GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Fakultät für
Physik

Master's Thesis

Entwicklung neuer Datenübertragungsmethoden für die nächste Ausbaustufe des ATLAS Pixeldetektor für den HL-LHC

Development of New Methods of Data Transmission for the Upgraded ATLAS Pixel Detector at the HL-LHC

prepared by

Frank Schuster

from Utgast

at the 2nd Institute of Physics

Thesis number:	II.Physik-UniGö-MSc-2013/01
Thesis period:	1st November 2012 until 1st May 2013
First referee:	Priv.Doz. Dr. Jörn Große-Knetter
Second referee:	Prof. Dr. Arnulf Quadt

Zusammenfassung

Wie in vielen Bereichen von Wissenschaft und Technik gilt auch in der Teilchenphysik das olympische Motto: 'höher, schneller, weiter'. Aus diesem Grund werden, beginnend mit dem zweijährigen Betriebsstillstand ab 2013, die Luminosität und Energie des Teilchenstrahls des weltweit größten energiereichsten Teilchenbeschleunigers, dem LHC in Genf, in mehreren Ausbauphasen in den nächsten 20 Jahren stark erhöht.

Zielsetzung der einzelnen Experimente ist es dabei, die Leistung und Effizienz der Detektoren in gleichem Maße mitzuerhöhen, während die Luminosität um einen Faktor Zehn erhöht werden soll. Dies stellt für die innerste Detektorlage, den ATLAS Pixel Detektor, die größte Herausforderung dar. Sowohl die Sensoren, als auch die Ausleseelektronik benötigen hierfür eine umfangreiche Weiterentwicklung.

Ein alternatives Auslesesystem, das die nötige Leistung liefert, das RCE system wird vorgestellt. Dieses wird um eine Datenkodierung erweitert, das sogenannte GBT Protokoll. Es wird mathematisch gezeigt, dass bis zu 16 aufeinander folgende Fehler nachträglich rekonstruiert werden können, wenn 30% zusätzliche Daten gemäß einer Reed-Solomon-Kodierung hinzugefügt werden. Dies ist insbesondere für hohe Luminositäten wichtig, um die Daten gegen strahlungsbedingte Bitfehler zu schützen. Dieser Effekt wird ausführlich in Simulationen und durch Messungen studiert.

Abstract

Just as in many areas of research and technology, the Olympic motto: 'faster, higher, stronger' holds also in particle physics. Therefore the energy and luminosity of the world's largest and most energetic particle accelerator will be upgraded even further within the next two decades, beginning with a two year shutdown in 2013.

The goal of the individual experiments is to keep the detector performance and efficiency the same, even though the luminosity is increased by a factor of ten. For the innermost layer, the ATLAS Pixel Detector, the challenge is thus the greatest. Both the sensors and the readout electronics require a significant upgrade in order to cope for the changes.

An alternative readout system that has the required performance, the RCE system is presented. In this thesis, it is extended by an encoding algorithm, the so called GBT protocol. It is proven mathematically that 16 consecutive errors can be reconstructed by adding 30% of additional data generated by a Reed-Solomon encoding to the bit stream. This is especially important for high luminosities to protect the data from corruption due to ionizing radiation. The effect is extensively studied both in simulations and by a measurement.

Contents

Nomenclature

ALICE	A Large Ion Collider Experiment
ARQ	Automatic Repeat reQuest
ASIC	Application Specific Integrated Circuit
ATCA	Advanced Telecommunication Computing Architecture
ATLAS	A Toroidal LHC Apparatus
BCH	Bose, Chaudhuri and Hocquenghem
BER	Bit Error Rate
BOC	Back Of Crate
BPM	Bi-Phase Mark
BRAM	Block Random Access Memory
CERN	European Organization for Nuclear Research
CIM	Cluster Interconnect Module
CLB	Custom Logic Blocks
CM energy	Center of Mass energy
CMS	Compact Muon Solenoid
COB	Cluster On Board
CSCs	Cathode Strip Chambers
DACs	Digital to Analog Converters
DCM	Digital Clock Managers

DPM	Data Processing Module
DTM	Data Transport Module
EOS	End Of Stave
FDK	Firmware Development Kit
FE	Front End
FEC	Forward Error Correction
FMC	FPGA Mezzanine Card
FPGA	Field Programmable Gate Array
GbE	Gigabit Ethernet
GbE	Gigabit Ethernet
GBLD	GigaBitLaserDriver
GBT	GigaBit Transmission
GBTIA	GigaBit Transmission Impedance Amplifier
GDAC	Global DAC
GLIB	Gigabit Link Interface Board
GPIO	General Purpose I/O
HDL	Hardware Description Language
HL-LHC	High Luminosity LHC
HSIO	High Speed I/O Module
I/O	Input/Output
IBL	Insertable Barrel Layer
IC	Integrated Circuit
IEC	International Electrotechnical Commission
IOB	Input/Output Buffers

IP	Intellectual Property
JTAG	Joint Test Action Group
L1 Trigger	Level 1 Trigger
LAN	Local Area Network
LFSR	Linear Feedback Shift Registers
LHC	Large Hadron Collider
LHCb	LHC beauty
LSB	Least Significant Bit
LVDS	Low Voltage Differential Signalling
MAC	Media Access Controller
MCC	Module Control Chip
MDTs	Muon Drift Tubes
MGT	Multi Gigabit Transceiver
MicroTCA	Micro Telecommunication Computing Architecture
MMCM	Mixed Mode Clock Manager
MSB	Most Significant Bit
PCIe	Peripheral Component Interconnect Express
PGP	Pretty Good Protocol
PICMG	PCI Industrial Computer Manufacturers Group
PIN	Positive Intrinsic Negative
PLL	Phase Locked Loops
PRBS	Pseudo Random Bit Sequence
PS	Proton Synchrotron
RAS	reliability, availability and servicability

RCE	Reconfigurable Cluster Element
ROD	Read Out Driver
RPCs	Resistive Plate Chambers
RTM	Rear Transition Module
SBC	Single Board Computer
SCT	Semiconductor Tracker
SEU	Single Event Upset
SFP	Small Form-factor Pluggable
SLAC	Stanford Linear Accelerator Center
sLVDS	sub LVDS
SoC	System on Chip
SPS	Super Proton Synchrotron
TDAC	Trim DAC
TGCs	Thin Gap Chambers
TIM	Timing Trigger and Control Interface Module
TRT	Transition Radiation Tracker
TTC	Timing Trigger and Control
UCF	User Constraints File
UI	User Interface
VCSEL	Vertical Cavity Surface Emitting Laser
VME	Versa Module Europe

1. Introduction

The Olympic motto *citius, altius, fortius* (lat. faster, higher, stronger) inspires not only sportsmen, but also science and technology. The credo is not only true for modern smartphone manufacturers but also for High Energy Physics aiming for faster detectors and higher energies to discover new physics phenomena.

At the *Large Hadron Collider (LHC)* at *European Organization for Nuclear Research (CERN)* in Geneva, Switzerland, protons are brought to collision with a *Center of Mass energy (CM energy)* of currently 8 TeV. While searching for New Physics phenomena, the CM energy will be upgraded to its nominal 14 TeV. Furthermore the *Luminosity* will be increased on a long-term basis by a factor of ten, thus being able to improve measurements in the Standard Model of Particle Physics in shorter measurement periods.

This thesis focuses on the upgrade of the *A Toroidal LHC Apparatus (ATLAS)* experiment. An increase in luminosity is always associated with an increase in detector performance. This especially applies for the Inner Detector where the density of tracks is the highest. In order to cope with these increased data rates, the current inner detector will be upgraded in several ways, including the sensors and the readout system. Especially the performance of the readout system is affected by the increase in luminosity, because of the increase in data rates coming from the sensors. Thus a completely novel design for the detector readout will be presented in this thesis and compared to the current system. Furthermore it is shown, how the readout chain can be improved even more to meet all the criteria necessary for operation in an LHC scenario with tenfold increase in luminosity.

The so called *Reconfigurable Cluster Element (RCE)* has originally been designed at the *Stanford Linear Accelerator Center (SLAC)* and is being prepared to be part of one of the next major upgrades of the ATLAS Detector. It has been set up at the 2$^{\text{nd}}$ Institute of Physics, Göttingen to expand the research group around the RCE system to the European continent. Beside the specific setting of the system in Göttingen, the results of the electrical stave[1] of the *Insertable Barrel Layer (IBL)* tests in Geneva, which have also been performed as part of this thesis, will be presented.

[1] A stave describes the mechanical structure that holds one row of detector modules, as will be described later.

The following chapters will firstly give an introduction into the fundamental concepts of Particle Physics and the experimental setups being used. In this context the focus will mainly be on the ATLAS experiment and the upgrades planned for the innermost layer, the Pixel Detector.

2. Fundamental Principles

In this introductory chapter, the fundamental principles of Particle Physics, especially the Standard Model and the experiments at CERN are described [1].

2.1. The Standard Model of Particle Physics

The Standard Model of Particle Physics describes a set of fundamental particles representing all known matter and four fundamental interactions not considering the gravitational force. It has been developed in the 1970s as a quantum field theory two groups of particles. The *fermions* obey the laws of the *Fermi-Dirac statistics*; their quantum number of spin is $\frac{1}{2}$. The *bosons* obey the *Bose-Einstein statistics* giving them a spin of 1. While the latter category describes the fundamental forces as shown in figure 2.1, the fermions assemble to form all currently known types of matter (e.g. a proton consists of two up quarks and one down quark).

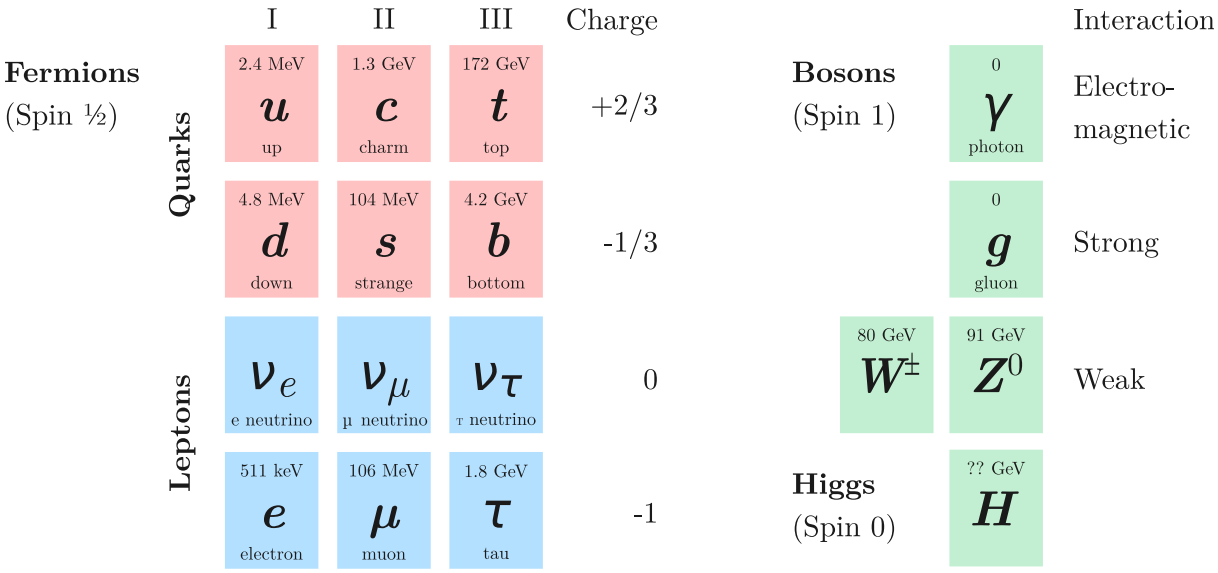

Figure 2.1.: The particles in the Standard Model categorized into quarks, leptons and gauge bosons.

The fermions can be divided into two groups, the leptons and quarks. Quarks interact via the strong interaction to form bound states, such as the proton described above. The gluon communicates this force, the so called *strong interaction* as described by quantum chromodynamics. The charge of the quarks is either $+\frac{2}{3}$ or $-\frac{1}{2}$, but quantum chromodynamics dictates that quarks always form bound states called *hadrons* that have an integer charge. Thus in nature, one does not encounter a free quark.

The leptons contain three particles with charge of -1 and their corresponding neutrinos, that do not carry a charge. The charged leptons can interact with each other via the electromagnetic force, while the neutrinos only interact via the weak force with other leptons or quarks. One example of weak interaction is the beta decay where a neutron (up, down, down) decays into a proton (up, up, down) emitting a neutrino and an electron via a charged W^- boson, which can also be found in figure 2.1.

Both the first generation of leptons and quarks are stable and are the building blocks of the visible matter around us, the second and third generation of particles are only produced in high energy regimes due to their high masses[1]. In nature these particles are present in cosmic radiation that originates from the Big Bang. They are unstable and decay via the three interactions described above. Note that leptons can only couple via the electromagnetic or the weak interaction, while the strong interaction is limited to quarks. The heaviest particle that has been found so far is the top quark, the third generation up-type quark. It has a mass of about 171 GeV and is thus almost as heavy as an entire Gold atom.[2]

While the properties of these particles and their interactions are well determined within the Standard Model (some of them can be seen in figure 2.1), the theory is not quite self-consistent. Since the Standard Model is a gauge theory and requires local gauge symmetry, the gauge bosons should be massless. However, it was found experimentally that the $W^{\pm}$ and Z^0 bosons are indeed massive. In 1964, Peter Higgs (among others [3]) adapted a principle known from solid state physics, the spontaneous symmetry breaking, explaining the masses of the bosons of the weak interaction. He thus predicted the Higgs field, an ubiquitous field interacting with each massive particle. The interaction of the field with itself creates another boson, the Higgs boson that can be measured and identified in modern particle accelerators.

[1] Except for the neutrinos, which have very small, but nonzero masses.

[2] Counting particles (core and shell) of ^{197}Au, the only stable gold isotope with an atomic mass of $m \approx 196.96\,\mathrm{u} \approx 185\,\mathrm{GeV}$.

[3] Francois Englert and Robert Brout from the Université Libre de Bruxelles submitted an independent paper at roughly the same time.

After almost half a century of searches at various experiments across the world, a Higgs-like particle was discovered at the LHC at CERN with a mass of about 126 GeV. The first evidence from LHC data was presented by both the ATLAS and CMS collaboration on July 4th, 2012 setting a mile stone in particle physics research.

But apart from finding the Higgs boson the by far most challenging and at the same time interesting task is to determine its properties and to study, whether the newly discovered particle is indeed in agreement with the Standard Model predictions. Thus the LHC and the large collaborations around each experiment still have a lot of work ahead.

2.2. The Large Hadron Collider

The LHC is the biggest particle accelerator in the world. It is located at CERN in Geneva and crosses the border between Switzerland and France approximately 100 meters underground [2]. The proton-proton collider reaches energies that are about three to four times higher than the previous world record[4]. It is a storage ring of about 27 km circumference and has been built to reach a CM energy of 14 TeV. Nowadays, it runs at 4 TeV per beam producing on average 20 collisions every 50 ns[5].

These amounts of energy can only be reached by a very carefully intertwined chain of preaccelerators. Mostly outdated particle accelerators from several decades ago serve their purpose as preaccelerators for the LHC. The chain starts with two linear accelerators injecting either lead ions or protons into the booster, the first preaccelerator. From the booster, the particles enter the *Proton Synchrotron (PS)*, which was built in the late 1950's. The next step is the *Super Proton Synchrotron (SPS)*, introduced in the 1970's. The (at the time highest) energy of 400 GeV led to the discovery of the $W^{\pm}$ and Z^0 boson, which won a Nobel prize for Carlo Rubbia and Simon van der Meer in 1984. Today the SPS injects proton bunches into the LHC at 450 GeV via two access points, such that two oppositely revolving beams are produced. The proton bunches of roughly 10^{11} protons are brought to collision after further acceleration to 8 TeV at the four major experiments indicated in figure 2.2.

The LHC's nominal luminosity[6] $L \approx 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ has set a world record for hadron collisions. Only the electron-positron collider KEKB in Japan achieved a higher luminosity of $L \approx 1.96 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, but because of the different architecture of the two

[4]The Tevatron near Chicago reached a CM energy of about 2 TeV.

[5]The nominal collision frequency is 40 MHz; in the current setup, only 20 MHz have been reached.

[6]The luminosity is defined as the proportionality factor between the number of interactions per unit time $\dot{N}$ and the interaction probability σ called *cross section* $\dot{N} = \sigma \cdot L$.

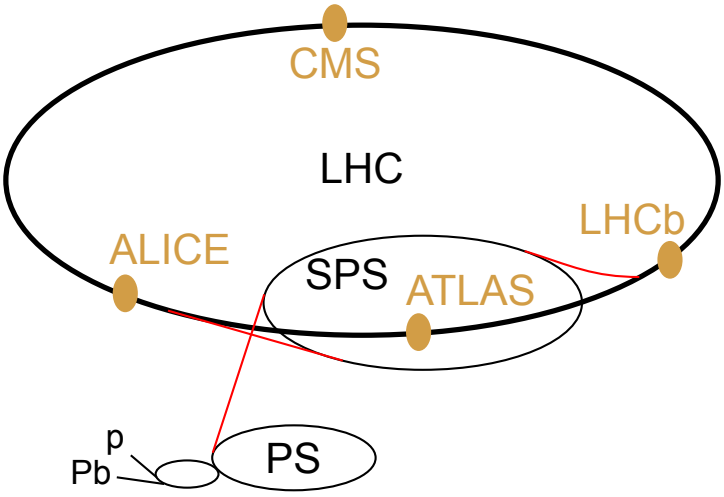

Figure 2.2.: The LHC with various preaccelerators and the four major experiments.

accelerators and different particles used, the luminosities in both experiments are to be judged quite differently.

The collisions are studied in four major experiments, ATLAS, *Compact Muon Solenoid (CMS)*, *A Large Ion Collider Experiment (ALICE)* and *LHC beauty (LHCb)*. ATLAS and CMS are general purpose experiments looking for new physics phenomena, whereas ALICE mainly investigates lead-lead collisions (creating quark-gluon plasmas), that the LHC delivers in a different operation mode. Lastly there is LHCb, investigating b-quark physics. All experiments are designed symmetrically around the beam pipe as will be shown later for the ATLAS experiment, except LHCb. This detector focuses mainly on the forward region of the collision shrapnel and thus it has most of its sensors perpendicular to the beam pipe.

Since the work presented in this thesis is performed as part of the ATLAS collaboration, further elaborations will focus on the ATLAS experiment, even though both ATLAS and CMS are similar in many ways.

2.3. The ATLAS Experiment

As mentioned before, the ATLAS detector is built symmetrically around the beam pipe, similar to the CMS and ALICE detector. It is organized in different layers similar to an onion skin. Each layer contains a different system fulfilling its own purpose. In figure 2.3 the different layers are displayed in a three dimensional rendered image of the detector [3, 4].

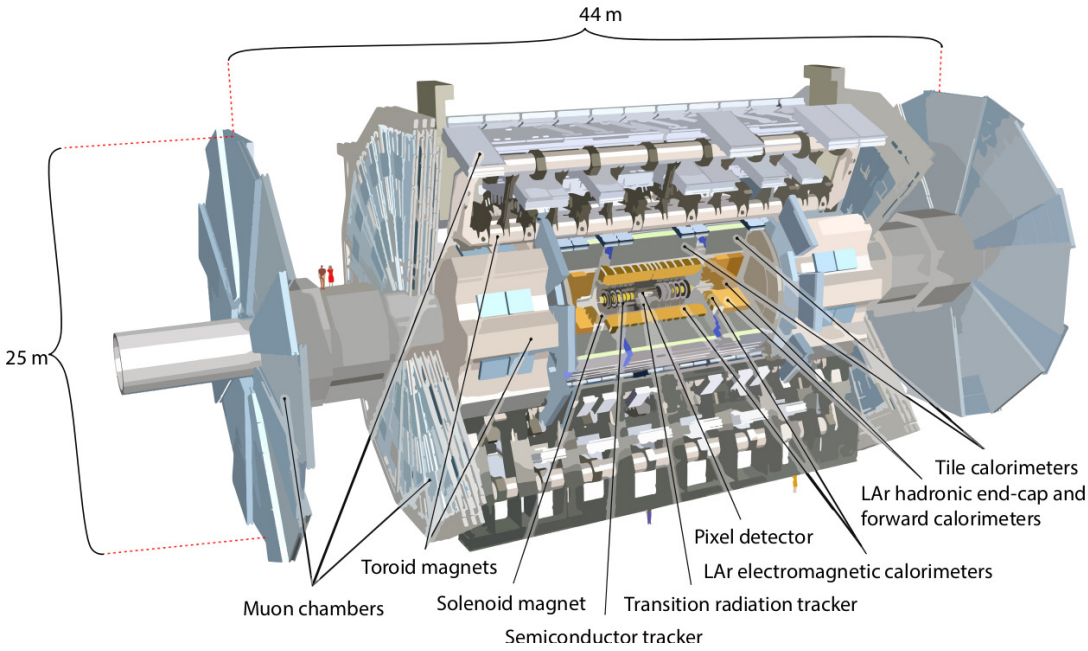

Figure 2.3.: The ATLAS Experiment as an onion skin detector covering almost the entire solid angle with its various systems.[5]

The outermost subdetectors are the muon chambers. Muons interact rather scarcely with matter, enabling them to leave the inner detector. Thus, the muon system can be placed far away from the interaction point. The muons are diverted by a toroidal magnetic field of 0.5 T to determine the momentum and charge. Inside the muon chambers and the toroidal magnet there are the hadronic and the electromagnetic calorimeters, whose purpose it is to measure the energy of hadronic and electromagnetic showers, respectively. At the very center of the detector, there is the so called inner detector tracking the created charged particles with a precision of a few micrometers. The inner detector is embedded in a 2 T solenoid magnet to determine momentum and charge of the particles very precisely.

The entire detector has a size of about 25 m by 44 m and weighs about 7000 tons. It covers almost the entire solid angle. The following paragraphs give a short explanation of the measurement principle for each of the mentioned subdetectors.

The Muon spectrometer measures the curvature of the muons with three different devices. The *Muon Drift Tubes (MDTs)* work similar to a Geiger-Mueller counter. Each of these aluminum tubes are filled with a gas that can withstand high doses of radiation. In the center there is a wire. By applying a high voltage between the wire and the tube, an electric field is generated for charged particles to drift in. A passing muon leaves a trail of positively charged ions and electrons that leave a measurable electric current when drifting to the electrodes.

