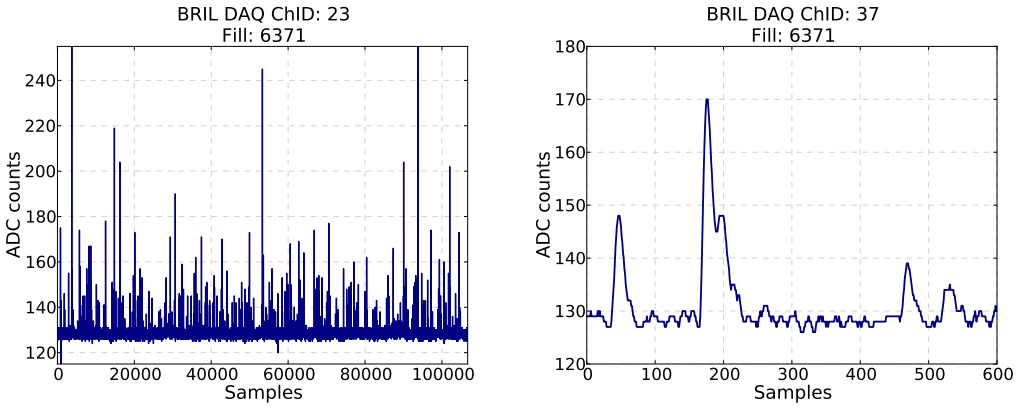


Figure 8.1: Simulation setup

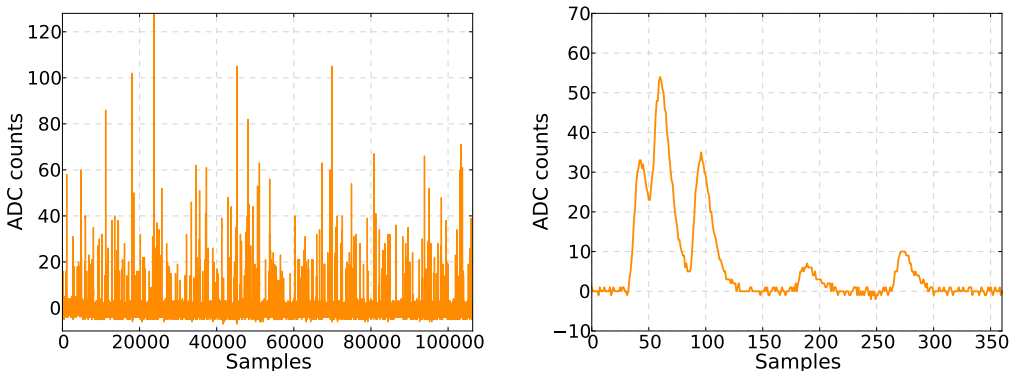
By the end of 2017 the LHC entered the Year-End Technical Stop, which lasts until spring 2018. In this period no beam is circulating in the accelerator and thus no collision data can be taken. Before the technical stop, raw data of the BCM1F detector was acquired with the MicroTCA system. As the firmware was not fully tested for a sampling frequency of 1.2024 GHz at that point, some data was corrupted and has been rated as invalid. Nevertheless, raw data of 8 different channels from Fill 6371 has been saved. Using this data, the implementation of the derivative based peak finding algorithm can be tested with real signals, without the need of an operational setup and collider.
By using the software model described in chapter 5, it is also possible to generate files with simulated data for evaluating the peak finder functionality. In this way, a known number of pulses as well as scenarios with multiple hits per bunch crossing can be created on demand, taking into account the transmission path characteristics and the possibility to add normal distributed noise. The advantage of the generated files is that the number, timing and amplitude of the pulses are known, for direct comparison with the peak finder output. When evaluating the peak finder output for raw data input files, the number of sensor hit pulses is ambiguous.
The setup of the simulation is shown in figure 8.1 and contains the raw and generated data files, the VHDL testbench of the module and Python scripts for comparison and analysis results.

The data files can be accessed by the VHDL testbench by converting them to TXT-files and using the TEXTIO package [47] that allows the reading and writing of ASCII text files from VHDL. The files are read into the testbench in chunks of 30 samples,



(a) One orbit of raw data for sCVD diamond **(b)** Raw data snippet for pCVD diamond

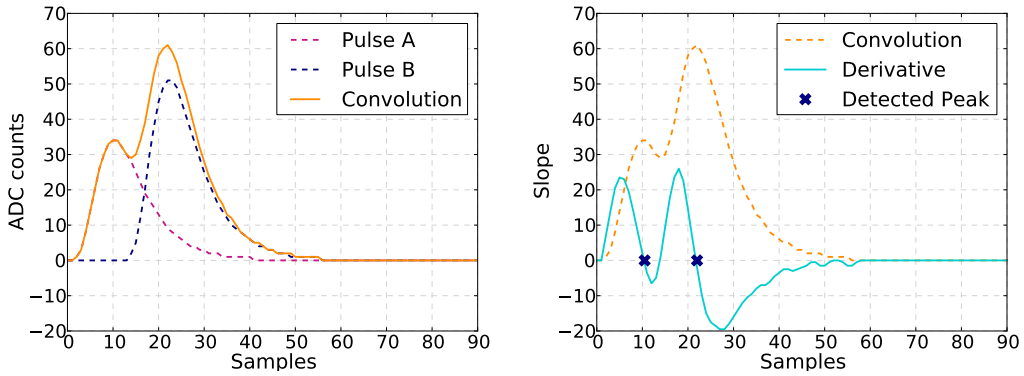
Figure 8.2: Raw data signal from Fill 6371



