Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Fast Optical Readout of the Mu3e Pixel Detector

Master Thesis

Simon Corrodi

March, 2014

Advisors:
Dr. Niklaus Berger
Department of Physics and Astronomy, Heidelberg University
Prof. Dr. Günther Dissertori
Department of Physics, ETH Zurich

Zusammenfassung

Das Mu3e Experiment sucht nach dem Lepton-Flavour-verletzenden Zerfall $\mu^+ \rightarrow e^+e^-e^+$ mit einer Sensitivität von besser als 1 in 10^{16} μ-Zerfällen. Um diese Sensitivität zu erreichen, sind über eine Messzeit von ca. 1 Jahr 2 Milliarden Zerfälle pro Sekunde notwendig. Die Trajektorien der Zerfallsprodukte werden von Pixel-, szintillierenden Faser- und Kacheldetektoren gemessen und in Echtzeit in einer auf Grafikprozessoren basierenden Filterfarm komplett rekonstruiert. Der für die schnelle Auslese der Daten im Detektor vorhandene Platz ist stark limitiert.

Das auf Kapton Flexprints, optischen Fasern und FPGAs basierende Auslesesystem verarbeitet 1 Tbit/s auf engstem Raum.

In der vorliegenden Arbeit wurden optische Verbindungen in Kombination mit FPGA Baugruppen auf ihre Bandbreiten bei möglichst kleinen Fehlerraten getestet.

Bidirektionale Übertragungen mit 8 simultan genutzten Kanälen auf einer FPGA Tochterkarte mit SFP Steckern sind mit Fehlerraten unter $< 10^{-16}$ (95 % C.L.) bei 6.4 Gbit/s realisiert worden. Optische Verbindungen im QSFP Standard können mit einer Fehlerrate von $(3.29 \pm 1.04) \cdot 10^{-16}$ bei 11.3 Gbit/s betrieben werden. Die optischen Datenübertragungen erfüllen die Anforderungen, die an das Mu3e Auslesesystems gestellt werden.

Zusätzlich wurde gezeigt, dass Kapton Flexprints grundsätzlich mit einem neu angeschafften Laserplotter an der Universität Heidelberg produziert werden könnten.

Abstract

The Mu3e experiment searches for the lepton flavor violating decay $\mu^+ \rightarrow e^+e^-e^+$ with a sensitivity better than 1 in 10^{16} μ-decays. To reach this sensitivity in a measurement period of approximately 1 year, 2 billion decays per seconds are required. The decay products' trajectories are measured by pixel, scintillating fibers and tile detectors and fully reconstructed online by a filter farm based on graphics processing units. The available space inside the detector for the fast data readout is strongly limited.

The readout system based on Kapton flexprints, optical fibers and FPGAs processes 1 Tbit/s in a very compact volume.

In the presented work, optical links in combination with FPGA boards are tested with respect to their bandwidths at minimal bit error rates.

Eight parallel duplex 6.4 Gbit/s links on one FPGA daughter board equipped with SFP plugs have been realized with bit error rates below $< 10^{-16}$ (95 % C.L.). Optical links in QSFP standard have been operated at 11.3 Gbit/s with bit error rates of $(3.29 \pm 1.04) \cdot 10^{-16}$. The optical data transmissions fulfill the requirements for the Mu3e data acquisition system.

In addition, it has been proven that Kapton flexprints can be manufactured in principle with a new purchased laser cutting system at the University of Heidelberg.

Contents

Part I

Introduction

Chapter 1

Introduction

The Standard Model (SM) of particle physics describes the constituents of matter as well as their interactions. It is described in more detail in a first section, followed by the observation of lepton flavor violation through neutrino oscillations and its consequences for the theory. These motivate the search of lepton flavor violating processes in charged leptons as described in another section.

The Mu3e experiment looks for the charged lepton flavour violating decay $\mu \to eee$. In a second chapter, the design of this experiment is discussed. Particularly, the experiment's readout chain, the main scope of this thesis, is presented in detail.

1.1 The Standard Model

The Standard Model (SM) of particle physics is a quantum field theory which describes the fundamental constituents and interactions of matter. As shown in figure 1.1, matter consists of six quarks and six leptons, and their anti-particles, which are arranged in three generations. The interactions between quarks and leptons are mediated by four types of gauge bosons.

The first generation consists of up (u) and down (d) quarks with electrical charges of $+2/3$ and $-1/3$ respectively , the negatively charged electron (e^-) and the neutral neutrino (ν_e). The lepton family number L_e is characteristic for the leptons of this family. The second and third generation consist both in each case of two quarks with the same charge as the first generation - these are charm (c) and strange (s) in the second and top (t) and bottom (b) in the third generation. Their associated leptons, again with the same electrical charge as the ones in the first generation, are muons (μ^-) and the neutrino (ν_μ), tau (τ) and the neutrino (ν_τ). Their characteristic lepton family numbers are L_μ and L_τ. In the SM neutrinos are massless and lepton flavour is a conserved quantity.

Quarks and leptons are spin 1/2 particles whose interaction is mediated by spin 1 particles, the gauge bosons. The eight gluons mediate the strong interaction, photons (γ) the electromagnetic interaction and Z, W^+ and W^- bosons the weak force.

The model has demonstrated huge and continued successes, particularly the recent discovery of the long predicted higgs boson in 2012 [1] at the LHC. Gravitation is not included in the standard model [2, 3].

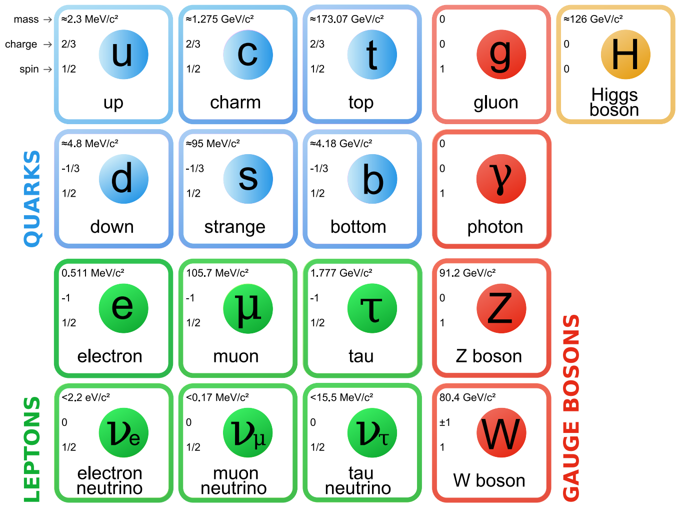


Figure 1.1: Standard Model Particles [4, modified].

Lepton Flavour Violation

Different experiments have observed mixing of neutrino flavours. Super-Kamiokande and others have observed [5] mixing in atmospheric and solar neutrinos, SNO [6] in solar neutrinos and KamLAND [7] in reactor neutrinos. The mixing angles in the Pontecorvo Maki Nakagawa Sakata (PMNS) matrix, the matrix which describes the neutrino mixing, are close to maximal [8].

Neutrino oscillation is only possible if neutrinos have a non-vanishing mass, which is not foreseen in the SM. An extension of the Minimal Standard Model by heavy right-handed neutrinos, called νSM, is required to incorporate neutrino masses consistent with oscillation experiments. The reason why the neutrino masses are significantly smaller than other particle's masses remains a puzzle [9].

Even though the PMNS matrix appears also in charged lepton currents, lepton flavour violation has never been observed in charged leptons. These flavour-changing neutral currents are suppressed by a mechanism described by Glashow, Iliopoulos and Maiani in 1970 [10].

Also, the νSM is not able to explain all observations such as dark matter, the baryon asymmetry of the universe or motivate the observation of exactly three generations of particles. This motivates theories beyond the standard model (BSM). Several of these, like supersymmetry or little Higgs models among others, predict large lepton flavour violation in the charged lepton sector.

Due to the fact that flavour violating processes in the charged leptonic sector are highly suppressed in νSM and predicted in many BSM theories, these processes are very interesting to search for BSM physics.

1.1.1 Lepton Flavour Violating (Muon) Decays

The lepton flavour violating (LFV) muon decay $\mu^+ \to e^+e^-e^+$ can be realised in extensions of Standard Models which include lepton mixing. Figure 1.2 shows this Feynman diagram with neutrino oscillation. The W^+ mass of 80.4 GeV$/c^2$ is much higher than the neutrino masses of $\mathcal{O}(0.01eV)$, hence the process is suppressed by a factor of $\sim \left(\frac{\Delta m^2}{m^2_{W+}}\right)^2$ which is of the order $\ll 10^{-50}$.

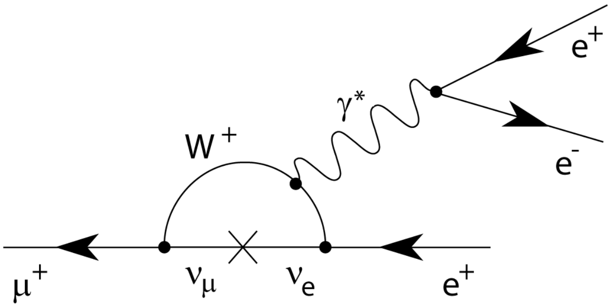
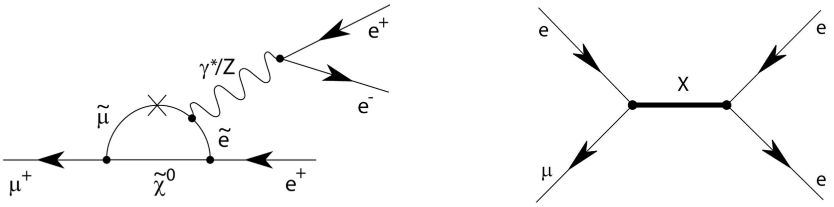


Figure 1.2: Feynman diagram for the $\mu \to eee$ process via neutrino mixing [11, Fig. 2.1].



(a) involving supersymmetric particles

(b) at tree level

Figure 1.3: Diagram for lepton flavour violation [11, Fig. 2.2,2.3].

BSM theories can introduce new possible diagrams, particularly loop contributions and new tree couplings. Figure 1.3a shows a diagram with a γ/Z-penguin with a supersymmetric particle in the loop, where LFV is introduced by slepton mixing. Figure 1.3b shows a diagram, where lepton flavour violation occurs on tree level via new heavy particles, coupling to both electrons and muons [11].

As described before, the process $\mu \to eee$ is sensitive to new physics and suppressed in the νSM. In contrast to $\mu \to e\gamma$, it is also sensitive to tree level processes.

Experimental Situation

The current upper limit of B$(\mu \to eee) < 10^{-12}$ at a 95% C.L. was set in 1988 by the SINDRUM experiment at PSI [13] .

Other decays such as $\mu \to e\gamma$ measured by MEG in 2009 to 2011 with B$(\mu \to e\gamma)< 5.7 \cdot 10^{-13}$ (90% C.L.) [14] and conversions in presence of a nucleus $\mu N \to eN$ as measured by SINDRUM II with B$(\mu \to e\, conversion\, in\, ^{27}Al) < 7 \cdot 10^{-13}$ are also sensitive to charged LFV.

For loop correction diagrams, MEG's sensitivity is two orders of magnitude higher due to the additional photon electron-positron vertex in $\mu \to eee$. But the experiment is not sensitive at all for tree level processes. Conversion processes are sensitive to both described types of diagrams and their sensitivity scales $\sim Z^2$ [15, Figure 2-5,2-6]. Figure 1.4 gives an overview over previously performed measurements in the search for LFV in charged leptons.

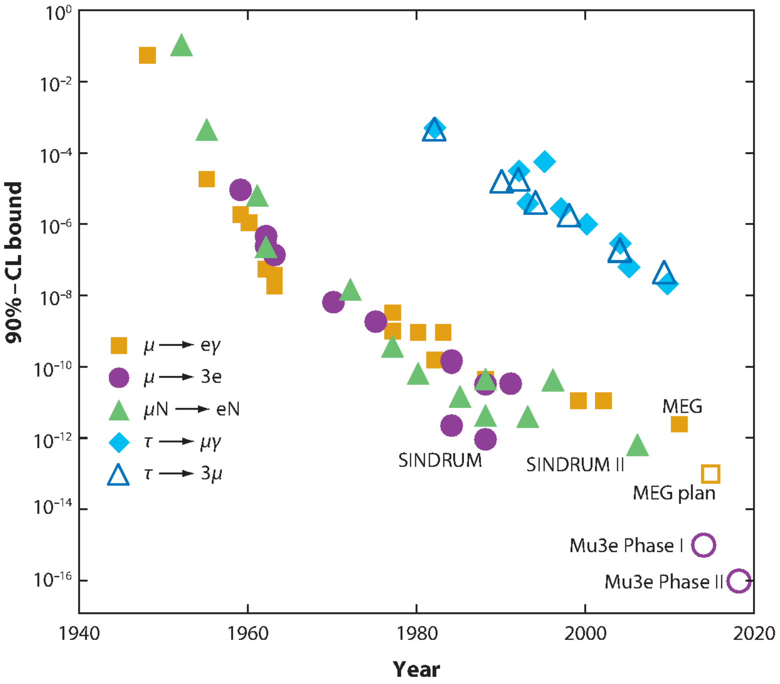


Figure 1.4: History of LFV measurements. Modified [12].

Backgrounds for a $\mu \to eee$ search

On one hand, background due to internal conversion $\mu \to eee\nu\nu$ with a branching ratio of $3.4 \cdot 10^{-5}$ [16], and on the other hand accidental background is present. Accidental background consists of a combination of events which produce one positron and an overlying electron-positron pair.

The internal conversion can only be resolved by a very good energy resolution, which is able to resolve the missing energy due to the additional neutrinos. Michel decays $\mu^+ \to e^+\nu\nu$, radiative muon decays $\mu^+ \to e^+\gamma\nu\nu$ with a branching ratio of $1.4 \cdot 10^{-2}$ and Bhabha scattered electrons contribute to accidentals. They are suppressed through good vertex fits and time resolution.

The pion decay $\pi \to eee\nu$ with a branching fraction of $3.2 \cdot 10^{-9}$ [16] is indistinguishable if the right momentum is met. A low pion contamination in the beam, small branching ratio and small probability to meet the right momentum suppresses this background source strongly.

1.2 The Mu3e Experiment

The Mu3e experiment searches for the lepton flavour violating decay $\mu^+ \to e^+e^-e^+$. It aims for an ultimate sensitivity of one in 10^{16} μ-decays. The experiment uses novel thinned silicon pixel sensors for high spatial resolution and scintillating fibres as well as scintillating tiles for high timing resolution. These technologies combined with a detector design for highest possible momentum resolution allow a background suppression below the targeted $\sim 10^{-16}$. To perform the measurement in a reasonable time scale, very high muon decay rates are needed. These high muon rate and background suppression are the main challenges for the experiment and define, together with a desired high acceptance, the design.

To suppress background events, a precise vertex fitting, better than 200 µm, momentum measurements, better than 0.5 MeV/c, and timing resolution, better than 100 ps, are required. Therefore, the material inside the detector is reduced to below 1 ‰ of a radiation length to minimize scattering. Furthermore, the innermost layers are very close to the target to improve vertex resolution.

In the experiment, muons decay at rest, hence the maximal available momentum is 53 MeV/c. Because no calorimeter is needed, a very compact detector design is favourable to detect on one hand electrons with a momentum as low as 10 MeV/c. On the other hand, electrons with a higher momentum are measured with high precision as recurlers after almost one full cycle in the 1 Tesla magnetic field. Additional scintillating fibers and tiles provide very precise timing information, which is needed for background suppression particularly at high rates. For a design as shown in figure 1.5 with a pixel size of 80x80 μm the momentum resolution is multiple scattering dominated.

The detector is composed of up to five 36 cm long cylinders with an outer diameter of 17 cm surrounded by a magnet and its shielding. To provide enough free space for recurling electrons of up to 53 MeV/c the minimal distance of the magnet to the experiment's central axis can not be smaller than 50 cm. For cooling the whole detector volume is flushed with gaseous helium with a flow of several m/s.

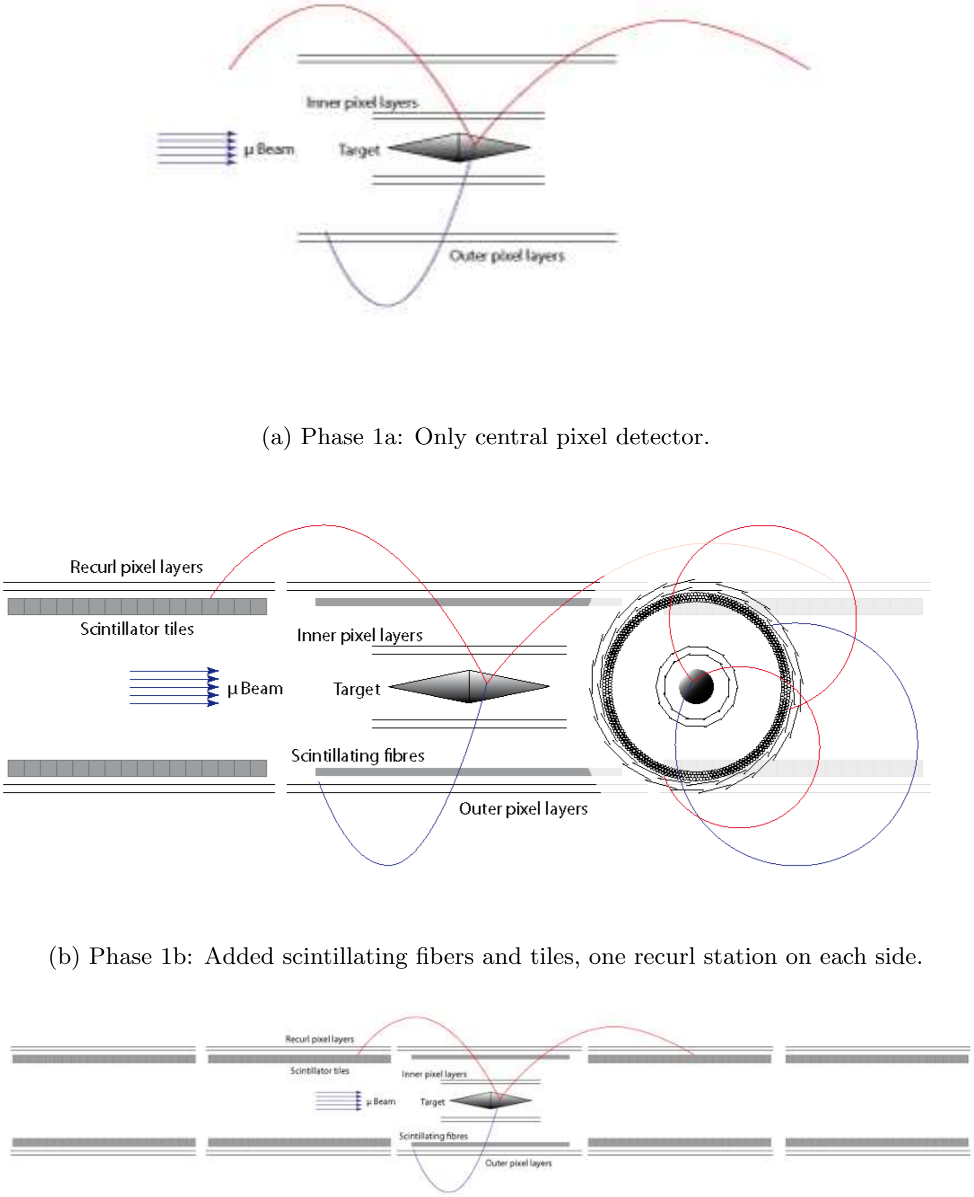
The detector will be built in phases. A first phase, called Ia, is composed only of the inner and outer layers of the center pixel sensors element. In phase Ib the scintillating fibers and recurl stations are added. Phase I will be operated with a maximum muon rate of $2 \cdot 10^8$ Hz. For phase II one additional recurl station on each side as well as tile sub-detectors will be added to handle rates up to $2 \cdot 10^9$ μ/s.

Muon Production and Stopping At the Paul Scherrer Institute (PSI) in Switzerland, a cyclotron produces a 2.4 mA proton beam with particle momenta of 590 MeV/c. The proton beam hits a graphite target rotating with 1 Hz, producing pions which decay on the surface to muons. The proton beam bulk remains and is shot to a spallation neutron target, which is built from lead-filled zircaloy tubes.

For phase I, the πE5 channel at PSI provides 28 MeV/c muons at a rate of 10^8 μ/s produced in target E. Their momentum is very close to the kinematic-edge of stopped pion decay and hence close to the maximum production rate. These muons can be stopped efficiently in the thin Mu3e target. For phase II, a new beam line also at PSI is being planned, the high intensity muon beamline (HiMB). The HiMB extracts muons produced at the existing spallation neutron target. This new beam is supposed to deliver up to $3 \cdot 10^{10}$ μ/s, $2 \cdot 10^9$ μ/s are needed for Mu3e.

In the Mu3e detector, the polarized muons are stopped in a 100 mm long hollow double cone target with a maximum diameter of 20 mm. The front cone is made of 30 μm and the back one of 80 μm aluminum.

Pixel Detector The Mu3e pixel tracker, here after called the MuPix sub-detector, is built from High-Voltage Monolithic Active Pixel Sensors (HV-MAPS) thinned to 50 μm [17]. The sensors are held by a Kapton support structure. Aluminum traces on Kapton flex-prints supply the chips and provide fast serial data links. The 150 mW/cm^2 heat from the sensors is cooled with a global gaseous helium flow as well as by small helium tubes in the support structure.



(a) Phase 1a: Only central pixel detector.

(b) Phase 1b: Added scintillating fibers and tiles, one recurl station on each side.

(c) Phase 2: Additional recurl stations on each side.

Figure 1.5: Mu3e experiment setup overview. Phase I consists of inner layer and corresponding outer layer including the fiber sub-detector. Phase II adds a recurler station on each side with pixel and tile sub-detectors. In (b) on the right side a front view with recurling electron and respectively positron tracks is shown [11].

In classical MAPS designs, ionization charges are collected by diffusion with a time constant of several hundred nanoseconds. Applying a high bias voltage, introduces charge collection by drift and increases the time resolution to the order of 10 ns. Deep N-wells allow to place the complete electronics inside the pixels. The per pixel electronics are accompanied by a per sensor digital serial readout part. The pixel sensors provide zero-suppressed hit information with an associated 20 MHz Gray code timestamp. HV-MAPS are produced in a standard technology mainly used in the automotive industry, AMS/IBM 180 nm HV-CMOS. Thinning silicon wafers down to 50 μm is also a standard procedure.

Two different types of sensors are used for the inner and the outer layers. Both have pixel sizes of 80 x 80 μm^2, the inner sensors have a size of 1.1 x 2 cm^2 and are equipped with three serial output lines, whereas the outer ones have a size of 2 x 2 cm^2 and provide only one line [18, 19].

Fiber Detector The pixel sub-detector's hit information is read out in 50 ns frames. To be able to handle rates up to $2 \cdot 10^9$ decays per second, which results in up to 100 tracks per frame, more precise timing information is needed. A scintillating fibre (Sci-Fi) hodoscope with a length of 36 cm and a radius of 6 cm and a timing resolution of 1 ns partly solves the problem. The fibers are a trade-off between a minimal material budget to decrease scattering and an efficient readout. Ribbons with three layers of 250 µm round fibers as well as 2 layers of 250 µm square fibers are under discussion. The light produced in the scintillating fibers is detected by silicon photo multipliers (SiPM) mounted at both ends of the ribbons. These devices are very compact, have a high gain factor and are insensitive to the presence of magnetic fields. They can be operated at very high rates [20].

Tile Detector The timing measurement in the recurl stations is performed with scintillating tiles right inside the pixel layers. Since this is the last measurement performed on the particles, more material can be used. The tiles achieve a time resolution of ≈ 0.1 ns and an efficiency close to 100%. Like the scintillating fibers they are read-out with SiPMs [21, 22].

Detector Environment All the above described elements of the detector are placed inside a homogeneous solenoid 1 T magnetic field. The whole detector volume is flushed for cooling with gaseous helium supplied by helium cooling channels inside the Kapton base structure. The read-out electronics is placed up- and down-stream directly on the beam pipe, which is cooled through embedded channels for liquid coolant [23]. Figure 1.6 shows a rendering of the phase 1 detector and shows the limited space available for readout electronics.

1.3 Mu3e Readout Concept

The Mu3e readout chain is designed in such a way that every graphic processing unit (GPU) in a filter farm receives data of the entire detector, but only of a small time slice. The raw data from all sub-detectors are buffered, ordered, bundled, merged, routed and transformed in the data acquisition system. Data reduction takes place only at the last node through complete track and event reconstruction. Finally, only selected events are stored.

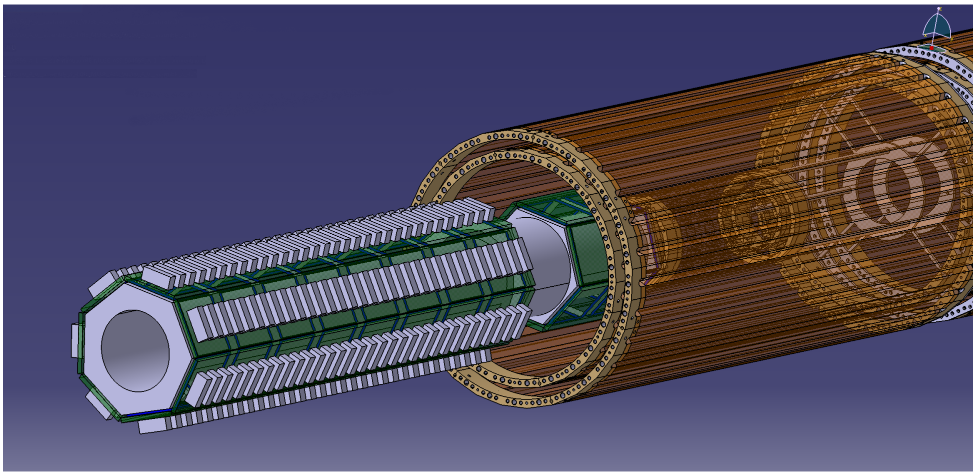


Figure 1.6: Mu3e phase 1 detector rendering with 4 layers of pixel detector, beam pipe and electronics in green. The available space for readout electronic is highly limited.

Figure 1.7 shows a data flow overview with focus on the MuPix sub-detector. MuPix pixel chips send zero-suppressed data over LVDS links to a front-end FPGA. The received hit data is time ordered, merged and routed via optical links to a read-out FPGA which routes it further on to a standard PC in the filter farm. Different sub-detectors are processed with separate read-out FPGAs. The third FPGA in the chain transforms the hit data into global coordinates and puts it through direct memory access (DMA) into a powerful GPU. Online event reconstruction is performed and selected events are stored. Slow control information is sent via the same links from a controller over read-out and front-end FPGA to the pixel detector.