For the outer parts of the muon system *Cathode Strip Chambers (CSCs)* are used, which have multiple wires, operating as a linear proportional chamber measuring charged particles in an electric field generated between the wires. This system can measure the muon's path very precisely, but is very slow compared to the following two systems.

The *Resistive Plate Chambers (RPCs)* and *Thin Gap Chambers (TGCs)* have a shorter dead time, because the charge collection is much faster. Thus, these systems can be used to trigger all other ATLAS subsystems upon particle detection. The overall muon system has a resolution of about 60 microns on the curvature of the muons enabling an equally precise measurement of the momentum.

The Calorimeters measure the energy of incoming particles when they are interacting with an absorber material. Via the electromagnetic interaction, high energy photons are created that split into an electron-positron pair if their energy exceeds the mass of two electrons[7]. The positrons on their part will interact with the cores of the absorber material, creating photons. Thus a cascade is created.

The calorimeters measure the number of times that the photons can produce electron-positron pairs, thus measuring the energy of the initial particle. Because of the different properties of the electromagnetic interactions (leptons) and the strong interaction (hadrons), this number of produced pairs for electrons and hadrons is different, such that two different calorimeter systems for hadrons and electrons are introduced to the LHC experiments.

In the ATLAS Experiment, sampling calorimeters are used, meaning that the absorber material and the measuring material are organized in layers. In the barrel region of the hadronic calorimeter, iron is used as absorbing material between layers of plastic scintillator. In the forward region liquid argon and tungsten or copper are being used because of the increased radiation dose in that area. The electromagnetic calorimeter consists of lead and liquid argon absorbers and similar plastic scintillators. The absorber material as well as the thickness of the systems are adjusted such that all the energy of the shower is deposited within the calorimeter.

The Inner Detector consists of three parts, each responsible to track the particles right after creation with different resolutions. While the *Transition Radiation Tracker (TRT)* usually contributes to the tracking with 36 points, the *Semiconductor Tracker (SCT)* provides four points. The Pixel Detector, that is located at the very center, provides an additional three space points.

[7] An electron has a mass of about $511\,\mathrm{keV/c^2}$.

The TRT works similarly to the muon system and contains very delicate straws that operate like drift chambers. Due to the very small diameter of 4 mm, the dead time of the detector is very small. The straws are located both in the barrel region and in the forward region. The straws have a maximal length of 144 cm and are mainly filled with Xenon gas. The system consists of 420,000 readout channels in two barrel layers and two end caps on each side. The spatial accuracy on the particle hits is about 130 µm.

The SCT aims to contribute an additional four hit points per track in radial direction to the tracking measurement of the inner detector. It has four barrel layers and nine disks on each side in radial direction. With silicon as detection material, the SCT works similar to the pixel detector, which will be described in detail in the next chapter. The main difference between the pixel detector and the SCT is the geometry of the silicon sensors. While the Pixel Detector consists of 2D pixels, the SCT consists of large strips of 80 µm by 12 cm. This geometry makes the coverage of large areas practical. The total covered area of the SCT is about 60 m^2 with almost 6.5 million readout channels achieving a resolution of particle hits of about 23 µm in azimuthal direction and 580 µm parallel to the beam pipe.

The Pixel Detector is the innermost detector in the ATLAS experiment. It provides the highest resolution of particle hits of about 100 µm in z direction[8] and 12 µm in the $r\varphi$ plane. Because this thesis deals with future read out systems for the ATLAS Pixel Detector, it is described in more detail in the following section.

2.4. The Pixel Detector

The Pixel Detector is the most central subsystem of the ATLAS detector and the smallest in size. It is merely 1.3 m long and has a diameter of 34 cm with a total mass (including all of the mechanical structure) of about 4.4 kg. Particle tracks are recorded with three barrel layers and three disks on each side as can be seen in figure 2.4.

The detector is built in such a way that multiple scattering is reduced to a minimum. The mechanical mounting seen in figure 2.4 is made out of carbon fiber, because it has a very low absorption. Additionally the designers of the staves and mounting fixtures spent years optimizing the stability vs. the material budget [4, 6].

The innermost barrel layer is mounted only about 5 cm from the interaction point and contains 22 staves. The modules are hosted on mechanical structures made of carbon fiber tubes to hold 13 modules in one row. It contains a cooling mechanism through the center of the tube and additional services, as well as the cables and electrical connections

[8] When looking at the detector in zylinder coordinates with the z axis aligned with the beam pipe.

Layer No.	Radius [mm]	No. of Staves	No. of Modules	Disk No.	Mean z [mm]	No. of Sectors	No. of Modules
0	50.5	22	286	0	495	8	48
1	88.5	38	494	1	580	8	48
2	122.5	52	676	2	650	8	48

Table 2.1.: The number of modules on the barrel layers and disks of the ATLAS Pixel Detector.

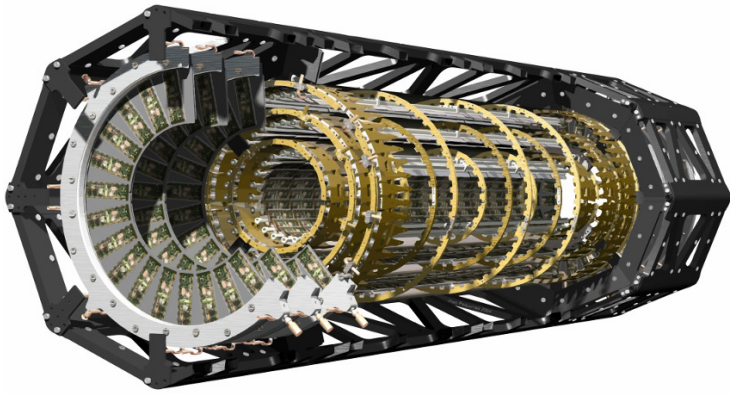

Figure 2.4.: The ATLAS Pixel Detector, a silicon tracking detector with three disks on each sides and three barrel layers [7].

needed to read out the modules. The modules are read out symmetrically to both sides via two flexible cables (flexes) that are also found in mobile phones, each flex connecting six or seven modules. As will be discussed later, one half stave is connected to the same flex and thus their readout chain is bundled towards the same components.

The other layers are built in the same structure. The number of modules, the number of staves and the distance from the interaction point can be found in table 2.1.

While the barrel layers contain an increasing number of staves (and thus modules) with growing distance from the interaction point, the disks are all designed identically. The total number of modules is 1744. Each pixel module contains about 4.6×10^4 readout channels, such that the barrel region counts about 67×10^6 readout channels and the endcaps amount to 13×10^6 individual channels.

The following section explains the front end electronics that make up this tremendous number of readout channels.

2.4.1. The Pixel Module

The smallest active unit of the Pixel Detector is called Pixel Module. The active region is 2 cm by 6 cm, the sensor is made of silicon. The geometry of the pixel module is shown

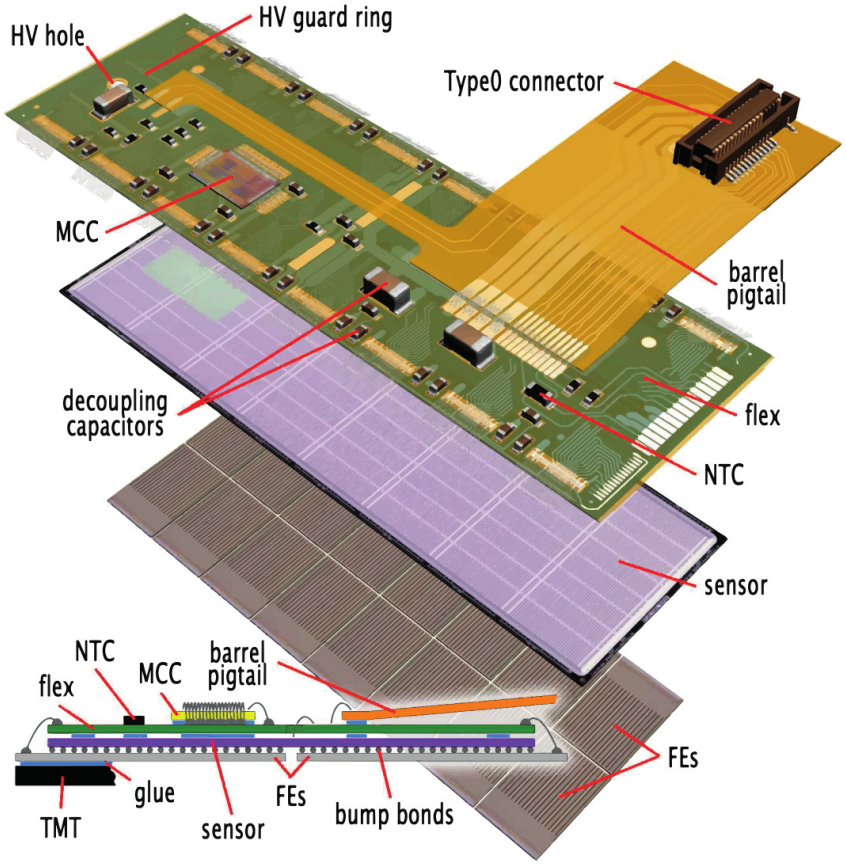

Figure 2.5.: The Pixel Module of the ATLAS Pixel Detector in an exploded view showing the Front-End chip, the sensor and the module flex. [4]

in figure 2.5. It consists of three layers: the sensor material is located in the central layer of the module, 16 *Front End (FE)* chips located on the bottom[9] of the module in figure 2.5. The top layer contains the *Module Control Chip (MCC)* and the connection to the off-detector components via a flexible cable. The following paragraphs describe each layer in detail to get a basic understanding of the design and data generation of the pixel modules, which will be relevant later on when discussing the readout system.

The Sensor is made out of silicon, which is a semiconductor. The electrical properties are in between a conductor and an insulator depending on the band gap between conduction band and valence band. By injecting foreign atoms into the crystal lattice of the element, one can manipulate the electric properties of the material very conveniently thus designing electronic components on the micrometer scale. This process is called *doping*.

[9]The FE chips are facing away from the interaction point in the actual Pixel setup.

By choosing a doping element that has less or more Wertigkeit than the bulk element, one can either create additional electrons (n doped) or additional positively charged holes[10] (p doped) in the conduction band, both changing the electric properties of the material.

One of the most fundamental building blocks of todays electronics is the diode, which the silicon sensor is much similar to. It consists of a p-n junction where a p doped zone and an n doped zone are connected. This assembly allows for current to only move into one direction, such that, the electrons move from the n doped zone to the p doped zone. If a voltage is applied in the other direction, the area around the junction gets *depleted* meaning that no charge carriers (holes or electrons) are present in that area.

The detector works similar to a diode operated in reverse bias. When a particle interacts with the depleted volume, it creates electron-hole pairs along its path due to its energy loss via ionization described by the Bethe-Bloch formula. The free charges drift towards the electrodes due to the bias voltage applied to the sensor[11] as displayed in 2.6 on the left. On the surface of the silicon bulk, the anode is continuous along the lower part of the sensor. The cathode has a pixeled surface, whereas each pixel has its own readout channel on the FE chip. The connection is done via bump bonding, a special soldering technique that can produce conducting connections on a very small scale.

The Front End Chips are located at the bottom of the module. To readout the entire sensor, 16 FE (ordered in an array of 2 by 8) chips are needed to collect the charge, digitize and format the collected information from each pixel cell. The chips have been manufactured on a sub micron level by IBM, such that all of the following operations can be done right on the chip. The design has further been optimized to be immune to radiation to guarantee functionality in the inner detector region.

The FE chip provides two cascaded amplifiers for each readout channel followed by a differential discriminator. The threshold of the discriminator can be adjusted in the range of 0 to 12×10^3 electrons. Typically the threshold is tuned to about 3×10^3 electrons using two *Digital to Analog Converters (DACs)*. One of the DACs is setting a common threshold for all readout channels at once called *Global DAC (GDAC)* and the other one, the so called *Trim DAC (TDAC)* is capable of addressing each channel individually to tune the pixels more homogeneously.

The *Module Control Chip (MCC)* is located on the top of the module and handles the communication between the FE chips and the off-detector electronics. It receives

[10] a positively charged hole is created when one electron is being moved from the conduction band to the valence band, which is, just as the electron, considered as a free charge carrier.

[11] For the ATLAS Pixel Detector the typical bias voltage for the sensor is around 150 V.

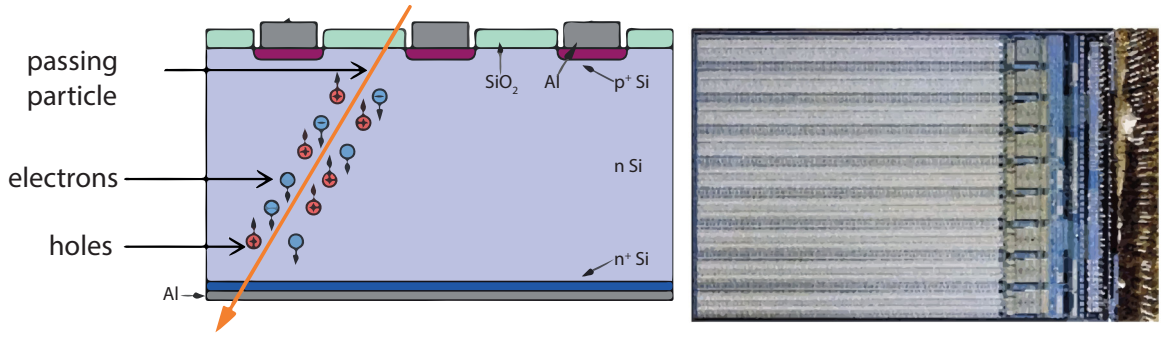

Figure 2.6.: Left: Principle of charge collection in a silicon detector.
Right: Front End chip with multiple readout channels and wirebonds.

configuration data and distributes it to the FE's. On top of that, it passes through and formats the digital data generated by the front ends and formats it for optical transmission by the optoboards at the end of the staves.

After the data has been processed by the MCC, it is transmitted via the ATLAS Pixel Readout Chain towards the off-detector side, where the analysis (tracking, reconstruction, etc.) is being performed. The current Pixel Detector readout system is described in the following section.

2.5. The Readout of the ATLAS Pixel Detector

To investigate the current ATLAS Pixel Readout System, firstly a short introduction to the nature of readout systems in high bandwith detectors is given. After the discussion of the current system, different upgrade scenarios will be introduced in the following chapters.

2.5.1. Fundamental Principles of Detector Readout Systems

The purpose of the rather complex readout systems designed individually for each detector component is twofold. During regular detector operation, while physics data is being taken, the readout system's main purpose is to gather the produced data from the detector modules and provide the first low level decision algorithms, called *Level 1 Trigger (L1 Trigger)* reducing the data rate. A very basic diagram is shown in figure 2.7 to illustrate the different inputs and outputs of the readout system. Collecting information like trigger information and time stamps from an LHC wide system, a physics event is built with the collected raw data from the Pixel Detector. This information is transmitted to the L2 trigger system, which takes different detector components, like the muon chambers, into

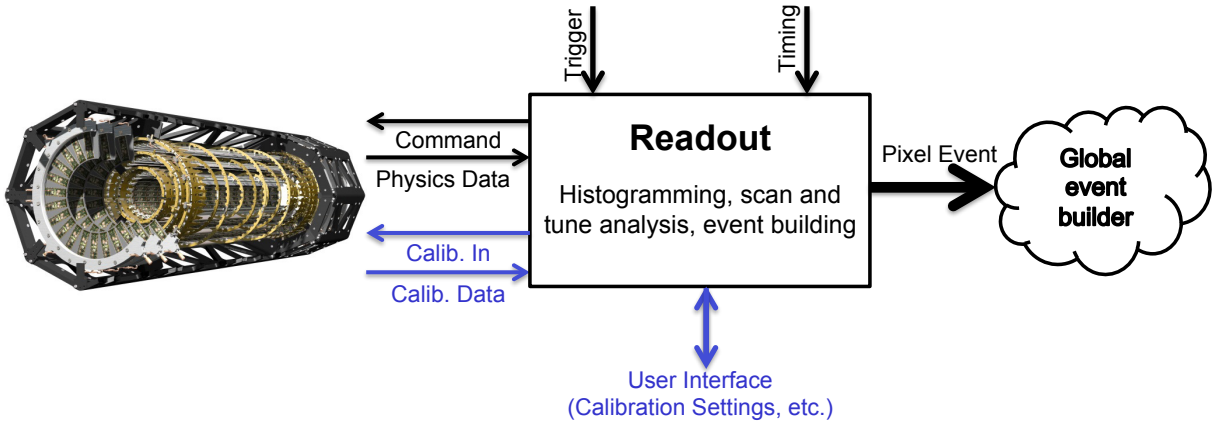

Figure 2.7.: Basic sketch of the two main purposes of the ATLAS Pixel Readout System, data acquisition (black) and calibration (blue).

account to either reject the data set or to build a global event for the entire pixel detector. The event data is sent to a computing farm for further processing.

If the detector is not in physics mode, the components of the Pixel Detector are being calibrated (shown in blue in figure 2.7). Via a user interface, one can start scans to determine characteristic quantities of the detector including operation functionality, thresholds for particle detection and many more. It also handles the GDAC and TDAC tunes described in the previous section. The calibration data is being evaluated from the detector modules' raw data directly on the readout boards via several very powerful *Field Programmable Gate Array (FPGA)* processing units. The data is passed through a histogrammer generating a discrete data set, which is then fitted to determine the properties required for calibration. The calibration mode is thus the more challenging state for the readout system than the physics operation mode, since the latter does not require any calculation, but merely a selection and book keeping.

2.5.2. Current Pixel Detector Readout Systems

The current readout system can be divided into two main parts [8]. The first one, the on-detector electronics has essentially been described in the previous chapter, explaining the MCC and front end chips inside the pit, where the detector is located. The data is formatted and transported to the off-detector side, where the histogramming, fitting and event building are performed. In the following discussion, the readout chain on the off-detector side is briefly introduced.

The communication between the modules on the detector and the off-detector electronics is done via optical fibers to ensure a connection without any capacitive interference.

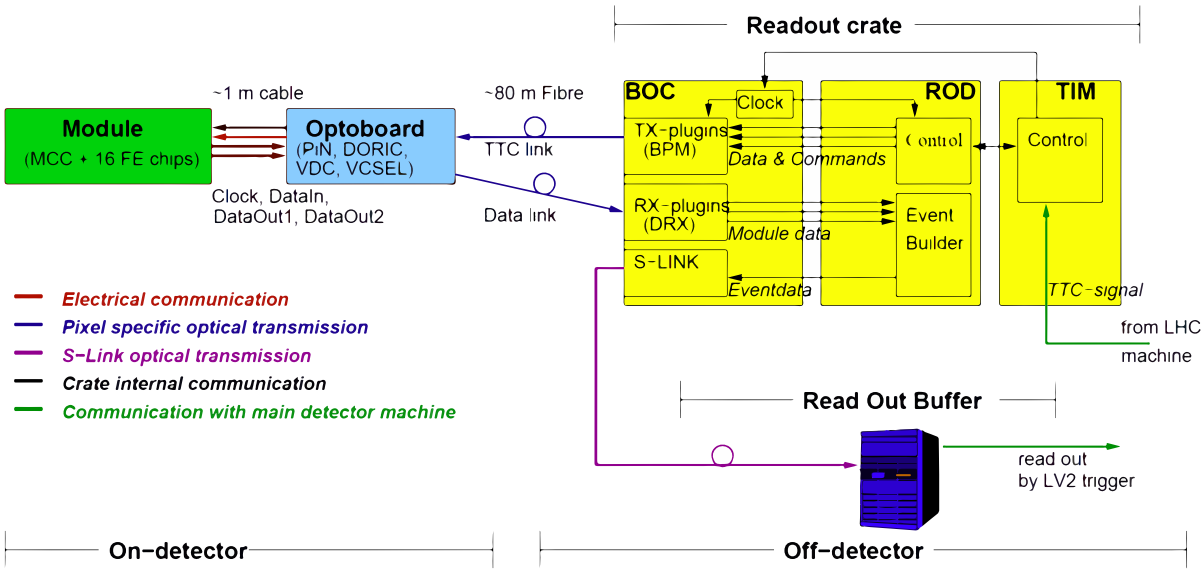

Figure 2.8.: Readout chain of the ATLAS Pixel Detector [4]

They are capable of a maximum data rate of about 3 GB/s. At the end of the staves in the inner side an optoelectrical interface is needed, as well as with the off-detector electronics. This is realized by an optoboard on the on-detector side and the so called *Back of Crate (BOC)* on the off-detector side.

The Optoboards are about 1 m away from the modules and receive the data electrically in a special type of the *Low Voltage Differential Signalling (LVDS)* format from one half stave[12]. This means that 6 to 7 modules are connected to the same optoboard. That way, if one of the optoboards fails, not too much information is lost. On top of that, the readout to both sides of the inner detector is symmetric.

The optoboards of the innermost layer contain two 8 channel arrays of *Vertical Cavity Surface Emitting Laser (VCSEL)* diodes to transmit data rates of $n \cdot 160\,\mathrm{MB/s}$[13]. Because the off-detector side can only handle 80 MB/s per channel, the data are split into two parts, each being transmitted at 80 MHz. The middle and outer layer transmits data at 80 MHz and 40 MHz respectively, such that only one VCSEL is needed, whereas the 80 MB/s signal also operates on a 40 MHz clock, but with data bits on the rising and falling edge.

To receive *Timing Trigger and Control (TTC)* information from the off-detector side, the optoboards contain an array of eight *Positive Intrinsic Negative (PIN)* diodes used as photo diodes.

[12]For the disks, the data are grouped by sector.
[13]The number n being either 6 or 7, depending on how many modules are connected.

To save resources and to make the data transmission more reliable, the (slow) control and the clock line are *Bi-Phase Mark (BPM)* encoded, meaning that the two independent signals are combined to one signal, whereas the command information is coded into the polarity of the transmitted bits. The coding further guarantees a transition from logical '0' to logical '1' within one clock period. This is a very important aspect of the coding, because it helps keeping the clock and command data synchronized more easily. Usually the command line transmits many zeros and then, for a short while, a few nonzero bits, such that most of the common phase aligning methods become unreliable. Additionally the transmitting lasers are kept at relatively constant temperature, because they are not shut down for long periods of time.