(a) One orbit of simulated data **(b)** Convolution of simulated pulses

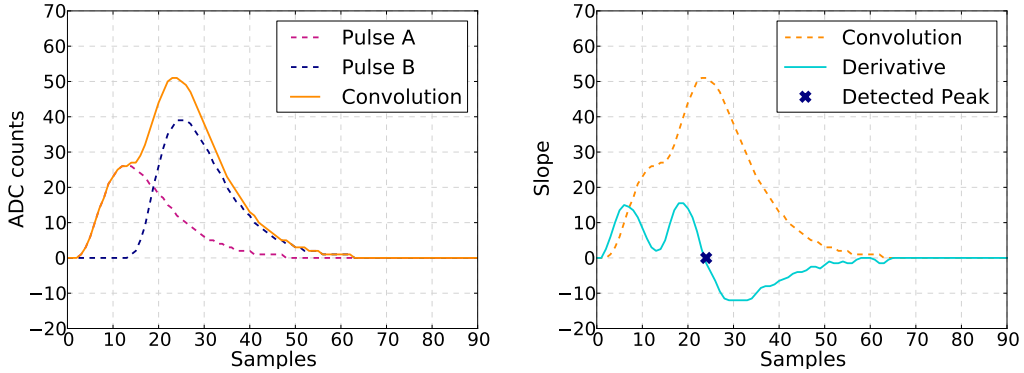
Figure 8.3: Simulated signals

reconstructing one bunch crossing and are fed into the *samples* input of the peak finder module. Before reading in the data, the module is configured for the desired parameters using IPbus commands. After simulation of all input data, a TXT-file consisting of the detected peak positions and amplitudes is produced.
For cross-checks, software implementations of the discriminator- and derivative-based peak finding algorithms also perform the peak finding, using the same data file as the VHDL testbench. Comparisons between the detected peaks of the derivative based algorithm for the VHDL- and Python-implementation as well as the software implementations of the discriminator-based approaches are performed using a Python script. The script also contains histogramming and plotting functionalities for visualising the performance of the peak finding implementations.
Figure 8.2a illustrates the raw data of one full orbit of a detector channel with a sCVD diamond, whereas figure 8.3a shows data of one simulated orbit. Particle hits at the sensor are visible as spikes in the plot. Figure 8.2b visualises an excerpt of the raw data of BCM1F channel 37, equipped with a pCVD diamond sensor, where the occurrence of two piled-up pulses can be seen around sample 200. A similar scenario is reproduced in the simulated data that is shown in figure 8.3b.



(a) Convolution, FWHM = 10.0 ns

(b) Derivative, FWHM = 10.0 ns



(c) Convolution, FWHM = 13.3 ns

(d) Derivative, FWHM = 13.3 ns

Figure 8.4: Convolution of two pulses with the derivative of the resulting signal for two different pulse widths at the same distance

8.2 Peak Separation Capability

The major feature of the derivative based peak finding algorithm is the capability to detect the peaks of piled-up pulses, even if they occur in the same bunch crossing. The discriminator-based approaches are vulnerable to piled-up pulses, which can lead to missed peaks in several consecutive bunch crossings, as described in chapter 6.

For two very close pulses, the resulting signal is the convolution of both pulses, as seen in figure 8.4a. Pulses A and B, which both have a FWHM of 10.0 ns, occur in a distance of 11 samples, resulting in the convoluted signal. Figure 8.4b displays the derivative of the signal that was calculated with the smooth noise-robust differentiator and the peak positions that were found by applying the derivative based peak finding algorithm are indicated. It can be seen that both peaks were detected correctly, although both pulses occur in the same bunch crossing and are piled-up. A derivative threshold of 8 ADC counts per Sample was applied. To detect both pulses with a simple discriminator approach, a very high threshold of 30 ADC counts would be necessary. However, this would prevent the detection of pulses below that threshold and thus lead to undercount of sensor hits. In figures 8.4c and 8.4d the pulses remain at the same distance as in the previous example, but the pulse FWHM is increased to 13.3 ns. In this

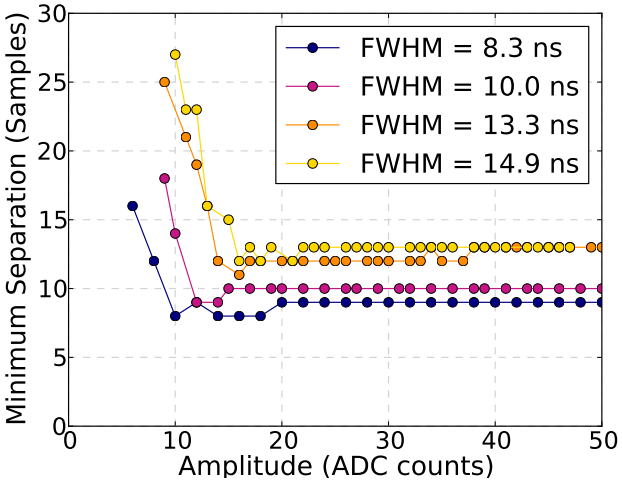


Figure 8.5: Minimum distance between two detectable pulses vs. pulse amplitude for different FWHM