Data links in Mu3e handle $\mathcal{O}(1\ \text{Tbit/s})$ through different technologies. The components required to handle this rate are shown in figure 1.8, in phase Ia only subfarm A is needed. The data from 1116 pixel sensors are divided into up- and down-stream and collected in 38 front-end FPGAs with 45 or 36 links each. The up- and downstream data sets are collected in two readout FPGAs, which deliver full detector information of a time slice to one of 12 PCs in the filter farm. Each PC is equipped with one FPGA and one powerful GPU [24].

In the following, each element of the readout chain is described in detail, where the focus lies on the MuPix sub-detector.

1.3.1 Pixel to Front-End Links

The MuPix pixel chips have an integrated digital logic, which provides zero-supressed 8b/10b encoded serialized hit data. They run without a trigger. Gray code timestamps can be mixed over multiple frames due to the internal pixel read-out scheme. 800 Mbit/s LVDS (see section 7.2) lines implemented with Aluminum stripes on Kapton foil transmit the hit data to front-end FPGAs. The innermost sensors of layer 0 and 1 use three, the others one link. Slow control signals are implemented in single aluminum Kapton flexprint lines. A global clock and reset is distributed over the whole system as a differential signal.

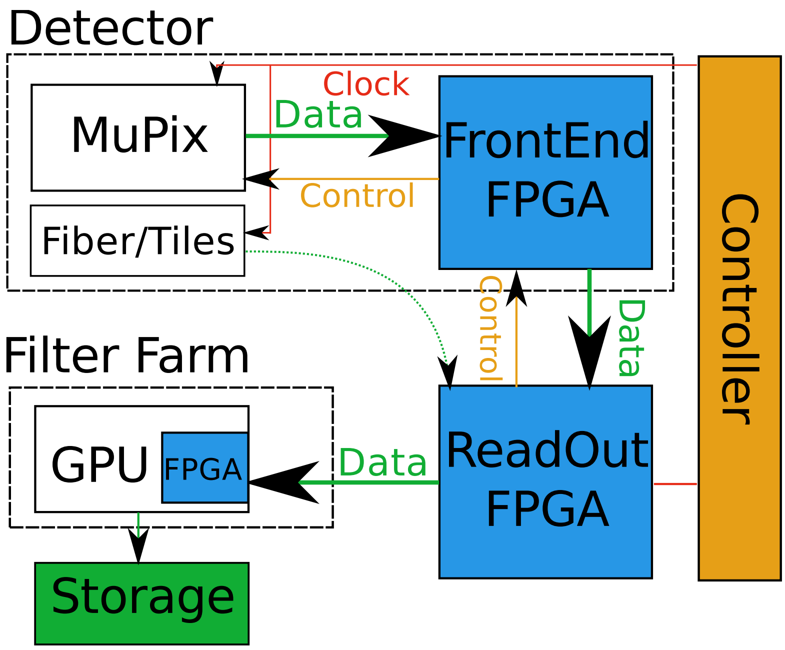


Figure 1.7: MuPix readout chain with data connections in green, control in orange, clock in red and all FPGAs used for the chain in blue.

MuPix Address Scheme

Hit information from the MuPix sub-detector is encoded in the pixel address in columns and rows of the corresponding chip. The smaller chips in the vertex layers encode the hits into 8 column bits and 7 row bits, whereas the sensors in the outer layers need 8 bits due to their double area. Both chip types add 8 bit Gray counter timestamp information. In total, a hit from a sensor consists of 23 bits respectively 24 bits. This is the amount of data that has to be transmitted over Kapton flexprints to the front-end FPGAs.

In the front-end FPGAs, information about the chips' position in the detector has to be added. 5 bits are used to address the chips position along the beam direction. Upstream chips get values between 0x7 and 0xF, downstream between 0x10 and 0x18. Another 5 bits encode the phi position and the 4 last bits the layer number. An overview of the address scheme is given in figure A.6.

1.3.2 Front-End FPGA

A total of 38 front-end FPGAs are located on both sides, up and downstream, directly outside the active area. For cooling reasons they are thermally connected directly to the beam pipe structure. They receive encoded zero-supressed pixel sensor data with Gray code timestamps (see 3.5) from 36, respectively 15 sensors, convert the timestamps and buffer the events time ordered before they are sent out again in frames. The exact data structure of these frames depends on the link performance and is a part of the scope of this work.

Simulations show an average of 0.05 hits per 50 ns frame per sensor in the busiest sensors for a muon rate of $2 \cdot 10^7$ and up to 5 hits per frame per sensor for muon

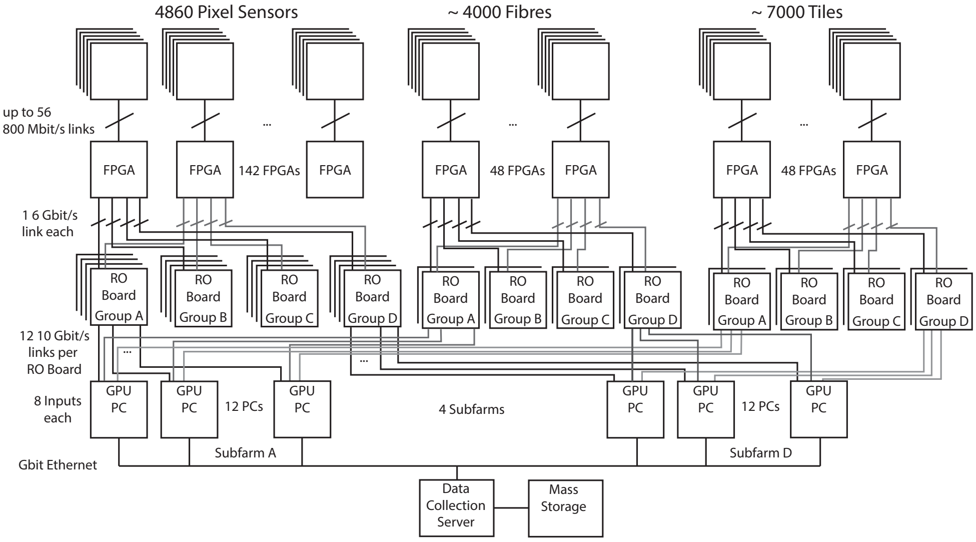


Figure 1.8: The Mu3e detector is read out with fast links in three stages: The first stage consists of the links from the detector chips of the pixel detector, the fiber tracker and the tile detector. These ASICs send zero-suppressed data over fast LVDS links to the front-end FPGAs. The second stage consists of fast optical links from the front-end FPGAs to FPGA driven readout boards in the counting house. A third set of links distributes the data from the readout boards to the filter farm PCs [25, Fig. 3].

decay rates of $2 \cdot 10^9$. This requires in phase 2 a bit rate of 1 Gbit/s if a 30 bit address scheme as described in 1.3.1 is used. The received events are not strictly time ordered, but in phase 1a all are distributed inside 16 frames with an exponential decrease for big delays. If the muon rate is increased to $2 \cdot 10^8$, in phase 1b, the maximal delay in timestamps reaches 23 frames. The delay depends strongly on the used readout speed as well as on the hit frequency of the busiest sensors. If 800 Mbit/s LVDS links are used in phase 1, more than 80 % of the link bandwidth is free. For phase 2, 1 to 1.25 Gbit/s LVDS links are planned [26].

1.3.3 Detector to Counting House Links

The MuPix front-end FPGAs as well as the front-end FPGAs of the other subdetectors send time ordered data over high speed optical links outside the detector. The optical links ensure a galvanic separation of the detector from the filter farm. Performance tests of these optical links are the main scope of this thesis. Additional slow control information has to be transmitted between the front-end FPGA and the counting house. Whether this requires additional links or can be added to the data stream is subject of investigations. A suggestion can be found in chapter 5.4.

1.3.4 Read-out FPGAs

The read-out FPGAs each receive data in time slices from one sub-detector partition. The already time ordered data sets of different read-out FPGAs are combined to packages which contain the whole detector information of such a slice and are routed as one package to one of 12 GPU equipped PCs in the filter farm. Therefore, high speed optical links are used again. Because the number of required links is much smaller than between front-end and read-out FPGAs, slightly faster links could be used.

1.3.5 GPU Filter Farm

The standard computers in the filter farm are equipped with a FPGA and a powerful graphic processing unit (GPU). The FPGA card receives the optical data, transforms from the local pixel address into global coordinates and pushes it over the PCIe interface via direct memory access (DMA) to the GPU. On the GPU online track and event reconstruction is performed. Only selected events are sent to a storage device.

Part II

Basics of Data Transmission

Chapter 2

Physical Layer

Communication, in particular digital, is the transmission of information from one point to another. The first part of this chapter describes the theory of transporting analog signals through space. The second one describes how the actual information, mostly digital states, can be encoded into the available physical channels. This is followed by a third part, which addresses techniques to check the quality of transmission lines.

2.1 Signal propagation

A signal in an mathematical approach is an abstract concept of knowledge. A totally deterministic signal, where the time evolution is known exactly by the observer, is useless for the transmission of information. According to the formulation of Wiener and Shannon, messages must be unpredictable to have an effective information content [27]. A physical signal is usually a certain condition of a physical medium that can be measured by the observer. Such physical signals are discussed in more detail in the following part.

Signals in the scope of this work are carried either as electrical signals in conducting wires or as electromagnetic waves. In both cases their propagation is described by Maxwell's equations.

2.1.1 Electrical Conductors

An electrical conductor obeys Ohm's law $V = I \cdot Z$, where V is the voltage, I the current and Z a complex impedance. The impedance of different elements is given by

$$Z_{resistor} = R \tag{2.1a}$$

$$Z_{capacitor} = \frac{1}{\omega C} e^{-i\frac{\pi}{2}} \tag{2.1b}$$

$$Z_{inductor} = \omega L e^{+i\frac{\pi}{2}} \tag{2.1c}$$

where C is the capacity and L the inductivity of the corresponding element. An electric wire's impedance Z_0 can be described as the sum of Z_R, Z_C and Z_L. Depending on the values of Z_0, different frequencies pass or are suppressed.

If two elements, for example wires, with different impedance are connected, a part of the signal, described as a wave, gets reflected. The reflection coefficient is given in equation 2.2 [28], where Z_a and Z_b are the impedances of the two elements. Note that the reflection coefficient is frequency dependent.

$$\Gamma = \frac{Z_b - Z_a}{Z_b + Z_a} \tag{2.2}$$

To ensure proper signal propagation, the impedance of all elements has to be matched to minimize reflections. It is common to use components with an impedance of 50 Ω.

Depending on the used signal frequency, different cable designs are in use. For relatively slow signals, copper wires are well suited. For faster signals, such as radio frequencies, coaxial cables are usually used. They consist of an inner conducting core surrounded by an insulating layer, all enclosed by a shield. The advantage is that the electromagnetic field exists only inside the cable. Many other cable concepts exist. Microstrips are thin flat strips parallel to a ground plate, striplines are stripes sandwiched by two ground plates and balanced lines consist of two identical wires. In the last one, differential signals are usually used. Such structures, which build a structure in between which electromagnetic waves propagate are called wave guides.

2.1.2 Optical Wave Guides

Electromagnetic waves with optical frequencies can propagate inside optical fibers. These fibers consist of a transparent material with a higher refractive index in the core than outside. All light which propagates with an angle smaller or equal than the critical angle given by Snell's law $n_1 sin(\theta_1) = n_2 sin(\theta_2)$ propagates due to total internal reflection along the fibers. Very often material with a refraction index gradient is used.

Generically, multiple discrete solutions of Maxwell's equations exist inside wave guides. The lowest possible frequency is called "cut-off frequency". Depending on the guide geometry, they support only one propagation path, called single mode, or multiple paths as well as transverse modes, called multi-mode fibers. Single mode fibers are used for signal propagation over long distances in the order of kilometers, whereas multi-mode fibers are usually used for distances up to 50 m. Different dispersion relations of the different modes lead to a degeneration of the signal.

Optical wave guides are usually fed by monochromatic laser pulses. The crucial point is the coupling of the not necessarily Gaussian modes of the input laser beam into the discrete Gaussian modes of the fibers. The efficiency is given by the overlap of the two mode shapes.

2.2 Encoding Schemes

The very simple concept of sending data from one point to another can be realized in a number of different ways. The following section describes how the data, typically represented in binary bit states, is translated into states of a carrier medium which can be back-translated into binary bit states. Line codes describe typically the encoding of data into physical states, whereas running disparity and scramblers are tools to improve the transmission quality. In addition, an overview of selected protocols which specify data transmission is given.

2.2.1 Line Codes

Line codes describe how bit states “1” and “0” are represented in a physical signal. Depending on the used transmitting medium and distance, the data rate and application, different schemes are applied. In the following, three widely used schemes are presented.

In *return-to-zero* codes, the signal always returns to zero between the transmitted bits, thus the two states are described by positive and negative signal states. Three possible states are required. In optical communication a two state *inverted return-to-zero* scheme is applied very often. Data pulses which are shorter than the underlying clock are used to represent a “0”-state, the absence of a pulse represents a “1”-state [29].

Another example of a line code is *Manchester* encoding, in which “1”-states are represented in a falling edge and “0”-states respectively in a rising edge of the signal. This scheme is very frequency error and jitter stable and due to the many transitions clock recovery (see A.1.1) is relatively easy. But the many transitions turn into a disadvantage at high data rates, because double the bandwidth is required compared to non-return-to-zero codes as described below [30, 31].

Non-return-to-zero schemes align different bit states next to each other without any intermediate states. Due to fewer transitions, this scheme allows higher data transmission rates. In exchange, the clock recovery and bit alignment are more difficult. This scheme is applied in all transceivers used in this thesis [29] .

Line codes are also used to encode fixed length data words into patterns with properties suited for data transmission. Line codes can add some additional information to the data and therefore can need additional bits and thus additional bandwidth [32, chapter 1.3]. The following four issues can be addressed:

Clock Recovery If the line code does not foresee an additional clock transmission, the transmission’s bit rate and phase has to be recovered from the serial data stream. How the binary states are translated into signals is dominated by considerations concerning the reconstruction of the clocking information encoded into the data stream. In general, a high frequency of transitions is desirable.

DC Balancing ensures a balanced number of ones and zeros over the long run. This leads to vanishing net current flow.

Data and Control Word The chosen data pattern or some dedicated bits in the encoded data words hold additional information whether the bits of the current word are to be treated as data, or as a predefined control sequence. Some protocols, for example Interlaken (see 2.2.4), know control words which also contain a data part.

Error Detection Line codes can set constraints on resulting encoded data words. Not all combinatorially possible bit patterns represent a valid pattern of the used encoding scheme. This fact allows invalid pattern detection, hence some errors due to bad transmission quality can be detected.

In the following, different line code schemes which map data words into dedicated bit patterns are described in detail.

Word	Data	dp=-1	dp=+1	Word	Data	dp=-1	dp=+1
D.00	00000	100111	011000	D.16	10000	011011	100100
D.01	00001	011101	100010	D.17	10000	100011	
D.02	00010	101101	010010	D.18	01010	010011	
D.03	00011	110001		D.19	01011	110010	
D.04	00100	110101	001010	D.20	01100	001011	
D.05	00101	101001		D.21	01101	101010	
D.06	00110	011001		D.22	01110	011010	
D.07	00111	111000	000111	D.23*	10111	111010	000101
D.08	01000	110001	000110	D.24	11000	110011	001100
D.09	01001	100101		D.25	11001	100110	
D.10	01010	010101		D.26	11010	010110	
D.11	01011	110100		D.27*	11011	110110	001001
D.12	01100	001101		D.28	11100	001110	
D.13	01101	101100		D.29*	11101	101110	010001
D.14	01110	011100		D.30*	11110	011110	100001
D.15	01111	01011	101000	D.31	11111	101011	010100
K.28	11100	001111	110000				

Table 2.1: 5b/6b encoding scheme, for certain 5 bit words two different disparity ($dp = \pm 1$) encodings exist. D.x are all 32 possible data words and K.x represent the predefined control words. The D.x* words can also be used to build control words.

2.2.2 Running Disparity

The disparity of a given data word is defined as the difference between ones and zeros in it. If a word consists of more ones than zeros its disparity is defined to be positive. The running disparity (rd) is a continuous sum over the disparities of all previously received words. In principle it is possible to calculate the rd after each received data bit, but this is usually not necessary.
Some protocols or encoding schemes, such as 8b/10b, restrict the running disparity to a given set of values.

8b/10b Encoding

In 1983 Al X. Widmer and Peter A. Franaszek [33] introduced for IBM a scheme to encode 8 bit words into 10 bit patterns to ensure DC balancing (see 2.2.1) and added at the same time the possibility to send a predefined set of control words. The 8b/10b encoded words consist of 10 bit patterns whose disparity is either ±2 or 0 and which have never more than five times the same bit state in a row. Out of the $2^{10} = 1024$ combinatorially possible patterns only 584 are valid in the sense of this definition. Because this number is bigger than $2^8 = 256$, which is the number of possible bit patterns which are to be encoded, some 8 bit values can be assigned to more than one 10bit pattern.
To achieve the above stated properties, the 8 bit pattern is split into two parts and encoded separately in a 5b/6b and a 3b/4b part. There are different ways to implement an 8b/10b encoding, in the following the commonly used version in IBM's patent [34] is explained in detail. All the possible outcomes as well as the possible valid control words are shown in tables 2.1 and 2.2.

During data transmission the disparity over all previous data is summed up, this

Word	Data	dp=-1	dp=+1	K-Word	Data	dp=-1	dp=+1
D.x.0	000	1011	0100	K.x.0	000	1011	0100
D.x.1	001	1001		K.x.1	001	0110	1001
D.x.2	010	0101		K.x.2	010	1010	0101
D.x.3	011	1100	0011	K.x.3	011	1100	0011
D.x.4	100	1101	0010	K.x.4	100	1101	0010
D.x.5	101	1010		K.x.5	101	0101	1010
D.x.6	110	0110		K.x.6	110	1001	0110
D.x.P7	111	1110	0001	K.x.7	111	0111	1000
D.x.A7	111	0111	1000				

Table 2.2: 3b/4b encoding scheme, for certain 2 bit words two different disparity ($dp = \pm 1$) encodings exist. For D.x.7 either P7 or A7 has to be chosen to ensure that in the resulting 10 bit pattern never more than five equal bits occur.

sum is denoted running disparity (rd). Depending on the current rd the new data pattern is assembled according to the following rules to ensure that the running disparity always has a value of ±1. Whenever the pattern assigned to the word to be encoded has only a neutral disparity pattern ($dp = 0$), the pattern is transmitted and the running disparity is kept in the same ±1 state. If the assigned pattern can be represented in a $dp = +2$ or $dp = -2$, the one with the opposite sign to the running disparity is chosen, the rd is thereby inverted.

64b/66b Encoding

The 64b/66b encoding scheme uses two extra bits to encode a 64 bit word into a data pattern with given properties [35]. The highest two bits, number 65 and 64, are either set to "10" or to "01". A "01" prefix states that the following 64 bits are entirely data, whereas a "10" is followed by an eight bit type word, which defines the function of the remaining 56 bits. The two patterns "00" and "11" are not used, their detection in a receiver denotes the occurrence of an error. These constraints to the transmitted patterns introduce an assured bit transition at least every 65 bits. The run-length of 64b/66b encoded data streams is 65. Most of the modern transceiver designs require transitions at least every eighty bits. This requirement is naturally met with this encoding scheme and it introduces the possibility to send control words.

The main difference between 64b/66b and 8b/10b encoding is the smaller overhead of the first one. However, 64b/66b does not introduce a bound DC balance, and has a much longer run-length. DC balancing is only given statistically and improved if a scrambler (see 2.2.3) or an additional disparity (see 2.2.2) control is added. When 64b/66b is mentioned, very often scrambler and disparity control are addressed implicitly as well [36].

Which types of control words are used and whether they need the whole remaining 65 bits or a control word data combination is allowed has to be specified in the used protocol. This is done for example in Interlaken (see 2.2.4) or the 10GE (see 2.2.4).

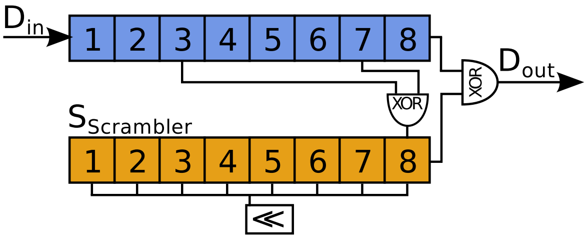


Figure 2.1: Self-synchronizing scramblers with taps 3 and 7. Data (D_{in}, blue) are scrambled/descrambled with the actual scrambler state $S_{Scrambler}$ (orange) which is updated by the received data. The $\ll$ stands for a left shift of each register's bit.

2.2.3 Scrambling

A scrambler modifies the payload data before the transmission in such a way that a descrambler on the receiving side can recover the original data. This method is used to give the transmitted data pattern desired properties or at least to decrease the statistical probability for undesired patterns. Scramblers are neither used nor suited for cryptography purposes.

Given random-like data, the occurrence of disadvantageous patterns can be reduced statistically by scrambling the data with a scrambler state. Such a scrambler state is a pseudo random number, generated at run time in a linear feedback shift register (3.4) which is x-ored to the data words.

The scrambler state can be generated either in a synchronous or a self-synchronous way. Synchronous scramblers need an initialization state and the states on the transmitter and receiver side have to be synchronized. On the other hand, self-synchronized scrambler states are calculated out of the data stream as shown in figure 2.1. This self-synchronizing scrambling scheme is synchronized after as many words are received as the scrambler state is long. Bit errors at the tap position of the scrambler state lead to error multiplication.

2.2.4 Protocols

For the targeted bit rate of several Gbit/s several protocols exist. Three of them are described below. Some concepts of these protocols are used in the following parts to either increase data transmission quality or to design a protocol tailored to the actual expected data.

Ethernet

Based on ideas from [37], Ethernet became standardized in 1985 in the IEEE 802.3 standard [38]. It is a very widely used local area network (LAN) technology that allows the connection of devices in a very flexible system. It is designed to share the same physical lines between multiple devices.

Ethernet evolved from a 10 Mbps to a multi-Gbit/s system which includes definitions on different abstraction levels, called layers. It is designed for networks where all devices are in principle connected in an arbitrary topology with all others.

[39] Ethernet is basically made up by four parts:

1. Frames are defined sets of bits which are sent over the network.

2. A media access control protocol manages the fair access to channels which are shared between multiple devices.
3. The component which physically sends the data.
4. A physical medium is used to carry the digital signals.

In order to be able to use the same physical layer to send information from one point to another, the above-mentioned frame structure is introduced. An Ethernet frame consists of a preamble, the destination address, as well as the source address. This information is used by all attached devices to identify frames destined for them. This header is followed by information about the frame size, the actual data and a cyclic redundancy check (CRC) hash as described in section 2.3.3.
Modern Ethernet networks operate with duplex lines which are rarely used by multiple devices. All devices are usually connected to a switch that handles potential collisions, which would occur if different devices access the same lane at the same time [39]. Even though Ethernet is designed for communication in LAN, where at least in principle multiple devices access the same lines, useful concepts can be extracted also regarding point to point connections.

10GBASE

Different higher level protocols describe how the above described Ethernet frames are transmitted in detail. A subgroup of such specifications is built by the 10GE technologies which specified an explicit duplex 10 Gbit/s transmission. Different versions for copper and optical physical layers exists. Here, the focus is on the optical versions as described in the IEEE standard 802.3ae [40]. Different physical specifications exist for different distances of data transmission. The focus is again on the short range version 10GBASE-SR that is specified to use 850 nm lasers, optical multi-mode (OM2) fibers, which have a maximal range of 50 meters, 64b/66b encoding as described in section 2.2.2 and specified in [41], and that is designed for a data rate of 10.3125 Gbit/s.
The available optical SFP hardware, as described later on in section 5.1.3, fulfills this specification and a 10GBASE PCS can be implemented very easily into the Stratix V FPGA IP hard cores (see 5.1.1). Nevertheless, it should be noted that whis would fix the data rate at 10 Gbit/s.

Interlaken

Contrary to the Ethernet protocol described above, the Interlaken protocol specifies a chip-to-chip interface for networking. This rather new protocol is designed as a successor of the XAUI [42] and SPI4.2 [43] protocols. The purpose of this short outline is to identify ideas which could reasonably also be used for a specifically designed protocol for the Mu3e data readout. Therefore, only the new Interlaken protocol is presented and not the two underlying older protocols.
Interlaken is designed to operate on multiple lines in parallel and its performance scales with that number. Nevertheless, it can also be operated with only one line. Interlaken uses 64 bit input data to generate 67 bit patterns. A 64b/66b encoding with additional running disparity control is applied and the generated data patterns are fed through a scrambler as described in section 2.2.3.
In the following the different Interlaken concepts are described in detail. The focus is on properties that are also important in single lane operation mode.

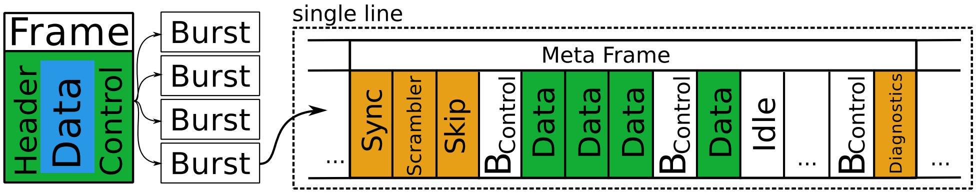


Figure 2.2: Interlaken protocol overview. The framed data is divided into different bursts. These bursts are sent within a meta frame on single lines.

In general, Interlaken communications are wrapped in frames. They are used for synchronization of the different parts and to share diagnostic information between the two devices. A frame's data is splitted into bursts, a package of data transmitted serial on one single line. The per lane communication is wrapped into meta frames. Figure 2.2 shows the different data wrappings.

Meta Frames Meta frames are used for synchronization and diagnostic purposes. An Interlaken meta frame consists of a synchronization part, the scrambler state, optional skip words for phase compensations, the payload, and some diagnostic at the end. The synchronization is implemented by sending the control word type "b011110" and the alignment pattern "h0F678F678F678F6". The skip words are dedicated words, which do not contain any data and therefore are skipped at the receiver side, which introduces a certain capability of rate matching. The type is specified by "b000111" and contains the fixed pattern of "h21E" followed by six times "h1E".
The diagnostic type is specified by the pattern "b011001" and contains mainly a CRC32 (see 2.3.3) hash over the whole frame where the three highest bits are never included and the scrambler state is set to all zeros for hash calculations because it can be different for each lane. The used CRC32 polynomial is given by

$$x^{32}+x^{28}+x^{27}+x^{26}+x^{25}+x^{23}+x^{22}+x^{19}+x^{18}+x^{14}+x^{13}+x^{11}+x^{10}+x^{9}+x^{8}+x^{6}+1 \tag{2.3}$$

Burst and Frames Interlaken distinguishes between bursts, which are associated to single channels and bound by two control words, and frames which can include multiple bursts and contain a package of data as described above.
The data package is sent over one or multiple channels in bursts whose length is variable, but limited by an upper and lower limit. Between two bursts, there is always a burst control word whose bits 65 to 64 are set, according to 64b/66b encoding (see 2.2.2), to "01" and the next lower bit 63 to "1". Table 2.3 shows how burst/idle words are built exactly.