Back Of Crate (BOC) The optical data is received on the off-detector side by the BOC. The BOC has been developed for both the SCT and the Pixel Detector. The hardware is equivalent, only the number of connected modules per BOC is different. For the innermost layer, one half stave is connected to one BOC, while in the other two layers either two or four half staves are connected[14], according to their bandwidth because of internal data rate and the processing limits in the bus system and processing units.

Via the s-link protocol the data is being forwarded to the global ATLAS Readout system (as can be seen in figure 2.8).

In total, 132 BOC boards are needed to receive all the data from the 272 optoboards on the on-detector side giving a ratio of about one BOC for every two optoboards. The 132 BOC boards are located in nine crates of the *Versa Module Europe (VME)*[15] telecommunication standard.

The VMEbus standard handles the communication between the RODs and BOCs. It has been designed in the 1980s and is an industrial bus system, that was mainly used in telecommunication infrastructures. Bus systems are mainly characterized by their so called backplanes, which the clients of the bus communicate through. It thus determines the data width and the maximum speed, that the system can handle. Additionally the bus provides a communication network managing data flow and distributing commands and interrupts. The VME standard has undergone a series of upgrades over the last years and has increased both in bandwidth and data rate. The latest version, VME64x provides a maximum data with of 64 bit and a maximum data rate of 160 MB/s. The

[14]Because the readout of the detector is symmetric to both sides there is never one full stave connected to the same BOC.

[15]VME denotes a data bus system developed by Motorola, Mostek, Philips, Thomson und Signetics, which has been standardized by several ISO norms and the *International Electrotechnical Commission (IEC)* unter IEC 60281.

backplane is capable of handling up to 20 slots in one crate on each side of the backplane. The *Back Of Crate (BOC)* is located, as the name suggests, behind the backplane, while the so called *Read Out Driver (ROD)*, that will be described later, is located at the front. Communication across the backplane on the VME bus is coordinated by the crate controller, a small *Single Board Computer (SBC)* that also handles communication with the user interface.

The *Read Out Driver (ROD)* The main part of the readout chain is the ROD. It is divided into various functional groups that fulfill their purpose mainly independently (see figure 2.8 for reference). It gets inputs from the *Timing Trigger and Control Interface Module (TIM)*, managing the synchronization of data with the external and global LHC clock, which is vitally important for the ROD's main purpose, building of a Pixel event, including reconstructing time stamps and processing trigger signals.

The Controller module is mainly using the TX plugin of the BOC to send commands to the on-detector side. The Controller generates commands from the interface of the user input and the TIM. The other and more challenging component is handling the event building. The event builder takes the timing information from the TIM and the event data collected from the optoboards and processes them. Additionally it delivers the computing capacity for the evaluation of the calibration data. By means of two Spartan 6 FPGAs from Xilinx Inc. the data for two half staves can be processed and analyzed.

The reconstructed pixel event is sent from the ROD via the BOC via s-link[16] to the global ATLAS readout.

2.6. Field Programmable Gate Arrays

The building blocks, of the readout systems, the *Field Programmable Gate Array (FPGA)* is introduced, because the entire readout chain relies on high speed formatting of data, which can be done best by FPGAs [9].

The FPGA is an *Integrated Circuit (IC)* that can be programmed after manufacturing by the user, hence it is *(application) field programmable*. Since it is programmable by assembling logic gates, the device is called a *Field Programmable Gate Array*. In figure 2.9, the general structure of the FPGAs is shown, containing all the basic building blocks, that are necessary for this thesis. In the following, the individual components are introduced briefly.

[16] *simple link interface (shortly s-link)* is a high performance data protocol developed at CERN. It operates on a 32 bit bus with a frequency of 66 MHz and can thus transfer 264 MB/s and is usually realized via optical fibres.

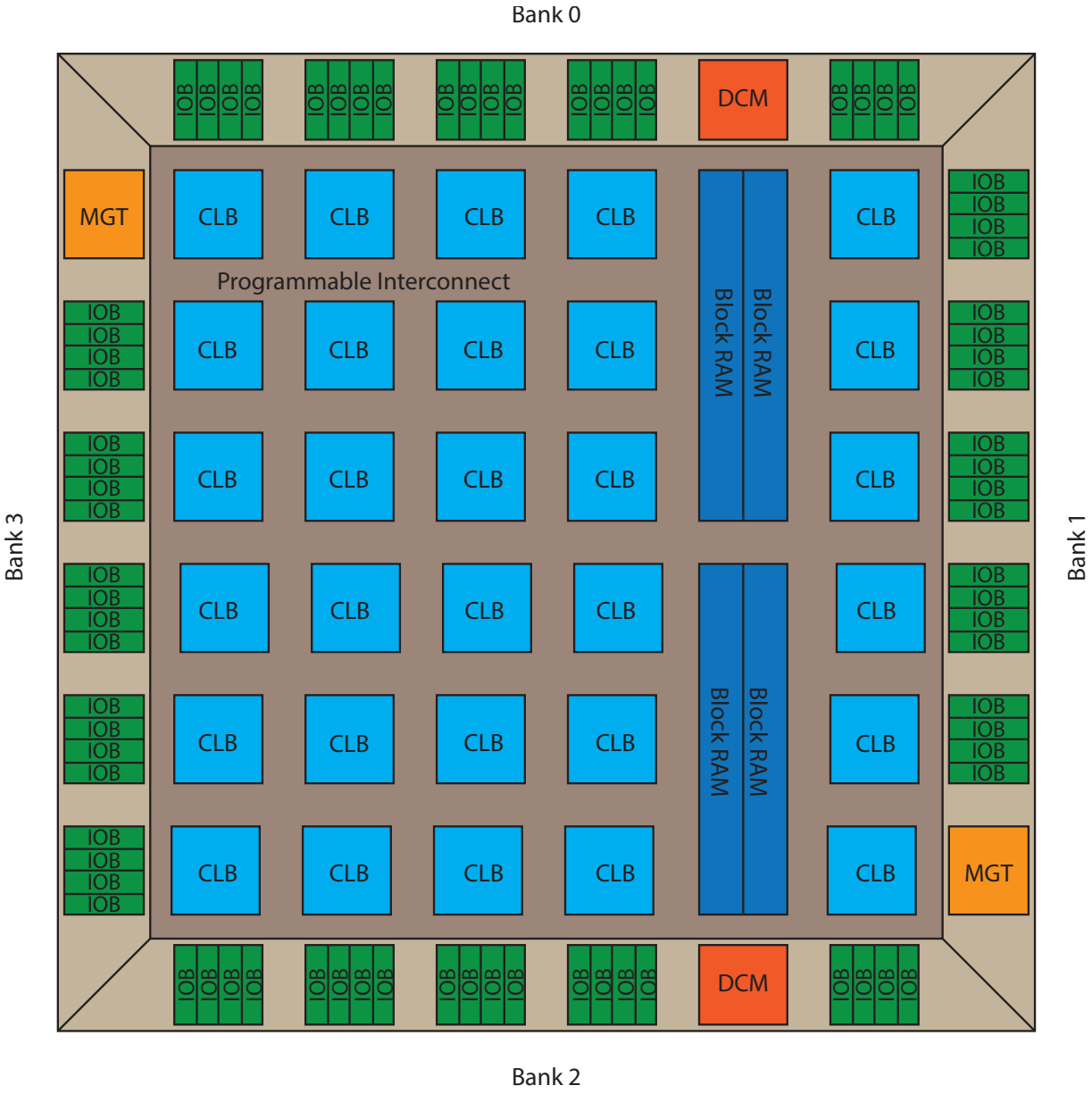

Figure 2.9.: Generic FPGA structure. Actual FPGA models have significantly more blocks and I/Os.

Input/Output Buffers (IOB) are located at the sides of the FPGA and are grouped into four banks. As the name suggests, they receive and transmit data to the external electronics. To ensure better signal quality and also to shield the FPGA from external voltage spikes, the signal is buffered before it exits or enters the FPGA. As additional precaution, most of the IOBs are terminated with external capacitors to eliminate dangerously high voltages.

Custom Logic Blocks (CLB) are located in the inner part of the FPGA. They host the logic programmed by the user. Each block contains a certain part of the firmware running on the FPGA. Using an *Hardware Description Language (HDL)*, one can program individual logic blocks in a modular fashion, as will be discussed later when the coding done in this thesis is described. The individual logic blocks are connected via the *Programmable Interconnect* through which all signals are routed and connected to the correct logic blocks or I/O buffers.

Block Random Access Memory (BRAM) is also located in the inner part of the FPGA and is used for short time storage of data. Depending on the size and the model of FPGA in use, the BRAM has different sizes. For memory-intensive applications, it is recommended to put external RAM onto the board and to address it via the I/O banks.

Digital Clock Managers (DCM) are usually located in the I/O banks. They are used for generating clocks as reference signals for the logic inside the FPGA. To be able to generate high quality clock signals, it is important that they have direct access to an external reference clock, usually coming from an on-board oscillator. They thus have to be located on the I/O banks, so the reference clock does not deteriorate before it is converted by the DCM.

Depending on the model of FPGA, different components are used for clock generation, which have slightly different properties. In the Virtex 4 from Xilinx, DCMs are still used, while Virtex 5 uses *Phase Locked Loops (PLL)* that have increased clock quality, but poorer phase shifting accuracy. The two concepts are combined in the Virtex 6, where the *Mixed Mode Clock Manager (MMCM)* is capable of very precise phase shifting and high quality clock generation.

Multi Gigabit Transceivers (MGT) are also located on the I/O banks. They are specially designed IOBs to allow for high speed serial communication. The module on the FPGA generates – again depending on the type of FPGA used – signals of up to 28 GB/s.

This block is also different for each generation of FPGAs. While the Virtex 2 to 4 rely on the GT10 and GT11 blocks, beginning with Virtex 5, the GTX design is used. The latter is easier to configure for the user logic data input and is capable of handling several industry standard I/Os, such as *PCI Express (PCIe)* or *Gigabit Ethernet (GbE)*. Special encodings can simply be selected by the user and are applied automatically by the GTX, such as the 8b10b encoding, that is used for ATLAS and will be described later.

The output ports are usually connected to either a *Media Access Controller (MAC)* to send the data via GbE or PCIe or to a *Small Form-factor Pluggable (SFP)*, that is

capable of transmitting high data rates. For the receiver, the GTX, as well as the GT11 are capable of active clock recovery, meaning that the transmitted data is encoded into the clock signal such that the transmitter and receiver are always synchronized.

3. Upgrade Scenarios for HL-LHC

This chapter firstly shows the general upgrade plans for the LHC in the near and far future aiming towards a *High Luminosity LHC (HL-LHC)* within the next two decades. Additionally the ongoing development of the hardware for the ATLAS Pixel Detector is studied.

3.1. Upcoming LHC and Detector Upgrades

The upcoming LHC upgrades planned to be implemented within the next decade can be combined into three major upgrade phases as done in [10]. The series of upgrades, starting with the first one in 2013, are all aimed for roughly a tenfold increase in luminosity – hence the name *High Luminosity LHC (HL-LHC)* – at the LHC's nominal *center of mass energy (CM energy)* of $\sqrt{s} = 14\,\text{TeV}$. So far, the LHC has been running at half the CM energy that it was designed for and has experienced a small energy upgrade to $\sqrt{s} = 8\,\text{TeV}$ in 2012.

Since quantum mechanical phenomena, like Standard Model processes, abide by the laws of statistics, increasing the number of events will mean that statistical analyses can be done much more efficiently. In figure 3.1 it is shown, how many years of continuous data taking are necessary to halve the error on the quantities in the Standard Model for both an LHC and a high luminosity LHC scenario. In this case, the plot only has symbolic meaning, because the errors does not solely depend on the luminosity, but much more on the analyses used, but it gives a good impression on the impact, a high luminosity upgrade will have on the physics analyses. One can deduce from this plot that in order for the LHC to be useful for state of the art analyses, a luminosity upgrade is indispensable, given that the detector hardware has a limited lifetime of approximately six years in current detector operation. It is also depicted how pausing the data acquisition for an upgrade will be caught up within one or two years after the upgrade, underlining the necessity for a recent upgrade.

In the following, the three major upgrade phases are sketched with the remark that the measures taken are still subject to changes and reconsiderations.

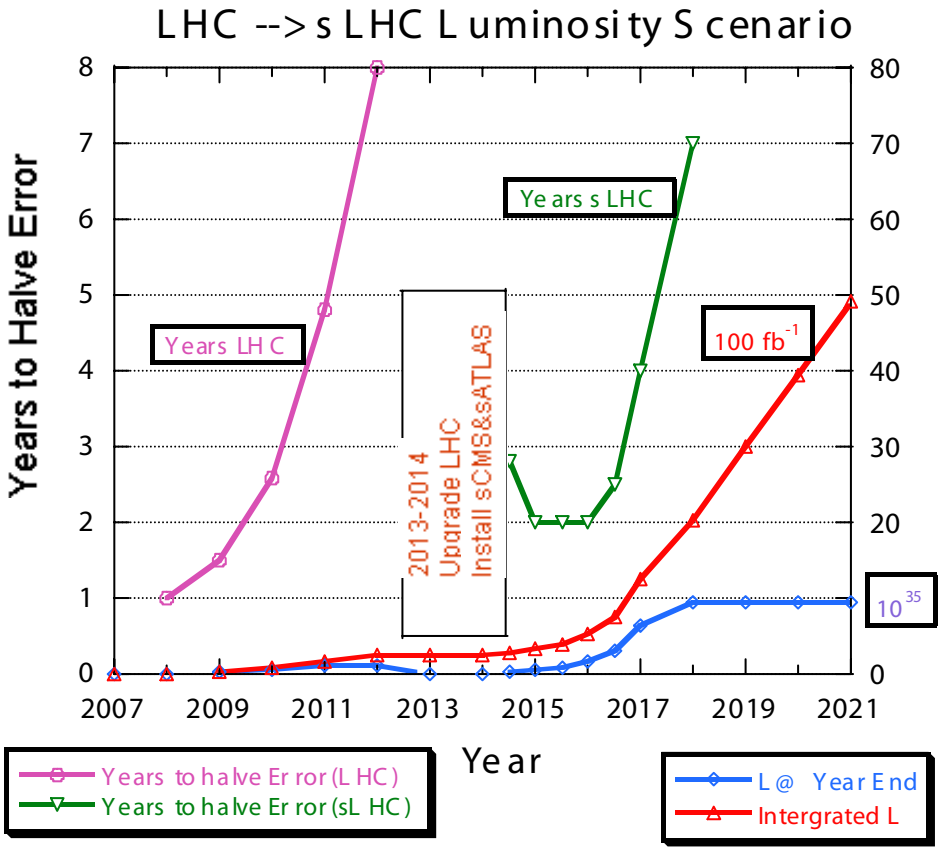

Figure 3.1.: Time of continuous data taking necessary to halve the error on Standard Model quantities as well as integrated and instantaneous luminosity as a function of time.

Phase 0 After a two year shutdown scheduled for 2013, the LHC is supposed to reach its nominal CM energy of $\sqrt{s} = 14\,\text{TeV}$ and luminosity of about $2 \times 10^{34}\,\text{cm}^{-2}\text{s}^{-1}$. Since the LHC was originally designed for operation in this mode, no major changes are necessary. To achieve the desired luminosity, the bunch size[1] is increased by approximately a factor of 1.5. To further increase the focus of the beam, it was suggested to only collide the beams at the main interaction points in ATLAS and CMS.

As can be seen from figure 3.1, the timing for the first upgrade phase at this time is chosen to be very good.

To prepare for the future increase in luminosity and to test some new hardware, the ATLAS Pixel Detector is upgraded in the course of the 2013 shutdown. The *Insertable Barrel Layer (IBL)* will be put into the inner detector as a fourth layer. Besides novel FE chip for the modules, about 25% of the sensors deployed will be 3D sensors[2]. This

[1]The protons in the LHC are organized in bunches that are brought to collision. Typical bunch sizes are around 1.1×10^{11} protons.

[2]The 3D structure of the sensor material will be described in more detail in 3.2.

upgrade will also affect the b-tagging efficiency in current jet algorithms and will help to deal with the expected increase in pile-up[3].

Phase 1 The phase one upgrade, scheduled for around 2017, is supposed to increase the luminosity by a factor of two. This cannot be achieved without additional hardware modification. For this update, costly changes throughout the entire accelerator are to be avoided though. At the interaction points at CMS and ATLAS, the quadrupole magnets focusing the beam will be exchanged by magnets having half the original focal length. That way the bunches are compressed more when they enter the experiments. The LHC arcs themselves remain unchanged. Minor changes like adjusting the bunch size and the amount of particles per bunch crossing have also been considered.

Phase 2 The final upgrade that is planned for around 2022 includes major changes to the accelerator infrastructure in front of the experiments, as well as at the LHC arcs, thus achieving another fivefold increase in luminosity. In this scenario, not only the LHC will have to be modified. Also the preaccelerator SPS will have to be equipped with superconducting magnets which will prevent defocussing of the beam during injection.

In each one of the mentioned upgrade phases, the detector performance is supposed to remain unchanged meaning that despite the increase in luminosity by a factor of ten, the efficiencies for tracking, calorimetry, etc. are supposed to remain the same. Thus the detector hardware needs significant upgrades throughout all parts of the detectors. In the following, upgrade plans for the ATLAS Pixel Detector, namely the FE chip and the read-out system are discussed.

3.2. The Insertable B Layer

In the beginning of 2013, the LHC ended the first physics run and began the phase 0 upgrade. Along with the accelerator upgrade, the ATLAS collaboration has decided to upgrade the Pixel Detector by adding a fourth layer to the current detector [7]. Because IBL is so close to the interaction point, the beam pipe has to be modified to fit the IBL into the Pixel Detector. This can be seen in the rendering of figure 3.2. The radius of the new layer is just 35 mm, leaving 15 mm of space to the B-layer between the centers of the modules. Including service lines and cables, the distance is even smaller. Including all of

[3]Pile-up is defined as the number of interactions per bunch crossing. The current detectors are designed for about 20 simultaneous interactions, but at higher luminosities and bunch sizes, the pile up is going to increase significantly.

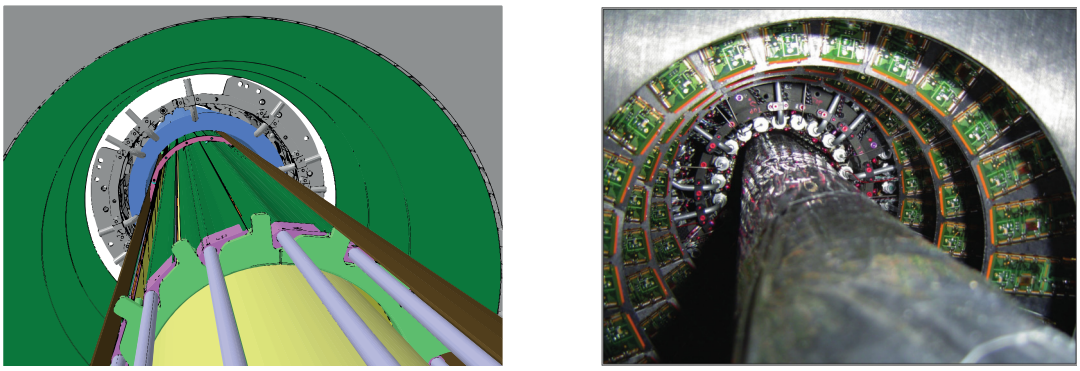

Figure 3.2.: Left: Rendering of the IBL project deployed into the original Pixel Detector. Right: Image of the ATLAS Pixel Detector with the inserted beam pipe [7].

the service panels and on-detector readout, the entire object is about seven meters long making the insertion of IBL a very delicate task.

Two main reasons justify taking the risk of inserting the IBL into the current detector. One of them is the amount of radiation the innermost layer has taken so far, which causes damage to the modules and diminishes the amount of signal that is received. Inserting IBL would thus provide the opportunity to put a detector with increased radiation hardness in between the damaged B- layer and the interaction point.

The second reason is to ensure that the detector can withstand the desired radiation fluence[4] of about 3 to $5 \times 10^{15}\,\mathrm{n_{eq}}\,/\mathrm{cm}^2$ and the increased data rates, the on-detector components that experience the highest amount of radiation dose, the FE chips and sensors have to be modified. This will be described in detail in the following chapter.

To complete the picture of the IBL, table 3.1 shows the most important characteristics of the device. It has 14 newly designed staves that hold 16 double chip modules or 32

Property	Value
Number of staves	14
Number of modules per stave (single/double chip)	32/16
Pixel size	50 by 250 µm
Active size per module (double/single chip)	16.8×40.8/ 20.4 mm^2
Center of the sensor radius	33.25 mm
Sensor material used	planar silicon, 3D silicon

Table 3.1.: Characteristic quantities of the IBL detector.

single chip modules (as opposed to the current staves holding 13 modules). This way, the

[4]The unit refers to the NIEL Scaling Hypothesis stating that radiation doses of an energy distribution can be scaled to the radiation dose of a reference particle, in this case referring to 1 MeV neutrons.

entire data infrastructure has to be redesigned to cope with the increase in modules and with the different data rates.

The main difference though between the current staves and the IBL staves is the newly designed FE chip and the fact that for the first time 3D sensors will be used.

Unlike planar sensors described in the previous sections, the n^+ and p^+ doped electrodes are not located solely on the surface but have a pillar structure reaching far into the bulk material. That way the distance between the electrodes can be adjusted without losing a lot of signal. Reducing the distance of the electrodes will reduce the depletion voltage and thus extend the lifetime of the sensor material when irradiated.

3.3. The FE-I4 Chip Generation

To cope with the higher data rates due to the closer proximity to the interaction point and – in the future – with higher luminosities, a new FE chip has been designed [11]. While the current detector hosts the FE-I3 chip generation, the IBL detector will host the FE-I4 chip. The design has been finished in 2010 and has two different versions, the FE-I4A and the FE-I4B. It has especially been designed to cope with higher data rates by using a smaller pixel size and by adjusting the regional architecture. The latter enables the information to be flushed faster along the columns towards the main readout. The main properties of the FE-I4B chip, the production version[5]for IBL, are described in table 3.2.