case the derivative peak finder only detects the second pulse peak. As the convoluted signal does not show a dip between the peaks, the derivative of the signal does not have a descending zero crossing at the position of the first peak and thus cannot be detected by the derivative based peak finder. Hence, two adjacent peaks need to occur at a minimum distance, to be detectable.
The minimum distance between two separable pulses rises with the pulse FWHM. Figure 8.5 shows the minimum distance between two detectable pulses in dependence of the pulse amplitude for different FWHM (assuming both pulses have the same amplitude and FWHM). No noise was added to these signals. Due to the short rising edge of pulses with small amplitude and the limited vertical resolution, the minimum distance between two pulses is higher for small pulses. For higher amplitudes, the minimum detectable separation distance tends to a constant value. The dependency of the minimum distance and the FWHM is shown, as the narrower pulses tend to a smaller minimum separation width than the wider pulses.
The minimum distance between two separable pulses also implies a maximum number of detectable peaks per bunch crossing. For pulses with a FWHM of 8.3 ns and an amplitude greater than 20 ADC counts a minimum distance of 9 samples is needed, which means that up to four peaks could be detected in one bunch crossing. In case of pulses with the same amplitude but a FWHM of 13.3 ns, only three peaks can be detected within one bunch crossing, as the minimum peak distance is 12 samples.

It was shown that the pulse FWHM is the limiting factor for the pulse separation capability of the derivative based peak finding algorithm. For pulses with amplitudes below approximately 15 ADC counts, the minimum distance between two separable peaks is significantly higher than for pulses with higher amplitudes, where the minimum distance tends to a constant value. A decrease of the minimum distance between two detectable peaks can potentially be achieved by determining local minima in the second derivative of the signal (see chapter 10).

8.3 Hit Rate Measurement Performance

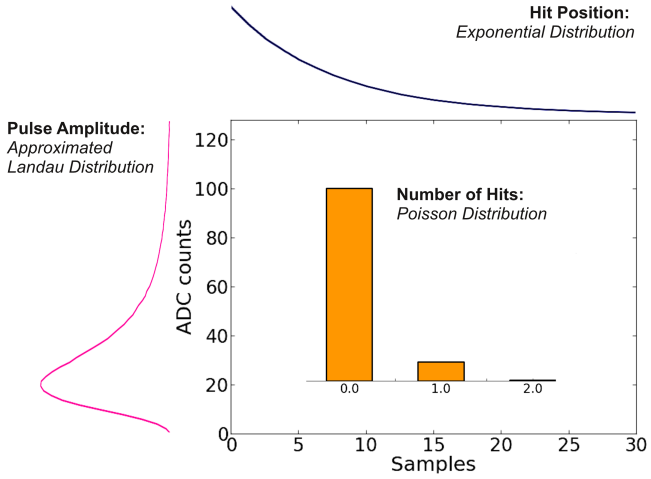
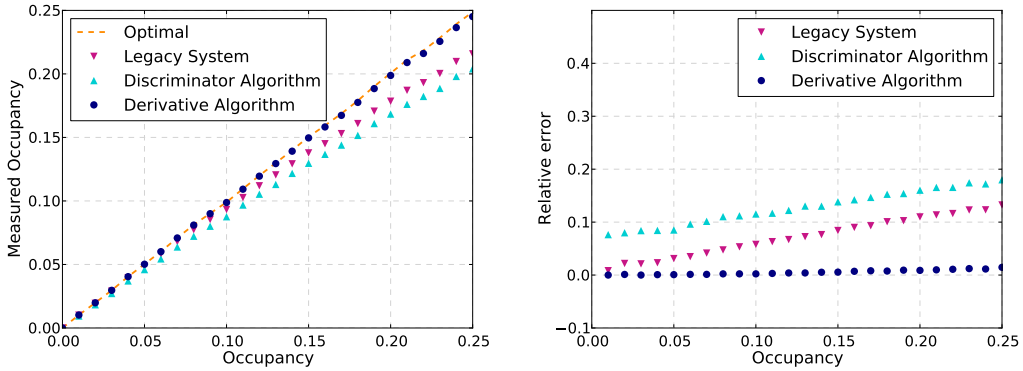


Figure 8.6: Distributions for the simulation of one bunch crossing

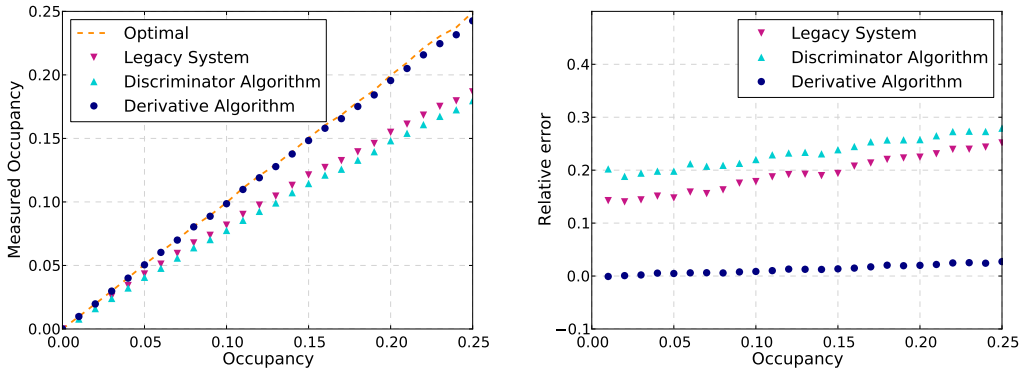
The capability of the derivative based algorithm to detect piled-up pulses as well as the ability to find multiple peaks in one bunch crossing were previously illustrated. Pulse pileup, which is favoured by high FWHM pulses, is one of the main limits of peak detection, for both the discriminator and derivative based approaches. The impact of pulse pileup on the hit detection accuracy is lower for the derivative based algorithm, than for the discriminator based approaches. With a higher per-bunch occupancy, the probability of piled-up pulses rises. Therefore, the hit rate measurement performance of the new algorithm should be better than the discriminator based hit rate measurement for higher occupancy. As the zero counting algorithm essentially calculates the luminosity using the measured hit rate, this should lead to a more accurate luminosity measurement.