The lower limit of the burst length can introduce data words which cannot be used. To avoid this, dedicated algorithms are described in the specification to find the optimal burst length given the allowed burst length range [44, p. 16].

Control Words As described in section 2.2.2 about 64b/66b encoding and in section 2.2.2 about running disparity, the first three bits are used for disparity control and to indicate control words. If a control word is detected, bit 63 indicates whether it is a burst or a framing control word. In the case of a burst control word, the next

Bit	
66	Inversion
65:54	framing “10”
53	Control “1”
62	Type
61	start of packet (SOP)
60:57	EOP Format
56:	Reset Calendar
55:40	In-Band Flow Control
39:32	Channel Number
31:24	
23:0	CRC24

Table 2.3: Structure of an Interlaken idle/burst word [44].

bit, number 62, indicates whether it is a burst control word with a following start of packet (SOP) flag or an idle statement to fill up unused data slots. Table 2.3 shows a control word overview.
If the current data is a burst control word, the end of packet is indicated in the bits 60 to 57 with a leading “1”. The following bits state how many data words, consisting of 8 bits, of the current word in the bits 55 to 0 are valid and still belong to the ending packet. The pattern “0000” in these dedicated bits indicates that the control word is not an end of package word and the pattern “0001” indicates the occurrence of an error in combination with the end of the package.
The last 24 bits of a burst control word contain a CRC24 hash of the previous data burst and the current control word. (see 2.3.3). The CRC24 is calculated with the following polynomial: [44, p. 18]

$$x^{24} + x^{21} + x^{20} + x^{17} + x^{15} + x^{11} + x^{9} + x^{8} + x^{6} + x^{5} + x + 1 \tag{2.4}$$

Synchronization The Interlaken specification explains exactly how the synchronization of each lane as well as multiple lines with respect to each other have to be synchronized. The single lanes synchronize to the clock data recovery (CDR) (see A.1.1), to the 64b/67b word boundaries and the scrambler state. The interface as a whole first synchronizes all single lanes and then aligns the lanes in addition.

Flow Control The protocol leaves open whether the flow control, a status about all used lanes, is incorporated into the data stream or whether an off flow solution is chosen. Once a channel is open, the transmitter is allowed to use it. No credit system is implemented. The in-band flow control is encoded into the burst and idle control words.

Scrambler In contrast to the 58 bit long scrambler, which is self-synchronized on the payload in the older Ethernet IEEE 802.3 [38] standard, Interlaken uses an independent synchronous scrambler for each line. This mainly reduces the danger of error multiplications (see 2.2.3). The scrambler state is payload independent and generated out of the taps 58 and 39. The downside of this scheme is the need for scrambler state synchronization, which is the reason why the scrambler state is transmitted in the meta frame header. The control word type, which contains the

58 bit of the scrambler state is indicated with the type “b001010”. The scrambler is never applied to the three highest bits which contain the parity and the type pattern [44, p. 30].

2.3 Signal Quality Check

Once the physical and digital encoding of data described in section 2.2 are implemented, online signal quality checks are a desired feature. In the first two subsections of this chapter, tools for physical signal quality checks such as eye diagrams and bathtub plots are described. In a second step, cyclic redundancy checks are introduced which allow an evaluation of the correct data transmissions by adding only a very small amount of extra data.

2.3.1 Eye Diagrams

Eye diagrams are a tool to screen the signal quality in fast data transmissions where non-return-to-zero (see 2.2.1) schemes are used. The different transitions from a “1”-state to a “0”-state and vice versa are folded into a single diagram. Perfect signals, where the transitions are performed instantaneously result in a square with the length $T_{bit} = \frac{1}{f}$ where f is the serial clock frequency of the data transmission and the height V_{diff} is the differential voltage.
The physical medium which propagates the signal as well as all included electronic circuits constitute a low pass filter and deform the signal. The folding of real signals looks much more like an eye. Figure 2.3 shows an example eye diagram. The presented signal shows a wide eye opening, very little jitter, a crossing level almost in the center and much faster falling times than rising times.

Jitter introduced either by the transmitter and receiver units or the clock recovery circuits (see A.1.1) result in misalignment of the data transition lines in the horizontal time axis. The eye width is an indicator how well the clock recovery is working. The eye height is the difference between the lower limit of the one-level and the upper limit of the zero-level inside the eye. Only if the eye is open enough, which means that both height and width cannot be too small, a secure recovery of the sent bits is possible. A further indicator is the level at which the falling and rising edges cross. Distortions in the clock cycle or signal symmetry problems manifest in a crossing level that is not located exactly in the middle between the one- and zero-level [45].

2.3.2 Bathtub Diagrams

Similar to the eye diagrams extracted from the pure signals, one can add a bit error rate test (BERT) (see 5.2.2). So-called bathtub plots can be produced by measuring the bit error rate (BER) for different values of the signal height thresholds or by adding an offset to the recovered clocks signal. Examples of such plots are shown in the lower part of figure 2.4. They show the clock offset and the signal threshold versus bit error rates. The desired eye opening for a targeted BER can be estimated with these plots. If the two variables clock offset and signal height threshold are varied simultaneously, 3d plots with clock offset, and signal threshold versus bit error rate can be extracted. 2d projected contour plots look very similar to the eye diagrams described above, although they are not exactly the same [46, 47].

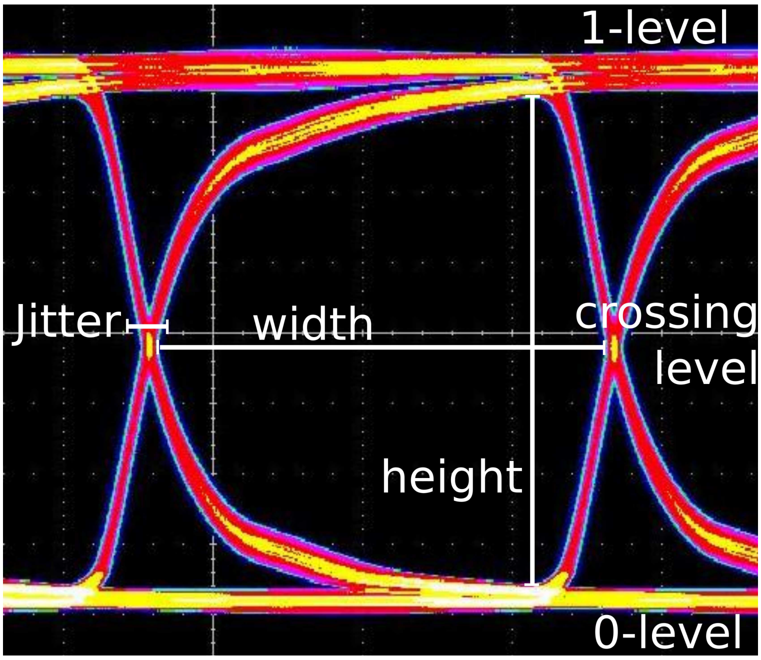


Figure 2.3: A typical eye diagram with indicated width, height, jitter and crossing level. This particular signal shows a much faster falling than rising time.

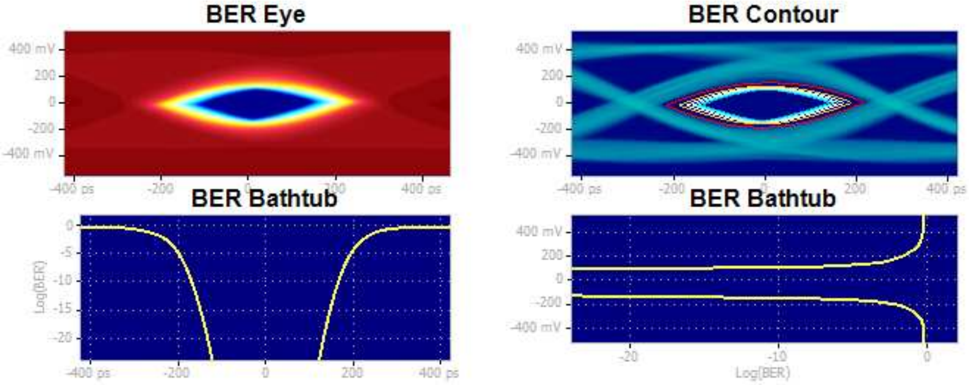


Figure 2.4: BER bathtub plots. The upper two plots show the projected 3d plots, where the lower two plots show the 2d projections which results in bathtub plots.

Even though the EyeQ circuits described in A.1.1 are called eyes, they represent more the second type of eye diagrams where a BER measurement is required.

2.3.3 Cyclic Redundancy Checks (CRC)

A cyclic redundancy check (CRC) is used for error detection in data transmissions or storage. It is a checksum with a set of very convenient properties, but it is not a cryptographic hash. CRC is essentially the remainder of a polynomial division which can be implemented very efficiently in hardware.

CRC as a Polynomial Division

Given data, represented in binary form, can be understood as a polynomial of the following form

$$a(x) = a_0 x^{l-1} + a_1 x^{l-2} + ... + a_{l-2} x + a_{l-1} \tag{2.5}$$

where $a_n \in \mathbb{F}_2 = \{0, 1\}$ are the bits of the given data and therefore $a(x) \in \mathbb{F}_2[x]$. The polynomial division of a polynomial $p(x)$ by another polynomial $q(x)$ can be expressed as finding $s(x)$ so that there is $r(x)$ a reminder polynomial with degree less than $q(x)$:

$$p(x) = s(x) \cdot q(x) + r(x) \tag{2.6}$$

The finite set of all possible $r(x)$ describes all possible CRC values given a fixed divider $q(x) = p_{CRC}(x)$. For technical reasons, the polynomial is defined after a multiplication with x^N

$$a(x) \cdot c^N = b(x) \cdot p_{CRC}(x) + r_{old}(x) \tag{2.7}$$

where N is the length of the CRC polynomial. Note that one is not interested in how $b(x)$ looks like [48, p. 3].

The above described polynomial division can be implemented with a register of the width N where the data bits are shifted in series. As soon as the bit shifted out of the other end is different from the current input bit, the register content is xored with the fixed CRC polynomial. Usually the register is filled with all ones to start. Alternative to this bit wise calculation, the CRC can be calculated out of tables where up to eight bits can be treated reasonably at the same time. Table 2.4 shows an example how a 4 bit CRC hash is calculated out of an 8 bit word.

	10011011	000
⊕	1011	
	00101011	000
⊕	1011	
	00000111	000
⊕	101	1
	00000010	100
⊕	10	11
	00000000	010

Table 2.4: CRC example. The CRC of the input data "10011011" is calculated with a CRC polynomial $x^3 + x^1 + 1$, which corresponds to "1011". The resulting CRC hash is "010". ⊕ stands for xor.

Online CRC Error Check

If the calculated CRC is added to the data out of which it is calculated and the CRC is evaluated again including the appended code, the CRC is always 0. Adding the $r_{old}(x)$ obtained from 2.7 to the data $a(x)$ to get the new data $a'(x)$ can be written as shifting $a(x)$ N bits and then add $r_{old}(x)$, hence

$$a'(x) \cdot x^N = \big(a(x) \cdot x^N + r_{old}(x)\big) \cdot x^N \tag{2.8}$$

$$= (b(x) \cdot p_{CRC}(x) + r_{old}(x) + r_{old}(x)) \cdot x^N \tag{2.9}$$

$$= b(x) \cdot x^N \cdot p_{CRC}(x) + 0 \tag{2.10}$$

the last step is true because $p(x) = -p(x)$ since the polynomials $p(x) \in \mathbb{F}_2[x]$. As it can be seen in equation 2.10 $r_{new}(x) = 0$ [48, p. 8].

Error Detection Strength

The error detection strength of a CRC code depends very strongly on the used polynomial. In general

- All single bit errors are detected by polynomials with order of at least two.
- All two bit errors are detected if the CRC polynomial does not divide the term $x^i(1 + x^{j-i})$ for $i > j$
- All odd numbers of errors will be detected if the polynomial is a multiple of $x + 1$.
- Burst errors of the length b, continuous patterns of the length b where the first and last bits are errors and the state of the bits in between are unknown, are detected of polynomials by the length b or longer and with a x^0 term [49].

There are many of standard CRC polynomials which are widely used although there may very often exist better choices. Particularly because the CRC error detection strength depends strongly on the used data width, a careful and application suited polynomial choice makes very often sense. In [50] and [51], a variety of polynomials is presented, and their performance with different data sizes performance is shown.

Chapter 3

Electronic Components

In this chapter, different electronic elements, which are referred to at various parts of this work, are explained in detail. This chapter's intention is to provide a reference.

3.1 Logic Gates

Logic gates are electronic circuits implementing Boolean functions. They build the smallest logic element of a digital circuit and are usually implemented by transistors. Table 3.1 shows the different types with the associated symbols.

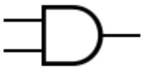
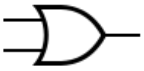
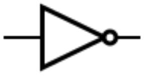
AND	OR	NOT
		

Table 3.1: Simplest logic gates with their symbols.

In addition to these Boolean function gates, tree-state gates allow the removal of an output from a circuit by a high impedance state. The same output can be shared by multiple circuits.

3.2 Memory Elements

The capability to store, respectively buffer, electric signals is a main ingredient for modern sophisticated electronics. Beside the naive and very simple approach of storing signals in long cables, different elements have been developed in the past. Particularly in clocked logic circuits they simplify the timing.

3.2.1 Flip-Flops

A flip-flop is used to store information in an electrical circuit. A flip-flop circuit has two stable states between which can be switched by applying a signal to a dedicated port. The simplest possible flip-flop, an SR latch, is shown in figure 3.1 where S is the signal, R a reset and Q and $\bar{Q}$ the signal, respectively the inverted signal [52].

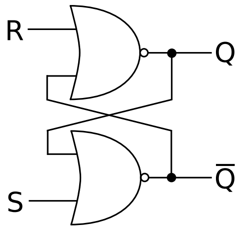


Figure 3.1: RS latch flip-flop.

3.2.2 Random-Access Memory (RAM)

This kind of storage devices provide addressed storage. That means, that through an address bus every memory cell can be read and written at any time. This feature is called random-access. When the device is not powered, the data is lost - RAM is volatile. It can be distinguished between static (SRAM) and dynamic (DRAM), where the first must not be refreshed periodically.

3.2.3 First In First Out (FIFO)

A FIFO is a memory unit which outputs the data which are put in at first, again at first. The name is an acronym for First In First Out. In electronics the storage can be implemented in various ways such as SRAM, flip-flops or others. It is distinguished between FIFOs with a common clock for read and write and FIFOs with two different clocks for the write and read process. Furthermore, FIFOs consisting of more than a few words, very often have a full and an empty port to indicate these two states which may cause errors.

3.2.4 Read-Only Memory ROM

In the contrary to RAM, read-only memory cannot be written to. This memory is used to store small programs, constants or look-up tables of mathematical functions. In 1956 programmable (PROM) memory was introduced, which cannot be written at run time but programmed before. This offers new flexibility [53].

3.3 Phase Locked Loop (PLL)

A phase locked loop (PLL) is a highly non linear circuit that outputs a signal whose phase is correlated to the input signal. There are many difference types of PLLs nevertheless the underling concept shown in figure 3.2 is always the same. A phase detector (PD) compares the phase different between the input signal S_{in}, which can be understood as a reference signal, and the feedback signal S_{fb}. The PD signal S_p is fed through a filter, in most cases a low pass filter (LPF), and is then used to control a variable frequency oscillator (VFO) whose signal is then looped back to the PD and serves as an output S_{out}. If all the components are properly tuned, the system locks itself in a stable condition [54, p. 4f].

In FPGAs cascading PLLs can be used to generate clocks with significantly different frequencies than those available from external oscillators. They are also heavily used in clock recovery circuits. In general, there can be three different types of

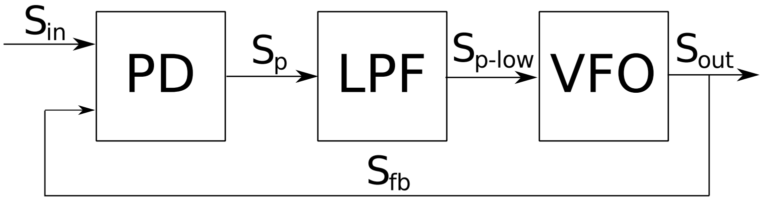


Figure 3.2: Block diagram of the basic PLL concept.

PLLs inside an FPGA. Clock Multiplier Unit (CMU) PLLs add one divider into the feedback loop (M) and one into to input signal (N) to achieve locked output signals with a multiplicity of the input frequency. Fractional PLLs (fPLL) add an additional delta sigma modulation into the feedback loop, which allows fractional values [55]. Altera has introduced auxiliary transmit (ATX) PLLs, which have the same building blocks as CMU PLLs, but are tuned for low jitter at high frequencies [56, p. 1-13].

3.3.1 Clock Data Recovery (CDR)

In serial data transmissions, the underlying clock is very often omitted. PLLs fed by the serial received data stream lock to data base frequency. This clock data recovery (CDR) is crucial for successful data transmission and requires enough transitions in the data stream.

3.4 Linear Feedback Shift Register (LFSR)

A shift register consists of a series of flipflops (see 3.2.1) with the same clock and whose output is fed into the next flipflops‘ input. The input bit of a shift register is only the one foremost bit [57]. In most general notation, a linear feedback shift register is a shift register whose input bit is a linear combination of its previous state.

3.4.1 Pseudo Random Number Generators (PRN)

For well suited linear functions, pseudo random periods with maximal cycle lengths of $2^n - 1$,where n is the number of bits in the shift register, can be achieved. In such a scheme, the bits at different positions, called taps, are xored to obtain the input bit. Very often, this is expressed in a polynomial in $\mathbb{F}_2[x]$ as shown in equation 3.1 for an 8 bit PRN. Where the exponents indicate the used taps and the $1 = x^0$ the insertion position of the new bit.

$$x^8 + x^6 + x^5 + x^4 + 1 \tag{3.1}$$

For many different PRN lengths schemes with two or four taps exist. For maximal length PRN the number of taps has to be even, and all used taps must not share any conmen divisor except for 1. Useful taps for LFSR up to 786 can be found for example in [58].

3.4.2 Counter

LFSR can also be used as counter. Because of their relative simple feedback logic with only a few xor gates they can be operated very fast. Counters based on LFSR are faster than Gray counters (see 3.5), but the advantage of only one bit flip is lost [59].

3.4.3 Other Uses

Linear feedback shift registers are also used to generate test patterns for data transmission (see 5.2.2) and used for data scrambling (see 2.2.3). They are not suited for any cryptographic use.

3.5 Gray Counter

> A binary Gray code of order n is a list of all 2^n n-bit strings such that exactly one bit changes from one string to the next [60, p. 32].

Hence, Gray codes are used for error minimizing in analog to digital encoding and to avoid errors due to readouts during flipping states. They are particularly useful between two clock domains. Gray code counters are very often used for fast counters in electronic devices.

Example The easiest way to convert binary to Gray and vice versa are explained below in 1 and 2. The here shown implementation of a Gray code is also called reflected binary code. Less demonstrative, but more efficient, decoding schemes exist.

Algorithm 1 Binary to Gray Code Conversion

```
$out_{gray} \leftarrow (in_{bin} \ll 1) \oplus in_{bin}$
```

Algorithm 2 Gray Code to Binary Conversion

```
$num \leftarrow in_{gray}$
$mask \leftarrow in_{gray} \gg 1$
for $mask \neq 0$ do
    $mask \leftarrow mask \gg 1$
    $num \leftarrow num \oplus mask$
end for
$out_{bin} \leftarrow num$
```

Chapter 4

Field Programmable Logic Gates (FPGA)

Field-Programmable Gate Arrays (FPGA) were developed in the late 1980s in an environment where electronic systems were mainly composed of standard components, such as microprocessors, memory chips or logic components, mounted directly onto multi-layer printed circuit boards (PCBs). The rapid increase of complexity in these systems led to much higher possibility of incorrectly connected components on such boards.

Programmable interconnections between components offer the possibility to test designs before production and introduce the new opportunity to adapt the functionality of components during operation to fulfill new requirements. In principle every input and output (I/O) pad can be connected to every logic component on the board. This programmability feature introduces an increase of latency between the single elements, which significantly reduces the maximum speed at which the devices can be operated.

The idea of adjusting the hardware to a given task is best met in application-specific integrated circuits (ASICs). They are commonly used and are known to operate most cost effective for large production numbers, fastest and with lowest energy dissipation [61, p. 1.1]. Nevertheless, the re-programmable properties of FPGAs overcome the run-time advantages of ASICs in many cases [61].

As shown in figure 4.1, FPGAs consist of arrays of logic blocks. These blocks can be simple transistors as well as much more complicated structures such as memory blocks, phase locked loops (PLLs ct. 3.3.1), or even a microprocessor. In modern FPGAs, most of the blocks are pairs or quartets of transistors, small gates, multiplexers, look-up tables or different AND-OR structures. The Interconnection of these blocks is programmed by electrical switches, which can be implemented with different technologies such as SRAM, EPROM or Antifuse [62].

Modern FPGAs are produced in the same technology also used for other micro processors. Various intellectual property (IP) hard cores are embedded inside the chips to fulfill highest performance and in some cases security requirements. Figure 5.1 shows a modern FPGA from Altera which includes many IP hard cores mainly with the aim to achieve the highest possible bandwidths with the ultimate flexibility. Particularly the use of IP hard core transceivers allows significantly faster data rates. Rates that could never be achieved by the, in relation, slow FPGA logic gate arrays.

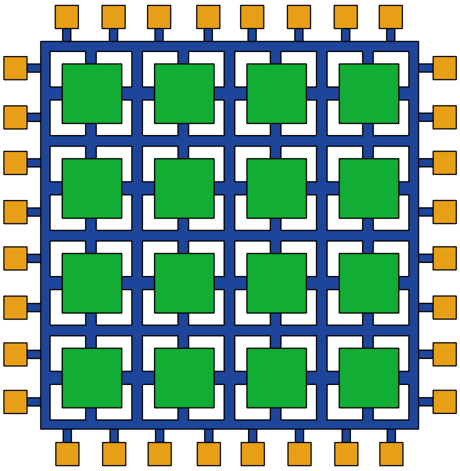


Figure 4.1: FPGA architecture: logic blocks in green, I/O pads in orange and programmable interconnections in blue.

Part III

Measurements

Chapter 5

Optical Links

5.1 Soft- and Hardware

The used soft- and hardware are assembled around Altera Stratix FPGAs development kit which build the core of every setup. Therefore, the used FPGAs are described in detail. In the following part, daughter boards and smaller components such as plugs and cables are described.

5.1.1 Altera Stratix V Development Kit

In all used setups, DSP version Stratix V GS Development Boards are used. Figure 5.2 shows such a board with the embedded inputs and outputs to evaluate various data transmission possibilities [63, 64]. The core component is a Stratix V FPGA.

Stratix V FPGA

The Stratix V from Altera is produced in a 28 nm structure size process. As shown in table 5.1 and the scheme in figure 5.1 many intellectual property (IP) hard cores designed for signal transmission, especially high bandwidth transceivers, are available.

In the following, selected components of the FPGA are explained.

Adaptive Logic Modules (ALM) Linked ALMs are used to implement any desired logic functions in the FPGA and make up by far the biggest part. The Stratix V FPGA has of 262400 adaptive logic modules (ALMs) which consist of 8 inputs, a look-up table (LU), two adders, as well as four registers [65].

ALM	262400
DSP blocks (18x18)	3926
M20K	2567
PCIe IP blocks	2
Transceivers	48 (up to 14.1 Gbit/s)

Table 5.1: Stratix V (5SGSMD5K2F40C2N) specifications.

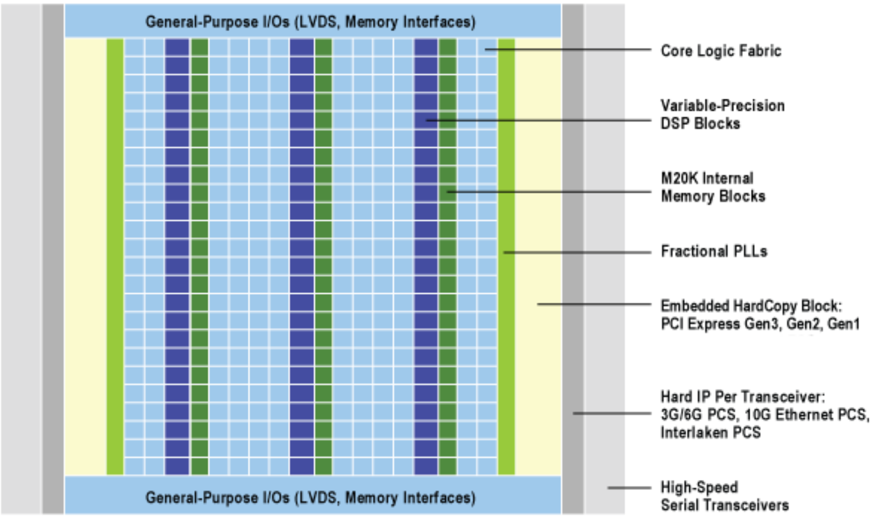

General-Purpose I/Os (LVDS, Memory Interfaces)
Core Logic Fabric
Variable-Precision DSP Blocks
M20K Internal Memory Blocks
Fractional PLLs
Embedded HardCopy Block: PCI Express Gen3, Gen2, Gen1
Hard IP Per Transceiver: 3G/6G PCS, 10G Ethernet PCS, Interlaken PCS
High-Speed Serial Transceivers
General-Purpose I/Os (LVDS, Memory Interfaces)

Figure 5.1: Altera Stratix V architecture [63].

Digital Signal Processing (DSP) Digital Signal Processing (DSP) blocks contain hard cores, spacial purpose logic circuits. The Stratix V device offers a huge variety of such cores ranging from encryption, video and audio handling, different signal modulators, fast Fourier transformations, hashing up to floating point addition and multiplication cores. Stratix V allows DSP blocks with configurable precision [66, 67].