	FE-I4B	FE-I3
Year of Development	2011	2003
Feature Size	130 nm	250 nm
Chip size	20 by 19 mm^2	7.6 by 10.8 mm^2
Pixel	80 by 336	18 by 160
Pixel Size	50 by 250 μm^2	50 by 400 μm^2
Data Rate	up to 320 MB /s	40 MB /s

Table 3.2.: The FE-I4B compared to its predecessor, the FE-I3.

To make the FE-I4 more radiation hard than the FE-I3, the structural size of the chip has been decreased from 250 nm to 130 nm. At the same time, the size of the chip and its number of readout channels have been increased by a factor of roughly four and two respectively. For IBL, modules have been designed with only one or two FE-I4 chip per sensor, unlike in the setup of the current Pixel Detector. The geometry of the FE-I4 and

[5]An FE-I4C version has already been planned and first prototypes are being tested, but FE-I4B is the newest chip generation that is available.

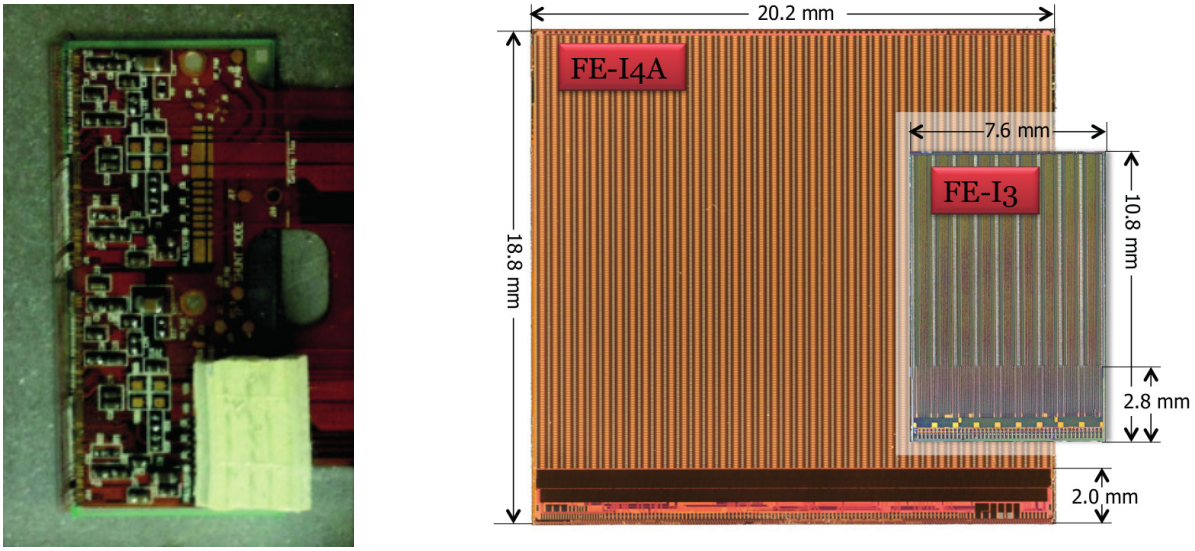

Figure 3.3.: Left: IBL module with FE-I4B and V3 Flex from INFN. Right: The FE-I4 chip to scale with the FE-I3 (the latter as in 2.6) [11].

an IBL module can be found in figure 3.3 as well as a photograph of an FE-I4 module used for IBL.

Even though the FE-I4 is capable of handling a readout rate of 320 MB/s, the current IBL setup only envisages a data rate of 160 MB/s. Here, the designers of the FE-I4 planned for the future. With further upgrade phases, the readout speed of IBL might not be sufficient any more, but the modules allow a doubling of the data rate. Thus, a HL-LHC scenario could be compensated by doubling the amount of readout channels or upgrade the readout altogether but the detector itself can remain unchanged.

Thus the IBL detector already requires an upgrade of the readout system, because of the fourfold increase in data rate. The ROD/BOC system in its current state is limited to a bandwidth[6] of either 40 MB/s or 80 MB/s.

3.4. The IBL-ROD

When the upgrade plans of the LHC were fixed, the Pixel community began the planning of the IBL and the concurrent readout systems. Since the time frame for the first upgrade in 2013 and the integration of IBL was quite short, it was decided that the readout system would not be rebuilt from scratch. Thus it was decided to proceed with an upgraded version of the RODs using up to date components and minor adjustments to

[6]The two streams of 80 MB/s are split into four signals of 40 MB/s internally on the ROD/BOC interconnect though.

cope with the higher data rate. The data distribution had to be redesigned to incorporate 32 FE-I4 chips with 160 MB/s each, as opposed to eight channels with 160 MB/s in the original design. The main bottle neck for this operation is the VME backplane limiting the communication between the RODs. The VME backplane has a small bandwidth compared to modern technology.

Beside the relatively small adjustments necessary, another advantage is the backwards compatibility of the ROD. Thus a unified readout can be designed using the same user interface and software packages. Also most of the firmware code from the original RODs can be reused. As a compromise, some of the processes usually done by the RODs themselves had to be moved towards an external server. E.g. the fitting procedures analyzing the histograms produced by the IBL-ROD are planned to be transmitted to an external server via *Gigabit Ethernet (GbE)*.

Even though the IBL-ROD has made a lot of progress lately, it is still behind schedule and there are some doubts that it will be finished in time for the beginning of operation of IBL. Thus, the system described in the next chapter, based on ATCA technology, is the official backup solution for IBL and will certainly be part of one of the next upgrades of the ATLAS Pixel Detector.

4. The Reconfigurable Cluster Element Setup

The *Reconfigurable Cluster Element (RCE)* setup has been developed from scratch using a completely different backbone structure than its predecessor. As opposed to the – rather outdated – VME based hardware, a complete set of replacement boards has been developed for the *Advanced Telecommunication Computing Architecture (ATCA)* which solves a lot of problems at the same time. The complete infrastructure is shown in figure 4.1.

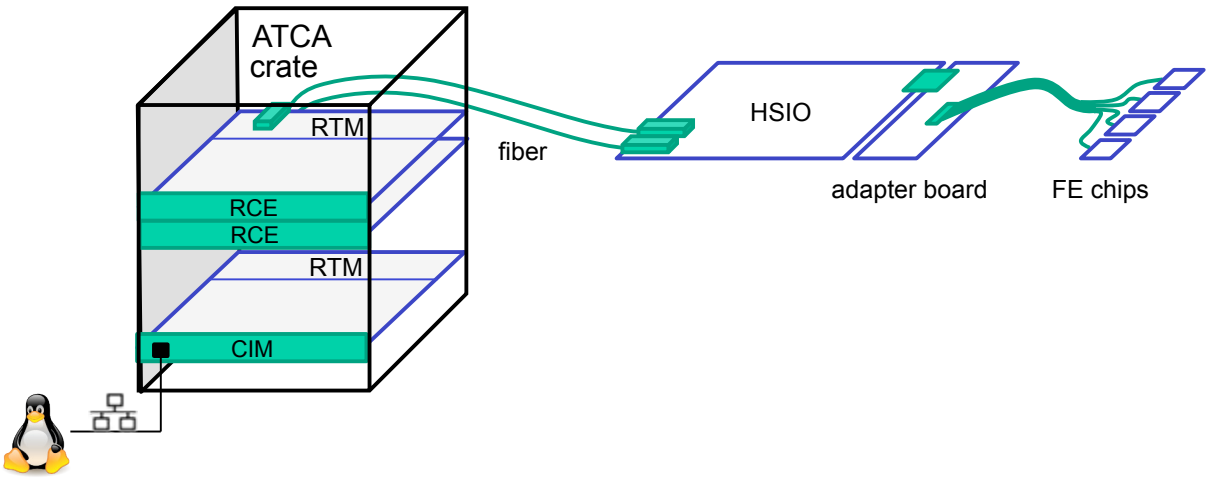

Figure 4.1.: Basic sketch of the RCE setup with all the vital components shown.

The setup is in many respects similar to the original setup but the components are different. In the following elaboration, the individual parts are described in detail. The elaboration is based on the publication from SLAC – the developers of the system – cited under [12].

The ATCA architecture defines a data transmission protocol that does not work like classical bus systems any more. Rather it uses a delocalized point to point connection that enables the slots within the crate for much faster data handling. A bus system, buffering the information (like it has been with the VME standard) does no longer suffice

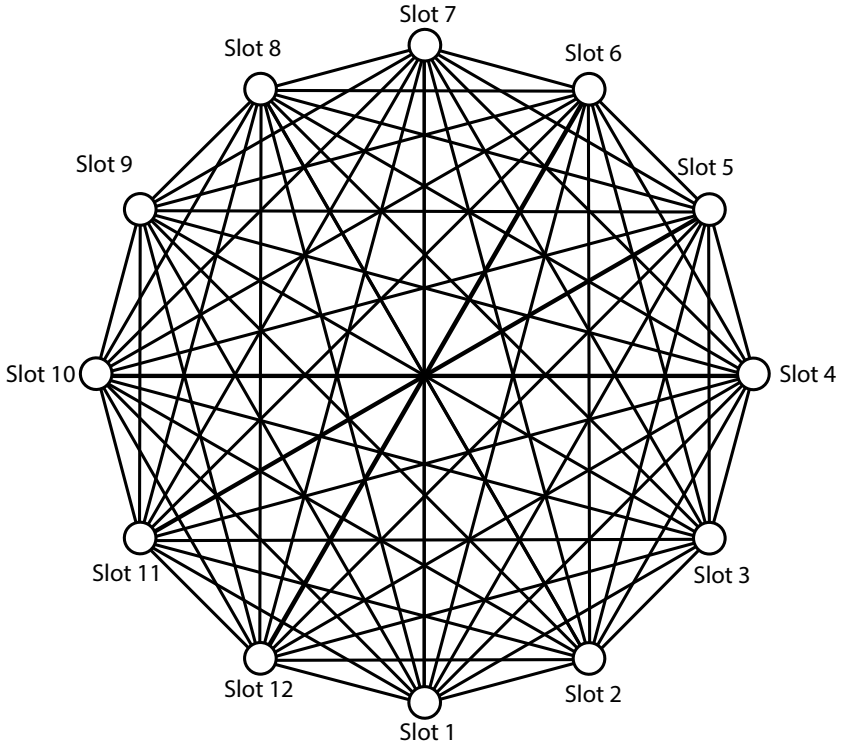

Figure 4.2.: Exemplary map of a twelve slot ATCA crate data transmission network.

to fulfill the increased data flow that has to be handled nowadays. An example of the connection lines for data flow is shown in figure 4.2.

Originally the ATCA standard was developed for handling multi channel data flow in telecommunication and Internet infrastructure applications. The ATCA standard has been developed by more than one hundred companies and has been standardized by the *PCI Industrial Computer Manufacturers Group (PICMG)* in 2002. This standard includes, besides the properties mentioned above, a high level of *reliability, availability and servicability (RAS)*[1].

The ATCA backplane is divided into three different zones that are mostly independent of each other. Zone-1 provides redundant power with a nominal voltage of $V_{DC} = -48\,\mathrm{V}$ and shelf management signals from a global crate controlling interface.

Zone-2 provides point-to-point connections between the individual clients of the ATCA crate as mentioned before. This zone is called 'fabric agnostic' meaning that it uses $100\,\Omega$

[1]The acronym was shaped by IBM and was used to describe the robustness of mainframe computers in the early days. A higher RAS level implies a multitude of features that help devices stay available for long periods of time without failure. The uninterrupted *uptime* is a characteristic property for any fault tolerant system though and is applicable to a multitude of digital systems.

differential signals, that are met by signals such as the LVDS[2] standard. This zone is used for data handling during detector operation and for communication between the different parts of the detector readouts.

Lastly there is a user defined connection interface in Zone-3 that enables the user to transmit signals that are not supported by the ATCA standard. Usually this zone is used to interconnect the *Rear Transition Module (RTM)* – located in the back of the crate – to the main electronics.

The three redundant zones of the ATCA infrastructure have been set up to increase the RAS level of the ATCA technology because powering, backbone and user lines are completely independent from each other and failure of one system does not automatically imply the failure of the entire infrastructure.

Rear Transition Module The RTM contains eight optical transceivers (operating at 3.125 Gbit/s), each converting a serial link from optical to electrical signals and vice versa to ensure communication with the optoboards on the on-detector side. Each of these optical fibers can connect to a half stave with the IBL setup. In the current setup, it is connected to the HSIO though which will be described in the following paragraphs. Via a Zone-3 customized connector the RTM is connected to the RCE module to transfer the electric signals through the ATCA backplane to the RCE boards for further processing. Beside the optical *Input/Output (I/O)* devices, the RTM also contains a GbE port enabling Zone 3 infrastructure access for user inputs. At this point this access point is only used for debugging purposes though.

Reconfigurable Cluster Element The RCE is a computational building block based on a Virtex-4 FPGA architecture with a PowerPC 450 core running at 450 MHz and 512 MB of RLDRAM-II. The architecture is based on the *System on Chip (SoC)* concept that can operate up to 24 I/O data lanes simultaneously with all of the needed computational blocks right in the FPGA logic and the PowerPC firmware. The lanes can be operated independently or combined together for higher data transmission rates. Each lane can process data at up to 40 Gbit/s. Interconnections are handled by the Zone-2 differential signal processing architecture that the ATCA crate provides. Each board groups together two independent RCE blocks that can each handle one half stave of pixel modules. One RCE board can therefore handle the same amount of FE-I4 pixel modules as its predecessor – the ROD – can handle FE-I3 modules (FE-I3 having a quarter of the FE-I4 data rate). A photograph of the RCE with a connected RTM is shown in figure 4.3,

[2]LVDS defines a certain voltage level for f signals that are widely used for robust circuits. The mentioned 100 Ω resistance is used to dampen signal reflections.

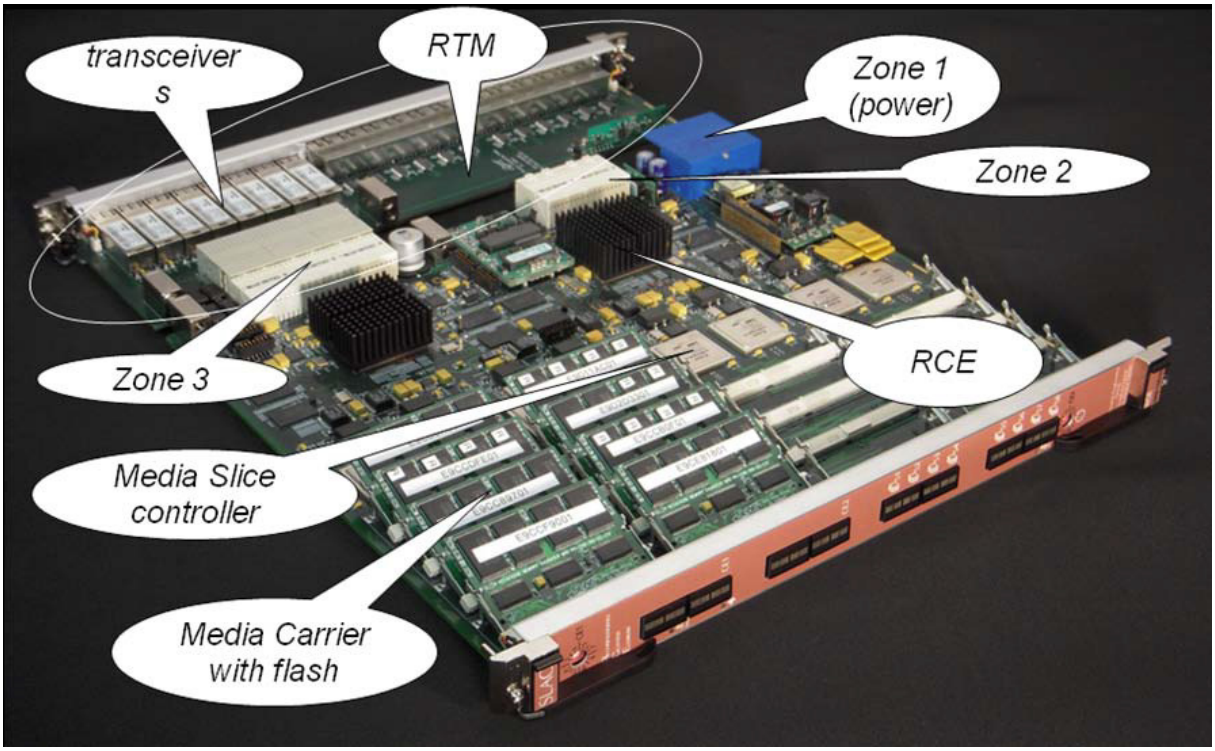

Figure 4.3.: Photograph of the Generation-I RCE with connected RTM. Media Slice Controller and Media Carrier with flash only relevant to the PetaCache Project[3] [12].

where all the important components of RTM and RCE are labeled. The RCE processors are underneath the black heat sinks.

Towards the front of the board, seven-segment-displays show the current operation mode of the RCEs and contain a hardware reset switch as well as some emergency pins. The board itself also contains a *Joint Test Action Group (JTAG)*[4] programming connector to update the firmware of the RCE and access a debug console for troubleshooting.

Cluster Interconnect Module The *Cluster Interconnect Module (CIM)* consists basically of a 10 GbE switch and is specifically designed to distribute user commands, like the initialization of a scanning or tuning algorithm to all the RCE boards in the ATCA crate. The architecture is realized by two PowerPCs that manage the data processing. This module is completely independent of the application that the RCE-system is being used for. Like a commercial ethernet switch, it is simply a generic data package distributing

[3]The PetaCache Project was initiated to produce a fast data access controlling unit that allows for instance LHC data to be accessed very quickly. More Information about this project can be found in the SLAC Today Newspaper [13].

[4]JTAG is a common transmission standard used by most FPGA testing boards to connect a debug device to evaluation boards.

system. The user can connect to the CIM via optical fiber or via GbE and uses TCP/IP to manage its clients within the crates.

High Speed Input/Output Module An additional auxiliary component that is already in use for tests involving FE chips with the RCE system is the *High Speed I/O Module (HSIO)*. This module handles the on-detector communication and could be considered an oversized optoboard. It contains a Virtex 4 FPGA for data processing. The main purpose is to receive the data from the FE chips and deliver commands, as well as a reference clock of 40 MHz or 160 MHz (depending on the FE type) to the FE electronics. The Virtex 4 serializes the incoming data and encodes it via the *Pretty Good Protocol (PGP)* to send the data to the RTM. The PGP was developed at SLAC as a frame based transmission standard compliant to *Peripheral Component Interconnect Express (PCIe)* that supports a lot of data formatting functions. As part of the physical layer of the PGP, the data is 8b10b[5] encoded and a phase recognition of the incoming signals is performed [14].

Since this thesis covers the on-detector hardware development for the RCE system, the HSIO plays a vital role in the further elaborations. Thus the firmware running on the FPGA as well as the encoding schemes and transceivers used will be described in section 6.2.

The Front End chips are connected via one incoming *sub LVDS (sLVDS)* line and two outgoing sLVDS lines for data transmission. The two outgoing connections deliver the clock and command signals to the FEs. Standard LVDS buffers terminated by a 100 Ω resistor (as described previously for the LVDS standard) are hosted on an adapter board that is available in several versions for different purposes of Front End testing. The incoming line receives the data coming from the FEs.

The entire setup and the interconnection of the parts in the ATCA crate will be discussed in the following two chapters introducing two different setups. Firstly, the system set up in the context of this thesis will be described, since it deviates slightly from the discussed setup shown in figure 4.1. The second application of this system describes the work done on the IBL stave electrical tests performed at the University of Geneva, which was also conducted as part of this thesis.

[5]Similar to the BPM encoding mentioned earlier, the 8b10b encoding codes clock and signal together mapping an 8 bit input to a 10 bit output, ensuring enough phase changes for decent clock recovery and good DC balancing.

4.1. RCE Setup in Göttingen

The RCE setup as described above, including RCE board, ATCA crate, CIM and HSIO have been ordered by the 2nd Physical Institute in Göttingen to get a research group in Europe to work on the development of this system and thus further promote the use of ATCA.

As the only available adapter board for the HSIO hosts a 100 pin connector usually used in mobile devices, a cable converting the 100 pin connector to the FE's RJ45 plugs was built. Details about the plugs, including the pinout and design can be found in appendix A.2.

Because of the increased interest in the system, and the limited number of prototypes that SLAC has produced, there was no CIM available to be shipped to Göttingen. The solution described in the following section has been executed in cooperation with Dr. Martin Kocian from SLAC.

4.1.1. RCE Operation without a CIM

As mentioned above, the CIM is merely a crate wide network switch for the RCE boards communicating to the outside world via *TCP/IP* and a few other protocols. The RCE boards are designed to only take commands from the CIM via the ATCA backplane. To be able to dispense of the CIM completely, an *Small Form-factor Pluggable (SFP)* capable off-the-shelf network switch was added to the system serving as an external CIM.

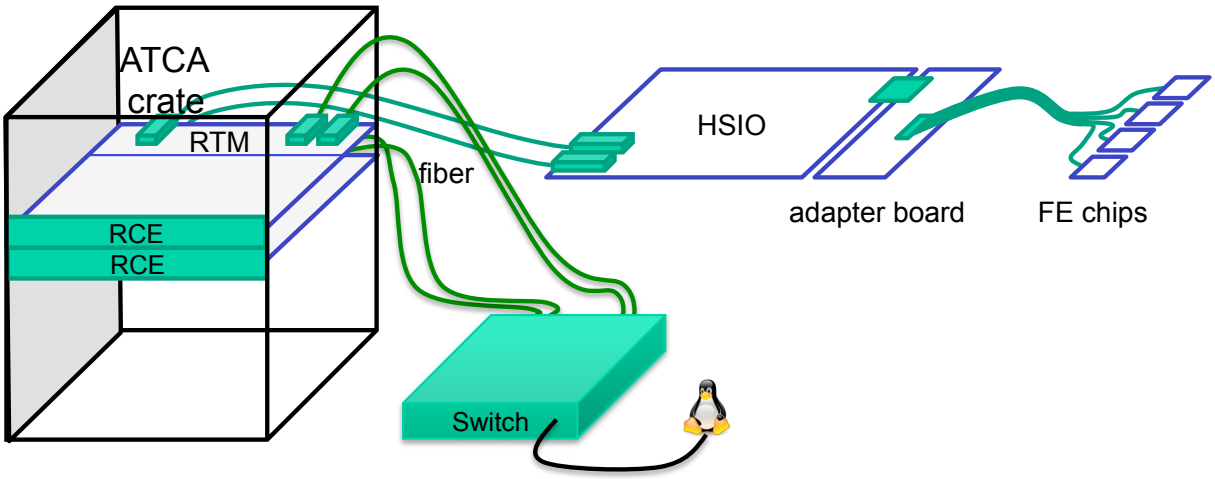

Figure 4.4.: Exemplary setup of the RCE configuration without a CIM as it was set up in Göttingen.