To evaluate the impact of the occupancy on the hit rate measurement accuracy, the particle hits on a sensor during collisions need to be simulated, taking into account the distributions of the hit probability, generated charge and hit arrival time. Therefore, a "train-sequence" of a so-called "8b4e-beam" [48] is created. The "8b4e"-beam has a duration of 12 bunch crossings, consisting of a sequence of 8 consecutive "filled" bunch crossings (8b), with actual collisions taking place, followed by 4 empty (4e) bunch crossings. The 8-bunch-sequence is also called a "train".
For the relatively small area of one of the BCM1F sensors, the probability of a particle hit in one bunch crossing is low. As described in chapter 3, the probability of a hit in a bunch crossing follows a Poissonian distribution. Provided the average number of hits per bunch crossing μ, a random number with Poisson distribution is generated to indicate the particle hits on the sensor. Reviewing the data of the BCM1F VME system of 2017, a maximum of $\mu = 0.12$ can be assumed.
In addition to the hit probability, the distribution of particle-hit induced charge generation in the sensor needs to be described. The passage of a fast charged particle through matter results in a statistical ionizing process with large fluctuations of the energy loss in an absorber that is small compared to the particle range [49]. The probability function of the energy loss in such an absorber can be described by the Landau distribution [50].



(a) Measured occupancy vs. actual occupancy **(b)** Relative hit counting error vs. occupancy

Figure 8.7: FWHM = 10.0 ns , σ_{noise} = 0.0 ADC counts



(a) Measured occupancy vs. actual occupancy **(b)** Relative hit counting error vs. occupancy

Figure 8.8: FWHM = 13.3 ns , σ_{noise} = 1.0 ADC counts

In the BCM1F diamond sensors, the hit of a charged particle creates several thousand electron-hole pairs, leading to a certain amount of charge that is read out by the front end ASIC and converted into a pulse with corresponding amplitude. For creating a realistic charge distribution, the acquired amplitude distribution of BCM1F channel 2 in fill 5737 is approximated, using the Von-Neumann method for modeling a continuous random variable [51]. A possible variation of the arrival time of a particle is modeled with an exponential function. Figure 8.6 illustrates the distributions of one simulated bunch crossing.

Knowing the probability distributions for hits, generated charge and arrival time, a simulation for an 8-bunch train sequence was implemented for an arbitrary amount of orbits, together with the BCM1F channel model. Per orbit and bunch crossing, a boolean value for representing a hit is generated using a Poisson distribution for a given occupancy value. In case of a successful hit, the generated charge is created using the approximated amplitude distribution and fed into the ASIC output pulse model. The signals of the bunches in each train are fed into the transmission path model, followed by the digitisation and noise modelling.
Figures 8.7 and 8.8 show the hit counting performance of the discriminator and deriva-

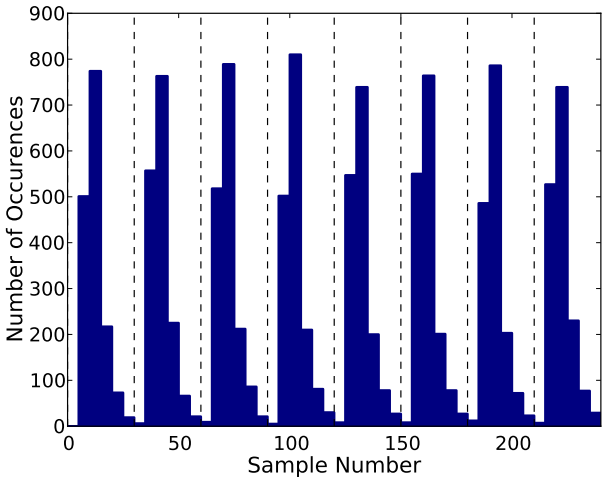


Figure 8.9: Occupancy histogram for FWHM = 13.3 ns , σ_{noise} = 1.0 ADC counts

tive approaches in dependency of the occupancy, for different FWHM and noise standard deviations. The occupancy, measured by the different approaches, over the actual occupancy is illustrated together with the ideal measurement in the left hand figures. The right hand figures show the relative hit counting errors over occupancy. The simulations were carried out for a duration of four lumi nibbles for two different FWHM values and two different noise standard deviations.