Embedded Memory (M20K) Stratix V devices have embedded MLAB as well as M20K memory blocks inside the device which both can be accessed with up to 600 MHz. The MLAB is a general-purpose dual-port memory array of 640 bits optimized for FIFOs and shift registers for DSPs. M20K blocks are dedicated memory blocks which are much larger (20 Kbit) [68].

Stratix V Development Board

The different components of the Stratix V development board, shown in figure 5.2, are explained in the following.

SubMiniature version A connectors (SMA) SubMiniature version A (SMA) connectors are differential coaxial radio frequency connector pairs designed for frequencies up to 18 GHz with an internal 50 Ω impedance. On the board they can be used to output electrical high frequency signals. Due to their small distance to the transceivers inside the FPGA, excellent signal quality is achieved [69].

High Speed Mezzanine Card HSMC The Stratix V development board provides two ports for high-speed mezzanine cards (HSMC). They are designed for multi-gigahertz data transfers and provide up to 192 pins. These standardized ports allow to access the full input and output (I/O) capability of the FPGAs by achieving

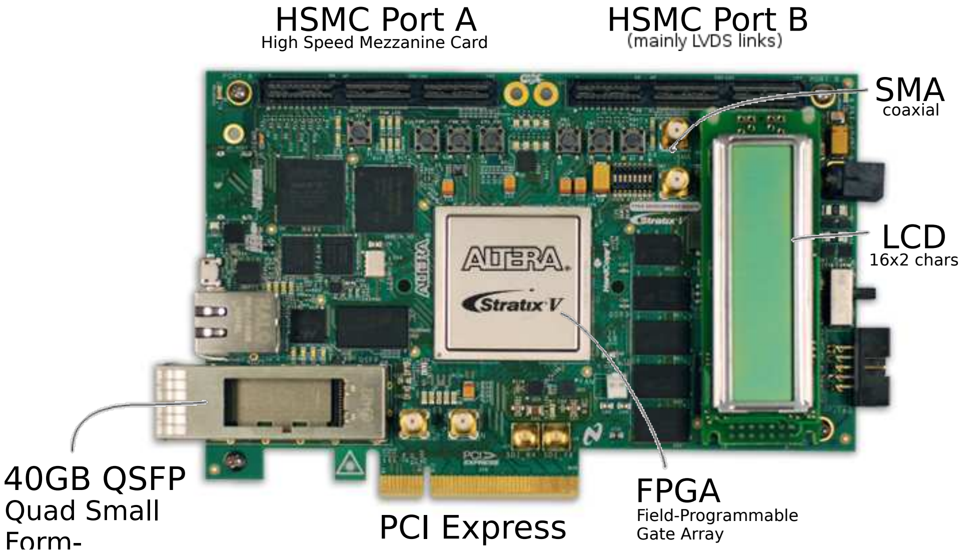


Figure 5.2: Altera Stratix V Development Board [63].

a maximum of flexibility at the same time [70]. Up to 8 multi-Gbit/s lines can be implemented per port.

Quad Small Form-factor Pluggable (QSFP) Quad Small Form-factor (QSFP) are hot-pluggable interfaces designed for high rate data transmission up to 4x28 Gbit/s [71]. The Stratix V development board supports data rates up to 4x10 Gbit/s.

PCIe, USB, Ethernat and JTAG The development board provides a backplane interface designed for PCIe slots and data transmission. Further communication and reprogramming of the FPGA are provided through an USB, an Ethernet and a JTAG interface. An USB blaster chip on the board support easy and fast reconfiguration of the device without any other hardware than a regular USB cable. In addition, it is possible to load FPGA programs via Ethernet into the on board flash memory and restore the setting after power up of the FPGA.

Oscillators The development board includes many programmable base oscillators [72, p. 2-23]. These oscillators are used to generate various clocks. Many of them are not available on the entire FPGA. Very often they can be accessed only within regions where the designated IP hard cores for usages the clocks are designed for are situated. For example, the 282.5 MHz clock labeled with REFCLK4_QL2_P generated with a Si5388 oscillator and designed for transceivers used in combination with the on-board QSFP interface are available only for these transceivers. This clock frequency cannot be used as input frequency for transceivers used with the HSMC interface.

Memory Different memory types and sizes are available on the board. A 1152 Mbyte DDR3 SDRAM with a 72 bit data bus, 4.5 Mbyte QDRII+ SRAM with a 18 bit data bus for writing and reading up to 550 MHz and a 72 Mbyte CIO RLDRAM II with a 18 bit data bus are available. Additional 512 Mbyte flash memory for non-volatile memory as mentioned already in the JTAG section 5.1.1 can be used.

Stratix V Transceivers

As described in section 5.1.1, the transceivers of the Stratix V FPGA are IP hard cores. This transceiver implementation provides a separation between the very fast serial part and the slower part where the data can be treated in parallel. Hard cores allow much faster rates than possible with logic gate arrays. Serial data rates up to multi-Gbit/s can thus be realized.

Each transceiver channel consists of a transmitter and a receiver part. Therefore, the channels are called full-duplex. The transmission, as well as the receiver part consists of a hard core physical media attachment (PMA) and a by-passable physical coding sublayer (PCS). The three different PCSs standard, 10G and PCIe Gen3 are available for each PMA. A detailed description can be found in section A.1.

5.1.2 SantaLuz Mezzanine Board

The SantaLuz Mezzanine card shown in figure 5.3 has been designed and produced at TU Dortmund. TU Dortmund developed this board in 2011 for a test bench of a fast data transmission line [73, p. 20]. The card is designed for data rates up to 8.5 Gbit/s [74] and it was shown that all eight channels are operational with a BER smaller than $3 \cdot 10^{-15}$ at 6.25 Gbit/s. All used SantaLuz boards are revision 0 boards [75].

The mezzanine card is designed for use as a Stratix daughter board attached to the HSMC ports to provide up to eight SFP+ plugs. Therefore, the SantaLuz card is equipped with female HSMC connectors. The SFP+ specification foresees ports to access an optional internal memory, this feature is not realized in the SantaLuz card. Although the plugs would allow a direct connection, high speed extension cables are needed due to the available space.

5.1.3 Plugs

The used Stratix V development boards and mezzanine daughter boards provide different types of plugs for data transmission. In the following extensively used electrical and optical plugs for SFP and there related QSFP cages are described.

SFP Plugs

Small form-factor pluggable (SFP) is a specification for transceivers developed by the multiple source agreement (MSA) group [76]. They build an interface to optical or copper networking cable, where their relative small size allows a high density of plugs on various devices. In 2006 an enhanced version of the SFP, the SFP+, was introduced, which supports data rates up to 10 Gbit/s [77].

In the following the different SFP plugs used are explained.

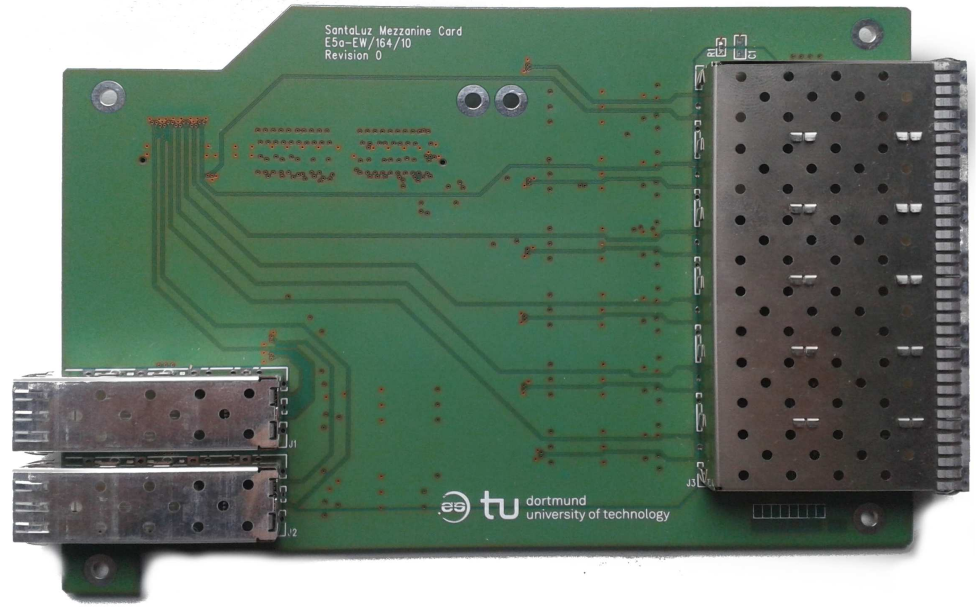


Figure 5.3: SantaLuz mezzanine board produced at TU Dortmund.

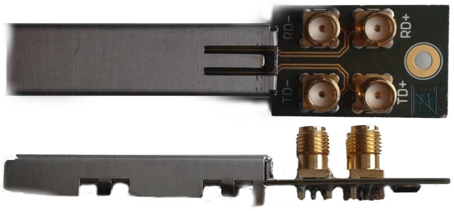


Figure 5.4: Top and side view of an electrical SFP plug.

Electrical SFP Plugs SFP to SMA adapters (SFP2SMA) from TrioFlex are used to access the fast signals at the SFP plugs electrically (see figure 5.4). The plugs consist of four PCB layers where the impedance matched high-speed lines of the transmitter and receiver channels are length matched. No DC block capacitors are present [78]. These plugs are mainly used to examine output signals on an oscilloscope.

Figure 5.5: Optical SFP plug overview. AFBR-57D7APZ

Optical SFP Plugs Optical duplex transceivers from Avago for multi-mode optical fibers are used in different setups (see figure 5.5). The Avago transceiver works at different rates with 850 nm wavelengths, where 2.125, 4.25 and 8.5 Gbit/s are ensured by the manufacturer. The rate selection is handled automatically by the device and although the SFP specification would provide rate selection, this is ignored. The transceiver is designed to work in a temperate range between $-10°$C to $85°$C with a supply voltage of 3.3V±10%. Both, the Rx and the Tx lines are AC coupled differential lines. The Rx lines need an external 100Ω termination which results in a voltage swing between 370 and 850 mV. The Tx lines are 100Ω terminated inside the module and accept a differential swing of 180 to 1200 mV.

The 850 nm laser pulses are generated in a vertical cavity surface emitting laser (VCSEL). The incoming differential high-speed logic signal is used to modulate the laser diode driver current. Provided the incoming data pattern is DC balanced, the transmitter laser driver circuit regulates the optical power at a constant level. Avago does not state over how many bits the signal should be DC balanced, but mention 8b/10b encoded (see 2.2.2) data as an example.

The receiver section includes a photodiode, an amplification and a quantization circuit.

Beside the basic functionality, the transceiver provides an internal memory which can be used to access real time temperature, supply voltage, bias laser current, laser average output power and received input power besides the standard SFP data. [79] As mentioned before, the used mezzanine daughter board does not support access to this transceiver internal memory.

5.1.4 Cables

A variety of different cables were used in different setups. On one hand there are commercial standard solutions either to connect components or for data transmission and on the other hand some cables were specially developed.

Commercial Cables in Setups

A HSMC cable from Samtec is used to connect daughter boards to Stratix V development board. The same type of cable with two female plugs could in principle be used to connect two development boards. With HSMC Altera developed an own standard, whose specifications are met by the Samtec 0.5 mm pitch Q Strip QSH QTH series high-speed board-to-board connectors. The used 228.5 mm long extension cables show a -7dB insertion loss at a frequency of 9.2 GHz [80].

Optical Cables for Data Transmission

As described in section 2.1.2, wave guides can be realized as single or multi-mode device. Single mode fibers allow the signal to propagate faster and farther than in multi-mode cables, which are in general much cheaper and easier to use.

Multimode Multi-mode duplex fiber patch cords (FPC) with lucent connectors (LC) on both sides are used in setups with SFP optical transceivers. The so-called low smoke zero halogen (LSZH) cable consists of typical fiber coated with a thermoplastic which does not contain any halogen and produces low smoke if burned. The LSZH feature is purely a safety aspect. This fiber shows an insertion loss below 0.14 dB and a return of the order of 37 dB. Cable lengths from 0.5m up to 50m are available, where for such multi-mode cables this represents the upper cable length limit at multi-Gbit/s.

Single Mode While the optical SFP transceivers are designed and operated with multi-mode optical fibers, the QSFP transceivers are mounted together with eight single mode fibers to one setup. Beside the QSFP plugs, no more connectors are needed, the fibers are directly glued on the lasers respectively photodiodes [81]. According to Molex, the used setups are designed to operate with BER below 10^{-18} per link in a range from 5 up to 10 Gbit/s. Due to the QSFP mechanical design with a cage, the devices are hot-pluggable. In addition, the setup with two QSFP uses only 0.78W per active cable [82].
As it can be seen from the facts stated above, these cable-plug systems are designed to work out of the box. Usually they are used in telecommunication applications. Point to point connections between devices, such as switches, servers, etc., with a very high bandwidth can be realized very easily up to several kilometers.

5.2 Firmware

Beside the physical hardware, the firmware with which FPGAs are loaded is a very crucial part of any actual setup including FPGAs. Therefore, the main concepts of the firmware developed are described in this chapter. In a first section the way how transmitter and receiver logic are synchronized is described, followed by an outline how a user bit error rate test (BERT) is implemented. In a third section, the firmware required by the Altera transceiver toolkit (ct. 5.2.3) and its possibilities are described.

The development of suitable FPGA firmware for link verification tests is a first part of this thesis. It will be followed by the performed measurements.

5.2.1 Data Transmission State Machine

Finite state machine (FSM) is a mathematical concept to design sequential logic circuits. A FSM maps through a transition function all states of a finite input alphabet depending on a set of finite internal states to an output function out of a finite set of an output alphabet [83, p. 11f.]. The machine is always in exactly one state, a transition to any other state is triggered by specified conditions. In the majority of cases, state machines are run synchronously, which means that a clock is present and both the check of the transition conditions and the transitions itself are performed at every cycle.
In the following the concept of the state machine which manages the data reception is outlined and illustrated in figure 5.6.

In- and Output The state machine gets the received deserialized data from the PMA together with the recovered clock as inputs. The output is composed of the bit error number from the BERT test, the total number of received words and the number of synchronization losses. Additional debug information, for example the last word patterns with errors are provided. Optionally, a flag for a bit slip in the PMA can be output.

Not Sync State The reset puts the state machine into this state. The input data is checked against a specified word alignment pattern (WAP).

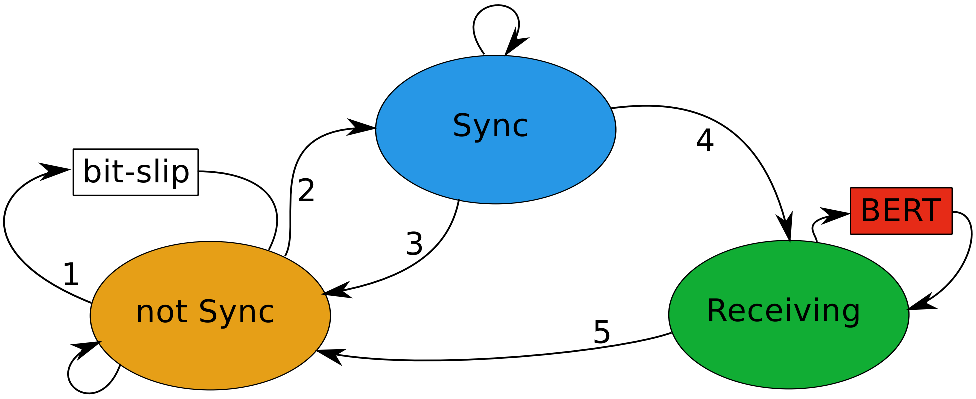


Figure 5.6: Transceiver state machine with the three different states: not sync (orange), sync (blue) and receiving (green). The conditions for the transitions 1 to 5 are adjusted to the needs of the current setup.

Transition 1) If the WAP is not found 400 (bitSlipNo) times one bit is slipped in the incoming data. This can be realized either with the bit slip function of the transceiver in the PMA or inside the receiving state machine through an additional register.

Transition 2) If the WAP is identified 300 (wapNo) times the state machine is set into the synchronized state.

Sync State In the synchronized state the input data is checked against WAP again. As long as this pattern is detected, the local state machine is synchronous to the data stream, but no valid data is sent by the transmitter.

Transition 3) In principle it is possible to loose the synchronization in the sync state and return to the non-synchronized state. A realization of this transition requires the ability to detect errors in the WAP pattern and distinguish it from valid data. Without any further protocol, this transition cannot be realized.

Transition 4) Whenever a data word different from the WAP is detected, it is assumed to be a valid data word. Hence, the state machine is put into its receiving state.

Receiving State In the receiving state any action with the data can be performed. In this setup, a bit error test is performed as described in section 5.2.2.

Transition 5) Based on the results of the BERT, different criteria lead to a loss of synchronization.

- More than a quarter (errRatio) bits of a word contains errors
- More than 256 (erWordNo) words with errors occur in a row
- More than a fixed numbers of errors have occurred. This condition is not used for BERT runs and exists only for debug purpose.

5.2.2 Bit Error Rate Tests (BERT)

The bit error rate is the number of received bits which are different than they were transmitted normalized to the total amount of received number of bits. This rate can be seen as the probability that a single bit state flips.
To perform such tests, data in a known pattern has to be sent on the transmitter side. To actually test for bit errors, the expected data pattern has to be known. The number of bits for which expected value and their actual received state differ can then be counted.
The crucial part is how the bit error test gets the information on how the received data pattern is supposed to look like. In the implemented versions, the same data generator is used on both transmitter and receiver sides, generating the next pattern based on the previous word.

Data Generator

The data generators implemented need the previous data pattern as input and calculate the new pattern. Various different patterns are used for testing purpose. The simplest pattern are fixed patterns, which do not depend on the previous words. They are used to establish a new connection, in this case word alignment patterns (WAP) are generated, or to test special patterns. By sending a fixed numbers of zeros followed by alternating one and zero patterns, the zero acceptance of a system can be tested.

A counter can be easily implemented to generate a deterministic pattern. This allows latency measurements of the system. It has to be mentioned that bit pattern representing numbers can contain a lot of zeros, a quite disadvantageous pattern. To avoid this disadvantage one can start the counter at a value where less zeros in a row occur or use gray counters. The problem however cannot be avoided in the long run.

Most of the time pseudo random number (PRN) generators (see 3.4.1) with periodicity maximizing taps are used. The length of the PRN determines the number of different possible states, the actual generated words can be of arbitrary length.

Test Pattern Quality

Pseudo random number generators, which generate the new bit state out of a 31 bit state, are commonly used for signal integrity tests. These pattern generators are often called PRBS31. Assuming an 8.0 Gbit/s data transmission with a PMA width of 80 bits, as it is used in the measurements with tuned transceivers in section 5.3.4, all of the $2^{31} = 2.1 \cdot 10^{9}$ possible words are transmitted once in 0.27 seconds. The single transmission of each possible state of this data is not enough to measure real bit error rates. To test also for random jitter and sinusoidal jitter, all patterns have to be transmitted many times because these effects behave independently of the bit pattern. Ransom Stephens stated in [84] that each pattern should be sent at least 20 times. If bit error rates are measured down to $< 10^{-16}$, each pattern is transmitted over 59'000 times. This is certainly enough to tests for all combinations of pattern and any kind of noise respectively jitter with reasonable statistics.

BERT Implementation

Figure 5.7 shows the implementation of the bit error test described above into the FPGA logic. The received data from the last cycle is sent to an instance of the same data generator, which generated the data on the transmitter side, to get the supposed data pattern. The received data is compared to the expected pattern.

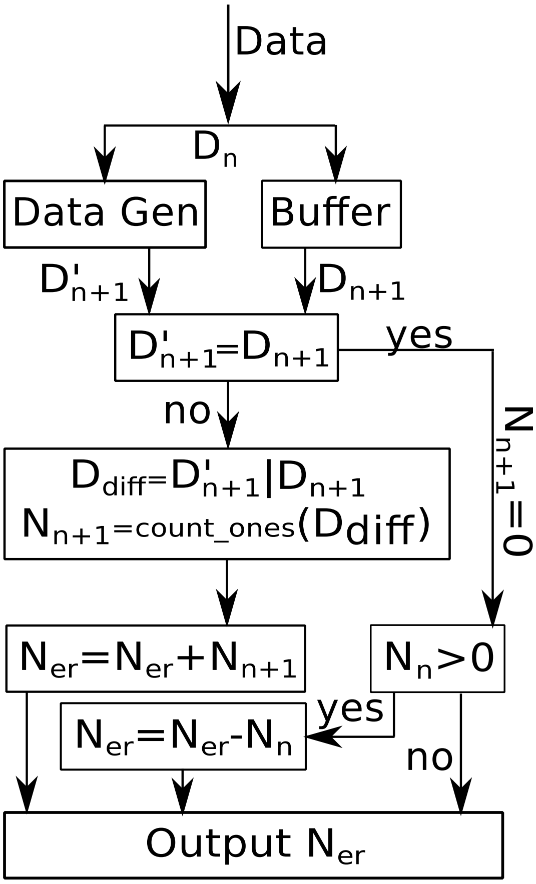


Figure 5.7: Simplified flow diagram of bit error rate test (BERT) with the most important counter N_{err}.

If the two patterns show any difference, their differing bits are stored in a register over which is summed in the following cycles. This process is quite timing critical and therefore split over multiple clock cycles. Once the result is calculated, the number of error bits is added to the overall error counter.

The expected pattern obtained from the data generator does not match the real pattern if the input pattern already contained an error. Hence, if only one error containing word occurs, it is counted twice. If a series of words contains errors, one word is counted too much. This problem is addressed by counting the error containing words in a row and always correct for the last one. Therefore, if the generated and received pattern match, but an error was indicated in the last word, this error was introduced artificially and has to be corrected for in the overall error counter. This procedure allows the error counter to display a too large value only for a couple of clock cycles. Too low values cannot occur.

5.2.3 Altera Receiver Toolkit

As described in section A.2 the Altera Quartus II software development kit (SDK) provides different useful programs for optimization of the logic designed. The Altera Transceiver Toolkit provides a GUI for dynamic reconfiguration of the transceivers on the FPGA. If the corresponding logic blocks are implemented in the user logic, an access via the system console interface is possible.

The Toolkit can be loaded with the same design as it is implemented in the FPGA. All physical transmitters and receivers are then auto detected by the software and can be controlled individually. To test channels, the individual transmitters and receivers can be grouped to a data link and then be controlled as data link.

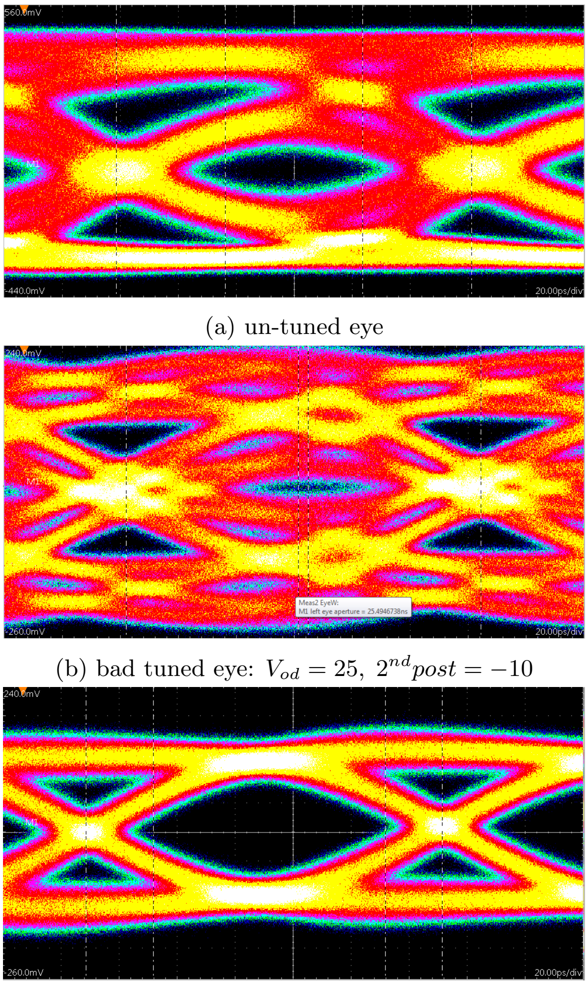
To access the functionality of the transceiver toolkit, a special firmware with the above described logic elements is used.

BERT Swap

The toolkit provides the possibility for a scan over different analog tuning settings. A BERT swap varies all analog parameters (see 5.1.1) inside specified ranges and performs bit error rate tests during a given time interval. This tool allows to find a suitable range for the analog tune values of a given setup. This tuning method is based on bit error rates, hence it needs a lot of time to improve the tuning of already well tuned systems.

Manual Eye Tuning

It is not possible to access the eye diagrams (see 2.3.1) of optical signals with the available hardware. To find on one hand a suitable range for the automatic tuning with the toolkit and on the other hand to understand the available transceiver parameters better, a manual tuning with an electrical signal was performed. The SantaLuz mezzanine card on the HSMA port equipped with an electrical SFP with the firmware described in 5.3.4 operating at 8.0 Gbit/s was connected to a digital serial analyzer from Tektronix to visually inspect the eye diagrams. Figures 5.8 shows a selection of eye diagrams with the corresponding tune values.



(a) un-tuned eye

(b) bad tuned eye: $V_{od} = 25,\ 2^{nd}post = -10$

(c) well tuned eye: $V_{od} = 26,\ 1^{st}post = 11,\ 2^{nd}post = 1,\ pre = -2$

Figure 5.8: Eye diagrams with different analog tuned transmitter routed over the SantaLuz mezzanine card and an electrical SFP plug.

For an electrical 8.0 Gbit/s setup with the SFP plugs on the mezzanine card, the settings $V_{od} = 26,\ 1^{st}post = 11,\ 2^{nd}post = 1,\ pre = -2$ work the best. The optimal tune values for an optical setup are different and stated in 5.1.3 in the measurement chapter.

EyeQ Measurements

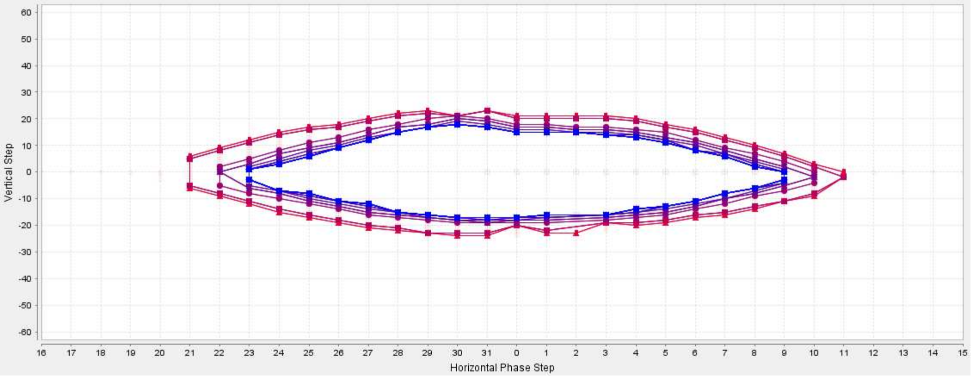


Figure 5.9: Example of a EyeQ diagram in the Altera Transceiver Toolkit. The different colored lines stay for different BER levels. The shown eye has an opening of 19/36 for BER of 10^{-12}. The horizontal time axis is divided into 32 units and the vertical threshold axis into 128 units.