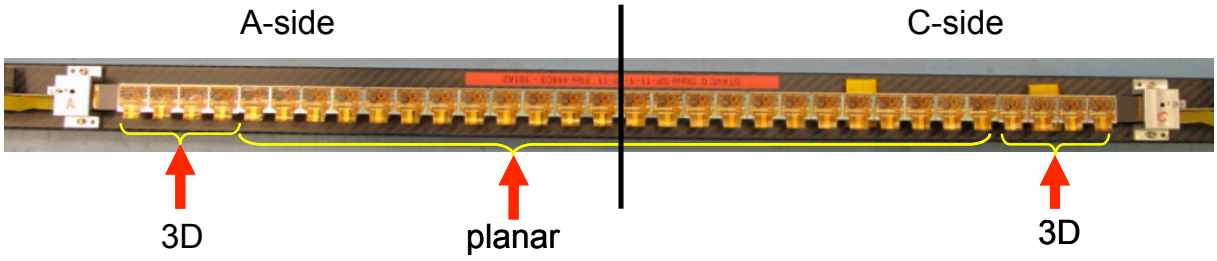

Figure 4.5.: Photograph of the Stave0 prototype during the first connectivity tests at the University of Geneva [15]

Additionally, the firmware of the Virtex 4 FPGAs on the RCE board was altered to map the network connection to a *Multi Gigabit Transceiver (MGT)* on the RTM that was unused so far. Equipping the network switch with optical transceivers, the CIM could be replaced by an off the shelf component using optical fibers to connect it to the RCEs. Each RCE computing unit thus requires an additional fiber cable as indicated in figure 4.4. For the user handling the system, nothing has changed in the current setup compared to the RCE setup with a CIM. The only difference between the setup in Göttingen and the regular procedure is that the number of RCE boards that are operateable at the same time is limited to the number of SFP slots in the network switch.

4.2. Stave tests for IBL Development

In the following section, the first large scale setup of the RCE system is described. Up to this point, only several modules were read out at once with the RCE system, mostly using custom made cables with RJ45 connectors. The Stave0 electrical tests at CERN in 2012 thus were the first large scale tests with a preproduction of an IBL stave.

The stave consists of eight single chip modules that have a 3D geometry and 24 planar double chip modules, as indicated in figure 4.5. In total there are eight single chip 3D sensors and 24 double chip planar modules. As described in previous sections about the current readout system, the staves are read out symmetrically to both sides, such that on each side an *End Of Stave (EOS)* card is located. The HSIO is capable of reading out the modules of two half staves simultaneously.

Due to a mismatch in the pinout of the 100 pin connector at SLAC and the University of Geneva, there was only one cable available to connect the EOS cards to the HSIO. Thus each test was performed for the A side and the C side (as indicated in 4.5) separately.

To simplify the setup in Geneva and to set up the connection of the modules through the flex more easily, some adjustments to the HSIO firmware were made in advance.

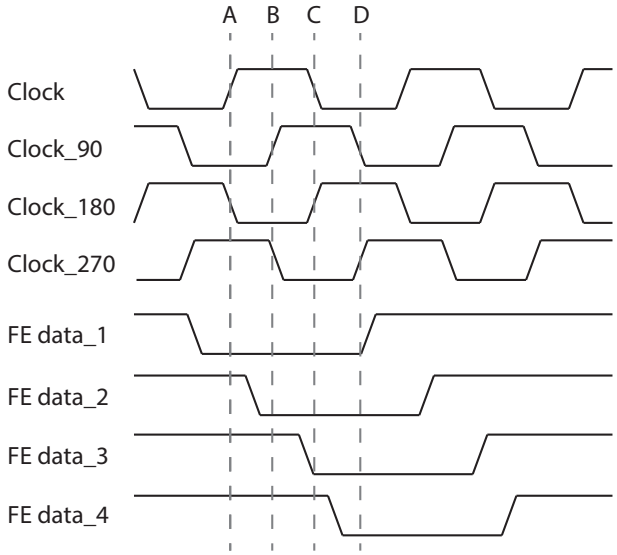

Figure 4.6.: Illustration of the principle used for phase recognition.

4.2.1. Preparation of the Stave Tests

Since the staves are read out through the flex cable connecting to each module from the end of the stave, the data path for each module is quite different in length. Given that electric signals usually travel[6] at around $v_{signal} = 0.7c$ at a rate of $f_{clock} = 160\,\mathrm{MHz}$, one clock period corresponds to a reasonably short distance of

$$l_{period} = \frac{v_{signal}}{f_{clock}} = \frac{0.7 \cdot 2.97 \times 10^8\,\mathrm{m\ /s}}{160 \times 10^6\,\mathrm{s}^{-1}} \approx 1.3\,\mathrm{m}. \tag{4.1}$$

The IBL stave has an active length of 0.65 cm and a data path length on the flex of about 0.8 m. Thus the signals are phase shifted significantly with respect to each other when they reach the FPGA on the HSIO. Thus far, the user interface corrected the shifts by manually selecting one of four equidistant phases. By trial and error, the required phase setting was determined. For large numbers of modules, this procedure is not practical, especially considering that all the tests had to be done for the A and C side separately. Thus the parameters would have to be changed every time the side is switched.

So an automated phase recognition and correction was implemented into the HSIO's firmware based on the four phases that have already been generated and selected manually. Using a *Digital Clock Manager (DCM)*, two clocks with a phase difference of 90° are generated. By inverting the clocks, one gets four equidistantly spaced clock signals covering the entire angle of 360°. To determine, which clock samples the data correctly,

[6] In an ideal conductor the propagation speed of electromagnetic signals is the speed of light, but due to finite resistances and capacitances, the speed of the signal propagation is reduced.

the idle patterns[7] coming from the FEs are analyzed. The incoming data from the FEs is sampled by four clocks, analyzing the pattern. If one of the four clocks has successfully identified the idle pattern, a counter for that specific flag – an indicator that a criterion is met – is set to '1'. This procedure is illustrated in figure 4.6 for a simple example. Depending on the phase of the incoming data, the data stream is sampled differently by the individual clock signals[8]. Suppose that a logical zero is expected for the FE's idle pattern. As illustrated in figure 4.6, with the four phase sampling, there is at least one of the phases sampling the data correctly.

Ensuring that the correct clock has been selected and to avoid that one clock might only be valid temporarily, the clock phase is ultimately selected when $2^8 = 256$ idle patterns have successfully been determined. The setting is not unique because of the finite rise times of the clocks and the signals. The signals are considered *high* when they are at 50% of their nominal *high* level but due to statistical noise, this is not always exactly at the same time, so a statistical measurement with 256 measurements is done.

If due to the phase shift an entire clock cycle of 360° is skipped or if the signal is one clock cycle too late, the difference is coped for, by putting the data through a shift register of variable length, delaying the data if it is one clock cycle early and accelerating the data if it is one clock cycle late.

4.2.2. First large scale test of the RCE system

The RCE system as described above has been tested under several circumstances and is already being used for IBL R&D and test beam read-outs. The first major test with more than a few modules at the same time was performed at the IBL Stave0 electrical tests in Geneva though. The work has been presented at the IBL General Meeting 2012, namely [16], [17]. The stave contains modules, which have been initially tested after production using the *USBPix* test system.

After some basic conductivity tests, ensuring that the communication to all the modules is working, the threshold of the FEs was determined via a threshold scan, as well as digital and analog connection tests. Besides testing the functionality of the modules and the interconnections on the stave, the test is also a good cross check for the results, that the RCE system gives compared to other readout systems.

In figure 4.7 the widely used USBpix system and the RCE are compared by performing threshold scans. As can be seen, the measured thresholds with USBpix and RCE agree

[7] When no commands or data are being transmitted between the FE and the HSIO, the FE transmits idle patterns that can be analyzed.

[8] Each clock only samples the data at the rising edge.

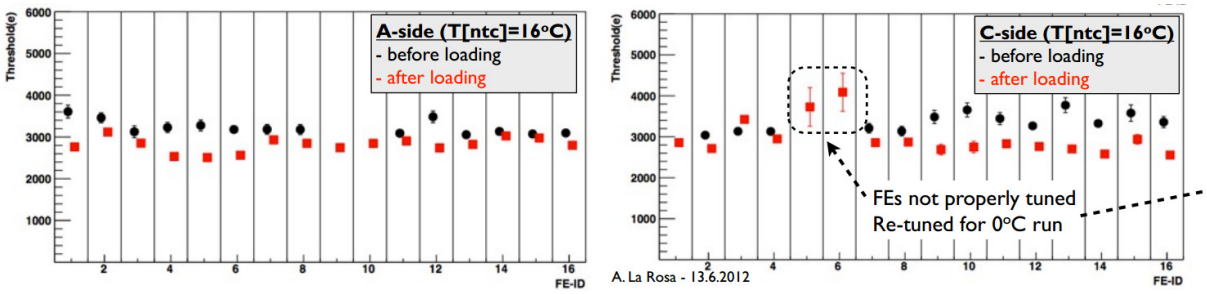

Figure 4.7.: Comparison of the USBpix system (in red, before loading onto the stave) and the RCE system (in black, after loading onto the stave) considering the FEs' thresholds [16].

very well. Some changes might have been caused by the loading process, including gluing the modules onto the mechanical support and wirebonding the module onto the module flex, as well as connecting the module flex to the global flex. Since the actual threshold depends on a lot of quantities, like the chip temperature, humidity, the measured threshold is equal within the expected discrepancies. For the RCE and USBpix comparison, one can say that due to the absence of any systematic deviation between the USBpix and RCE data, both systems yield equally good results.

In context of this thesis, the measurements have been performed at the University of Geneva in cooperation with a research group consisting of people from SLAC, CERN and the University of Geneva [16].

The RCE technology has also been used in several testbeam setups both for cosmic rays and for high energy beam sources e.g. at DESY in Hamburg. For that purpose a specially designed HSIO firmware for a cosmic telescope setup has been developed with less channels than the original setup and external trigger handling.

The HSIO has a special input for trigger signals that has been connected to a trigger card for the testbeam measurements. Thus the HSIO can simulate regular Pixel Detector operation modes during the test beam. Apart from external triggering, there is also a possibility to use self triggering mechanisms with the HSIO.

In this setup, the current RCE infrastructure has yielded very satisfying results as well as for the Stave0 testing. Currently, the RCE system is thus becoming more popular, especially in the testbeam community. Both test setups can be simplified by the planned RCE upgrades that are currently being implemented.

4.3. Upgrades of the RCE Readout

Since the RCEs have been developed over several years, even though it is still in the prototyping stage and not yet used, several upgrade plans have been developed to improve the system before it is deployed to the ATLAS experiment. This chapter will focus on the software upgrades, while an overview over the upcoming hardware upgrades can be found in the appendix A.1.

A lot of improvements in performance can be done by solely software upgrades and also by updating the firmware of the FPGAs in the various components.

One of the major upgrades necessary for full detector operations is the development of the user interface. At the moment, the *User Interface (UI)* for the RCE system is based on the common Pixel readout library called *PixLib*, but is a standalone software with limited functionality, not supporting all of the tuning algorithms that *Pixlib* provides. Thus one of the major software upgrades planned is the integration of the RCE controller into the *Pixlib* framework and to unify the readout chain with the currently used UI called *calibrationConsole*. In this context it was also discussed to write the software package for the RCE system from scratch, thus avoiding the PixLib framework altogether, but no final decision has been made about how to proceed with the software packet.

Concerning the firmware upgrades to the various parts, there are two main aspects to consider. Firstly firmware upgrades are always required when changing the hardware of a system, so certain firmware upgrades for the RCE and the HSIO are intertwined with their corresponding hardware upgrades.

In addition to the required firmware upgrades, one major interest for the future is making the data transmission from the FEs to the HSIO more secure. Thus, a special encoding algorithm based on Reed-Solomon encoding has been developed at CERN to allow for an error correction. The so called *GigaBit Transmission (GBT)* protocol allows for such reconstruction algorithms. In this thesis, the first implementation of the system on the HSIO firmware was performed. Since the protocol is an essential component of the work done in this thesis, the following chapter is devoted to a theoretical approach to the GBT protocol, followed by the implementation on actual FPGAs for detailed analysis.

5. Theoretical Principles of GBT Encoding

In section 2.4.1 it has been described, how particle detection in silicon bulk materials can be performed by detecting free charges that ionizing particles create when traversing the sensor. Since the entire FE electronics is also based on silicon, similar effects can happen in the electronics components of the detector readout, thus corrupting the data. Such occurrences are called *Single Event Upset (SEU)* and describe the active change of a stored bit from logical zero to one or vice versa.

To prevent SEUs from irreparably damaging the data, special redundancy mechanisms have to be implemented when designing radiation tolerant electronics. There are basically two independent approaches to protect the data from being corrupted, each useful in different settings. A hardware based solution would be the so called *majority vote principle*. It is based on the assumption that two transistors contained in different CLBs in an FPGA (or any other IC in an irradiated environment) are very unlikely to experience an SEU at the very same time. Thus, if the user logic is implemented in at least three different logic blocks, two should most likely give the same – the correct – result, while only one may be off.

Of course, the probability of SEUs is very much related to the amount of radiation in the detector, thus the majority vote criterion becomes unsafer for high luminosities. In addition to that, not all of the silicon based electronics can be implemented threefold. Amplifiers and LVDS drivers are also subject to SEUs and cannot be implemented multiple times, because the measured signal is only available once.

The second approach can add extra reliability via line encodings capable of error reconstruction. The data that is being transmitted is made more reliable by systematically adding *redundancy digits* to the data stream. The bits are chosen such that the original data can be recovered, even if a finite amount of bits have been altered by radiation.

In this chapter, the mathematical foundation of error correction algorithms is introduced and the GBT protocol is explained in detail.

5.1. Mathematical Principles of Error Correction

There are two methods of error control that are useful depending on the architecture of the system, as described in [18]. The first procedure is applying an error *detection* code, that is feasible for a two way channel. If the receiver detects the sent data to be faulty, it sends a request to the source to repeat the last data sent. This procedure is called *Automatic Repeat reQuest (ARQ)*. For high energy physics application, this procedure is not practical, even though there are bidirectional links between the on-detector side and the off-detector side. Storing all of the data in the inner detector until they are analyzed by the decoder on the off-detector side on the one hand adds extra electronics to a system that is supposed to be as lightweight as possible. On the other hand, data storage blocks (namely BRAMs) are especially prone to SEUs. The killing argument is the communication of the commands for the modules though. They are transmitted via broadcast[1], such that no individual ARQs can be handled.

The more efficient way to handle the data is to use an error *correction* algorithm, meaning that the decoder does not merely detect the error, but can also pinpoint its exact location and reconstruct it. This procedure is called *Forward Error Correction (FEC)*.

There are two main families of FEC algorithms, convolutional and block codes. Since convolutional decoders again require the use of BRAM, they are not practical in HEP applications. This technique is mainly used by satellite communication, because the encoders are usually very simple. Block codes do not require any BRAM; it divides the message into blocks of k bits. Each of the blocks is then transformed into a code word with length $n \geq k$. Each block is encoded independently, such that the decoder can reproduce the original message by logic blocks, not using any memory.

Let the message rate – the data rate actually containing information – be called r_b and let the code rate (or the code ratio) be $R_c = k/n$, then the actual data rate is given as

$$r = \frac{r_b}{R_c} = \frac{r_b n}{k} \geq r_b,$$

clearly indicating that the capability of error reconstruction requires part of the transmission bandwidth. This is due to the principle of error detection itself. The number of code words corresponding to an actual message must be small compared to the number of elements in the coded set. If the number of errors per message is small, an erroneously received block does generally not belong to the set of valid code words[2]. In the follow-

[1] Broadcast means that all clients receive the same data from the source.

[2] There are 2^k code words for a message of k bits, but 2^n possible words in the coded space.

ing we write the bit sequences as vectors of length k (for the messages) and n (for the code words). The closest valid codeword among the 2^n possible code vectors is selected, whereas distance in coded space is defined by the number of different bits between two code words $\mathbf{x}$ and $\mathbf{y}$, namely the *Hamming distance*

$$d(\mathbf{x}, \mathbf{y}) = |\{i : x_i \neq y_i | i \leq n, i \in \mathbb{N}_0\}|.$$

When taking two vectors that contain actual code words, say $\mathbf{a}$ and $\mathbf{b}$, one can define the minimal distance of the code, namely

$$d_{min} = \min_d d(\mathbf{a}, \mathbf{b}),$$

then one can show that the code can either detect up to $d_{min} - 1$ errors or correct up to $t := \lfloor (d_{min} - 1)/2 \rfloor$ errors found in the transmitted data. It can further be shown that for block codes mapping k bits to n coded bits, the minimum distance is limited[3] to $d_{min} \leq n - k + 1$.

Error Detection Criterion Now that the number of detectable errors has been defined properly, an algorithm has to be presented to detect the occurring errors. In the following an algorithm is presented that allows error detection for *linear* block codes. A code is called linear, if it includes the vector $\mathbf{x}_0 := (0, ..., 0)$ and if the sum of two code words $\mathbf{c} = \mathbf{a} + \mathbf{b}$ is also a code word.

In this case one can define a set of linearly independent code words written into a $k \times n$ matrix $\mathbf{G} := (\mathbf{g}_0, \mathbf{g}_1 ..., \mathbf{g}_{k-1})$ that form a basis for the coded vector space $\mathcal{G}$. Thus every code word $\mathbf{v}$ can be written as a linear combination of the basis:

$$\mathbf{v} = \mathbf{G}\mathbf{u}, \text{whereas } \mathbf{u} := (u_1, u_2 ... u_{k-1}), u_i \in \{0, 1\}$$

Let now $\mathbf{H}$ be another matrix of order $k \times n$, such that $\mathbf{H}\mathbf{G}^T = \mathbf{0}$. This means that all of the elements in $\mathcal{G}$ are orthogonal to the row vectors of $\mathbf{H}$ and vice versa. This leaves us with a criterion for $\mathbf{v}$ being a code word:

The vector $\mathbf{v}$ is in $\mathcal{G}$ if and only if $\mathbf{H}^\mathbf{T}\mathbf{v} = \mathbf{0}$.

[3]This is ostensively clear when considering that for 2^n possible words, only 2^k are used in coded space. The ratio of which is (2^{n-k}). So on average every $n - k$th vector is another code word.

Now consider a received word $\mathbf{r}$ that has changed from the transmitted word $\mathbf{v}$ by $\mathbf{e}$, meaning $\mathbf{r} = \mathbf{v} + \mathbf{e}$. Then the *syndrome* $\mathbf{s_r}$ of $\mathbf{r}$ is defined as

$$\mathbf{s_r} := \mathbf{H^T r}.$$

If $\mathbf{s_r} \neq \mathbf{0}$, then $\mathbf{r}$ was not a code word and an error has occurred during transmission. Since we assumed linearity for the code, the matrix $\mathbf{H^T}$ is dependent on the code and the syndrome does not depend on the transmitted vector $\mathbf{v}$, because $\mathbf{H^T v} = \mathbf{0}$ by definition, thus $\mathbf{s_r} = \mathbf{H^T e}$.

As a final comment note the following property: If the error vector $\mathbf{e}$ is again a code word, then the error is undetectable. Thus it is very important to choose a large enough vector space for the code words compared to the message vector space to increase the minimal distance d_{min} and thus minimize the probability of an error being a code word. To further simplify the error detection, the structure of the matrix $\mathbf{H}$ is analyzed. If the code is designed such that the message blocks are left unchanged and merely additional code bits are appended to the data, then the matrix $\mathbf{H}$ breaks down to $\mathbf{H} = \mathbf{P I_k}$, where $\mathbf{P}$ is a $k \times (n-k)$ matrix and $\mathbf{I_k}$ is the unit matrix of dimension k. Thus, the number of calculations needed to evaluate the syndrome is reduced significantly.

Having found that error detection algorithms always require mapping a message to a code word in a larger vector space and given a means to detect an error by multiplying the received word with a constant matrix, in the next chapter, the *Reed-Solomon (RS)* encoding is introduced. This encoding type is used in the GBT transmission protocol and is thus explained in detail.

5.1.1. Reed-Solomon Encoding

The Reed-Solomon code was developed by *Irving Reed* and *Gustave Solomon* in 1960 based on finite field symbols using *Galois theory*. The family of coding is part of the *Bose, Chaudhuri and Hocquenghem (BCH)* codes that are a special type of linear and cyclic[4] block codes.

In the following we only consider finite linear codes that, even if they do not necessarily have to be binary any more, need to be cyclic. The latter part is important, because it can be proven, that if a code is cyclic, it can be uniquely defined using a *generator polynomial*. In other words, a code is cyclic, if the code space $\mathcal{G}$ is isomorphic to the extended ring of polynomials $R[n-k]$ of order $n-k$. In the following we will make use of this property in finite fields to derive the RS encoding algorithm.

[4]Codes are called cyclic if every cyclic permutation of a code word is again a code word.

To understand the nature of RS encoding, in the following Galois fields are introduced and it is shown how to extend a Galois field such that the coded space is much larger than the message space.

Galois Field Algebra A field K is a set of elements in which addition and multiplication are defined, both satisfying the commutative, associative and distributive laws. In addition to that, a neutral element must exist for multiplication $1 \in K \setminus \{0\} : a \cdot 1 = a$ and for addition $0 \in K : a + 0 = a$. Also an inverse element for both operations must exist, such that $a \cdot a^{-1} = 1$ and $a + (-a) = 0$.