With a FWHM of 10.0 ns and no noise, the simulated conditions in figure 8.7 are close to optimum. It can be seen, that the legacy system and the discriminator algorithm show an increasing deviation of the ideal measurement with increasing occupancy. Especially the discriminator based algorithm shows a high systematic undercount, even for very low occupancy values. Remarkably, the derivative based algorithm has a very small relative hit counting error, that only increases slightly for higher occupancies. This shows that the new algorithm has near-optimal hit counting efficiency for near-optimal conditions and is almost linear with the occupancy in the used range from 0 to around 0.12.
In figure 8.8 the FWHM is increased to 13.3 ns and the noise standard deviation is set to 1.0 ADC counts. Similar conditions can be found in BCM1F channel 1, where the average pulse FWHM was measured to be 13.5 ns and the noise standard deviation is 1.01 ADC counts (see chapter 5). As expected, the near-linear range of the discrimination based approaches has drastically decreased and the relative hit counting error is partially more than twice as high as in the previous example. Also the legacy system has a high systematic error for very low occupancy, similar to the discriminator based peak finding algorithm. On the contrary, the derivative based algorithm maintains an almost optimal hit counting efficiency for very low occupancy. Over the accounting occupancy range, the new algorithm only shows a small non-linearity.
The above figures suggest the superiority of the derivative based peak finding algorithm in terms of relative error and linearity with occupancy. Even with increased noise and pulse FWHM, a high hit counting efficiency at high occupancy and improved linearity

over the full used occupancy range was achieved.

Figure 8.9 shows the occupancy histogram for the simulation of four lumi nibbles with a FWHM of 13.3 ns and a noise standard deviation of 1.0 ADC counts. The bunch structure of the 8b4e-beam is clearly visible in the figure, with the exponential distribution of the peak arrival time within the bunch crossing. A bin width of 5 samples was configured, thus having 6 bins for one bunch crossing. This corresponds to a timing resolution of 4.15 ns per bin. With this configuration of the bin width, the timing resolution is about 1.5 higher compared to the legacy system, as the VME system is fixed to 4 bins per bunch crossing, corresponding to a timing resolution of 6.25 ns.

8.4 Discussion

The previous simulations can only be understood as a coarse approximation to the real conditions during accelerator operation. They serve the purpose of comparing the different approaches with each other under various conditions. The determined hit counting efficiencies do not necessarily reflect the real performances. Still, the simulation of four lumi nibbles of a 8b4e-beam revealed the higher hit detection efficiency and linearity of the derivative based algorithm in comparison to the legacy VME system and an implemented discriminator based algorithm, under given conditions. The absolute hit rate measurement performance of the new algorithm needs to be evaluated with a finally completed detector setup and colliding beams.
One aspect that is not covered by the simulation is the presence of machine induced background (MIB) [52]. These particles enter the experiment together with the beam and occur at a typical rate of 10^5 lower than the collision rate. They arrive 12.5 ns apart from the collision products, due to the position of the BCM1F detector plane along the beam pipe [22]. This can lead to more than one hit per bunch crossing, if a MIB particle arrives at the same sensor in the same bunch crossing, as a particle originating from a collision. In theory, the VME system is able to detect pulses that are at least 12 ns seconds apart. In practice however, the presence of a collision and MIB hit in the same bunch crossing most likely results in puled-up pulses, with only a small dip between their peaks and will therefore not be detected by the legacy system. As the discriminator based algorithm does not foresee the detection of more than one hit per bunch crossing by implementation, the hit counting error for occurring MIB hits would further increase. The derivative based algorithm is capable of detecting more than one hit per bunch crossing, as long as the peaks have a minimum distance. It can handle background hits in the same bunch crossing as collision hits, if the separation distance and amplitudes of these pulses are meeting the minimum requirements from figure 8.5.

9. Conclusion

The BCM1F detector measures the luminosity in the CMS experiment at the LHC. Currently, the detector is read out by a VME back end electronic system, which detects pulses by applying a leading edge discrimination. This system lacks sufficient pulse detection efficiency in case of closely occurring pulses, due to pulse pileup. A newly developed MicroTCA back end system will soon replace the legacy VME system to enable a more reliable hit counting measurement. This back end system provides the capability of digitising and processing the detector signal. To fully use the potential of the new system, a highly efficient peak finding module needs to be implemented.
This work presented a new peak finding algorithm for the MicroTCA system and its implementation in an FPGA. The algorithm detects the pulse peaks by determining the position of descending zero crossings in the derivative of the detector signal, using a differentiator with implied smoothing to suppress the high frequency noise. By applying different thresholds, the algorithm is capable of robust peak detection in noise-affected signals. It was shown, that the algorithm is also capable of separating piled-up pulses with a minimum peak distance that is depending on the pulse FWHM and amplitude. The implementation of the new peak finder includes the possibility of finding more than one pulse per bunch crossing as well as the possibility to find the peaks of pulses that start or end in the previous or next bunch crossing, respectively.