In the Stratix V FPGA used, circuits denoted as EyeQ are available as IP hard cores. These units described in A.1.1 are accessed by the Altera Transceiver Toolkit to sample for an eye diagram. It is important to mention that eyes in this setup are not exactly eye diagrams as mentioned in 2.3.1 but are rather bathtub diagrams (see 2.3.2). The BER is tested at different points in the plane spanned by an offset in the CRC recovered clock and different thresholds for detecting a zero or a one respectively. In the eye figure, isolines with the same BER are drawn. These contours indicate the eye opening for this specific BER.
If such EyeQ diagrams are calculated for different runs in an analog tuning scan, the eye opening of the EyeQ diagrams can be used to find the best suited analog tune values. Figure 5.9 shows an example of such an EyeQ diagram. It is to be mentioned that the circuits only provide eyes down to BER of the order of 10^{-12}. This is due to the fact that more precise measurements would need too much time and limits the exact tuning if much smaller BER are aimed for.

5.3 Measurements

This chapter describes all performed measurements with their corresponding setups using the hardware described in chapter 5.1 and firmware based on ideas which are described in chapter 5.2. Various bit error rate tests (BERTs) have been performed to test the high speed optical links of the MuPix readout chain. In a first part, the statistical approach used for measurements without any error is introduced. Followed by single and multi-channel measurements with optical SFP links. In a fourth chapter, measurement with optical QSFP links are presented.

5.3.1 BER Upper Limit and Error Calculations

The performed measurements of BER are classical counting experiments. There are two ways to understand these measurements. On one hand, one can consider N binary measurements which check whether a bit error occurs or not. The sum of errors is denoted k and the probability of a single bit to be an error is p. The two quantities k and N are actually measured. The best estimator for p is then $\hat{p} = \frac{k}{N}$. The variance of such a measurement is given by

$$Var_{binom}[X] = Np(1-p) = k(1 - \frac{k}{N}) \tag{5.1}$$

for large N and relative small k the binomial distribution converges to a Poisson distribution and the variance is approximated by k.

On the other hand, one can consider that one single measurement of the quantity $k = p \cdot N$ is performed and assume that the rare process of bit errors is Poisson distributed. In this case the parameter s of the distribution described in 5.5 is given by $\hat{s} = E[X] = k$. The variance can be written as

$$Var_{Pois}[X] = s = \hat{s} = k = \hat{p} \cdot N \tag{5.2}$$

where $\hat{p}$ is the best estimator for p.

In both perspectives described above, the best estimator for BER p, with its error is given by:

$$BER = p = \frac{k}{N} \pm \frac{\sqrt{k}}{N} \tag{5.3}$$

Where the stated error is valid for Poisson distributions or if p is small. BERs are here considered small.

According to [85] there is no standard prescription for setting upper limits for rare signals. In a Bayesian approach the probability density function (pdf) of a unknown parameter is given by equation 5.4

$$\pi(s|n) = \frac{f(n|s)| \cdot \pi(s)}{\int_0^\infty f(n|s)\pi(s)ds} \tag{5.4}$$

where s is the given signal rate, n the number of observed events, $f(n|s)$ the conditional probability to observe n events by given signal rate s and $\pi(s)$ is the prior pdf.

If very rare events, such as decays or bit errors, are observed, the signal is Poisson distributed:

$$f(n|s) = \frac{s^n e^{-s}}{n!} \tag{5.5}$$

Bayes [86] and Laplace [87] stated that a non-informative prior for any parameter must be flat.

For a given confidence interval $[\alpha, \beta]$ equation 5.6 holds

$$1 - \alpha - \beta = \int_\alpha^\beta \pi(s|n)ds \tag{5.6}$$

Regarding the special case of a Poisson distributed random variable and the case in which $n_{obs} = 0$, the confidence interval can be transformed into a upper limit at a given confidence level (CL) β:

$$1 - \beta = \int_0^\beta \pi(s|n)ds = \sum_{n'=0}^{n'=n_{obs}} \frac{s^{n'} e^{-s}}{n'!} = e^{-s} \tag{5.7}$$

If no error is measured and the measurement is considered background free, the shown Bayesian approach is equivalent to the Frequentist. Given that in N independent Poisson distributed events no errors $n_{error} = 0$ were measured and requiring a C.L. of $\beta = 95\%$ a upper limit of

$$BER \leq \frac{log(-\beta)}{N}(95\% C.L.) \approx \frac{2.996}{N}(95\% C.L.) \tag{5.8}$$

can be stated.

5.3.2 Optical SFP Links

The performance of data transmission with multi-mode optical fibers in a setup with Stratix V development boards and SantaLuz mezzanine cards are intensively studied with the following base setup:

- Stratix V Development Board FPGA (see 5.1.1)
- IP hard cores PMA (see A.1.1)
- Optional IP hard core PCS (see A.1.2)
- State machine and BERT as described in 5.2.1 and 5.2.2 where the bit slip option of the transceiver PMA is used for word alignment.
- Different data generators which will be explained for each measurement
- SantaLuz mezzanine card (see 5.1.2) connected via Samtec HSMC cables (see 5.1.4)
- Mainly Optical SFP modules (see 5.1.3)

An overview of such a base setup is shown in figure 5.10.

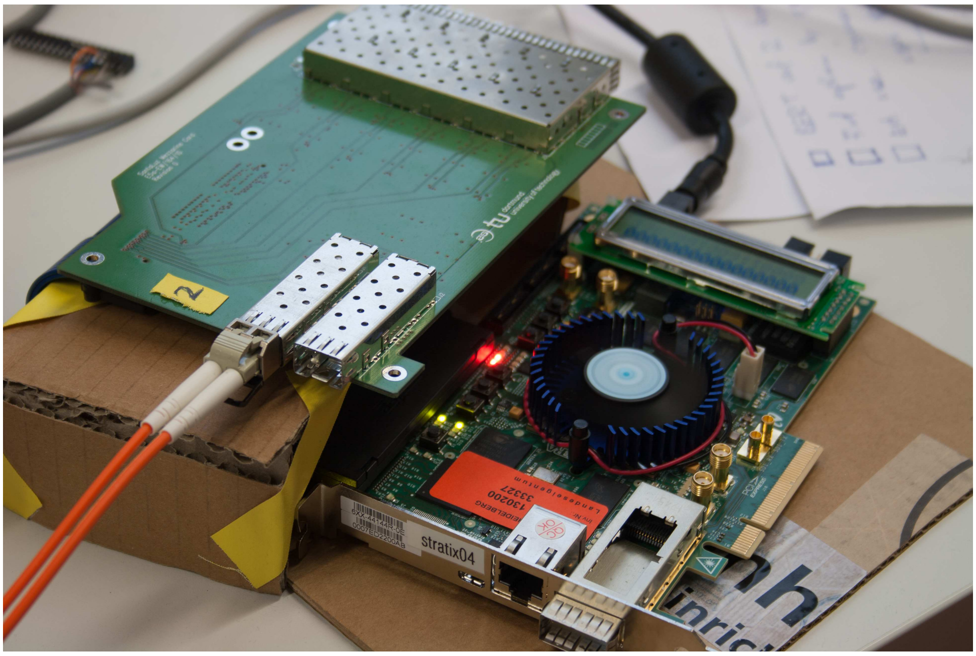
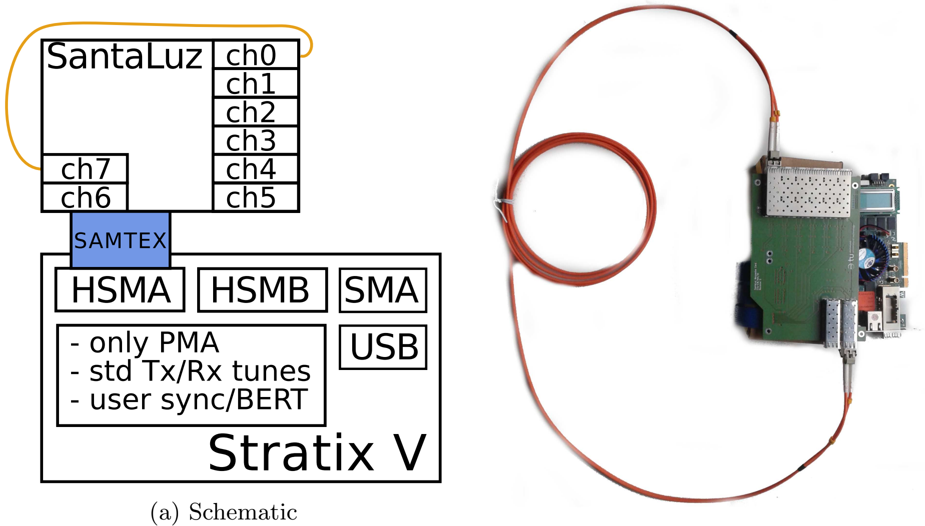


Figure 5.10: Setup to test optical SFP links. A SantaLuz mezzanine card (labeled with 2) is connected with blue Samtec high speed cables on HSMC port A of a Stratic V development board. The LCD display, a fan right above the FPGA, Ethernet and QSFP plugs are visible. The mezzanine board is equipped with one optical SFP transceiver which is connected with a duplex multi-mode optical fiber in orange.

In a first step, the signal quality and the performance of the single components is studied with standard settings. Particularly, no analog tuning of the transceivers in the PMA is performed. In a second step, the overall system is optimized to achieve the best possible data transmission quality at a given rate. In this particular step, the analog tune values of the PMA are in focus. Furthermore, different data encoding schemes are tested under the aspect of transmission quality.

5.3.3 Single Channel SFP



(a) Schematic

(b) Picture

Figure 5.11: Setup for single channel measurements with a single Stratix V board.

Figure 5.11 shows the base setup for single channel measurements with only one Stratix V FPGA board. The hard core PMA with non-duplex transceivers and configured for a 64 bit interface width are the only used IP hard cores in the FPGA. The PMA is not user tuned, which means that the standard values are set by the Quartus II software package. The SantaLuz mezzanine card is connected with a Samtec cable on the HSMA port, which provides an interface to all eight duplex channels of the board. Once the FPGA is programmed, it runs independent of any USB connection. The on-bard electrical SMA outputs are used only for debugging purposes.
Different firmwares for data rates of 2.4, 4.8, 5.0, 6.4 and 8.0 Gbit/s were used.

A data generator in the user logic provides the following data pattern:

Pseudo Random Numbers (PRN) are used to simulate data and to test for every possible data pattern within the length of the PRN generator length. Linear shift feedback registers (see 3.4), with taps chosen to achieve the maximal periodicity, with 8, 16 and 32 bit are used.

Zero Pattern which consist of a fixed numbers of zeros in a row followed by alternating ones and zeros, e.g. “000000101010”

Counter which allows to measure bit error rates and latency at the same time. It should be noted that a counter can produce quite a lot of zeros in a row.

In addition to these different patterns which can be selected during run-time, different optical cable lengths between 0.5 m and 50 m are used to investigate the effect of this variable system.

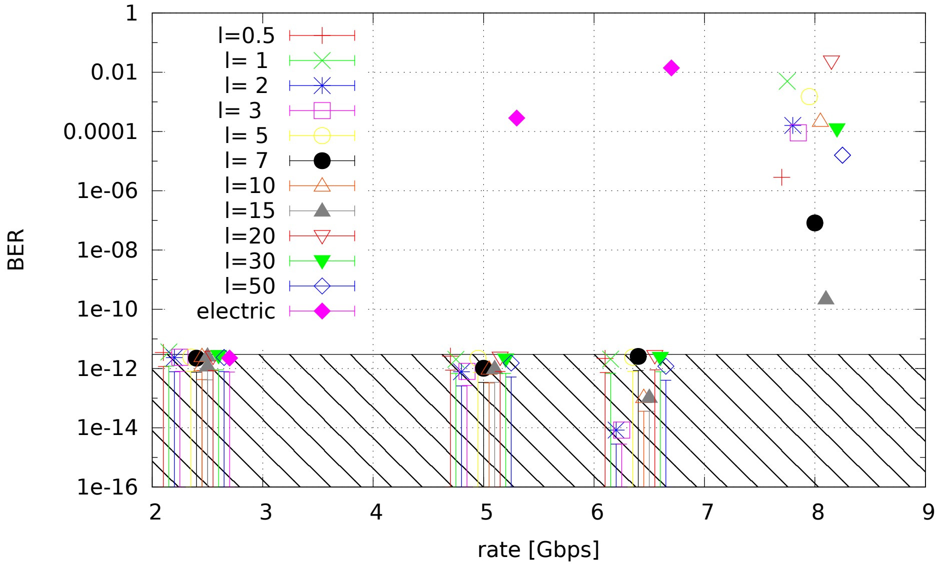


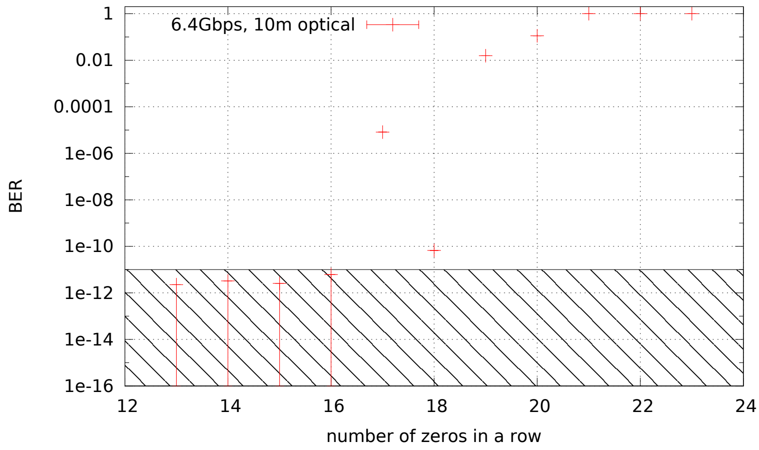
Figure 5.12: Bit Error rate (BER) dependency on the data rate. The different cable lengths are shifted around the used rates of 2.4, 5, 6.4 and 8 Gbit/s for better viability. The cable length l is given in meters.

BER Dependency on Rate Figure 5.12 shows the BER dependency on a given rate, where the BER is measured with a 31 bit PRN generator. In this measurement, the BER is measured only down to a level of $3{\cdot}10^{-12}$. All values below this BER value in the hatched box are upper limits. In addition to testing the optical links, one run was performed with electrical SFP plugs as described in section 5.1.3, which show an error rate higher than optical. It has to be mentioned, that electrical measurements are performed only without DC balance. No significant difference due to the cable length is observable for the optical cables. With BER $> 10^{-8}$ at 8 Gbit/s no data transmission is possible with the given setup.

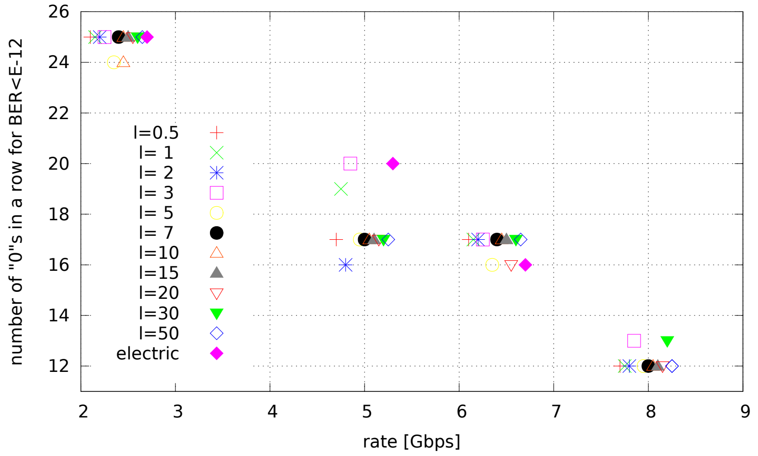
In addition to the data shown in figure 5.12, long time measurements of certain setups are performed to get a much lower upper limit for the BER. One long run with a 3 m and one with 50 m optical link at 6.4 Gbit/s were performed and no errors were detected at all. In the two runs a total of $1.31 \cdot 10^{15}$ bits in the 3 m run and $1.001 \cdot 10^{16}$ bits in the 50 m run were transmitted, which results in:

$$BER_{3m} < 2.2 \cdot 10^{-15} (95\% C.L.) \tag{5.9}$$

$$BER_{50m} < 2.9 \cdot 10^{-16} (95\% C.L.) \tag{5.10}$$



(a) BER dependent on the test pattern used. In this case zeros in a row followed by alternating zero-one pairs filled up to 64bit words.



(b) Number of maximal zeros followed by a alternating zero-one pattern filling up 64bit words at a given rate. The data is shifted around the actual used rates of 2.4, 5, 6.4 and 8 Gbit/s to achieve a better visibility.

Figure 5.13: Tolerance of zeros in a row dependency on rate. The cable lengths l are given in meters.

Zeros in a row To test the zero acceptance of the setup, the number of zeros in a row followed by an alternating one-zero pattern filling up 64 bit words is increased and the BER of this pattern is recorded. Figure 5.13a shows such a measurement for a fixed rate at 6.4 Gbit/s and a 10 m optical cable. The BER is measured down to 10^{-12}. With increasing number of zeros the BER reaches at a certain point a level above the arbitrary fixed value of 10^{-12}. The biggest number of zeros in a row before the BER increases over the threshold of 10^{-12} is detected for every setup and summarized in figure 5.13b.

Latency Beside the BER at different rates and different cable lengths, a latency measurement was always performed as well. This allows to fit the measured latency to a model which consists of three parts. A latency part due to the wave propagation in the optical line which scales with cable length and a part which comes from the FPGA logic as well as the optical transmitter and receiver which consists of a fixed number of clock cycles and hence scales with the data rate. The observed latency lat is described by

$$lat = lat_{cable} \cdot L + lat_{logic} \cdot \frac{1}{R} + lat_{traces} \tag{5.11}$$

where $lat_{cable} = (5.00 \pm 0.05)\frac{ns}{m}$ is the latency per cable length unit, L the cable length, $lat_{logic} = (246 \pm 4)\frac{ns}{Gbit/pma_{width}}$ the latency per cycle, R the parallel data rate and $lat_{traces} = (5.2 \pm 1.7)ns$ the latency from the connection between FPGA and optical transmitter as well as optical transmitter and receiver latency. It can be seen that the signal propagation inside the multi-mode fibers is $v_{signal} \approx 2/3 \cdot c$. The second part of equation 5.11, the latency introduced by the logic, depends on the implemented logic.

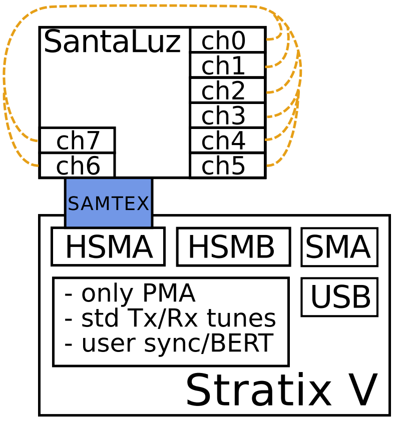


Figure 5.14: Setup scheme used to measure the performance of the different cages. Only one optical fiber is used at the time.

Different Cages The performance of the different cages on the SantaLuz mezzanine board were compared with an 8.0 Gbit/s data stream generated with a 8 bit data generator. The use of 8.0 Gbit/s rate ensures to get measurable BER and not only upper limits. Only in the presence of an error rate the different cage's performance can be compared. The setup used is shown in 5.14. In this measurement, only one channel is connected to another at a time, but both directions are tested simultaneously. Therefore, duplex transceivers were implemented in the PMA. The only non-tested permutations are the loop backs because the available cables are assembled in a way that does not support such a setup. Both measurement types, receiving and transmitting, are performed with the transmitting respectively receiving from cage A0.

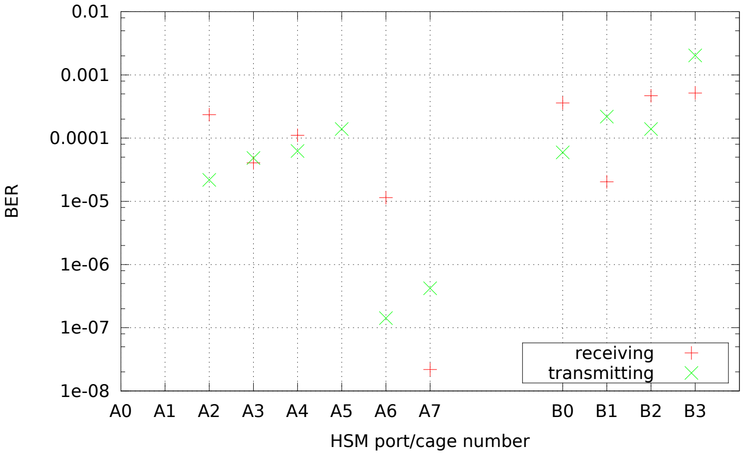
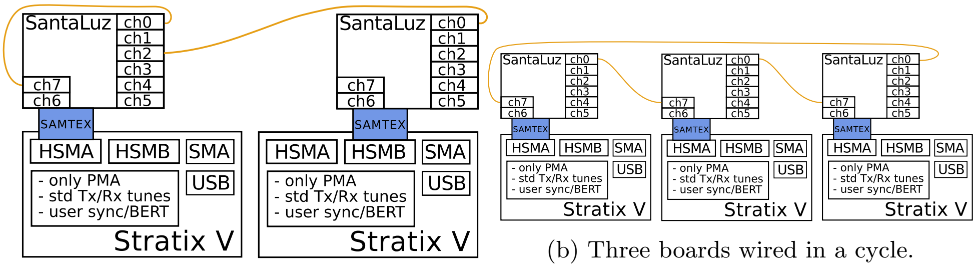


Figure 5.15: Performance of the different cages on the SantaLuz mezzanine card. For all receiving measurements channel 0 on HSMA port was used as transmitter channel. For all transmitting channels the same A0 receiver was used as receiving channel.

In figure 5.15 the performance of the different point to point connections on the same mezzanine boards are shown. Where the mezzanine boards were connected to the HSMA as well as to the HSMB port.
The significantly better performance of the channel pair 0-7 was also observed in many other measurements.

As shown in figure 5.15, no significant difference between the two ports HSMA and HSMB exist in the accessible cages.

Complementary to the base setup shown in figure 5.11, a setup with two FPGA boards is used. In this particular setup, on one board the HSMB plug is used instead of the HSMA as before. The HSMB port allows access only to the SFP cages 0 to 3 on the SantaLuz mezzanine card because the others are not connected to a transceiver pin in the FPGA. This setup is shown in figure 5.16.



(a) Two boards, with HSMB in use.

(b) Three boards wired in a cycle.

Figure 5.16: Setup for single channel measurements with two and three Stratix V board.

Beside the performance test of the different cages a long time run for BER was performed with 6.4 Gbit/s rate with the multi-board setup shown in figure 5.16. In a total of $1.03 \cdot 10^{16}$ transmitted bits not a single error was detected, which leads to a $BER < 2.9 \cdot 10^{-16} (95\% C.L.)$ at 6.4 Gbit/s with setup 5.16a.

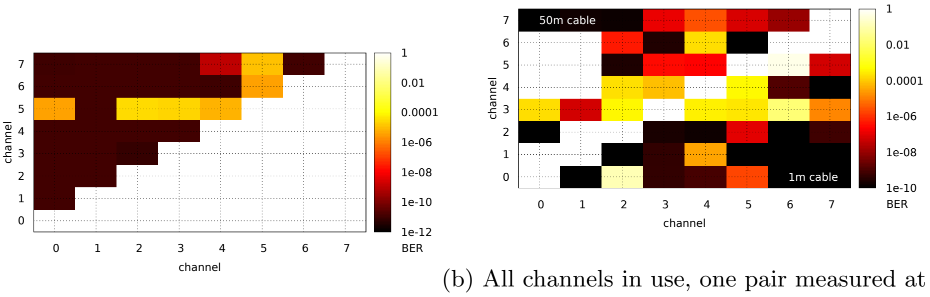
5.3.4 Multi-Channel SFP

The main feature of the SantaLuz mezzanine card is to provide up to eight channels on one card, which is attached with a HSMC cable to the Stratix V development board either via the HSMA or the HSMB port. Besides the performance of single channel optical links, the performance of the board equipped with multiple optical transceivers is investigated. To this end, the different cages are used at the same time as well as all cages at the same time.

A setup of two Stratix V boards is used to measure the BER behavior if multiple channels are used. Therefore one development board with a SantaLuz card attached to the HSMA port is used to transmit 6.4 Gbit/s random data generated out of a 8 bit PRN (see 3.4.1) generator on multiple lines. All data streams of this setup used the same random number generator with the same seed. This has to be taken into account when the following data from these measurements are interpreted. In a later setup described in 5.3.4 this disadvantage was removed and different seeds were used for each channel.

Extensive tests were performed with permutations using between one and up to five lines at the same time. Figure 5.17 shows the performance if two receivers are operated at the same time and the data are transmitted from a second FPGA with 6.4 Gbit/s. Some channel combinations introduce bit errors. Figure 5.17b shows the behavior of the cards if all channels are in use, but only two channels are used to measure their bit error rate. The results of the complete measurements with up to five parallel channels are listed in table A.1 in section Appendix.

The measurements show that some channels behave much worse than others with error rates up to 10^{-6}. With this setup, the measurements with four and five channels operating in parallel show a better performance than the measurements where only two or three channels are used. This observation is not completely understood. It can be speculated that this behavior is related to the fact that the data generators produce all the same pseudo random numbers in phase. The bottom line of the measurements is that the setups cannot be operated with multiple transceivers without any further analog tuning or DC balancing.