If the field K has a finite amount of elements, it is called *Galois Field* $GF(q)$, where $q \in \mathbb{N}$ specifies the number of elements of the field. Probably the most famous Galois field is $GF(2) = \mathbb{F}_2$ or the residue class field with two elements. It contains the set of elements $\{0, 1\}$, where the following relations hold:

$$0 + 0 = 0 \qquad 0 + 1 = 1 \qquad 1 + 1 = 0 \tag{5.1}$$

$$0 \cdot 0 = 0 \qquad 0 \cdot 1 = 0 \qquad 1 \cdot 1 = 1, \tag{5.2}$$

as well as the above mentioned attributes of commutativity, associativity and distributivity. As a matter of fact, the Galois field $\mathbb{F}_2$ is the most commonly used field in digital data processing, because it describes binary encoding very well and has some very special properties. Applying linear algebra, one can show that for the finite field $\mathbb{F}_p$, where p is a prime number, there exists an *extension* $\mathbb{F}_{p^m}$ such that $\mathbb{F}_p \subseteq \mathbb{F}_{p^m}$ and $\mathbb{F}_{p^m}$ is again a linear finite field.

This is very helpful for finding a proper code word. Since 2 is a prime number, one can thus easily find an extension of the field $\mathbb{F}_2$ for the code words. Because $\mathbb{F}_{p^m}$ such that $\mathbb{F}_{p^{m'}} \subseteq \mathbb{F}_{p^m}$, whereas $m' < m$, the code word can contain the original message plus an appendix to simplify the syndrome determination as mentioned earlier.

With this idea of field extensions in mind, we can now define the RS codes and present an algorithm for encoding and decoding the information.

RS Code Construction To construct an RS code, let t be the number of correctable errors. Now choose a Galois field $GF(q)$ with the corresponding generator polynomial $\mathbf{g}(X)$. Let α be a root of $\mathbf{g}(X)$, such that with $\alpha, \alpha^2, ..., \alpha^{2t}$ a number of $2t$ powers of α exists, factoring the generator polynomial to

$$\mathbf{g}(X) = (X - \alpha)(X - \alpha^2)...(X - \alpha^{2t}) = g_0 + g_1 X + g_2 X^2 + ... + g_{2t-1} X^{2t-1} + X^{2t}.$$

There are $2t$ elements of $g_i \in GF(q = 2t)$ corresponding to the generator polynomial of degree $2t = n-k$, because $n-k$ bits are appended to the message of size k resulting in an n bit long code word[5]. The definition of the Galois field $GF(q)$ gives $n = q-1$ independent code words, thus we have a possible message length $k = q - 1 - 2t$. It can further be proven that the minimal distance between two code words is given by $d_{min} = 2t+1$. Thus the code rate is given by

$$R_{RS} = \frac{k}{n} = \frac{q-1-2t}{q-1} = \frac{2^m - 1 - 2t}{2^m - 1}, \tag{5.3}$$

where the last step was made assuming that we have a binary data stream, so $GF(q) = \mathbb{F}_{2^m}$ is an extension of the field with two elements $\mathbb{F}_2$. Thus each message word is divided into symbols of length m forming a code word of length n. Every symbol $x \in GF(2^m)$ is thus m bits in length.

Finally note that even though this approach suggests fixed block lengths of $n = 2^m - 1$ due to the binary nature of the transmitted data, the designer is not required to maintain the natural size of the coded bits. It is completely legitimate to set l digits to zero *before* encoding the data stream. This has an impact on the code rate leaving

$$R_{RS,short} = \frac{q-1-2t-l}{q-1-l}$$

as the ratio between transmitted data and transmitted information.

After the basic idea of RS codes has now been presented, one of the most interesting parts is how this process is implemented. As has been said in the beginning of the elaboration, the encoder and decoder require only logic blocks; no physical memory is used. In the following paragraphs, the RS encoding procedure is presented.

Linear Feedback Shift Registers Shift registers are a commonly used structure in electronics used for buffering data. By a series of flip-flops[6], the data is shifted through m stages of the flip-flop, so after m clock cycles, the data exits the shift register. By manipulating the connections between the flip-flops, one can change the data while it is in the shift register, allowing for mathematical operations to take place.

To implement generator polynomials of finite fields into actual hardware, *Linear Feedback Shift Registers (LFSR)* are a very convenient tool. Because the information is binary,

[5]Formally the degree of the polynomial comes from the fact that the generator matrix $\mathbf{H} = \mathbf{PI_k}$ is a $k \times n$ matrix and splits into a $k \times k$ unity matrix and a $k \times (n-k)$ remainder.

[6]The complete theory behind shift registers and flip-flops are described in detail in [19] and are not fully elaborated in this thesis.

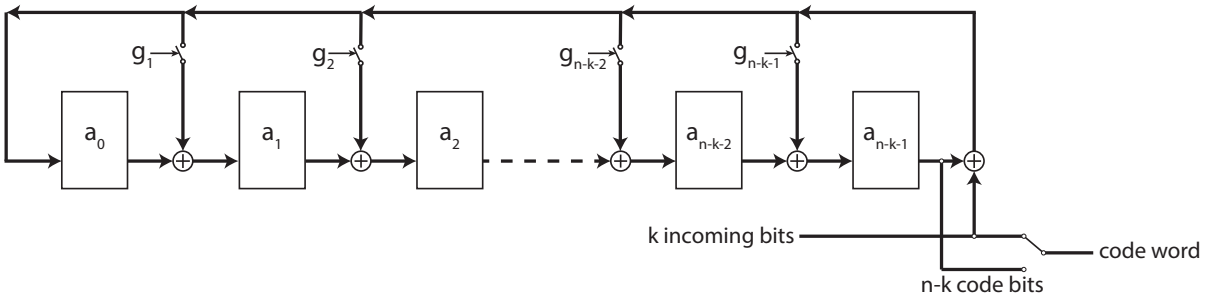

Figure 5.1.: Representation of the polynomial $g(X) = 1 + g_1X + g_2X^2 + ... + g_{n-k-1}X^{n-k-1} + X^{n-k}$ as an LFSR.

some very important features hold, helping the LFSR to represent polynomials in $\mathbb{F}_2$. In figure 5.1 an LFSR of length $(n-k)$ is shown representing the generator polynomial of $\mathbb{F}_{2^m}$ given by

$$g(X) = 1 + g_1X + g_2X^2 + ... + g_{n-k-1}X^{n-k-1} + X^{n-k}.$$

Looking a bit more closely at $\mathbb{F}_2$, one can see from equation 5.1 that addition of two elements $x, y \in \mathbb{F}_2$ is equal to the truth table of the *xor (exclusive or)* relation, represented by the symbol $\oplus$ in figure 5.1. The multiplication is defined in equation 5.2 and represents the truth table of the logical operation *and*. Moreover, when looking at the polynomial's coefficients $g_i \in \{0, 1\}$, the addition can be translated into a switch, telling the LFSR whether the coefficient g_i is zero (not connected) or one (connected). This way, each connection between the flip-flops represent a mathematical $\oplus$ operation and each flip flop represents one monome of the polynomial. This is only true for the binary case though, because in binary, with the multiplication defined as in equation 5.2, especially the relation $a^2 = a$ holds for $a \in GF(2)$.

One might wonder why the monome of degree zero does not have a switch g_0. This is due to the dimension of $\mathbb{F}_{2^m}$. If the constant bit was zero, the polynomial could be factorized into two lower order polynomials making the actual dimension of the field $2^m - 1$, thus by definition of the generator polynomial for $\mathbb{F}_{2^m}$, $g_0 = 1$.

For the extraction of the actual code word, there are two stages represented by the settings of the two way switch in the lower right. While the k message words are being filled into the LFSR, they are also written to the output line labeled with *code word*. After k clock cycles, the switch is changed into its lower position mapping the $(n-k)$ parity check bits to the output port. That way, a coded block of length n is generated.

Before getting to the actual GBT protocol used, another easy to implement, but very powerful tool in data encoding is introduced, the so called *interleaving* process.

Interleaving describes a technique that permutes the transmitted and decoded data in such a way, that an error burst of length n_e can be recovered, even if $n_e > t$, where t was the number of reconstructible errors per block. This is achieved by separating the errors into different blocks. Consider a data stream

$$a_1a_2a_3a_4b_1b_2b_3b_4c_1c_2c_3c_4d_1d_2d_3d_4e_1e_2e_3e_4,$$

where $a_i, b_i, c_i, d_i, e_i, f_i, g_i \in \mathbb{F}_2$. So there are five blocks of four bit each. The process of interleaving rearranges the bits by filling the information into a two dimensional array, writing the information row wise and reading them column wise as illustrated here:

$$a_1a_2a_3a_4b_1... \xrightarrow{\text{Write by row}} \begin{pmatrix} e_1 & e_2 & e_3 & e_4 \\ d_1 & d_2 & d_3 & d_4 \\ c_1 & c_2 & c_3 & c_4 \\ b_1 & b_2 & b_3 & b_4 \\ a_1 & a_2 & a_3 & a_4 \end{pmatrix} \xrightarrow{\text{Read by column}} a_1b_1c_1d_1e_1a_2b_2...$$

This technique is very useful for error bursts. If multiple errors in a row are caused by an error burst, the finite number of reconstructible errors by the RS code might not suffice, but by interleaving data before transmission and de-interleaving the received data, the errors get spread evenly throughout multiple blocks. Thus the amount of errors per block is diluted and can still be reconstructed. The cost for this technique is an increase in latency, because reading and writing of the matrix take a certain amount of clock cycles.

5.2. The GBT Protocol

In this section, the different steps of the encoding and decoding procedure of the GBT protocol are explained. They are illustrated with testbench simulations from the FPGA code written for this thesis. The code is based on the *Gigabit Link Interface Board (GLIB)* project's code for the Virtex 6 architecture, which will be introduced in this section. The code can be found in [20] and a starter kit firmware found in [21].

5.2.1. Encoding

For the simulation and for the initial tests on actual hardware, a pattern generator was used to generate signals that can easily be validated by looking at the received bits via an oscilloscope. In this case, a counter is generated, serving as a data packet of length k.

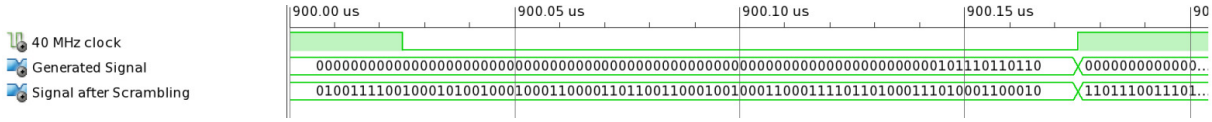

Figure 5.2.: Scrambling procedure as a behavioral model simulation generated with Xilinx iSim[7].

The generated data is going through various stages of transformation to make the GBT protocol very reliable.

Scrambling Much like the BPM encoding described in section 2.5.2, the GBT protocol is designed to generate a duty cycle of about 50% to ensure stable operation and reliable clock recovery. In this case, the scrambling of the data is done by another LFSR. A special generator polynomial for the LFSR (by choosing the positions of the *xor* operations) is selected. If the polynomial is irreducible, then it can be shown that the bit sequence only repeats after $2^m - 1$ bits, m being the degree of the generator polynomial (which is equal to the length of the shift register). Thus a *Pseudo Random Bit Sequence (PRBS)* is generated. In this case, the fact that it is a deterministic 'pseudo' randomization is very important, because the process has to be reversible. For a shift register, this can be done quite easily, by simply inverting the data flow, meaning the direction of the LFSR is inverted, while all other parameters remain the same.

The so called *taps* – the positions of the *xor* operations – for different lengths of the LFSR can be determined from a table for easier handling, i.e. provided by Xilinx [22].

In figure 5.2 the scrambling process can be seen for an input counter at $c = 2998$ and following. Note that even though the counter only fills eight out of 84 bits with a logical one, the output contains 48 logical zeros and 36 ones, leaving a duty cycle of 75%, which is improved quite a bit from the original 90%.

Reed Solomon Encoding After the data has been randomized, it can now be translated into the Reed Solomon code. In case of the GBT encoding the RS code is a very simple one to keep the decoding logic as small as possible. Thus we choose $t_{RS} = 2$ reconstructible bits per block. This is a compromise between the reduction of usage of logic blocks and the number of detectable errors.

As the RS encoding does not require to be binary any more, we define symbols of length $m = 4$. Because the RS code does not care whether it encodes elements of $\mathbb{F}_2$ or any other

[7]Please note, that the timing units are multiplied by a factor of six to ensure integer numbers for all the generated clocks. If this isn't done, the high rate clocks get out of sync due to rounding effects by the simulation software iSim.

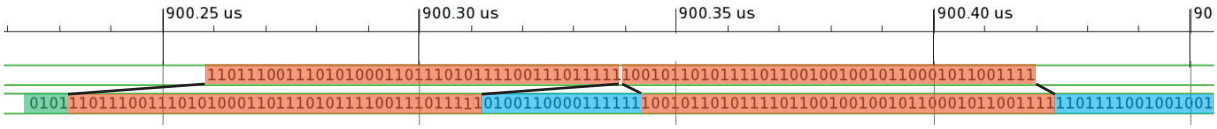

Figure 5.3.: RS encoding procedure. The red data is the message as it was returned by the scrambling routine, the blue bits are the RS appendix and the green data is the header.

finite field, one can replace – without changing anything in the encoding algorithm – one bit with an m bit symbol, which is in the finite field $\mathbb{F}_{2^m}$. By this mathematical trick it is very easy to scale the message word length and the encoded word length to values other than powers of two. The error message block length then modifies from equation 5.3 to $n = (2^m - 1)m$ and $k = (2^m - 1 - 2t_{RS})m$.

To define the packet size, one has thus only to choose one more variable ($t_{RS} = 2$ has already been fixed). As can be seen in figure 5.2 and others, the packet size for user data is 84 bits. The best way to achieve this number is by choosing two independent RS encoders, each with $m = 4$ and thus $k = 44$ and $n = 60$. We get a total message block length of 88 bits. Considering four header bits used for alignment, there are 84 bits left for user data within the message stream. This amounts to a total of 32 redundancy bits. The resulting bit stream can be seen in figure 5.3. Note that the bit sequences are only below each other for graphical reasons. They are generated at different clock ticks due to the latency of the RS encodings.

Interleaving Thus far, it is possible to reconstruct symbols of up to four bits each out of 44 message bits in each half packet of data. By interleaving, a total of 16 bits out of 88 bits can be achieved. To combine the individual two reconstructible errors out of a half frame, the total frame is interleaved into each other, by generating the following interleaving structure.

$$a_0a_1a_2a_3...a_{119} \xrightarrow{\text{Write by row}} \left(\begin{array}{cccc} a_{119} & a_{118} & a_{117} & a_{116} \\ \vdots & \vdots & \vdots & \vdots \\ a_{15} & a_{14} & a_{13} & a_{12} \\ a_7 & a_6 & a_5 & a_4 \\ \hline a_{115} & a_{114} & a_{113} & a_{112} \\ \vdots & \vdots & \vdots & \vdots \\ a_{11} & a_{10} & a_9 & a_8 \\ a_3 & a_2 & a_1 & a_0 \end{array}\right) \xrightarrow{\text{Read by row}} a_0a_1a_2a_3a_8a_9... \tag{5.4}$$

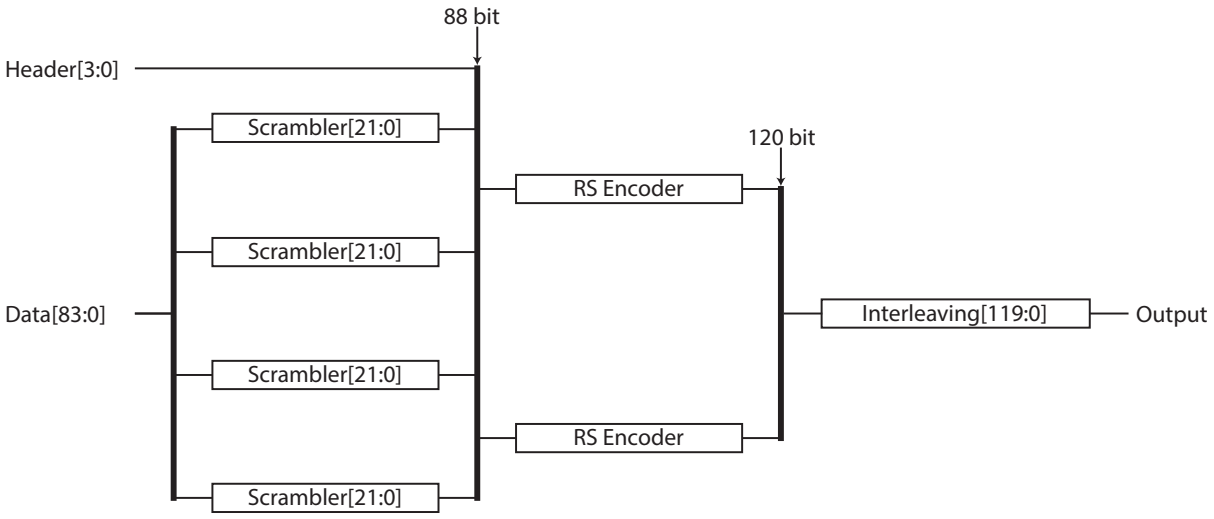

Figure 5.4.: GBT encoding scheme.

For the interleaving the 4-tuples that are used for the RS encodings are not being disrupted, thus the interleaving breaks down into an even mixing of symbols from the first and second RS encoder. Due to the even spread, the total number of reconstructible errors shifts from 8 per RS encoder to 16 in total. This is true even though the used interleaving method is not quite as described in the previous sections, but the permutation is homogeneous, such that the same arguments hold.

The final parameters that have been described in the individual paragraphs are summarized in figure 5.4 completing the elaboration about the encoding process. The output frame indicated in figure 5.4 is split by a multiplexer into three chunks of 40 bit. It is then fed into a GTX – a high speed I/O, as described in the section 2.6 – on the Xilinx Virtex 5 FPGA or similar and serialized. The serial bit stream is transferred via optical or electrical cables through the SFP to the decoder, that is described in the next section. The decoding is a bit more sophisticated because of the alignment routines and will be described in the following section.

5.2.2. Decoding

The decoding process consists, besides the error detection module and the inversion of scrambling and interleaving, of a sophisticated frame alignment process, that precedes all further processing of the received frame.

Frame Alignment Fortunately the GTX and also the GT11 transceivers inside the Xilinx Virtex series support the active clock recovery from the received data. Thus the

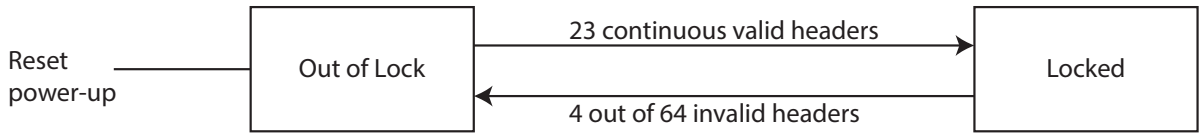

Figure 5.5.: Frame synchronization via finite state machine.

decoder and encoder logic is always synchronized via the transceivers. Still the beginning and end of the individual frames has to be determined.

Here the scrambling procedure of the encoding hampers our attempt to use one specific bit sequence that is unique for the header, because, due to the randomization, it is very unlikely that the header sequence only shows up at the beginning of the frame. Thus a continuous frame recognition has to take place in order to ensure that the frames are aligned throughout the entire decoding process via a finite state machine as shown in figure 5.5.

Clearly, the longer the header sequence is, the less likely it is to mistake a scrambled sequence for the header. This probability has been calculated in [18], showing that 23 successfully recovered headers are necessary to determine the correct location of a 4 bit header with a chance of error of less than 10^{-20}.

Once the locked state is acquired, it should not be abandoned when encountering only one false header. This is due to the fact that the probability for an SEU falling on the header is approximately 10^{-10}, while the probability of mistaking a scrambled word for a header is 10^{-20}. Thus it has been chosen that 4 out of 64 headers have to be incorrect to lose the locked state again.

In addition to the precaution concerning the locking mechanism, the header is chosen such that the Hamming distance d between each valid header is at least three. If the header is recognized such that $d(H_{desired}, H_{recognized}) \leq 1$, then the data is still passed on as valid. The header is still counted as invalid though if $d = 1$. This mechanism ensures that the locked state is achieved and maintained even if the header is corrupted, because the best error reconstruction is not worth much, if the frames cannot be identified and aligned correctly.

Error Reconstruction Error reconstruction algorithms are usually based on the syndrome evaluation mentioned in the last chapter. After the deinterleaving process, which reverts the process displayed in figure 5.4, the data is passed through the error detection logic. All the algorithms follow the same logical steps that are displayed in figure 5.6. If the syndrome $\mathbf{s} = 0$, then the data is correct and the bit sequence can be reconstructed, piecing the two blocks marked in red in figure 5.3 together.

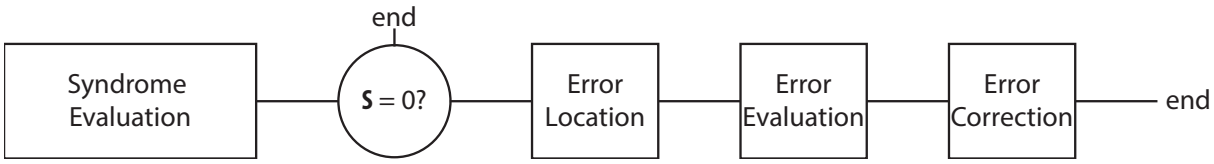

Figure 5.6.: Error reconstruction logic via the syndrome evaluation.

There are many different algorithms for the reconstruction of the errors from Berlekamp [23], Forney [24] and Chien [25], which do not differ much in computing power and logic usage for this scenario with $t = 2$ reconstructible errors. Thus they are not further compared and described in this thesis. It is rather assumed that the errors have been successfully reconstructed and the red bit sequence of figure 5.3 has been found again.

The remaining steps do not require very complex algorithms any more; the data is put through another LFSR to descramble the data stream.

As a final remark, it is noted that due to the frame alignment done, the latency between transmitted and recovered frame is at least 23 frames.

5.3. Case Studies

To illustrate the functionality of the GBT encoding and to show the response to different error scenarios, a few more simulations regarding the error handling are shown in this section.

In figure 5.7 one can see the standard latency, that is due to the frame alignment, even though the cable length in simulations is assumed to be zero. As explained above, a counter is incremented with every clock tick and is transmitted.

Simulating errors, the data is sent through a shift register of length three, which does not affect the latency much, but by a switch, one can invalidate n bits to simulate error bursts. Even though the bits are marked invalid (neither 0 nor 1) they should be overwritten by the error reconstruction and also detected as wrong (since they are in the wrong state). It is merely being invalidated for better visibility in the simulation. In real applications, there is no such state as 'invalid', but the bits caused by radiation are just inverted. In figure 5.8, the resulting bit stream is shown with an error burst of length ten.