As the LHC is not in operation during the genesis of this work, a simulation framework for the validation of the new peak finding algorithm was developed. The detector signal was successfully simulated by using an implemented BCM1F software model, that is based on the characteristics of the different detector modules. The resulting pulse shapes resemble the original pulses and can be freely adjusted in pulse width and amplitude. Additionally, the noise can be modeled with adjustable distribution parameters, by using a provided software routine. Acquired raw data, together with the BCM1F software model, offered the possibility to study the behaviour of the algorithms and to determine the limits of the peak detection capability.
To evaluate the efficiency of the developed algorithm in dependency of the occupancy in comparison to the discrimination based hit counting approaches, the hits during four lumi nibbles of an 8b4e-beam were simulated. The distributions of the particle hit probability, charge generation and particle arrival time were taken into account. It was shown that the new algorithm has a near-optimal behaviour for narrow pulses in a noiseless system, whereas the other approaches show a significant inefficiency that is increasing with the occupancy. In a simulation of a noise-affected system that resembles the conditions in a real detector channel, the derivative based algorithm showed an inefficiency of more than 10 times less than for the currently operating VME system. In contrary to the

discriminator based approaches, the new algorithm has an almost linear relation between the hit counting measurement and the occupancy, throughout the used occupancy range.

The new peak finding algorithm for the BCM1F detector has proven superiority over the legacy system as well as a discriminator based peak finding algorithm, in terms of hit counting efficiency and linearity with occupancy. It is therefore capable of significantly increasing the accuracy of the online bunch-by-bunch luminosity measurement at the CMS experiment. Due to a highly adjustable implementation, the new peak finder module can adapt to every channel's specific characteristics and thus maintain optimal performance for all channels, sensor types and extent of radiation damage.

10. Outlook

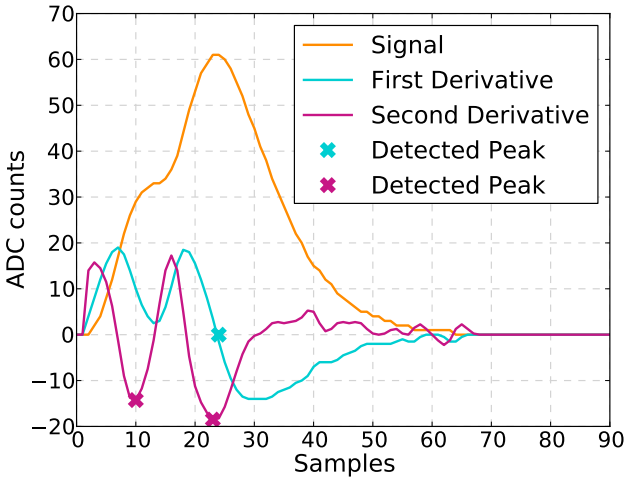


Figure 10.1: Detecting peaks at the local minimum of the second derivative

In spring 2018 the LHC will be recommissioned after the Extended Year End Technical Shutdown in winter 2017/2018. According to the schedule, the BCM1F MicroTCA back end development and implementation work will be completed by then. As the system will be ready for data acquisition, it will participate in the foreseen calibration processes. This implies that the newly developed peak finding algorithm must be completely integrated into the GLIB firmware. With the first beams, every detector channel needs to be characterised by its noise, gain and pulse width. Taking these channel properties into account, the parameters of the peak finder module have to be set accordingly, to assure maximum peak detection efficiency. A detailed evaluation of the derivative based peak finder performance will be possible after the Van-der-Meer scans.

After the Long Shutdown 2 in 2019 and 2020, the LHC will enter Run 3 with increased performance in 2021. The foreseen Run 3 upgrade of the BCM1F detector from DC-coupled diamond to AC-coupled silicon sensors promises to provide higher signal gain and thus better signal-to-noise ratio. This should lead to an increased performance of the newly developed derivative based algorithm, as it has the best efficiency for high amplitude pulses.

Possible improvements of the developed derivative based algorithm can be derived when reviewing figure 8.4d. Here, the developed algorithm was not able to detect the two

peaks due to the close pileup. A potential improvement of the derivative based algorithm could be the detection of local minima close to zero, additionally to zero crossings. This would enable the detection of pulses that violate the minimum peak distance, that was determined in section 8.2. A maximum value for the detected local minimum is necessary to prevent false peak detection due to noise.
Another approach for further decreasing the minimum distance for detectable piled-up pulses is to use the second derivative of the signal. Figure 10.1 recreates the situation in figure 8.4d, but additionally displays the second derivative of the signal. By detecting local minima in the second derivative, both peaks can be found, whereas the algorithm based on the first derivative only detects the second pulse. However, the detected position of the first peak by locating the minimum in the second derivative does not exactly match the actual peak position. This could be improved by combining indications from the first and the second derivative. Limitations for this approach could be the signal noise in combination with insufficient noise suppression and increased demand in logic and DSP resources in the FPGA.