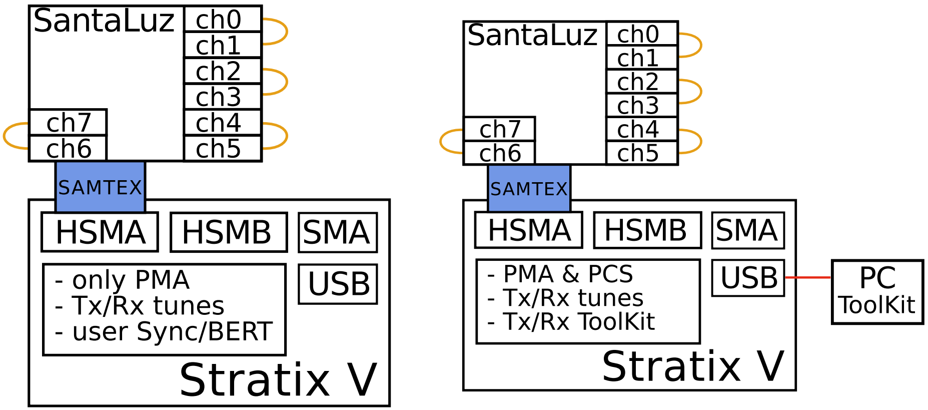


(a) BER with two active receivers. The data is sent from a second FPGA.

(b) All channels in use, one pair measured at the time with 1 m (under diagonal) respectively 50 m cables (upper diagonal). No measurements were performed for the white area in the upper part.

Figure 5.17: BER measurements at 6.4 Gbit/s to quantify cross talk between the channels.

Multi-Channel Analog Tuned BER Tests



(a) analog tuned BERT

(b) Toolkit setup

Figure 5.18: Setup for analog tuned transceivers.

To test the system as a whole, a setup as shown in figure 5.18a is used with different parameters. The range for optimal analog tune values for the transmitter in the FPGA PMA were obtained from scans with the Altera transceiver toolkit described in 5.2.3. These values are optimized further during the set up to decrease BER in all channels. To have full control over all elements, the PMA is the only used IP hard core, no hard core PCS is used. As before, the synchronization and bit error test are performed in user logic according to the concepts described in section 5.2.1 and 5.2.2. The data generator produces pseudo random numbers based on sequences of different length. For each channel, a different seed is chosen to ensure that the bit streams on all channels are different. The length of the PRN sequence has a direct influence on the maximal number of identical bits in a row as well as on how DC un-balanced the data stream is.

channel	V_{od}	$1^{st}pre$	$2^{nd}pre$	*post*	*DC Gain*	Lin. Eq.
toolkit BER						
0	18	4	0	-3	4	15
1	18	4	0	-3	0	1
2	18	4	0	-3	0	12
3	18	4	-2	0	0	0
4	18	4	0	-2	0	15
5	18	4	0	-2	4	15
6	18	4	0	-3	0	15
7	18	4	0	-3	4	15
toolkit EyeQ						
all	18	6	0	-2	0	5
manual perturbation						
0	18	6	0	-3	-	5
1	18	6	0	-3	-	5
2	18	6	0	-3	-	5
3	18	6	0	-1	-	5
4	18	6	0	-2	-	5
5	18	6	0	-2	-	5
6	18	6	0	-3	-	5
7	18	6	0	-3	-	5

Table 5.2: Analog SFP tune values for 8.0 Gbit/s setup. The EyeQ values provide an eye opening of 8/38.

To check the reliability of the user logic, tests with a loopback inside the IP hard cores, with a loopback on the HSMA port as well as an user error injection were performed. If errors are injected into the system, the error counters count them exactly one time. The two loop-back measurements show no error over their entire run time of several 10^{17} transmitted bits.

Tuning Procedure As already mentioned above, the tuning was performed using the Altera transceiver Toolkit as described in section 5.2.3. The tuning of the following measurements is performed with the PRBS31 random number generator, which is available as hard core and access able via the system console (see section A.2) by the toolkit. Once the parameters are roughly known from a manual electrical run, a scan over different values can be performed automatically by the software. The best values can be found by pure BERT or with the help of the opening of the eye, measured by the EyeQ circuit as described in section A.1.1. Both methods are applied to find the best possible settings by scanning first all channels in parallel with the same settings. This parallel feature is not supported by the software and all the scans have to be started by hand. Therefore small offsets are very likely. After such a parallel scan, the channels which still produced errors are scanned individually for better settings. These settings are then implemented into the standard firmware, which is used for all the BER measurements. In a third iteration, the tuning of channels which still produce the most errors is slightly varied to find even better settings.
The parameters finally used for the setup described in section 5.3.4 are the following:

loopback	transmitted bits	BER limit (95%C.L.)
SFP internal	$1.2 \cdot 10^{17}$	$< 1.7 \cdot 10^{-17}$
SFP external	$1.3 \cdot 10^{17}$	$< 1.9 \cdot 10^{-17}$
QSFP internal	$5.1 \cdot 10^{17}$	$< \cdot 10^{-17}$

Table 5.3: 8.0 Gbit/s SFP and 11.3 Gbit/s QSFP loopback measurments.

Loop Back Measurements Measurements without any physical attachment are performed to review the performance of the user logic as well as the one of the Stratix V. One setup with a loopback inside the IP hard core as well as one with a loopback on the HSMA port are performed. In a loopback the receiver channels are attached directly to the transmitter of the same channel. For the IP hard core internal loopback the same tune values as for all the other tests are used. The HSMC loopback measurement with a loopback header could only be performed without any tuning. This can be explained by the fact that in the setup with optical transmission, the optical receiver acts as an electrical amplifier (see 5.1.3).

This measurements together with the correct detection of all induced errors show that the used setup is reliable for BERT down to an order of 10^{-17} if all channels are used at 8.0 Gbit/s.

PRN length dependency The BER of 8.0 Gbit/s data transmission with the above described setup, where the transceiver are tuned as described, depends highly on the used data patterns. Pseudo random generators are a good tool to test for all possible data patterns. At data rates around 8.0 Gbit/s even all permutations of 31 bit PRN are tested once in less than a minute. Without any further modification, the synchronization conditions described in chapter 5.2.1 are not met for PRN which are generated out of more than 14 bits. Simulations of the running disparity behavior of the different patterns used are shown in figure 5.19.
If a running disparity controller is added to the logic, the transmission quality increases as is shown in table 5.4. In addition to this controller, tests with an additional scrambler are also performed.

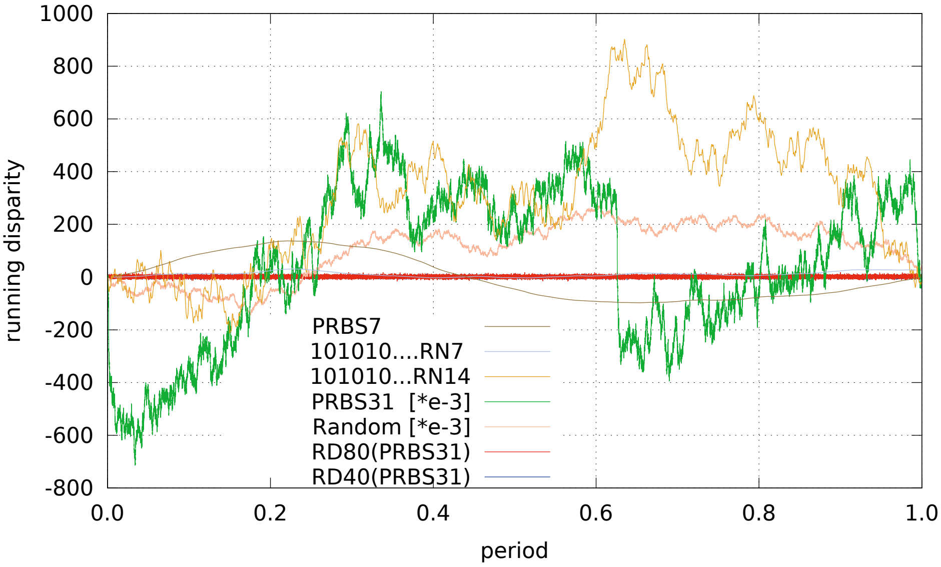


Figure 5.19: Running disparity of different data patterns. The period length for each pattern generated out of l bits is $2^l - 1$. The absence of the only zero pattern in random number generators (see 3.4) based on LFSR introduces a systematical disparity offset, which is corrected for in this figure. Zero-one alternating pattern followed by 7 bit generated RN run error free without any further disparity control at 8.0 Gbit/s, for 14 bit generated patterns synchronization still holds according to the definitions in section 5.2. For patterns with larger l disparity control (RD) are required.

In figure 5.19 a much better random number generator with 80 bits (RD80) is added to the simulation. It can be seen, that it performs better than the PRBS31 generators regarding running disparity .

Running Disparity This option controls the disparity of each 80 bit word sent. As described in 5.3.4, the parity of a data word is the inbalance between ones and zeros. The running disparity control uses the highest bit of each transmitted data word to indicate whether the data of this particular word is inverted or not. Depending on the running disparity, the continuously summed up disparities, the control decides whether the actual word is inverted or not. This method adds an overhead of 1 bit and some additional latency in the transmitting logic. Summing over all bits in FPGA logic is a time critical process. The gain is a limitation of the running disparity to values between ± (wordlength-1). In the used firmware, the full PMA width of 80 bits is utilized in RD80 and two parallel controllers each with 40 bits are implemented in RD40. Longer words decrease the speed of the whole user logic, which allows an easier adaption to higher serial data transmissions.

Scrambler As described in section 2.2.3 a scrambler ensures that the probability of getting a very unfavorable bit pattern in random distributed data is decreased strongly. The implemented scrambler is a very primitive version whose state is calculated out of two taps from the incoming data. In such a scheme, the scrambler is aligned after the number of data length bits and does not need any additional synchronization. The use of a scrambler can multiply errors due to bit flips if they occur at the bit, which is used as tap for the scrambler. An error out of such a bit flip would stay in the scrambler state for the number of word lengths and count as one error for each word.

BER Values In table 5.4 longterm measurements with tuned transceiverss at 8.0 Gbit/s with different cables, with and without scrambler and disparity controller are summarized. Beside this measurements with 31 bit generated random number generators, different other patterns are evaluated. Patterns composed of alternating ones and zeros, e.g. “10101010”, followed by a dedicated number of bits with a random number generator are used. Transmission without any errors ($BER < 6.0 \cdot 10^{-17}$) are possible with random number generators up to 7 bits, synchronization is achieved with generators up to 14 bits. For all pattern generated with longer generators, disparity control is needed.

ch	RD80, no Sc l=50m	RD40, no Sc l=1m	RD80, Sc l=50m
0	$< 1.5 \cdot 10^{-13}$	$(7.24 \pm 5.12) \cdot 10^{-16}$	$< 1.4 \cdot 10^{-15}$
1	$< 1.5 \cdot 10^{-13}$	$(4.56 \pm 0.05) \cdot 10^{-13}$	$< 1.4 \cdot 10^{-15}$
2	$(1.03 \pm 0.07) \cdot 10^{-11}$	$(4.61 \pm 0.04) \cdot 10^{-12}$	$(1.83 \pm 0.008) \cdot 10^{-12}$
3	$(8.32 \pm 2.08) \cdot 10^{-13}$	$(4.00 \pm 0.12) \cdot 10^{-13}$	$(4.20 \pm 0.14) \cdot 10^{-13}$
4	$< 1.5 \cdot 10^{-13}$	$(1.45 \pm 0.02) \cdot 10^{-12}$	$< 1.4 \cdot 10^{-15}$
5	$< 1.5 \cdot 10^{-13}$	$(6.72 \pm 0.16) \cdot 10^{-13}$	$(2.93 \pm 1.12) \cdot 10^{-15}$
6	$< 1.5 \cdot 10^{-13}$	$(2.55 \pm 1.04) \cdot 10^{-15}$	$< 1.4 \cdot 10^{-15}$
7	$(3.95 \pm 0.45) \cdot 10^{-12}$	$(1.70 \pm 0.85) \cdot 10^{-15}$	$(6.08 \pm 0.06) \cdot 10^{-12}$
total	$(1.85 \pm 0.11) \cdot 10^{-12}$	$(4.56 \pm 0.05) \cdot 10^{-13}$	$(1.041 \pm 0.008) \cdot 10^{-12}$

Table 5.4: BER test with 31 bit PRN patterns at 8.0 Gbit/s with analog tuned transceivers. Measurements with and without scrambler (SC). Measurements without disparity controller RD are not possible because no synchronization according to the conditions in 5.2.2 is achieved. All stated upper limits are $95\% C.L.$ values.

In addition to the measurements shown in table 5.4 with 8.0 Gbit/s, the rate to with the transceivers are tuned, measurements at 6.4 Gbit/s with the same analog tune values were performed. The results of these measurements are summarized in table 5.5. Setups with running disparity control, but non-tuned transceivers, stay synchronized according to the requirements mentioned in section 5.2. Nevertheless they show BER of the order 10^{-4}, hence they can not be used with this settings.

5.3.5 Optical QSFP Links

The QSFP cages on the Stratix V development board are used to setup a card-to-card connection between two boards. This is realized with the Molex transceiver and optical fiber system as described in section 5.1.3. The used firmware corresponds to the one used for all multi-channel test in section 2.1.2 for the SFP channels. A 80 bit width hard core PMA is fed with pseudo random numbers generated out of a 31 bit state. The running disparity is in addition controlled on a ±40 bit base. Table 5.6 summarizes the performed measurements. The 9.3 Gbps transmissions are performed without any analog tuning. However the 11.3 Gbps transceivers are tuned according to the values obtained in section 5.3.5. Synchronization is only achieved with a 80 bit disparity control.

ch	no RD l=1m	RD80 l=1m	RD80 l=50m	RD40 l=1m
0	$(6.98 \pm 0.01) \cdot 10^{-3}$	$< 7.8 \cdot 10^{-16}$	$< 4.9 \cdot 10^{-16}$	$< 7.8 \cdot 10^{-16}$
1	$(8.81 \pm 0.01) \cdot 10^{-3}$	$< 7.8 \cdot 10^{-16}$	$< 4.9 \cdot 10^{-16}$	$< 7.8 \cdot 10^{-16}$
2	$(7.79 \pm 0.03) \cdot 10^{-11}$	$< 7.8 \cdot 10^{-16}$	$< 4.9 \cdot 10^{-16}$	$< 7.8 \cdot 10^{-16}$
3	$(2.23 \pm 0.10) \cdot 10^{-2}$	$< 7.8 \cdot 10^{-16}$	$< 4.9 \cdot 10^{-16}$	$< 7.8 \cdot 10^{-16}$
4	$(9.57 \pm 0.01) \cdot 10^{-10}$	$< 7.8 \cdot 10^{-16}$	$< 4.9 \cdot 10^{-16}$	$< 7.8 \cdot 10^{-16}$
5	$(9.72 \pm 0.10) \cdot 10^{-12}$	$< 7.8 \cdot 10^{-16}$	$< 4.9 \cdot 10^{-16}$	$< 7.8 \cdot 10^{-16}$
6	$(3.81 \pm 0.01) \cdot 10^{-10}$	$< 7.8 \cdot 10^{-16}$	$< 4.9 \cdot 10^{-16}$	$< 7.8 \cdot 10^{-16}$
7	$(1.61 \pm 0.01) \cdot 10^{-10}$	$< 7.8 \cdot 10^{-16}$	$< 4.9 \cdot 10^{-16}$	$< 7.8 \cdot 10^{-16}$
total	$(4.85 \pm 0.01) \cdot 10^{-3}$	$< 9.8 \cdot 10^{-17}$	$< 6.1 \cdot 10^{-17}$	$< 9.8 \cdot 10^{-17}$

Table 5.5: BER test with 31bit PRN pattern at 6.4 Gbit/s with transceivers which are analog tuned for 8.0 Gbit/s. Upper limits are 95%$C.L.$ values.

rate	cable length	BER
9.3	3	$< 9.29 \cdot 10^{-17}$
9.3	20	$< 9.56 \cdot 10^{-17}$
11.3	20	$(3.29 \pm 1.04) \cdot 10^{-16}$
11.3*	20 (turned)	$(3.71 \pm 0.31) \cdot 10^{-15}$
11.3^{+}	20	$< 9.45 \cdot 10^{-17}$

Table 5.6: QSFP BER measurements with 80 bit disparity control. *Replaced bad receiving FPGA and turned cable. $^{+}$Used two new FPGAs.

The operation at 11.3 Gbit/s with tuned transceivers shows errors in one of the eight, four in each direction, channels for some FPGAs. The performance of this channel is $BER_{S04} = (3.32 \pm 0.74) \cdot 10^{-15}$, $BER_{S11} = (2.97 \pm 0.24) \cdot 10^{-14}$ and $BER_{S11} < 9.45 \cdot 10^{-17}$, depending on the Stratix V board used. Different cables and setups have been used to identify the source of the problem. Out of the 4 used FPGAs, 2 show errors in the same receiving channel. Once a problematic channel is identified, the others can in principle still be operated with a BER below $< 4.8 \cdot 10^{-17}$ (95 % C.L.).

QSFP Analog Tuning

Analog tune values are gained out of a swap test with the Altera transceiver toolkit in the same way as for the optical SFP links. Several tune value ranges with excellent BER are listed in table 5.7. For the 11.3 Gbit/s tests the values at the table bottom are used.

5.4 Discussion

Transmission quality checks have been performed with two different device families. On one hand, optical SFP transceivers on SantaLuz mezzanine boards with up to 8 channels are tested. On the other hand, on-board QSFP cages with cable assemblies which bundle four fibers have been used.

V_{od}	$1^{st}pre$	$2^{nd}pre$	*post*	*DC Gain*
12 to 13	2	-1	0	0 to 4
13 to 16	3	0	0	0 to 4
17 to 18	4 to 7	0	0	0,1,2,4
19	3 to 7	-5 to -3	0	0 to 4
20 to 25	4 to	0	0	0 to 4
26	3 to 5	-9 to -7	0	0 to 4
26	3 to 5	-9 to -7	0	0 to 4
27	8	0	0	0

Table 5.7: Analog tune values for 11.3 Gbit/s QSFP links. Different parameter regions show good BER behavior. Only a small selection of randomly chosen value ranges are shown here. The values at the bottom are actually used.

Plug	Gbit/s	tuned	no RD, no SC	RD, no SC	no RD, SC	RD, SC
SFP	6.4	no	part. sync $\approx 5 \cdot 10^{-4}$	part. sync $\approx 3 \cdot 10^{-5}$	part. sync	part. sync
SFP	6.4	yes	$\approx 6 \cdot 10^{-4}$	$< 10^{-16}$	$\approx 6 \cdot 10^{-4}$	-
SFP	8.0	no	not sync	not sync	-	-
SFP	8.0	yes	not sync	$\approx 10^{-12}$	not sync	$\approx 10^{-12}$
QSFP	9.3	no	not sync	$< 10^{-16}$	-	-
QSFP	11.3	no	not sync	not sync	-	-
QSFP	11.3	yes	not sync	$< 10^{-16}$*	-	-

Table 5.8: Overview over BER with different settings. RD stands for disparity control, SC for scrambler. *Depends on the used hardware. The best possible result is quoted.

5.4.1 Summary

Table 5.8 summaries the overall performance of the tested systems with all available channels operating at the same time.

It is possible to operate single SFP links with Stratix V development boards and SantaLuz mezzanine cards error free (BER $< 3 \cdot 10^{-16}$) at 8 Gbit/s with cables up to 50m length. The parallel operation of all eight available channels on one daughter board introduce errors with a rate of 10^{-12} errors/bit. All channels can be operated simultaneously with 6.4 Gbit/s with fiber lengths up to 50m, if the disparity of 80 bit words is controlled. Whether the gained speed in 8 Gbit/s operation balances the errors depends strongly on the application. To achieve the here stated rates, the transceivers have to be analog tuned. Suited analog tune values can be evaluated with the Altera transceiver toolkit.

QSFP links can be operated without any tuning at 9.3 Gbit/s with 80 bit disparity controlled data. If the channels are tuned, 11.3 Gbit/s can be achieved with an error rate below $3 \cdot 10^{-16}$. All errors have been measured in the same single channel and depend strongly on the used FPGA card. This bad behavior of the same single channel occurred also with different hardware, different FPGA boards and cables. Although, only some FPGA cards introduces bit errors. If the channel with bit errors is excluded and only three of the four fibers are used, the links can be operated error free as well. How to proceed with these slightly worse channels is application dependent.

The data rates achieved in this thesis are sufficient to realize the Mu3e readout chain as proposed and shown in figure 1.8 for phase II. Only these very high data rates resulting in a high total bandwidth of the DAQ make the Mu3e trigger-less detector and GPU reconstruction design possible.

5.4.2 Crucial Points

Based on the measurements in this thesis as well as the experience with the hardware the following points are considered crucial.

Cable Length

A BER dependency on cable length has never been observed. Multi-mode fibers between 0.5 m and 50 m and single mode fibers between 1 m and 20 m have been used. It has to be mentioned, that no cables longer than specifications allow have ever been used. Particularly, multi-mode fibers of the given quality (OM2) are specified exactly up to the used 50 m.

Cross Talk in SantaLuz Boards

The SantaLuz boards are the bottlenecks in the current SFP setup. The Samtec cables are specified for higher data rates than used, only the mezzanine board has never been used at these rates. It seems that the bit errors are introduced by cross talk on the board. This effect can be minimized by an analog tuning of the transceivers. Unfortunately, this affects only the transmitting part. The optical SFP receivers output signal can not be modified. Cross talk induced by the receiving lines on the boards can thus not be addressed at the moment.

Improvement Due to Disparity Controller and Scrambler

Different measurements with a simple scrambler and disparity controllers are performed. Disparity was controlled within 40 and 80 bits. Neither 8.0 Gbit/s nor 6.4 Gbit/s links work properly if all channels are used without a disparity controller. At 8.0 Gbit/s the state machines synchronization criteria are not met, at 6.4 Gbit/s synchronization is achieved but bit errors occur with a rate of 10^{-3} errors/bit which makes the links unusable.

Simulations with the used pseudo random number generators show that the running disparity can be very unbalanced. Results are shown in figure 5.19. The absolute running disparity is not of interest, the inbalance over a given time range is the critical value. How inbalanced signals over which time ranges are acceptable, is unknown at the moment. But a disparity control on a 80 bit base seems to be sufficient for all kinds of data patterns. Patterns without disparity control are possible for alternating zero-one pattern followed by 7 bit generated random numbers. Synchronization without disparity control is achieved up to an alternating pattern followed by 14 bit generated random number. For rates of 8.0 Gbit/s a slight decrease in BER can be observed if the running disparity is controlled within 40 bits. Nevertheless, the result does not improve significantly with a factor of ~ 5 at 8 Gbit/s. Only changes of orders of magnitudes are considered significant in this scope.

As described in 5.2.2, every possible bit pattern of 32 bits is tested hundreds of times. An additional scrambler which would only decrease the probability of disadvantageous pattern does not improve the signal quality.

For every transmission with the used optical links a running disparity control over maximum of 80 bits is required. This is also true for QSFP links.

Physical Coding Sublayer (PCS)

Different PCS have been used in the development process of the firmwares. The available PCS are very useful if a existing protocol is implemented. Standard logic performs equally well for parts where the data can be handled parallel and offers more flexibility. PCS are very device specific, whereas user logic is more easily portable. The hard core PCS free some logic cells.

BERs Relevance for the Experiments

The performed measurements have aimed at a bit error rate below $3 \cdot 10^{-16}$. This order of magnitude is mainly motivated by the time needed for the measurements. Single lines at 8.0 Gbit/s have to be operated for two weeks. Assuming a total data rate of the whole detector of 1 Tbit/s this rate limit translates to approximately 1 error per hour, which is orders of magnitudes less than for example noise hits.

Chapter 6

Readout Chain Firmware Components

Besides the fast optical transmission between different components of the Mu3e readout chain, some selected firmware components are investigated. There are on one hand the implementation of the event time ordering in the front-end FPGAs for the MuPix sub-detector, and on the other hand the potential coordinate transformation on FPGA cards inside the filter farm computers. These two readout chain aspects are described and tested in the following.

6.1 Front-End FPGA

Front-end FPGAs receive hit data from MuPix pixel sensors. The hits are encoded in 24 bits, 8 bit each for row and column information as well as an 8 bit Gray code timestamp. The timestamps have to be decoded into binary numbers in order to determine the relative timing of hits. In addition to the hit data, the FPGA has to know the current timestamp of the whole system. This can be realized either with an own Gray counter or by receiving timestamps from the sensors. A combination of these two concepts would allow some cross checks whether the counters are still aligned.

The timestamp sorting is realized in a buffer. The goal is to output time ordered hit information packages. How the ordering is realized best depends on the structure of the received hits and the hardware used. Due to space requirements and costs, it is foreseen to use relatively small FPGAs without large memory blocks.

In the following, the structure of the hit data is discussed. Followed by two different concepts how the data could be buffered and time ordered. This is completed by a comparison of the two previously introduced concepts.

6.1.1 Hit Data Structure

When a pixel in the MuPix sensor is hit, its address is stored together with a corresponding timestamps in a column wise queue for further readout. The readout process (see A.4) loads one hit per column into a readout buffer. These hits are transmitted one after another via the serial interface. New hits occurring during this time, are added to the queue. Once the readout buffer is empty, maximal one hit is load again from each column. This readout scheme results in disordered timestamps. Many hits per sensor introduce hits with long delayed timestamps.

The received hit data have always a timestamp of the current or older timestamps. The maximal delay is denoted D. After each readout of a bucket of data, the offset between the timestamp of the received hits and the one at the start of a new bucket is called T. If the buffer is read every D frames, T is in the range of $-D$ to $+D$. The number of hits per frame has an influence on the number of hits which has to be stored. The average number is denoted $\hat{N}$ and the maximal expected number N_{max}.