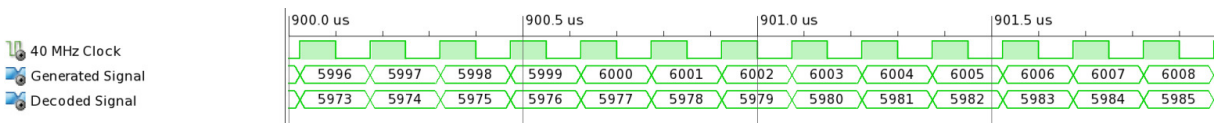

Figure 5.7.: Latency between transmitted and sent bit.

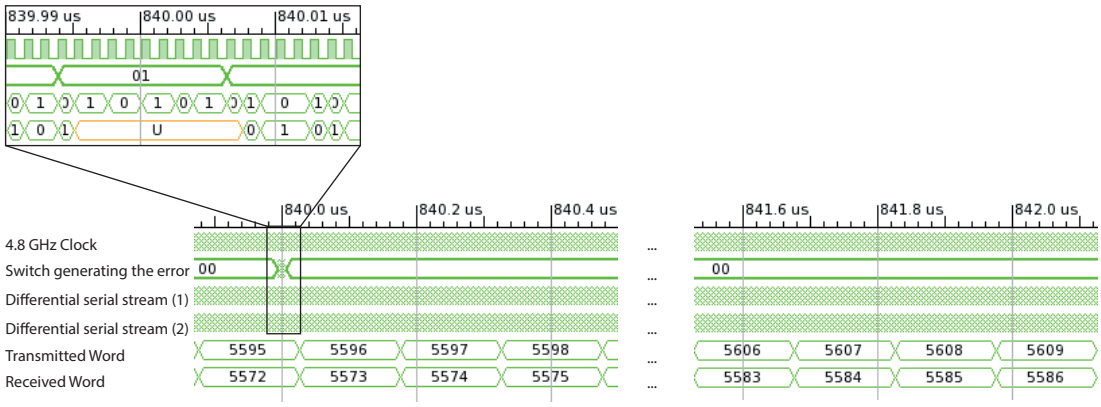

Figure 5.8.: Transmission scenario injecting an error burst of ten undefined bits (left) and the result (right).

The errors are generated in the transmitted frame number 5595, not disrupting the header (which is '0101'). Within the expected latency of 23 frames, the data is reconstructed and, as can be seen in figure 5.8, the frames up to 5595 have been recovered successfully.

On the other hand, in figure 5.9, twenty errors have been produced. They have been chosen in such a way, that the errors are affecting the header. As described earlier, the decoder does not lose the header locked state, but due to misinterpretations, two complete frames have to be sacrificed, even though each individual frame (before and after the header) received less than 16 disrupted bits.

This behavior is due to the fact that the header is marked invalid if the Hamming distance between the desired and the actual header is greater than one. The frame is then regarded as invalid and because there is wrong data being generated afterwards,

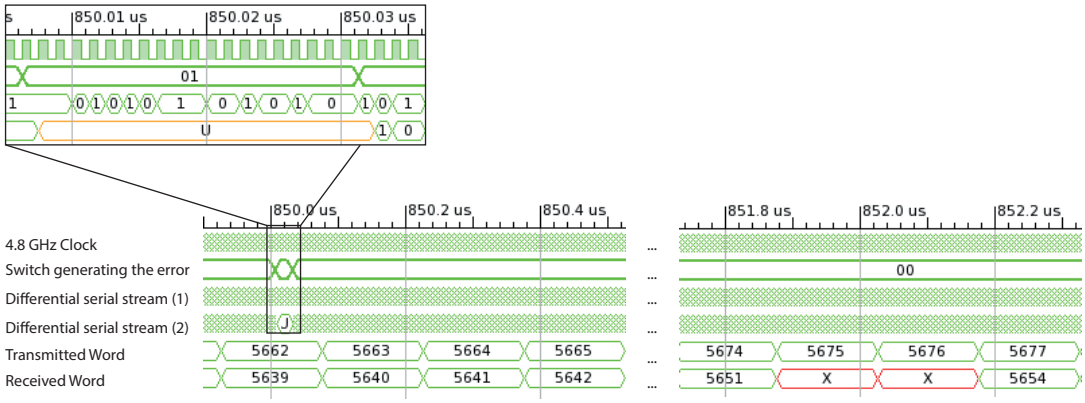

Figure 5.9.: Transmission scenario injecting an error burst of twenty undefined bits (left) and the result (right).

another frame is marked erroneous. Only when another regular header and data have been received, the regular frame reconstruction is valid again.

6. Measurements with the GBT Protocol Between Two Clients

After the thorough evaluation of the GBT protocol, in this chapter the implementation of the decoder and encoder routine for two different devices will be discussed and first test results will be shown. So far, the GBT protocol has only been applied to one single board, using loopbacks for testing. Setting up communication between two different devices, especially using different FPGA models, has been done for the first time in this thesis.

6.1. Desired Setup in Phase II Upgrades

For the Phase II upgrade and the high luminosity, the RCE system is a candidate to replace the current readout system, because there are only few competing systems at the moment, that are being developed like the RCE system is.

On the on-detector side there is – at the moment – still the very powerful and large in size HSIO that will have to be replaced by very small optoboards that only convert signals from optical to electrical signals and vice versa without any additional logic. To convert the raw FE data to the GBT format, a specially designed *Application Specific Integrated Circuit (ASIC)* has been developed. It is a custom made IC that has been designed in a structural size of 130 nm.

6.1.1. The GBT Chip

As can be seen in figure 6.1, the on-detector side, where the data is GBT encoded, consists of three different ASICs. The *GigaBit Transmission Impedance Amplifier (GBTIA)* receives input from the photodiode and amplifies the signal. In a similar way, the *GigaBitLaserDriver (GBLD)* connects the signals to optical transmitters, e.g. the VCSELs.

The heart of the on-detector GBT hardware is the GBTX chip applying the GBT encoding to the data stream and decoding the commands. It contains a high speed

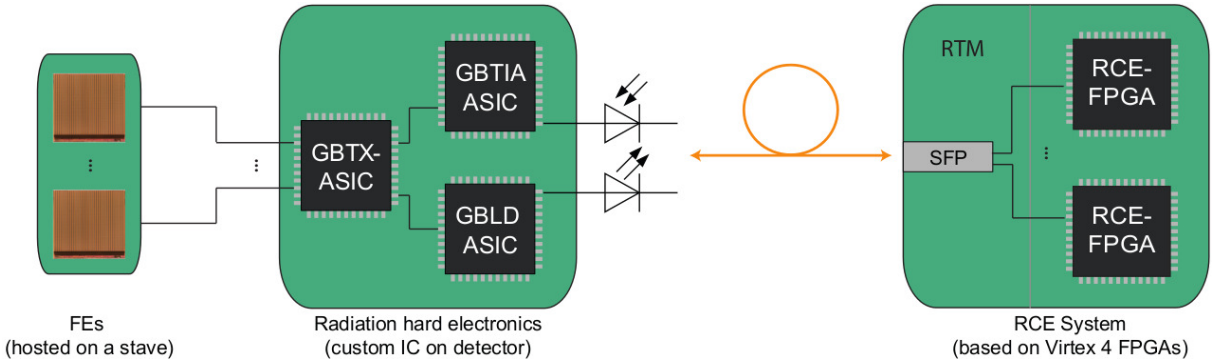

Figure 6.1.: The desired connection setup for high luminosity between the on-detector and off-detector side.

serializer and deserializer and contains the logic blocks necessary for applying the protocol. For more details on the GBTX, including prototyping and the exact design, consult [26]. Studies on the GBLD can be found in [27] and more information on the GBTIA can be found in [28]. The hardware contains additional precautions for radiation hardness, that are described in detail in the mentioned citations.

The data is then transmitted optically to the off-detector side, which can also be seen in figure 6.1. On the off-detector side, off-the-shelf electronics can be used, because the electronics does not have to be radiation hard. Here, regular FPGAs are used. This means that the off-detector side of the GBT communication will be performed by the RCE boards in the ATCA crates.

Since the GBTX chip was not available for this thesis, mainly because the final version has not been completed yet, a different approach was chosen to emulate the on-detector side. Another challenge was that the FPGAs on the RCE boards have a very high usage of CLBs running the current firmware that is responsible for fitting and histogramming procedure. Thus the GBT protocol could not be implemented on the RCE.

As an alternative solution, it was chosen to implement the GBT protocol on the HSIO's Virtex 4 FPGA, where not so many CLBs are used.

For the emulation of the on-detector side, there is a solution developed by the GBT group to enable other research groups to build custom hardware using the GBT protocol, even though the final components have not been finished yet. The *Gigabit Link Interface Board (GLIB)* [29] will be described in the following paragraph.

The GLIB is an emulator board for the on-detector side of the GBT readout, that handles the encoding process on a Virtex 6 FPGA. It is a specially designed board that comes with a multitude of high speed I/Os and several bus system standards like I^2C, Wishbone etc. to handle the data flow. Figure 6.2 shows a photograph of the GLIB. It

Figure 6.2.: The GLIB board [20].

contains four SFP connectors for high speed optical connection of up to 6.25 Gb/s and two *FPGA Mezzanine Card (FMC)* connectors for user I/O. With a suitable adapter card, the FMC connectors can be used to connect several modules to the GLIB. On top of that, the board is compatible with the *Micro Telecommunication Computing Architecture (MicroTCA)*, which is the new generation of the ATCA technology.

For this thesis, the GLIB project was used as an example and a similar project was built using a Xilinx evaluation board containing a Virtex 5 FPGA, the so called ML507. The firmware was rewritten for the Virtex 5 from the GLIB project and starter kit mentioned earlier, because the GLIB was not available for this thesis, due to unexpectedly high demand for the boards.

6.2. GBT on the HSIO

Alternatively, the setup described in figure 6.3 is best suited to the current hardware setup, both shifting the number of logic blocks to a less used FPGA and replacing the GBT chip by an emulator board.

The FEs are hosted on carrier cards that connect the clock, command and data signals via six pins of the *RJ-45* connector. Thus, the connection between the FEs and the Virtex 5 FPGA can be done by *Local Area Network (LAN)* cables. For the connection between the GBT emulator and the HSIO, an electrical connection was chosen, because the HSIO, serving as the powerful optoboard handles all of the optical communication. Still, the SFP connectors were used, because this was the safest way to connect a multi-gigabit

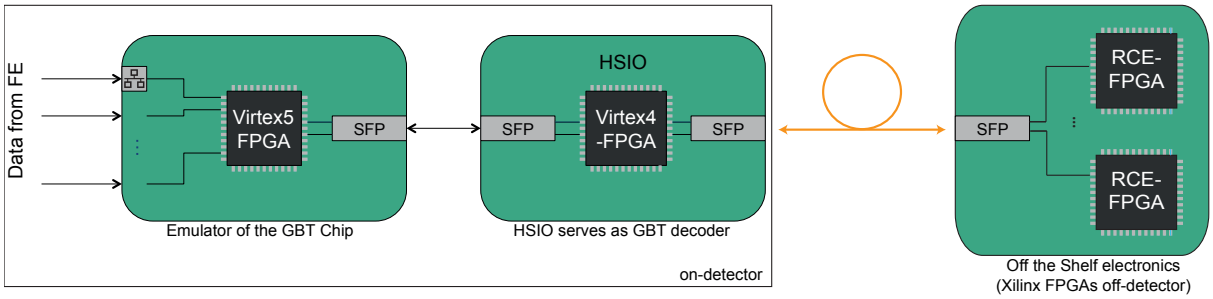

Figure 6.3.: Setup emulating the on-detector side with the ML507 evaluation board and the HSIO.

signal to the HSIO. There are off-the-shelf cables that come with an SFP connector and that are shielded enough to deliver reasonable signal quality over short distances.

This causes problems not only for the HSIO, but also for the current version (Generation 1) of the RCE boards, because they are also based on Virtex 4. In the upcoming version, the Generation 2 RCE, the problem will no longer persist, because the new version of RCE is based on Virtex 5.

The GBT chip transmits the frames of 120 bits at a nominal frequency of 40 MHz, such that the data rate after serialization is 4.8 Gb/s. In figure 6.4 the supported bandwidths of the Virtex 4 are displayed. It can be seen that 4.8 Gb/s, as well as halves and quarters of this data rate are not supported by the device. Both the transmitter and the receiver are affected by this problem, that is noted in the Xilinx Errata Notification [30].

When setting up the high speed transceiver GT11 that is present in the Virtex 4 for such speeds, the *Intellectual Property (IP)* cores predefined by the Xilinx *Firmware Development Kit (FDK)*, the compiler returns an error demanding the user to change to

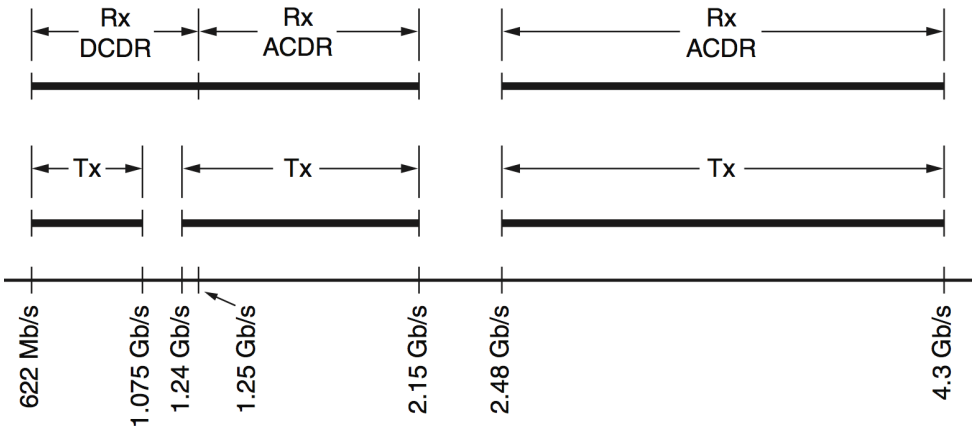

Figure 6.4.: Supported data rates of the Xilinx Virtex 4 FPGA, especially the version used on the HSIO [30].

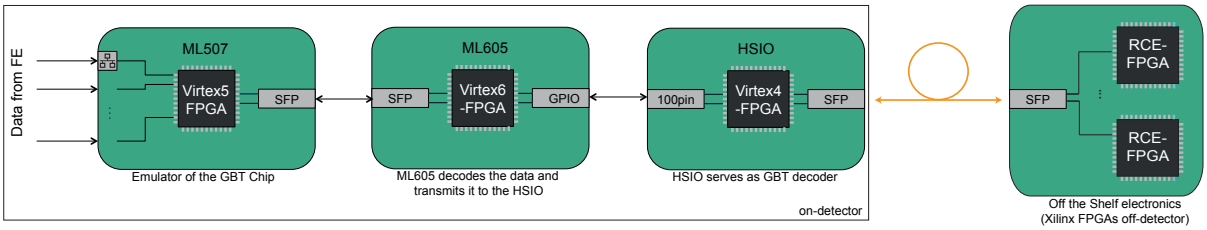

Figure 6.5.: Setup using two evaluation boards. The ML507 handles the encoding of the data and the ML605 handles the decoding.

another data rate. The error message could be avoided by setting up the GT11 manually, but due to the explicit warnings from the developer of the FPGAs, another solution was chosen.

To find a valid workaround for the usage of the Virtex 4, it was chosen to use another available evaluation board from Xilinx, the ML605 hosting a Virtex 6 FPGA to serve as the decoder, which then transfers the data to the HSIO.

6.3. Setup on two Evaluation Boards

The encoder and decoder have been implemented on an ML507 and an ML605 respectively, which presents a great possibility for cross checking, because the GLIB project is also based on Virtex 6. The firmware for the Virtex 6 is thus likely to work very well. The entire chain of devices can be seen in figure 6.5. The connection between the two evaluation boards transports the data at 4.8 Gbps, so the connection is again made through an electrical cable using the SFP port. The connection to the HSIO can then be done by an array of *General Purpose I/O (GPIO)* ports and then adapted to the 100 pin connector on the adapter board, that was described in chapter 4.

In this chapter, the implementation on the ML507 and ML605 are compared, because due to the difference of the boards, the internal data handling is slightly different.

Clock Domains of the GBT Firmware The GBT firmware is divided into several clock domains, where the data is sampled by different reference signals. In figure 6.6 each clock domain has been displayed in a different color. The clock domains for the GBT encoding process can be seen in figure 6.6 marked in blue and red. The scrambling and encoding take place at the nominal frequency of 40 MHz. The data is then formatted for the transceivers, that cannot take 120 bit wide frames as input at the same time. Thus the data is sent through a multiplexer, writing 120 bits at 40 MHz and reading 40 bits at 120 MHz. This clock domain is indicated in red in figure 6.6.

The serializer and deserializer have their own reference clock, indicated in orange, which is also chosen to be 120 MHz. This is a valid reference clock for generating a 4.8 GHz output signal[1].

As has been mentioned before, the GTX supports active clock recovery, meaning that the recovered clock from the deserializer is available for the user logic. The entire decoding process (indicated in yellow and pink in figure 6.6 is thus dependent on the recovered clock. As a first step, the 40 bit wide output of the GTX receiver are combined to recover the 120 bit GBT frames. The recovery is done by a demultiplexer having an 120 MHz input and a 40 MHz output. The 120 bit frames are then decoded and an error detection is performed at a frequency of 40 MHz.

The data can then be transferred to the HSIO, where a phase recognition similar to the one described in section 4.2.1 can be performed.

If everything is set up properly, it is thus only necessary to provide one external reference clock for the entire setup. It is recommended to set the GTX's reference clock from an external quartz to have a high quality reference for the generation of the 4.8 GHz signal.

Using a clock manager as described in section 2.6, the different frequencies can be generated for the various other clock domains. In a similar fashion, one can take the recovered clock from the GTX and generate the necessary frequencies for decoding.

The GBT firmware on the ML605 has basically been taken directly from the GLIB code, except that all the input buses were removed, such that only the decoding, encoding and high speed transceiver part remained. All of the IP cores needed for the multiplexer and the transceiver were already generated for the correct device, the Virtex 6.

Thus the only major issue was to generate a reference clock that could be used to derive all the other clock domains from. Unfortunately, unlike the ML507, the ML605 does not come with an integrated quartz that serves only as a reference clock for the GTX. Instead,

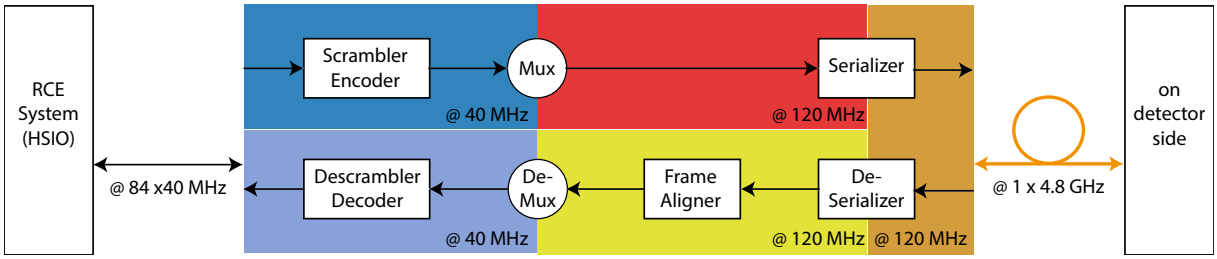

Figure 6.6.: The clock domains of the GBT firmware marked in different colors. On the emulator on the on-detector side, there are the same clock domains, but in mirrored order.

[1]The Xilinx IP Core Generator provides a list of recommended reference clocks for the MGTs.

a connector for an external clock source is provided. On the other hand, it does provide a 200 MHz quartz for general purposes that could be fed into an MMCM to derive a 120 MHz clock. Fortunately the MMCM provides very high signal quality for the derived clocks, such that the user clock can replace the GTX reference clock.

The GBT firmware on the ML507 required a regeneration of the IP cores for a different FPGA family, such that particularly the GTX had to be regenerated. For the list of properties that were used for setting up the GTX, refer to section A.3 of the appendix.

Additionally a reference clock had to be generated on the ML507 as well. In this case, there was a reference clock generator on board, whose frequency could be set by an array of dip switches. This was performed according to the instructions described in [31] to generate a 125 MHz clock that was reduced to 120 MHz by the integrated PLL of the GTX.

6.3.1. Stable Data Transfer via SFP

After all of the connections to outside pins had been specified in the *User Constraints File (UCF)* and the clock settings had been provided correctly, the data stream through the electrical SFP cable could be monitored using a software from Xilinx called Chipscope. It has been designed to monitor the changes of signals inside the FPGA without having to route them to the outside via debug pins. The monitoring can be done using the JTAG connector that is also used for programming the firmware to the FPGA.

The results can be seen in the following figures 6.7 for the data that is being received by the ML507 and figure 6.8 for the data received at the ML605. Both devices transmit the value of counters, as already seen in the simulations to the other device, the number transmitted in each frame increases by one. For the sake of clarity, only the lowest seven bits are displayed.

The displayed transmission with dummy data does not represent the detector operation very well, because the *Least Significant Bit (LSB)* sequences oscillate much more frequently than the *Most Significant Bit (MSB)* sequences. This is because in binary counting, each digit oscillates with half the frequency of its preceding bit, such that the 84th bit changes only after several years of operation. The scrambling does change the order of the bits, but the number of fast and slowly oscillating bits remains equal. Especially because the RS encoders are spread into two parts, it is better to generate the data homogeneously throughout the entire frame. To ensure this, in the following measurement, four counters are being transmitted, each 21 bit wide. This way, there are

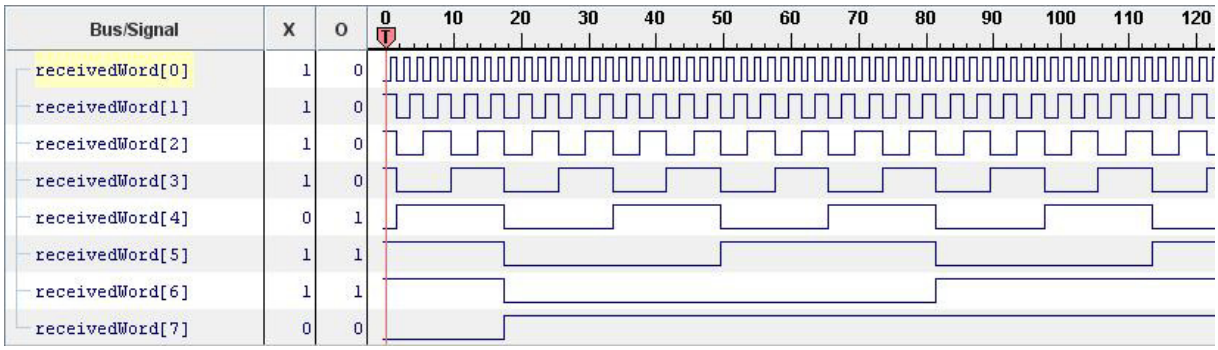

Figure 6.7.: The first seven bits of the data received at the ML507 illustrated with Xilinx Chipscope.

always four bits – located anywhere in the frame due to the scrambling – changing at the same frequency. The 21st bit – the highest bit of the counter – changes about every 60 milliseconds.