Appendix

SNRD Implementation

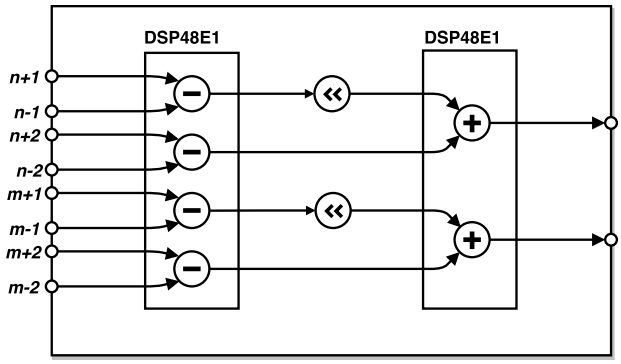


Figure 10.2: SNRD block for N = 5

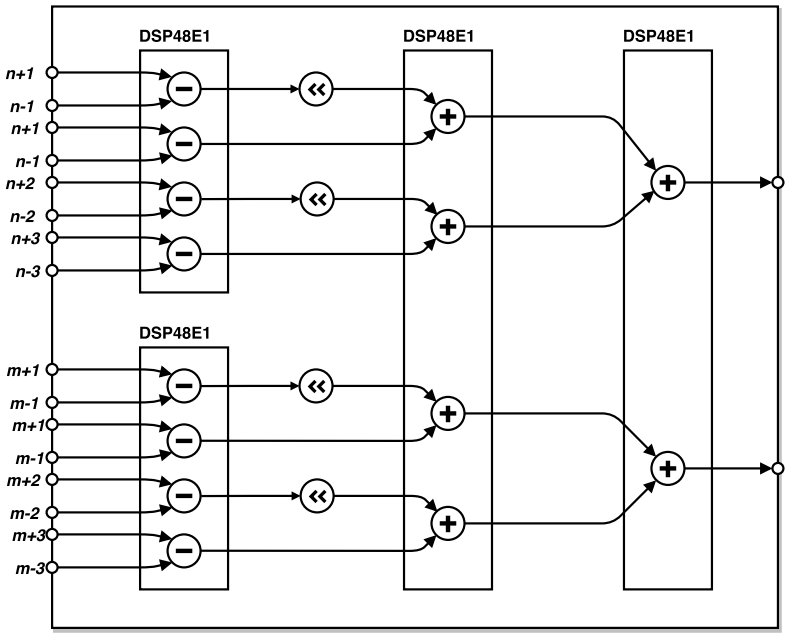


Figure 10.3: SNRD block for N = 7

Bibliography

[1] https://cds.cern.ch/record/1621583/files/.

[2] O. Bruning, et al, *LHC Design Report*, CERN Yellow Reports: Monographs, Geneva, 2004.

[3] *The Standard Model*, https://home.cern/about/physics/standard-model, Seen at 10:04 pm, February 7, 2018.

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

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

[6] The ALICE Collaboration, *The ALICE Experiment at the CERN LHC*, JINST 3 S08002, 2008.

[7] The KHCb Collaboration, *The LHCb Detector at the LHC*, JINST 3 S08005, 2008.

[8] T. Sakuma and T. McCauley, *Detector and Event Visualization with SketchUp at the CMS Experiment*, J. Phys.: Conf. Ser. 513 022032.

[9] CMS Collaboration, *Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC*, Physics Letters B 716 (2012) 30–61, 2012.

[10] Bailey, R and Collier, Paul, *Standard Filling Schemes for Various LHC Operation Modes*, LHC-PROJECT-NOTE-323, CERN, 2003.

[11] W. Herr and B. Muratori, *Concept of Luminosity*, Proceedings of the CERN Accelerator School, Vol. 361, 2003.

[12] O. Bruning, L. Rossi, *The High Luminosity Large Hadron Collider: The New Machine for Illuminating the Mysteries of Universe*, World Scientific Publishing Company, 2015.

[13] S. de Capua, et al, *Luminosity measurements at LHC*, Il Nuovo Cimento, Vol. B123, 2008.

[14] Olena Karacheban and Sergej Schuwalow, *Luminosity measurement at CMS*, DESY-THESIS-2017-043, 2017.

[15] S. van der Meer, *Calibration of the Effective Beam Height in the ISR*, CERN-ISR-PO-68-31, 1968.

[16] CMS Collaboration, *The CMS Pixel Luminosity Telescope*, Proceedings of the 13th Pisa Meeting on Advanced Detectors, 2015.

[17] A. Bell, et al., *Fast beam conditions monitor BCM1F for the CMS experiment*, Nuclear Instruments and Methods in Physics Research A 614 (2010) 433 - 438.

[18] CMS Collaboration, *CMS Hadron Forward Calorimeter Phase I Upgrade Status* Journal of Physics: Conference Series, Vol. 587, No. 1, 2015.

[19] CMS Collaboration, *CMS luminosity measurement for the 2015 data taking period*, CMS Physics Analysis Summary LUM-15-001, 2015.

[20] CMS Collaboration, *The Phase-2 Upgrade of the CMS Tracker*, CERN-LHCC-2017-009, CMS-TDR-014.

[21] A. A. Zagozdzinska, et al., *Architecture of the upgraded BCM1F backend electronics for Beam Conditions and Luminosity measurement*, JINST 10 C02020, 2015.

[22] J. L. Leonard, et al., *Fast Beam Condition Monitor for CMS: performance and upgrade*, arXiv:1405.1926 [physics.ins-det].

[23] M. Friedl, *Diamond Detectors for Ionizing Radiation*, University of Technology, Vienna, Diploma Thesis, 1999.

[24] M. Guthoff, *Radiation Damage to the diamond-based Beam Condition Monitor of the CMS Detector at the LHC*, CMS-TS-2014-043, CERN-THESIS-2014-216.

[25] D. Przyborowski, J. Kaplon, P. Rymaszewski, *Design and Performance of the BCM1F Front End ASIC for the Beam Condition Monitoring System at the CMS Experiment*, IEEE transactions on nuclear science, Vol. 63, No. 4, August 2016.

[26] M. Friedl, *Analog optohybrids for the readout of the CMS silicon tracker*, Nuclear Instruments and Methods in Physics Research A 518 (2004) 515 - 518.