6.1.2 Concept I

For each possible time offset T between $-D$ and $+D$ memory for N_{max} hits, column and row corresponding to 16 bits, is reserved. The timestamp is encoded in the storage address. The required memory size is given by:

$$B_{RAM} = N_{max} \cdot 2D \cdot 16bits \tag{6.1}$$

6.1.3 Concept II

The hit data, row and column, are stored in a ring buffer with a data depth of 16 bits and a length of $D \cdot \hat{N}$. In addition one register per possible time offset T with a depth of the length of the previously mentioned hit data memory exists. In the register corresponding to the timestamp offset of a hit, the bit number which corresponds to the address the data has in the hit memory is set to 1. In the reading process, the registers are read descending in the register and the hit data of all memory addresses, which are marked with a 1 in the current register, are read (see figure 6.1b). This results in time sorted outputs. The memory required for this concept is given by:

$$B_{RAM} = \hat{N} \cdot D \cdot 16bits \tag{6.2}$$

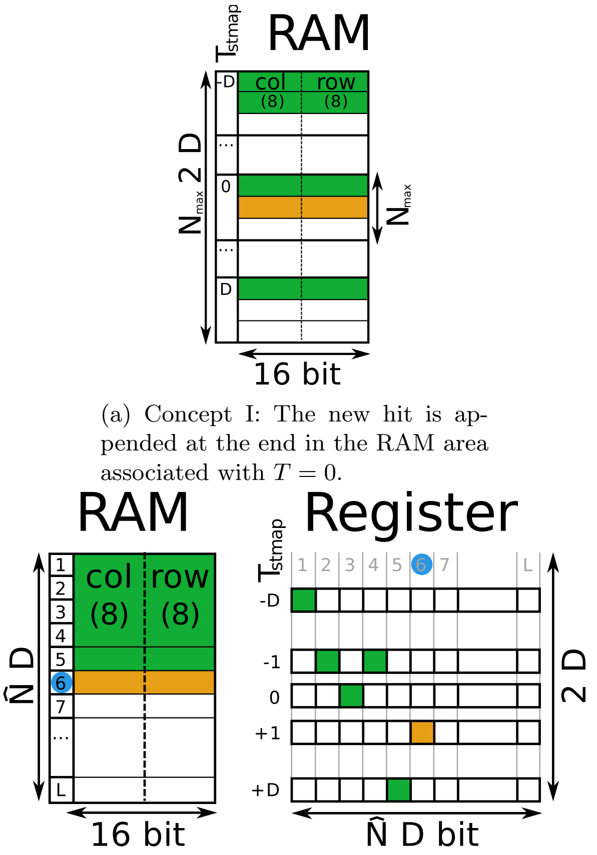
$$B_{REG} = \hat{N} \cdot D \cdot 2D \cdot bits \tag{6.3}$$

6.1.4 Comparison

Simulations have shown that $\hat{N}$ is maximal of the order 5 and up to $3 \cdot 1.25$ Gbit/s LVDS data links cause bandwidth limitations the possible read out of hit data (24bits) per 50 ns frame of 2.1 hits/frame. Nevertheless, the maximal number of hits with the same timestamp can be much larger, because they can be distributed over D readouts. A value of $N_{max} = 30$ seems to be on the conservative side. Maximal delays D of 30 are expected, as described in 1.3.2. Under these assumptions, concept II is favored with 11'400 bits over 28'800 bits of concept I. Concept II has been implemented into a Stratix V FPGA and has been simulated with ModelSim.

6.2 Coordinate Transformation on FPGAs

As described in section 1.3 the pixel hit data will be transformed on the FPGA located inside every PC in the filter farm from pixel address and hit coordinates into a global coordinate system. The conversion from integer pixel addresses and hit coordinates into single precision floating point numbers on FPGAs is discussed



(a) Concept I: The new hit is appended at the end in the RAM area associated with $T = 0$.

(b) Concept II: The new hit is appended into the ram, in the $T = 0$ register the bit corresponding to the RAM address is set.

Figure 6.1: Front-End time ordering concepts I and II. The orange hit with a timestamp offset $T = 0$ is buffered in the corresponding concept. Previously buffered data is represented in green.

in this chapter. In the end, the FPGAs will handle data from all sub-detectors, but for this first proof-of-principle tests the focus was on the pixel sub-detector. On one hand, because this is the detector which will be operated from phase Ia on and on the other hand, because the data structure of this sub-detector is known already the best.

6.2.1 Coordinate Systems

As described in section 1.3.1, the pixel sub-detector hits are represented in 14 sensor address bits and up to 8 bits for row and column coordinates. No timestamp conversion is considered at the moment, this is probably performed already in the front-end FPGAs. It is likely, that the frame timestamp will be transmitted in the event header and is therefore separated from the actual hit data. The global experiment coordinates consists of a right-handed Cartesian coordinate system, with positive z-axis parallel to the downstream beam line. The x-axis is parallel to the experimental hall floor and y points towards the roof. Distances are stored in single precision IEEE floating point format and given in mm.

6.2.2 The Transformation

The transformation of a hit in sensor s in the pixel $\vec{k} = (col, row)^T$ is given by

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \vec{x} = \vec{O}_s + T_s\vec{k} = \begin{pmatrix} o_s^x \\ o_s^y \\ o_s^z \end{pmatrix} + \begin{pmatrix} x_s^c & x_s^r \\ y_s^c & y_s^r \\ z_s^c & z_s^r \end{pmatrix} \begin{pmatrix} col \\ row \end{pmatrix} \tag{6.4}$$

where $\vec{O}_s$ is the origin vector of sensor s and T_s the sensors orientation. This transformation is implemented in 4 steps in hard core DSP blocks on the FPGA.

Integer to Float 6 cycles are needed to transform the integer col and row coordinates to single precision floating point numbers (F_{max} = 515 MHz [66]).

Multiplication x, y and z coordinates from col and row contribution are calculated in parallel. This 6 parallel multiplications, corresponding to the first part of the matrix multiplication in 6.4, can be implemented in 5, 6, 10 or 11 cycles (F_{max} = 445 MHz, for 11 cycle implementation [66]).

Addition 1 The contributions for each Cartesian coordinate from col and rows are added up. This step finishes the matrix multiplication and can be implemented selectively between 7 and 14 cycles (F_{max} = 495 MHz, for 14 cycle implementation [66]).

Addition 2 In a last step, the sensors origin vector is added in parallel for each coordinate component. This addition can be implemented again with between 7 and 14 cycles (F_{max} = 495 MHz [66]).

A transformation needs between 26 and 46 clock cycles. The more cycles are spent, the faster clock rates are possible, but more logic cells are consumed.

6.2.3 The Implementation

The above described transformation is implemented with VHDL and its simulation is shown in 6.2. To achieve the maximal speed, the shown version is implemented fully pipelined with 46 cycles. The sensors origin vectors and their orientation are stored in ROM (see 3.2.4). For each sensor, addressed with 14 bits according to the proposed address scheme, 9 times a 32 bit single precision floating point number is stored. Namely, the origin vector with its three coordinates as well as row and column direction with another 3 coordinates each. This results in a ROM size of $2^{14} \cdot 9 \cdot 32$ bits $= 2^{14} \cdot 288$ bits $= 4'718'592$ bits.

Multiplications as well as additions provide an overflow and underflow detection witch rises an error flag at the output.

The above described implementation uses $3'227$ ALMs (1.8 %), $4'740$ registers (1.3 %), 6 DSP (0.4 %) hard cores and $4'718'592$ (11.4 %) bits block memory per transformation component. The values in brackets represent the fraction used in a Stratix V FPGA with 230'000 ALMs (see 5.1).

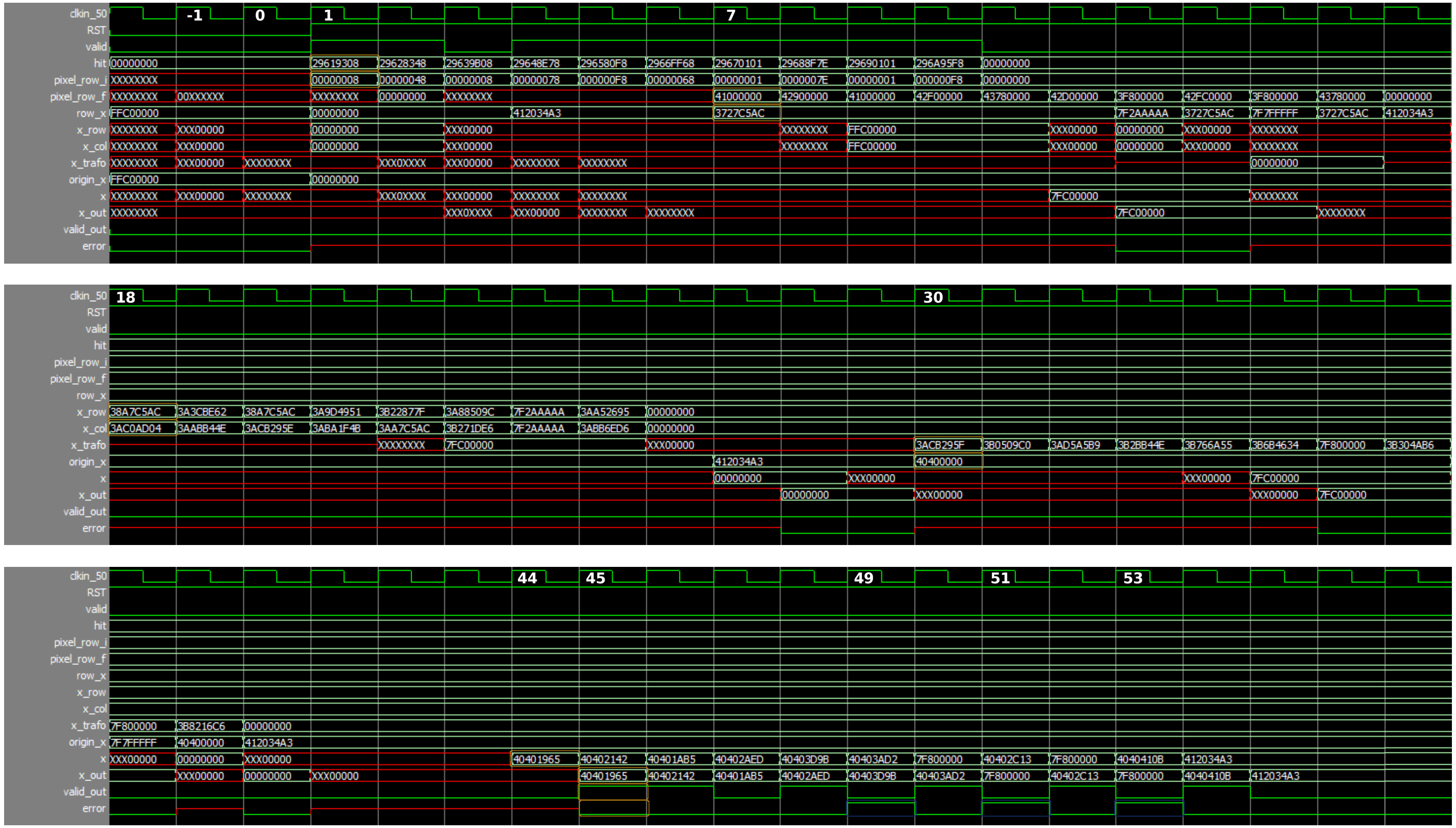


Figure 6.2: ModelSim simulation of a coordinate transformation on a FPGA. After the release of the active low reset RST in cycle 0, hit data are loaded. In Cycle 1 the address, col and row information are separated. pixel_row_i shows that the eighth row of the sensor was hit. This integer is converted to an single float representation in 6 cycles. In cycle 7, the row coordinate pixel_row_f in single float representation 0x4100000, which represents the number 8, as well as the transformation coefficient for row-x row_x 0x372C5AC which corresponds to $1 \cdot 10^{-5}$ are ready. The multiplication takes 11 cycles and is ready in cycle 18 as x_row with a hexadecimal value of 0x38A7C5AC, corresponding to $1 \cdot 10^{-5}$. All other row, col and coordinate combination are ready at the same time. The col and row contribution to each coordinate component are summed up in the next 14 cycles. There product x_trafo is ready in cycle 30 and has the value 0x3ACB295F, corresponding to $1.55 \cdot 10^{-3}$. In the same cycle the sensors origin coordinate are loaded and added to the previously calculated coordinate contributions in another 14 cycles. In cycle 44 the final coordinates are ready and output in the next cycle together with a valid and error flag. If the multiplication or the two additions result in an overflow or underflow, simulated in hit data loaded in cycles 5, 7 and 9 the error flags in the rorresponding output cycles 49, 51 and 53 indicate this.

ROM address depth	1	2	3	4	6	8	12	24
15 (2 x MuPix)	448	360	381					
14 (MuPix)	**412**	445	367	344	285			
13 (1/2 MuPix)	464	**431**		424	363	307		
12 (1/4 MuPix)	469			**417**		319	292	328

Table 6.1: F_{max} for different coordinate transformation implementations in a Stratix V FPGA with different number of parallel implementations and different ROM depths. 14 bits corespondents to the MuPix sub-detector address space. The configurations in bold could handle the whole detector.

6.2.4 Performance

The maximal clocking frequency of the different components are given in brackets in the list 6.2.2. It has to be mentioned that for F_{max} of adders and multipliers specifications only for Stratix IV and lower are available [88]. The ROM used to store the transformation coefficients is specified up to 600 MHz [69, p. 14]. In principle operation speeds up to 445 MHz should be possible. Table 6.1 shows TimeQuest results of different implementations. Multiple coordinate transformation components have been implemented in parallel as well as components with smaller ROM. This is motivated by the fact, that if the filter farm PCs get the raw data on different lines, the lines probably deliver data only from a sub-part of the detector. To include the other sub-detectors a version with twice the address space was used as well. The number of parallel implemented components is limited by the available internal memory to 7.

6.2.5 Conclusion

The implementation and simulation of coordinate transformation components in an Stratix V FPGA has proven the concept and confirmed the speed stated in the Altera manuals. The same order of speed is still achievable if multiple components are implemented in parallel. The limitations of how many parallel components can be used, comes from the available FPGA internal memory that is supposed to store all the transformation coefficients. External memory in Stratix V development boards are specified almost with the same speed as the internal. Hence, this memory could in principle be used as well.

The possible clock rates of approximately 400 MHz is sufficient to handle 30 bit long hit data transmitted with 12 Gbit/s. This transmitting speed is the very upper limit that will ever be used in the Mu3e experiment. One coordinate transformation per transceiver is sufficient for all phases of the Mu3e experiment.

Chapter 7

LVDS on Kapton FlexPrints

The Mu3e pixel sub-detector readout chain includes LVDS links on Kapton flex prints besides fast optical links. As a part of this work, a proof of principle is provided that such cables can be produced in-house with freshly acquired tools. This chapter introduces the used materials and tools, explains the used test pattern as well as the minimal achieved structure sizes. At the end, concrete ideas for subsequent works are provided.

7.1 Kapton

Kapton is a polyimide film developed by DuPoint with a unique combination of electrical, thermal, chemical and mechanical properties [89]. As described in section 1.2, Kapton is used in the detector's active region as mechanical support structure as well as substrate for aluminum traces. Kapton is used mainly to minimize the amount of material inside the detector, and consequently to reduce multiple scattering [11].

Regarding thermal properties, no melting point is known for Kapton Type 100HN films [90, table 2]. Furthermore, it remains stable in a temperature range between −269 °C to 400 °C [91]. Regarding electrical properties, Kapton's dielectric constants varies linearly with the relative humidity between 3.0 and 3.8 and drops in a temperature range between 0 °C and 200 °C nearly linearly from 3.5 down to 2.9 [90, figure 11, 15].

In the present thesis 25 µm Kapton Type 100HN metallized with aluminum of 50 nm respectively 25 µm is used.

7.2 Low-Voltage Differential Signaling (LVDS)

Low-voltage differential signaling (LVDS) is the specification of a serial differential signal scheme. LVDS transmitting lines consist of a pair of wires. The transceiver injects a constant current of 3.5 mA. The current direction indicates the logic state of the transition line. The impedance of the two lines is 100 Ω. To minimize reflections, they are terminated with an identical resistance. Therefore, the differential voltage, which is detected on the receiver side, is $V = R \cdot I = 350\,mV$. The polarity of this voltage indicates the logic state.

The implementation of two traces close to each other that carry an equal current in opposite directions minimizes the emission of electromagnetic noise. The induced

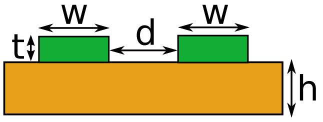


Figure 7.1: LVDS cable profile used for impedance calculations in equation 7.1.

noise is reduced as well because it is approximately the same in both wires, which is canceled by the differential approach. Furthermore, LVDS consumes very little power and the power supply requirements can easily be met due to the constant current flow.

The impedance of a LVDS wire pair is calculated as shown in equation 7.1 [28] for fractions of w/h between 0.1 and 3.0. Figure 7.1 shows the corresponding LVDS line and ϵ_r represents the substrate's dielectric constant.

$$Z_d = \frac{174}{\sqrt{\epsilon_r + 1.41}} ln\left(\frac{5.98 \cdot h}{(0.8 \cdot w + t)}\right)\left(1 - 0.48exp\left(-0.96\frac{d}{h}\right)\right) \tag{7.1}$$

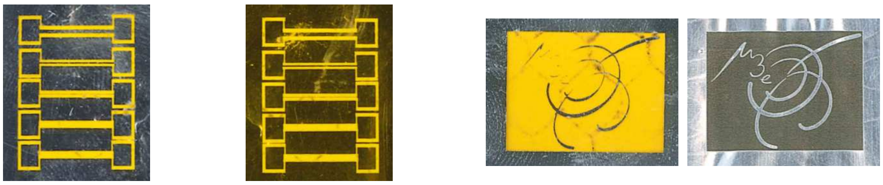
For the 25 µm Kapton film coated with 50 nm respectively 25 µm aluminum and Kapton's dielectric constant of $\epsilon_r = 3.5$ two LVDS lines have to be $w_{7\mu m} = 60\mu m$ respectively $w_{25\mu m} = 50\mu m$ width for d between 0.1 mm and 1 mm. These values are required by the fact that LVDS foresees a 100 Ω differential impedance.

7.3 Laser Platform

The Kapton aluminum processing was performed with a PLS6MW laser platform from Universal Laser System, which supports the use of both a 50 W CO_2 laser at 9.3 µm and a 40 W fiber laser at 1.06 µm. Both laser types are operated in a pulsed mode. The platform allows the adjustment of the laser movement speed, power, frequency, focus position in z direction and type of wave. If not only single lines (one movement of the laser) are required, but also areas are treated, additional values for contrast, definition and density can be set. These values characterize the raster behavior [92].

Beside a powerful smoke funnel, the apparatus provides a gas flow directed at the cutting, i.e. evaporation, point. The system was operated with air. In principle every other gas, advisably fire-proof, could be used easily. If light materials, such as plastic films, are used this additional airflow prevents the material from burning, but can also blow it away. Hence, sticky rubber mats are used as support structure.

The delivered software and drivers allow the use of the laser platform like any arbitrary printer. In principle, any vector format can be used as an input.

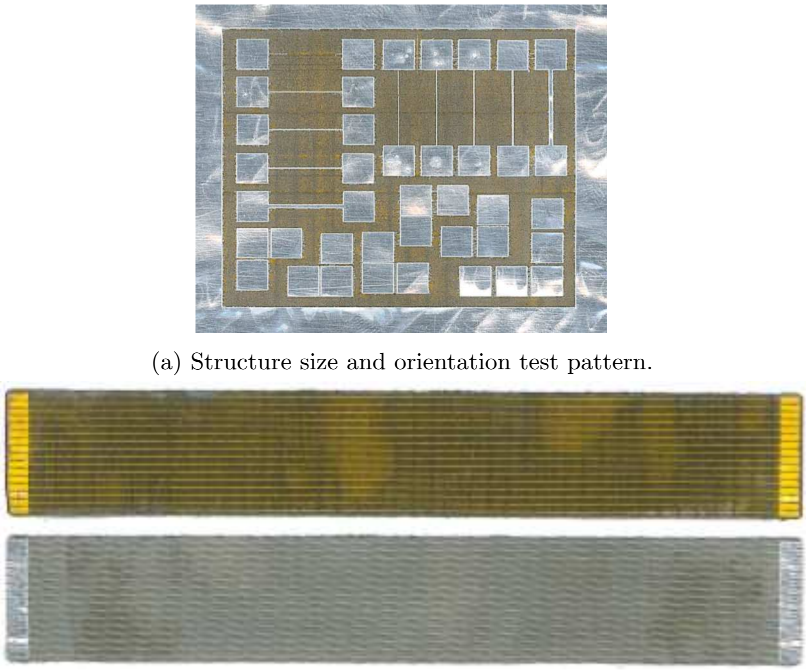


(a) Aluminum side.

(b) Kapton side.

(c) Comparison 50 nm (l.) and 25 µm (r.) Al.

Figure 7.2: 25 µm Kapton film with 50 nm/25 µm aluminium on top.



(a) Structure size and orientation test pattern.

(b) Kapton cable structures test pattern.

Figure 7.3: 25 µm Kapton film with 25 µm aluminum on top. Different test pattern.

7.4 Proof of Concept

The laser cutting machine described before could be used for Kapton flexprint fabrication. To produce reasonable LVDS links on a Kapton support structure, one should be able to create structures on a 50 µm scale. This is required by the targeted differential impedance of 100 Ω as well as space considerations inside the detector. The goal is to evaporate the aluminum with the laser system without damaging the underlying Kapton structure. For cables, the remaining aluminum has to be connected over relatively long distances of up to 30 cm. In a first step, the evaporation of the aluminum while preserving the Kapton is tested. The treatment of areas rather than simple cuts need significantly different settings. Once a good setting range is found, tests with different structure sizes and orientations are performed.

type	power [%]	speed [%]	freq. [MHz]	z [mm]	wave	contrast [%]	definition [%]	density [%]
area, $50nm$	100	45	76	1.5	0	20	10	80
cut, $50nm$	100	7	30	1.5	-	-	-	-
area, $25\mu m$	95	40	30	1.5	0	20	10	80
area*, $25\mu m$	100	30	30	1.5	0	20	10	80
cut, $25\mu m$	100	7	30	1.5	0	-	-	-

Table 7.1: Used values for laser cutter to produce small structures in Al on Kapton. *These values are better suited for small structures, e.g. between two pads in figure 7.3b.

Aluminum Evaporation

Aluminum's melting point is at 660 °C and it evaporates at 2519 °C. Its reflectivity around 1 µm lies in a range between 0.86 and 0.97 whereas around 10 μm it lies around 0.98. To deploy as much laser energy as possible into the aluminum, the fiber laser with its wavelength of 1.06 µm is used [93, Fig 1].

Different Evaporation Schemes

The two tested foils have to be treated in very different ways. While the thinner one can be handled with the Kapton side on top, this is not possible for the thicker one. The Kapton foil seems to be almost transparent at the used laser wavelength, so the energy is deposited in the aluminum layer, which then is evaporated. Figure 7.2 shows the clean results which could be achieved. The comparison in 7.2c shows the clean 50 nm vaporized Al sample, in the contrast to the thicker 25 µm glued Al sample on the right. The right Mu3e logo, manufactured with the aluminum side on top, shows residues of burned glue.
If 25 µm aluminum films are treated in the same way, not the whole Al layer evaporates and the gas escapes through the Kapton film. The Kapton is destroyed while still some aluminum remains and no electrical separation is achieved.
Hence the thicker 25 µm Al foil is handled with its aluminum layer to the top. Table 7.1 shows the settings used for the different foils. It has to be mentioned that the focus settings in z direction are very crucial and can vary from day to day. In order to vary the focus, the laser cuter moves the whole stage up and down. Presumably, the zero adjustment does not work properly.

Structure Sizes

The test patterns shown in 7.3a are used to evaluate the minimal scale for producable structures. With the patterns in figure 7.3a on one hand the ability to separate two elements electrically and on the other hand how small structures still provide electrical connectivity are tested. Both kinds of scale tests are performed in both directions, parallel and perpendicular to the laser carrier's motion. With the best settings as stated in table 7.1, the following minimal structure sizes could be realized:

scale [μm]	separation	connection
parallel	50	100
perpendicular	400*	50

Table 7.2: Minimal scales for separation respectively connection of $25\mu m$ Al components. *Settings for normal area not the one for dedicated small ones from table 7.1 are used.

The values stated above are the ones from the drawing, the actual scale after the process has to be measured in detail in a future work. The cable test patterns in figure 7.3b are measured by scanning them with a 600 dpi resolution. This allows an estimate on a 50 µm base. 10 cm long parallel cable patterns are produced with a minimal line width w of 150±50 µm. 10 cm long perpendicular patterns could be fabricated with a minimal width of 50 µm. For these samples, the connection between wire and pads is broken.

Slower LVDS links are planned with cables with a w ≈10 µm scale could be evaluated properly. Cables of both directions show a resistance of $2\Omega/10cm$. Taking the Al thickness of 25 µm and a specific electric resistance of aluminum of $\rho = 2.65 \cdot 10^{-2}\Omega \cdot mm^2/m$ into account, the measured resistance indicates a line width of 53 µm. In the cable test pattern, the separation between two pads and the connection between wire structure and pad are frequently broken and require particular care in the future.

7.5 Future Work

The presented work is a successful proof of principle. Aluminium coated Kapton seems to be processable with the present laser platform from Universal Laser Systems. The possible structure scales are just sufficient to create reasonable LVDS links. In future, an intensive study of laser parameters could lead to slightly better results.

The following tasks are still open and could be checked in a subsequent project:

1. Once structures suitable for LVDS links are producible, their assembly to cables with plugs has to be evaluated as well.
2. Kapton films with an intermediate aluminium layer thickness of 12 µm are available. It has to be evaluated if they still can be processed with the Kapton layer to the top.
3. A suitable solvent to remove all residues of the 25 µm Al layers and glue has to be found.
4. Furthermore, Kapton coated on both sides with an aluminium layer is available. Whether the treatment with the laser system works with this kind of foil has to be studied.

Part IV

Outlook

Chapter 8

Outlook

In this thesis, fast optical data transmission as well as selected aspects of the readout chain have been successfully tested. In chapter 5, the performance of different optical transmission methods are presented. Chapter 6 introduces concepts how the data structure could be manipulated on different FPGA levels and chapter 7.5 proofs the concept of in house Aliminum-Kapton flexprint production. These considerations covering very different aspects of the readout chain lead to the following recommendation for the data structure and for the phase Ia concept.

8.1 Readout Chain in General

It has been shown in general, that the components tested in this thesis fulfill the requirements proposed for the Mu3e read out chain. Particularly, the optical bandwidths is sufficient, Kapton flexprints could in principle be produced in house and coordinate transformation could be moved from filter farm CPUs or GPUs to FPGAs.

8.2 Data Structure

Based on the experience with the optical links, a data structure recommendation for the optical links of the Mu3e readout chain is presented. This recommendation focuses on the MuPix sub-detector.

8.2.1 Starting Point

A single pixel hit can be encoded into 38 bit according to the address scheme described in 1.3.1. Multiple hits from a couple of sensors are buffered and time sorted in the front-end FPGAs. This data is sent as a package outside the detector via optical links. As performance tests have shown, a disparity control over 80 bit is required. A 64b/66b-like encoding scheme would provide the possibility to distinguish between data words and control structures. An additional scrambler does not improve the signal quality significantly, but complicates the synchronization process. Therefore, a scrambler is optional.

8.2.2 Error Detection

Due to the very low error rate, an error detection with a CRC hash could be implemented over rather long packages. Whenever an error is detected in such a package, the whole package has to be thrown away. If, because of bandwidth limitations, higher rates such as 8.0 Gbit/s are required, the data packages over which a single CRC code is applied should be reduced. If errors occur more frequently, one would like to drop smaller data packages.