Thus the dummy data is closer to the actual data coming from FEs. In the following, a bit error rate has been measured. For this measurement, it is important to transmit a representative bit frame in order to be able to compare the measured rate to the actual application.

6.3.2. Bit Error Rate Measurements

To complete the discussion on the GBT transmission between two clients, the communication between the two clients has been analyzed for stability. Thus an estimate for the bit error rate has been performed by letting the communication run for a long time, transmitting 84 bit sequences with four 21 bit counters. The error reconstruction algorithm

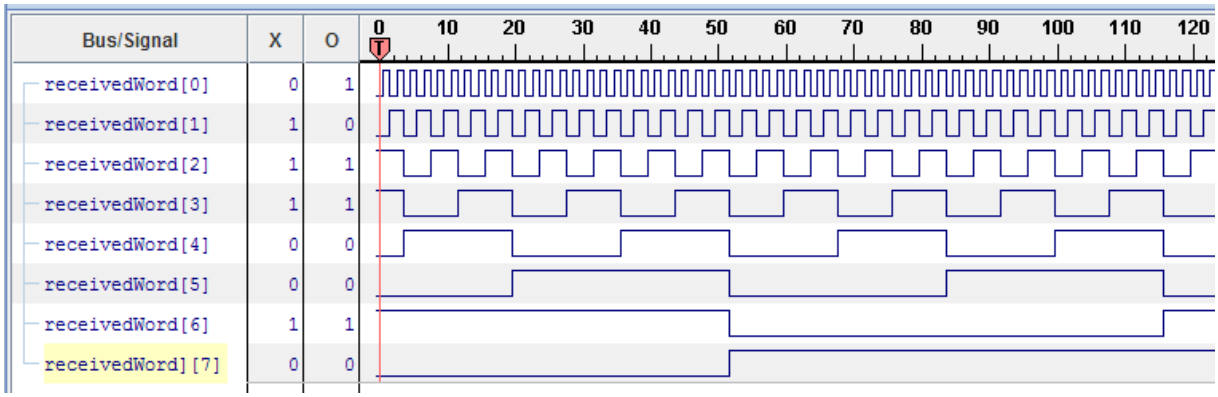

Figure 6.8.: The first seven bits of the data received at the ML605 illustrated with Xilinx Chipscope.

outputs a flag when an error is detected, which was mapped to an LED on the evaluation boards.

The counters have been transmitted for $t_m = 7.25\,\mathrm{h} \pm 5\,\mathrm{min}$. Within this time frame, no errors were detected[2].

The data is being transmitted in frames of $w = 120\,\mathrm{bit\ s}$ at $f = 40\,\mathrm{MHz}$, so the overall *Bit Error Rate (BER)* is given by

$$BER_{total} \leq \frac{1}{w \cdot f \cdot t_m},$$

if no error has been found during the measured time period t_m. This yields a total BER estimate of

$$BER_{total} \leq 1/(113.94\,\mathrm{TB} \pm 1.44\,\mathrm{TB}),$$

considering only the measurement time as an imprecise measurement. Because the frames consist of 84 data bits and 32 redundancy bits, the BER changes by a factor of the inverse code rate $R_c = \frac{k}{n}$ leaving

$$BER_{data} = BER_{total} \cdot R_c^{-1} \leq 1/(79.76\,\mathrm{TB} \pm 1\,\mathrm{TB}).$$

For the current Pixel Readout, the ROD/BOC system, BER measurements have been performed [32] for several parts reaching the ATLAS Pixel limit of $1/(1 \times 10^{14}\,\mathrm{bit}) \approx 1/(91\,\mathrm{TB})$.

The provided measurement of the BER is a very good first indicator, that stable communication between two clients can be achieved even for two devices using different electronics. Further tests on the final versions of the hardware, using the GBTX on the on-detector side are thus likely to give positive results and should be conducted as soon as the individual components are finished.

[2]The error detection was tested beforehand by unplugging the cable during transmission raising the error flag because the data stream suddenly stops.

7. Summary and Outlook

In this thesis it has been shown that for an upgrade towards high luminosities of the LHC, the readout system for the innermost tracker, the ATLAS Pixel Detector has to be upgraded to higher performance. Only by increasing the bandwidths and data rates, as well as higher computing performance, the detector can operate with the same efficiency as in the current setup.

The most likely replacement candidate for the current system, the RCE system has been introduced and it was shown that especially due to the ATCA backplane architecture and the modern processing units, it is a viable replacement option for the current readout system.

In this context, the first large scale test of the RCE system was explained in section 4.2.1, as well as the phase alignment, that was implemented in this thesis.

Part of the development of the RCE system is the integration of the GBT protocol that is based on an interleaved twofold Reed-Solomon encoder. It was mathematically shown that the RS encoding is capable of reconstructing 16 bits of consecutive data by means of 32 bit or redundancy. This was demonstrated in simulations in section 5.3.

Furthermore the desired setup with the radiation-hard on-detector electronics and the off-the-shelf off-detector electronics has been introduced. The communication between two independent clients, one serving as the emulator for the GBTX, the other as a replacement for the HSIO, was established and analyzed for stability. In this context, first BER measurements were made with dummy data. The tests were performed on different FPGA models, namely the Xilinx ML507 and the ML605 evaluation boards. It was shown that promising first results were found in section 6.3.2.

Based on these first steps, further measurements with the actual hardware and new versions of the RCE components, as well as actual communication with FEs through the GBT ASICs for the on-detector side have to be performed, as soon as the components are available in their final versions. In this regard it would also be interesting to prove the radiation hardness of the on-detector components by using ionizing radiation to provoke SEUs and measure the BER again.

For the RCE and the entire readout system development community these results are the first indication that the desired setup can be used to handle the communication between the detector and the off-detector readout electronics and that the GBT protocol is capable of improving the stability of the data transfer in such a way, that the performance of the detector does not deteriorate in an HL-LHC scenario.

It can be concluded that this project is a step towards a virtually failsafe communication between the ATLAS Pixel Detector and the off-detector readout electronics.

A. Appendix

A.1. Hardware Upgrades of the RCE Readout System

The upgrade scenarios take almost all of the hardware components mentioned in chapter 4 into account. All of the information about the upgrades is gathered from [12] and [33] as well as an outlook given at the IBL General Meeting in Febuary 2012 [34] and in June 2012 [17].

Cluster on Board The *Cluster On Board (COB)* is an upgraded version of the RCE board and contains not just two RCE processing units, but four. These are hosted on mezzanine[1] boards that are completely independent of each other. The first prototypes of this concept are being tested; figure A.1 shows the version 4 of the COB, which is still an intermediate model. Version 5 will be the first one to be distributed to other Universities, especially to the ones contributing to the stave testing and production for IBL.

If one of the modules malfunctions during regular operations, it can be replaced easily. Additionally the increased density of processing units saves precious space within the crate. In addition to the new architecture and number of processing units, the *Data Transport Module (DTM)* manages the data distribution on the board. It also uses an RCE on a mezzanine board with a different firmware.

Mezzanine RCE with HSIO host (DPM) Another idea that has been around quite some time and that approaches completion is an HSIO that allows for an RCE to be hosted on it. The new device will then be called *Data Processing Module (DPM)*. Now that the mezzanine RCE boards are at production level stage, it is a small step to allow for the HSIO to host it.

The combination would yield a single board testing interface similar to the USBpix system, but with a greatly increased speed. Due to the USB data transfer, the USBpix

[1] A mezzanine board or sometimes called piggyback board is a modular extension of a main board that is hosted via the mother-daughter principle.

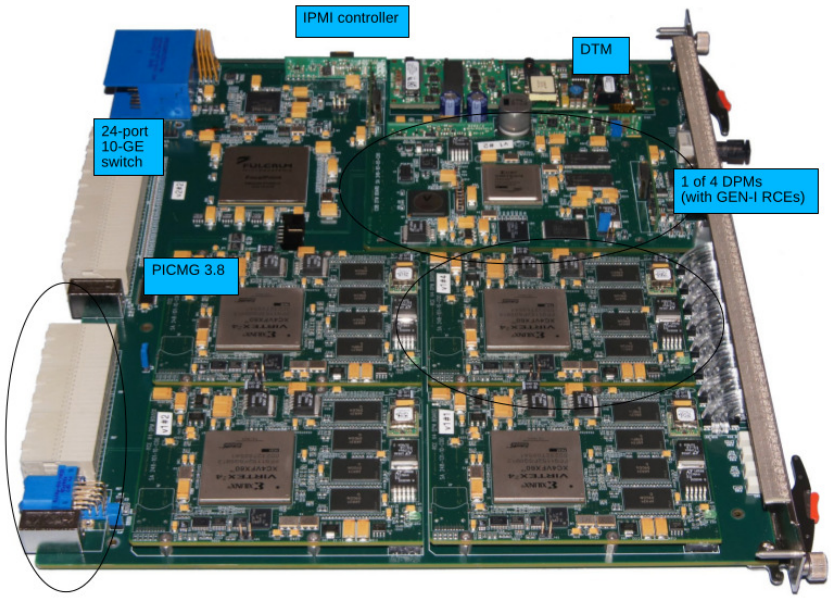

Figure A.1.: Photograph of the RCE upgrade called COB with four mezzanine RCE boards and one DPM.

system is quite slow, the HSIO on the other hand transfers the data at 3.125 Gb/s via optical fibers, increasing the speed of histogramming and further operations greatly.

Especially for testbeam setups as described before, the RCE-HSIO combination is very convenient, since no sophisticated setup is required. That way, less assembly time is needed and and thus costs and loss of measuring time is reduced. Due to the increased speed of the measurements, another temporal advantage is gained.

Gen-II RCE For the mezzanine RCE version to fully function, the RCE design has to be fully independent of ATCA standards, especially the infrastructure the crate provides. To accomplish this first step, the RCE has been redesigned. Apart from that, the RCE Gen-II will have a multitude of component upgrades including the following:

- The RCE core technology, the Xilinx FPGA, will be upgraded from Virtex-4 to Virtex-5, which does better I/O handling due to an integrated crossbar. That way, the Dual Core PowerPC processing unit can be better put into play. The firmware can be ported fairly easily to this new architecture.

- The RCEs will get a significant RAM upgrade. The slow and nowadays very expensive 512 MB RLDRAM-II blocks are being swapped for 4 GB custom DDR-II RAM found in regular PCs. That way, the RAM is both easier to replace when faulty

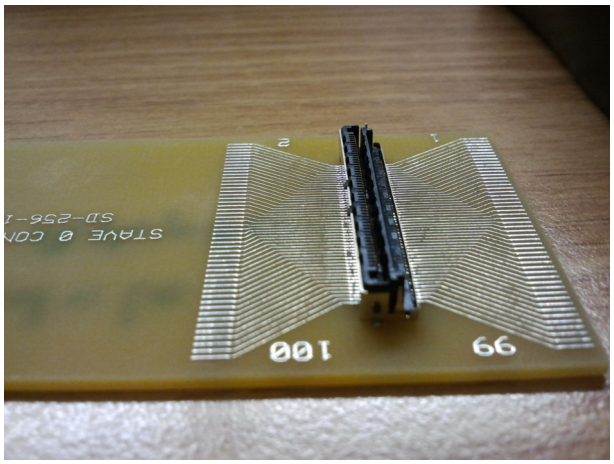

Figure A.2.: The 100 pin connector.

and much less costly. The capacity is increased widely, even though the current applications do not put that much RAM to use.

- A few more ethernet controllers are added to support higher data rates through the ethernet port. This is especially important, if the RCE is running outside of it's ATCA framework to ensure high speed data transfer.

- Another important improvement is the integration of the timing and trigger connection that is used at the LHC, the TTCrx ASIC. This port will be connected to a generic timing and control module similar to the TIM in the ROD/BOC setup. This way, it will not be necessary any more to use one slot in the crate for TIM modules. All the required data for building a pixel event will be on the chip.

- The most significant change is obviously the size. The new RCE modules will no longer require the physical space of an ATCA slot. Both the geometric factors and the electrical standards of the ATCA system will be abandoned to make an independent mezzanine board. In addition, this functionality will allow an RCE to act as a standalone trigger and timing emulator.

A.2. The 100 Pin Connector Cable

The 100 pin connector is used to connect the FE modules to the adapter board of the HSIO. It is hosted on a small board with the soldering pads already connected. Usually SLAC delivers these connectors with the rest of the RCE system. A picture can be seen in figure A.2.

To be able to connect the FEI4B modules, several RJ45 cables have been cut in half and attached to the correct pins. The pinout can be seen in table A.1.

Note that two of the four connected plugs share the clock and command lines. This has to be set in the RCE user interface software. They have the same *inlink*, but different *outlinks*.

The complete pinout of the 100 pin connector as well as a short explanation on how to operate the RCE user interface can be found on the Wiki of the II. Institute of Physics.

A.3. The GTX Core Generation Settings

The GTX is best set up via the CoreGen form Xilinx, because a manual setup is a lot of work. Please note that the CoreGen for the Virtex 4 GT11 does not work for the given frequency. For Virtex 5 and Virtex 6, the Transceiver Wizard can be used to generate the cores. They are a little bit different for Virtex 5 and Virtex 6, but basically have the same parameters.

Before starting to set up the GTX, it is important to find out, which one of the GTX blocks on the evaluation board indeed connects to the desired output. For the ML507 and ML605, this could be looked up in the respective user guide, e.g. under [31]. The wizard always activates GTX Duals, meaning that there are always two GTX on one bank of the FPGA. Because usually only one of the GTX Duals is needed, just *no TX* and *no RX* for the second GTX, which is not connected to the correct output.

It is important that no special predefined protocol like PCIe is used, instead select *start from scratch*, selecting the correct data rate (4.8 Gbps) for the GBT for both transmitter and receiver. The rest of the wizard is straight forward. Make sure to set all of the other additional protocols such as comma detection to disabled and all of the control flags as constant. This is usually the default setting though. For the GBT firmware, ensure that the transmitter and receiver both have individual reset flags.

Signal name	RJ-45 Pin	Plug 1	Plug 2	Plug 3	Plug 4
CLK-p	1	57	57	53	53
CLK-n	2	59	59	55	55
DCI-p	4	58	58	54	54
DCI-n	5	60	60	56	56
DTO-p	7	80	76	72	68
DTO-n	8	78	74	70	66

Table A.1.: Connections to the four connected plugs of the adapter cable in Göttingen

Bibliography

[1] S. F. Novaes, *Standard Model: An Introduction*, arXiv **hep-ph/0001283** (2000)

[2] O. Brüning et al., *LHC Design Report*, CERN (2004)

[3] ATLAS Collaboration, *ATLAS detector and physics performance : Technical Design Report, 2*, CERN (1999)

[4] M. George, *Implementation of an Electrical Read-Out System for Multi-Module Laboratory Tests of the ATLAS Pixel Detector*, Master's thesis, Georg-August Universität Göttingen, II.Physik-UniGö-Dipl-2009/03 (2009)

[5] ATLAS Collaboration, *The ATLAS Experiment*, `https://cds.cern.ch/tools/mediaarchive.py/copyright_notice?recid=1095924&master_path=https://mediastream.cern.ch/MediaArchive/Photo/Masters/2008/0803012/0803012_01.jpg&ln=de&reference=CERN-GE-0803012&tirage=01`

[6] N. Wernes et al., *Pixel Detector Technical Design Report*, volume ATLAS-TDR-11, CERN (1998)

[7] M. Capeans et al., *ATLAS Insertable B-Layer Technical Design Report*, volume CERN-LHCC-2010-013, CERN (2010)

[8] N. Krieger, *Development of the readout for the IBL Upgrade Project of the ATLAS Pixel Detector*, Ph.D. thesis, Georg-August Universität Göttingen, II.Physik-UniGö-Diss-2012/08 (2012)

[9] Xilinx, *Spartan-3 FPGA Family Data Sheet*, `http://www.xilinx.com/support/documentation/data_sheets/ds099.pdf`

[10] F. Giattoni et al., *Physics Potential and experimental challenges of the LHC luminosity upgrade*, arXiv **hep-ph/0204087** (2002)

[11] L. Caminada, *ATLAS FEI4 ASIC*, in *VERTEX 2012*, `http://vertex2012.knu.ac.kr/slide/s4/ATLAS%20FE-I4%20ASIC.pdf` (2012)

[12] R. Bartoldus, *High Bandwidth DAQ R&D for ATLAS Upgrade*, ATLAS Pixel Upgrade Proposals (2011)

[13] K. Tuttle, *PetaCache: Use That Memory*, SLAC Today (2006)

[14] R. Herbst, *Pretty Good Protocol - Design Specification*, Technical report, SLAC, `https://www.slac.stanford.edu/exp/npa/design/pgp_design.pdf` (2006)

[15] D. Ferrere, *Module loading activities report*, in *IBL General Meeting*, `https://indico.cern.ch/getFile.py/access?contribId=9&sessionId=2&resId=0&materialId=slides&confId=174800` (2012)

[16] A. L. Rosa, *Stave 0*, in *IBL General Meeting*, `https://indico.cern.ch/getFile.py/access?contribId=45&sessionId=6&resId=0&materialId=slides&confId=174800` (2012)

[17] M. Kocian, *RCE System Update*, in *IBL General Meeting* (2012)

[18] G. Papotti, *Architectural Studies of a Radiation-Hard Transceiver ASIC in 0.13 um CMOS for Digital Optical Links in High Energy Physics Applications*, Ph.D. thesis, UNIVERSITÀ DEGLI STUDI DI PARMA (2007)

[19] J. Reichardt, *Lehrbuch Digitalelektronik - Eine Einführung in VHDL*, Oldenbourg Verlag (2011)

[20] CERN, *The GLIB Project*, `https://espace.cern.ch/project-GBLIB/public/default.aspx`

[21] CERN, *GBT-FPGA project*, `https://espace.cern.ch/GBT-Project/GBT-FPGA/Lists/Starter%20Kit%20Firmware%20Releases/AllItems.aspx`

[22] Xilinx, *Efficient Shift Registers, LFSR Counters, and Long PseudoRandom Sequence Generators*, xapp052 edition

[23] E. R. Berlekamp, *Bit serial Reed-Solomon encoders*, IEEE Transactions on Information Theory **IT-28**, 869 (1982)

[24] G. D. Forney, *On decoding BCH codes*, IEEE Transactions on Information Theory **IT-11**, 549 (1965)

[25] R. T. Chien, *Cyclic decoding procedure for the BCH codes*, IEEE Transactions on Information Theory **IT-10**, 357 (1964)

[26] K. Wyllie et al., *A Gigabit Transceiver for Data Transmission in Future High Energy Physics Experiments*, Physics Procedia (2012)

[27] G. Mazza et al., *A 5 Gb/s Radiation Tolerant Laser Driver*, Proceedings of the Topical Workshop on Electronics for Particle Physics (2009)

[28] M. Menouni et al., *The GBTIA, a 5 Gbit/s Radiation-Hard Optical Receiver for the SLHC Upgrades*, Proceedings of the Topical Workshop on Electronics for Particle Physics (2012)

[29] P. Vichoudis et al., *The Gigabit Link Interface Board (GLIB) ecosystem*, IOPscience **JINST 8 C03012** (2012)

[30] Xilinx, *Virtex-4 XC4VFX20, XC4VFX40, XC4VFX60, XC4VFX100, and XC4VFX140 Production Errata*, Xilinx Errata Notification **EN070** (2008)

[31] Xilinx, *ML505/ML506/ML507 Evaluation Platform User Guide ML505/ML506/ML507 Evaluation Platform*, Xilinx User Guides **UG347** (2011)

[32] T. Flick, *Studies on the Optical Readout for the ATLAS Pixel Detector*, Ph.D. thesis, Universität Wuppertal (2006)

[33] D. Nelson, *HSIO Development Platform Users Guide*, ATLAS Pixel Upgrade Proposals (2010)

[34] M. Kocian, *RCE for Stave-1/0*, in *IBL General Meeting* (2012)

Acknowledgments

I would like to thank Dr. Martin Kocian for his help in setting up the RCE system in Göttingen and for coaching me both when this project started at SLAC and at the University of Geneva. For the GBT implementation on the HSIO he was the most important contact person.

Furthermore I would like to thank the people of the GBT project, namely Paulo Moreira, Sophie Baron and Paschalis Vichoudis for introducing me to the GLIB project and for helping me, when I got stuck.

Appreciation also goes to the X ray physics group in Göttingen and Johannes Agricola, who kindly provided the ML605 board, so the testing and measuring could be completed in time for the deadline of this thesis.

Dr. Jens Weingarten and Christian Wehrberger kindly commented on this thesis and helped ironing out most of of the imprecise phrases and other mistakes.

Finally there is a big thanks for PD Dr. Jörn Große-Knetter and Prof. Dr. Arnulf Quadt for allowing me to work on this thesis and for making the entire project possible. They enabled me to go to CERN and SLAC to get a basic understanding for the RCE system and FPGA programming. This did not only help the completion of this thesis; rather I gained experience for life.

Thanks to the entire research group of the II. Institute of Physics for the good times, the funny moments and the helping hand, whenever it was needed.

Erklärung nach §18(8) der Prüfungsordnung für den Bachelor-Studiengang Physik und den Master-Studiengang Physik an der Universität Göttingen:

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe.
Darüberhinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, im Rahmen einer nichtbestandenen Prüfung an dieser oder einer anderen Hochschule eingereicht wurde.

Göttingen, den June 9, 2013

(Frank Schuster)