[27] G. Cervelli, A. Marchioro, P. Moreira et al., *A radiation tolerant laser driver array for optical transmission in the LHC experiments*, Proceedings of the 7th Workshop on Electronics for LHC Experiments, Stockholm 2001

[28] F. Vasey, et al., *A 12-Channel Analog Optical-Receiver Module*, Journal of Lightwave Technology Vol. 23, pp. 4270-4276, 2005.

[29] A. A. Zagozdzinska, *Identification of fast signals from the upgraded single crystal diamond detector for the Online Beam Monitoring and Precision Luminosity Measurement for the CMS experiment at the LHC accelerator*, PhD Thesis draft, University of Technology, Warsaw, November 2017.

[30] T.Bauer, F.Vasey, *A Model for the CMS Tracker Analog Optical Link*, CMS NOTE 2000/056, September 2000.

[31] P. Vichoudis, et al., *The Gigabit Link Interface Board (GLIB) ecosystem*, JINST 8 C03012, 2013.

[32] R. Frazier, G. Iles, D. Newbold and A. Rose, *Software and firmware for controlling CMS trigger and readout hardware via gigabit Ethernet*, Phys. Proc. 37 1892, 2012.

[33] Intel, Hewlett-Packard, NEC, Dell, *Intelligent Platform Management Interface Specification v2.0 rev. 1.1*, Version 2.0, Document Revision 1.1, 2013

[34] PCI Industrial Computer Manufacturers Group, *Micro Telecommunications Computing Architecture Base Specification*, PICMG Specification MTCA.0 R1.0, 2006

[35] V. Brigljevic, et al., *Using XDAQ in application scenarios of the CMS experiment*, CHEP-2003-MOGT008, 2003.

[36] V. Arbet-Engels, et al., *Characterization of optical data links for the CMS experiment*, Proceedings of the 3rd Workshop on Electronics for LHC Experiments, London, 1997.

[37] S. Ruscak, L. Singer *Using Histogram Techniques to Measure A/D Converter Noise*, Analog Dialogue, Analog Devices, Volume 29, Number 2, 1995.

[38] *NumPy v1.14 Manual*, https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.nor Seen at 6:09 pm, February 5, 2018.

[39] S. W. Smith, *Digital Signal Processing: A Practical Guide for Engineers and Scientists*, Newnes, 2002.

[40] Dominik Przyborowski, *Zastosowanie submikronowych technologii VLSI w rozwoju wielokanałowych układów scalonych do odczytu detektorów cząstek jonizujących*, PhD Thesis, AGH University, Krakow, 2015.

[41] e2v Technologies Limited, *EV8AQ160 Datasheet*, 2016

[42] M. Penno, et al, *A Real-time Histogramming Unit for Luminosity Measurement of each Bunch Crossing at CMS*, Topical Workshop on Electronics for Particle Physics, DESY-2013-00940, 2013.

[43] CAEN S.p.A., *V895 Technical Information Manual*, Revision n. 2, NPO: 00101/97:V895x.MUTx/02, 2012.

[44] A.U.A. Bin Ab Maalek, O. Karacheban, *Finding Multiple Peaks Signal in Fast Beam Conditions Monitor (BCM1F)* , CERN-STUDENTS-Note-2017-135, 2017

[45] A. Savitzky and M. J. E. Golay, *Smoothing and Differentiation of Data by Simplified Least Squares Procedures*, Analytical Chemistry, 36 (8), 1964.

[46] Pavel Holoborodko *Smooth Noise Robust Differentiators* http://www.holoborodko.com/pavel/numerical-methods/numerical-derivative/smooth-low-noise-differentiators/, Seen at 11:43 am, February 2, 2018.

[47] 1076-2008 - *IEEE Standard VHDL Language Reference Manual*, IEEE, 2009.

[48] Y. Papaphilippou, et al, *Operational Beams For The LHC*, Proceedings of Chamonix 2014 Workshop on LHC Performance, 2014.

[49] S. Meroli, et al, *Energy loss measurement for charged particles in very thin silicon layers*, JINST 6 P06013, 2011.

[50] L. Landau, *On the energy loss of fast particles by ionization*, J. Phys. USSR 8 (1944) 201, 1944.

[51] I. M. Sobol, *A Primer for the Monte Carlo Method*, CRC Press LLC, 1994.

[52] G.Corti and V.Talanov, *Aspects of Machine Induced Background in the LHC Experiments*, 3rd LHC Project Workshop, Divonne-les-Bains, 2006.

Thesen

1. Das aktuell operierende System sowie ein einfacher, Schwellwertbasierter Spitzenfinder weisen Ineffizienz bei Pulsüberlappung und Pulsen mit hoher Breite auf

2. Mittels der Ableitung des Detektorsignals lassen sich die Positionen von Pulsspitzen bestimmen

3. Der Ableitungsbasierte Spitzenfinder-Algorithmus weist weniger Ineffizienz bei Pulsüberlappung und Pulsen mit hoher Breite auf, als die Schwellwertbasierten Methoden und ist dadurch in der Lage mehr Pulse zu detektieren

4. Die Linearität der Messung der mittleren Einschlagsrate wird durch den neuen Algorithmus deutlich gesteigert, auch bei hoher mittlerer Einschlagsrate