8.2.3 Proposed Format

I propose to send the data in rather long packages, which include several frames. The exact data width has to be adapted to the available PMA. One bit should be used for disparity control, further two to indicate data or control words. One dedicated control word could indicate the start of a new frame and include the corresponding timestamp. Followed by the raw hit data with stripped-off timestamp completed with an end of event (EOE) code word and a CRC hash. If the CRC hash does not match, only one frame has to be dropped. Each front-end FPGA needs slow control as well. Small form factors are required by the experiments design. Therefore, it seems to be reasonable to include slow control information into the same optical links. This also ensures galvanic isolation of the detector.

8.3 Phase Ia Readout Chain

The results of this thesis show that the rate requirements of the proposed Mu3e readout chain are met. Furthermore, the used hardware is almost sufficient to build a readout chain for phase Ia. Figures 8.1 shows the idea of such a concept.

The 38 front-end FPGA boards could be equipped with MicroPOD 14 Gbps twelve channel parallel fiber optic modules from Avago [94]. The fibers are connected to SFP transceivers on SantaLuz boards of 4 Stratix V development boards. 10 respectively 9 of the available 12 SFP cages are used as inputs. Two of this front-end FPGAs collect up-stream, two down-stream data. One SFP output can be used to connect all four Stratix boards to a fifth one. SFP to QSFP cables from Cisco could be used therefore [95]. This fifth FPGA distributes the data with two SantaLuz cards, which allow all together 12 SFP links to 12 PCs in the filter farm. There, the FPGA receives the data with the same SantaLuz daughter cards and put it via PCIe and direct memory access into the GPUs memory.

The bottleneck of the proposed scheme is the single fifth Stratix board which limits the data rate of the whole system to 4 times the operation speed of the SFP plugs. If the links are operated at 6.4 Gbit/s this results in 25.6 Gbit/s. This FPGA could easily be doubled as indicated in green in figure 8.1. With single QSFP to quad QSFP plugs, also available from Cisco, the number of these FPGAs could even be six-folded. If only one additional Stratix is used, one SFP plug per read-out FPGA could be used for slow control, managed by a run control computer equipped with a Stratix development board either with QSFP to SFP or SFP to SFP links. Particularly, because one Stratix V board is sufficient for the expected data volume of the pixel tracker.

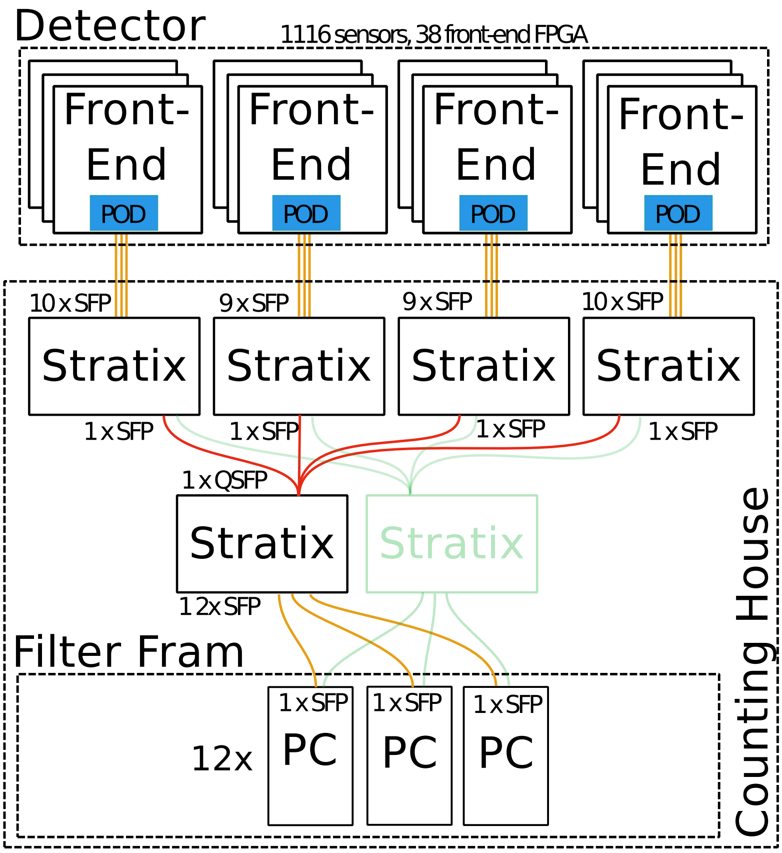


Figure 8.1: Proposed topology of phase Ia readout chain. The MicroPODs in blue, as well as the QSFP to 4 SFP cables in red have to be tested in a subsequent work. The bottleneck of the proposed system is the fifth DAQ FPGA, which can easily be doubled. All Stratix boards have to be equipped with two SantaLuz cards.

The MicroPOD as well as the QSFP to SFP cables have to be tested in a subsequent work.

Appendix A

Appendix

A.1 Stratix V Transceivers

The different components of the Stratix V transceivers are listed and described in this section.

A.1.1 Physical Media Attachment (PMA)

The PMA provides all the necessary (essential) functionality for high rate serial data transmission. Parallel data with a given data width is processed to a well shaped serial data stream. Figure A.1 shows the different PMAs divided in a transmitter (Tx) and a receiver (Rx) part.

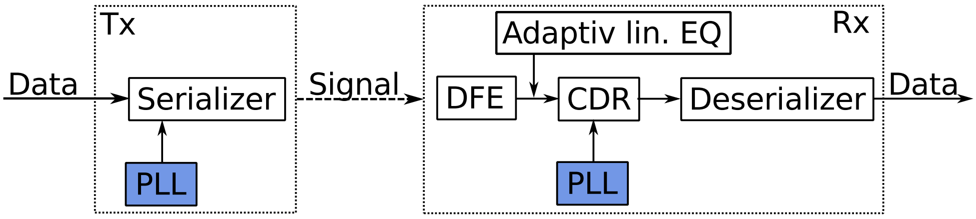


Figure A.1: PMA transmitter (Tx) and receiver (Rx) overview.

Serializer The Serializer converts incoming parallel data to a fast serial data stream. This task requires the use of different PLLs (see 3.3.1) which are built directly into the hard core. The Stratix V GS FPGA supports 8, 10, 16, 20, 32, 40, 64 and 80 bit parallel data inputs [56, p. 1-20].

Analog Settings of the Tx Signal The Tx hard core supports analog tuning to shape the output signal for higher performance of the whole system. The following values can be set:

Differential output Voltage V_{OD} can be adjusted to the requirements of the different physical media. The differential output is $V_A - V_B = V_{OD}$. If single ended channels are used, the signal is modulated between $-V_{OD}$ and $+V_{OD}$ [56, p. 1-21].

Pre-Emphasis allows to boost the high frequencies in the transmitted data signal. Three different taps are supported. Pre-tap with values between 0 and 15 influences the pre-emphasis on the bit before the transition. The first and second post-tap with 32 respectively 16 different values, set the pre-emphasis in the transition bit and the following bit [56, 96, p. 1-22].

DC Gain and Continuous Time Linear Equalization (CTLE) The first part of the PMA Tx unit is a circuit to boost the whole incoming signal independent of frequency. This is referred to as DC gain. Furthermore, circuits to boost high-frequency part of the signal are available. The settings for this five staged frequency dependent boosting is adjusted during run time to adjust a linear equalization (CTLE). Once the system finds a stable setting the values are locked and not adjusted any longer. Alternatively, the values for the linear equalization can be set manually at compilation time.

Decision Feedback Equalization (DFE) In addition to the above explained components which are suppose to improve the signal quality, a supplementary module for decision feedback equalization exists. Similar to the CTLE this system boosts the high-frequency part of the signal, but under the additional consideration of intersymbol interference (ISI). The boost amplitude depends in this case on the previously received data bits [56, p. 1-12].

Clock Data Recovery (CDR) Out of the incoming serial data flow the CDR recovers the clock frequency of the incoming stream. Once this system is locked the Rx unit is ready to process the fast serial data stream into parallel data which can be processed with a much slower clock frequency. The Stratix V CDR provides a hybrid mode. In addition to the data stream, a reference clock is used to lock a PLL circuit as described in chapter 3.3.1 much faster to the expected data rate. Once the PLL is locked to the reference clock and the phase difference between PLL, driven by the reference clock, and the incoming data doesn't differ more than a given value, the system changes to a lock to the data state. In this state, the clock data recovery unit is driven by the incoming data stream and the reference clock is only used to check the frequency stability of the system. The CDR unit is a very crucial part of the whole PMA [p. 1-15f. 56, 97, p. 6f.].

EyeQ In addition to the circuits essential for data transmission, the PMA includes an EyeQ circuit which fulfills purely debugging purposes. The EyeQ circuit shifts the recovered clock from the CDR unit by a specified fraction and compares the resolved data with a fixed known pattern. This does not allow to measure classical eye diagrams as described in section 2.3.1, but allows the eye openings in the time and signal direction to be sampled over a continuous data stream by looping throw all available fractions of offset to the original recovered clocks. This procedure relies on data pattern generation units and bit error tests in the different PCSs. Despite the misleading name EyeQ, this measurement concept corresponds rather to bathtub plots as described in 2.3.2. How this method is used to tune the analog settings in detail is described in section 5.3.4 [98, 99].

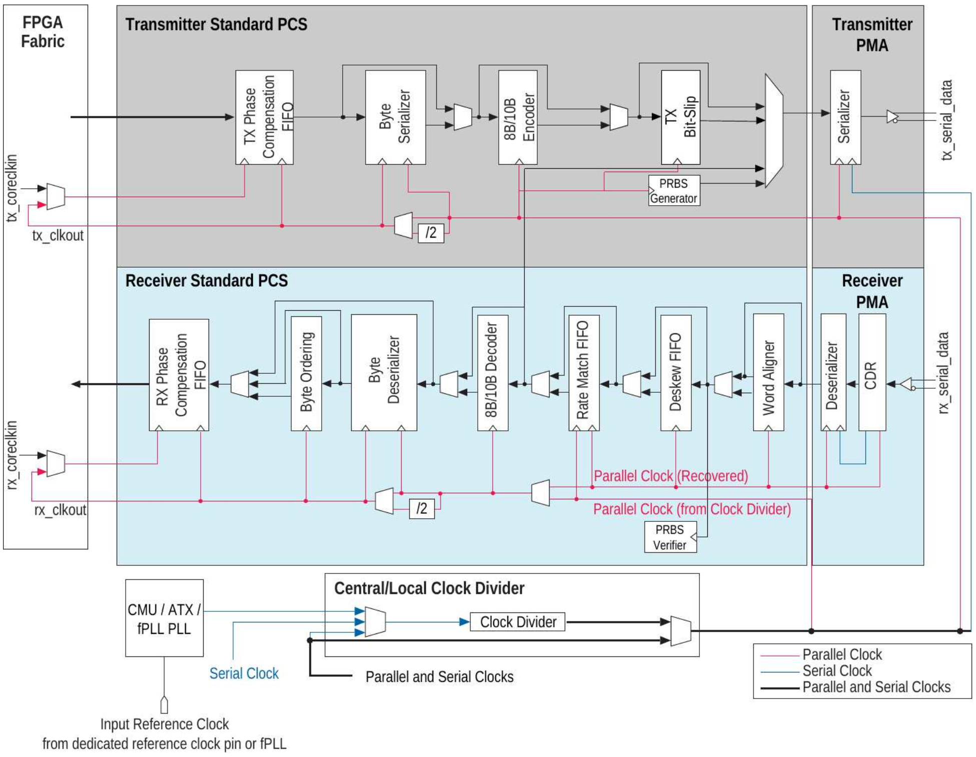


Figure A.2: Stratix V Standard PCS overview [56, Figure 1-19].

Dynamic Reconfiguration Dynamic reconfiguration IP hard cores allow the adjustment of transceiver's parameters at run time. The reconfiguration is managed via a mapped memory and has to be synchronized to a strict transceiver reset procedure.

A.1.2 Physical Coding Sublayer (PCS)

The three different available physical coding sublayers (PCSs) standard, 10G and PCIe provide additional optional hard cores for data transmissions. The use of hard core PCS saves logic resources in the FPGA and decreases the power consumption of the device. All individual components can also be bypassed. Figure A.2 shows all available elements of the standard PCS, which are described in the following.

Phase Compensation FIFOs Both channels, transmitter and receiver, offer a FIFO section 3.2.4 between the PMA and the user logic clock network. The only possible compensation is the one of a phase difference between the networks. To compensate for slightly different clock frequencies by adding or removing dedicated elements from the data stream, a separate deskew FIFO exists in the Rx part.

Byte Serializer and Deserializer The possibility to serialize two bytes in the PCS offers more flexibility in the data width for the PCS part. The byte serializer respectively deserializer allow to double the input bandwidth.

8b/10b Encoder and Decoder The PCS provides a hard core for 8b/10b encoding and decoding. This circuit implements an encoding scheme as described in section 2.2.2. The running disparity is calculated 8 bit word by 8 bit word. If a data width of more than 8 bit is chosen, the encoder has to be operated at much higher rates to calculate the running disparity word by word. This very strict calculation of the running disparity limits the maximal frequency with which this module can be used. Alternatively, an user implemented encoding, where multiple 8b/10b encoders are used in parallel and therefore multiple running disparities of 8 bit words are used. The serial DC balance is less strong and only balanced over the time.

Bit-Skip Units As described in section A.1.1, the CDR unit and the deserializer reconstruct the parallel data out of the serial data stream. The bit slip unit in the Rx part skips one bit to slip the reconstructed parallel word. This feature can be used to align the receiver logic to the transmitter. Alternatively this operation could be performed in a user designed buffer. The bit slip unit on the transmitter side can be used to align different Tx lines.

Clocks The PCS provides the parallel clocks on the Tx as well as the Rx side for further use in the user logic. The Tx part requires a reference clock to which the provided parallel data have to be in phase. However, the Rx part does not require any additional input clock because the timing is recovered out of the received data stream. The different required, optional and output clocks are shown in figure A.2.

Debug Elements In addition to the above described elements which serve the data transmission, additional elements for debug purpose exists. A pseudo random binary sequence generator is a linear feedback shift register (LFSR) as described in section 3.4 to generate different test patterns which can be fed directly into the data stream. This modules supports the following patters: [p. 11-5 99, 100, Table 34–12]

PRBS7 LFSR $x^7 + x^6 + 1$

PRBS15 LFSR $x^{15} + x^{14} + 1$

PRBS23 LFSR $x^{23} + x^{18} + 1$

PRBS31 LFSR $x^{31} + x^{28} + 1$

high frequency “010101010...”-pattern

low frequency “111110000” pattern for 10-bit words or “11110000” for 8-bit words

where the maximal supported data width is 40 bits. The PRBS verifier checks whether the incoming data corresponds to the generated pseudo random pattern. The output of these circuits can be access with memory mapped reconfiguration as described in section A.1.1 and is used by the EyeQ circuit incorporated in the PMA.

10G PCS

The alternative 10G PCS provides the same base functionality as the standard PCS. As the name indicates, it is used for data transmissions around 10 Gbit/s. Therefore, some additional components which are primarily used for faster data transmissions are implemented and some components also present in the standard PCS are slightly adapted to the requirements of the high data rates. Figure A.3 shows an overview over all 10G PCS components.

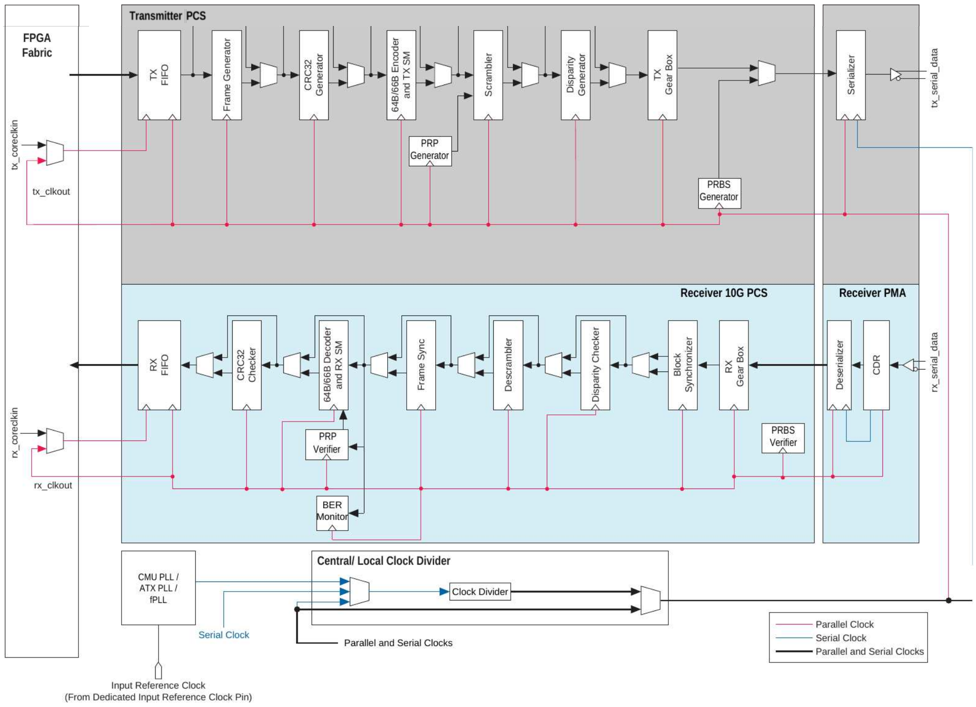


Figure A.3: Stratix V 10G PCS overview [56, Figure 1-42]
.

CRC32 Generator and Checker A Cyclic Redundancy Check (CRC) hash as described in 2.3.3 can be calculated during run time with this dedicated hard core unit. CRC32 generator and checker are used in many different communication standards and serve manly as an error detection method.

64b/66b Encoder Analogously to the the 8b/10b encoder in the standard PCS a 64b/66b encoder is available in the 10G PCS. 64b/66b encoding is described in section 2.2.2 and is used in several different modern 10G data transmission standards such as 10GE (see 2.2.4) or Interlaken (see 2.2.4). In contrast to 8b/10b encoding, this encoding does not ensure DC balancing, hence an additional disparity controller is needed.

Disparity Generator and Checker To limit the running disparity, the continuous sum of super-numerous ones, a disparity checker is built into the 10G PCS. This module decides whether the actual data word should be inverted or not and indicates this in one additional bit. This method ensures that the running disparity is never larger than plus, respectively minus the used data width. Disparity generation in combination with 64b/66b encoding is sometimes quoted as 64b/67b encoding.

Scrambler and Frame Management A scrambler as well as a frame generation and synchronizer unit as described in section 2.2.3 about the Interlaken protocol are available as hard cores. These units are designed for this very specific protocol.

Gear Box The 10G PCS and the attached PMA have very often different data widths, particularly because some 10G PCS components such as 64b/66b encoding or disparity checker change the used data width. The gearbox mediates between these two parts with possibly different data widths. It converts for example a 67 bit data word from a Interlaken protocol to a 64 bit or 80 bit word in the corresponding frequency to put it through the PMA.

PRB and BER Monitor The 10G PCS adds some additional features for debugging and signal quality monitoring. Some of this components are very protocol specific, for example the PRB checker for the 10GBASE-R protocol described in 2.2.4 [56, p. 1-45].

PCIe PCS

The third available PCS is explicit designed for Gen3 PCIe connections. This topic is not treated in this work and therefore no further description of this PCS is provided at this point. The documentation can be found in [56, p. 1-51ff].

Available Configurations

Altera provides a set of defined PHY IP blocks for a huge variety of different protocols. These IPs are managed through Altera MegaFunctions which use the above described PCSs and PMA with some predefined values to protocol specific IP cores. If nothing else is stated, in the following, the native PHY IP core is used. The native PHY IP reveals the whole functionality to the user and reconfiguration as well as resets have to be managed by the user through other IP cores or user own designed logic.

A.2 Quartus II and ModelSim

Since Altera FPGAs (see section 5.1.1) are used for the whole setup, the corresponding Quartus II software package, also from Altera is used. In Quartus II, the complete functionality for FPGA programming is united in a single software development kit (SDK), even though the SDK very often only provides a starter for quite independent software packages such as ModelSim or TimeQuest Timing Analyzer.

Quartus II provides on one hand the possibility to describe the desired logic in a hardware description language (HDL) such as VHDL or Verilog and provides on the other hand a graphical interface to wire logic blocks with the mouse. Although the graphical design process can be very fast for smaller projects, the advantages of being cleaner, more easily maintainable and reusable of the HDL scheme dominate. It is possible to use VHDL and Verilog in parallel in the same project.
The question whether to use very high speed integrated circuits HDL (VHDL) or Verilog HDL is a matter of taste. Both languages provide the same functionality where VHDL seems to be sometimes more explicit where as Verilog is in some cases

slightly faster to write.

Once the desired logic is described with any HDL and the output and input pins are assigned to the right physical pins on the FPGA, Quartus II is used to synthesize the user logic, followed by fitting it into the available logic cells in a design which meets the required timing constraints. The netlist which explains the connectivity of the design is generated. Depending on the size of the project and the used hardware, this process can take several minutes up to several hours.

The Quartus II SDK provides a number of small programs which can optimize the current logic design regarding various aspects. In the following a selection is presented with a short introduction what they are used for.

TimeQuest Analysis Software used to analyse the timing of a given logic design. Besides of reporting the slack of failing paths, this tool allows to access information about how fast the current design could be clocked.

Power Analysis A software package which calculates the power consumption as well as the predicted temperature of the device.

Floorplan Editor The Floorplaner allows the user to manipulate directly the implementation of the logic in the device.

Device Programmer The small program sends the compiled software file (.sof) over the on-board USB Blaster (see 5.1.1) circuit direct into the FPGA. The software provides in addition a possibility to validate the current programming through a hash tag.

MegaWizard The MegaWizard manager allows to access Altera's Mega Functions. These functions are a combination of user logic and IP hard cores. The wizard offers a GUI to configure these predefined functions and generates all required files. Particularly the generated HDL files are fully, transparent and can be viewed and manually edited.

QSys Altera's System Integration Tool QSys provides similar functionality as the MegaWizard whilst adding more flexibility. QSys allows to connect different components, very often IP hard cores, in a graphical fashion. In the end they are synthesized and included into the logic as one logical unit.

System Console The System console is an interface over which Altera provides online debugging. If the required logic is connected in the user design, the system console allows to access FPGA resources such as memory and transmitter reconfiguration over an USB connection. The receiver toolkit described in section 5.2.3 is built on this console.

A.2.1 ModelSim

ModelSim is a software package developed by Mentor Graphics which provides simulations of behavioral, register transfer level (RTL) and gate level modeling based on HDL files, respectively on the produced netlist. In RTL simulations, all external signals and their time dependencies are specified by the user. Figure A.4 shows the output of a RTL simulation of a MuPix Emulator (see Appendix A.4) where the

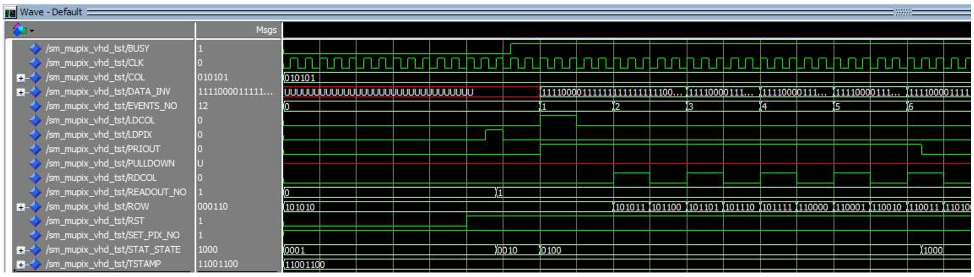

Wave - Default
Msgs
/sm_mupix_vhd_tst/BUSY 1
/sm_mupix_vhd_tst/CLK 0
/sm_mupix_vhd_tst/COL 010101
/sm_mupix_vhd_tst/DATA_INV 1111000011111...
/sm_mupix_vhd_tst/EVENTS_NO 12
/sm_mupix_vhd_tst/LDCOL 0
/sm_mupix_vhd_tst/LDPIX 0
/sm_mupix_vhd_tst/PRIOUT 0
/sm_mupix_vhd_tst/PULLDOWN U
/sm_mupix_vhd_tst/RDCOL 0
/sm_mupix_vhd_tst/READOUT_NO 1
/sm_mupix_vhd_tst/ROW 000110
/sm_mupix_vhd_tst/RST 1
/sm_mupix_vhd_tst/SET_PIX_NO 1
/sm_mupix_vhd_tst/STAT_STATE 1000
/sm_mupix_vhd_tst/TSTAMP 11001100

Figure A.4: An example view of ModelSim where a MuPix emulator RTL simulated is performed.

clock clk and the input signals PULLDOWN, RDCOL, LDCOL are generated by the software.

ModelSim RTL simulations are often the only way to verify FPGA designs because they allow complete control over all input signals. Beside the output signals, the simulation allows access to internal signals which yields a deeper insight into the logic.

A.3 Multi-Channel Results

used rx channels	BER
0,1,6	$< 2.3 \cdot 10^{-12}$
0,1,7	$< 9.0 \cdot 10^{-12}$
0,1,5	$1.7 \cdot 10^{-11} \pm 6.2 \cdot 10^{-12}$
0,1,4	$< 1.0 \cdot 10^{-11}$
0,1,3	$< 9.8 \cdot 10^{-12}$
0,1,2	$< 8.1 \cdot 10^{-12}$
0,2,5	$1.9 \cdot 10^{-12} \pm 1.9 \cdot 10^{-12}$
0,4,5	$1.1 \cdot 10^{-5} \pm 5.5 \cdot 10^{-9}$
0,2,3	$< 1.0 \cdot 10^{-11}$
0,2,4	$< 1.0 \cdot 10^{-11}$
0,3,4	$< 9.3 \cdot 10^{-12}$
1,3,4	$< 1.0 \cdot 10^{-11}$
2,3,4	$< 9.4 \cdot 10^{-12}$
1,2,3	$< 8.5 \cdot 10^{-12}$
6,1,3	$< 9.5 \cdot 10^{-12}$
6,1,2	$< 7.0 \cdot 10^{-12}$
6,2,3	$< 9.3 \cdot 10^{-12}$
6,3,4	$< 7.2 \cdot 10^{-12}$
6,4,5	$2.2 \cdot 10^{-8} \pm 2.1 \cdot 10^{-10}$
7,6,2,0	$< 2.2 \cdot 10^{-12}$
7,6,1,0	$< 5.2 \cdot 10^{-12}$
7,6,3,0	$< 4.0 \cdot 10^{-12}$
7,6,4,0	$< 2.9 \cdot 10^{-12}$
7,6,5,0	$< 3.5 \cdot 10^{-12}$
7,6,0,2,5	$2.2 \cdot 10^{-12} \pm 1.3 \cdot 10^{-12}$
7,6,0,2,4	$< 5.4 \cdot 10^{-13}$
0,1,2,3,4	$6.2 \cdot 10^{-14} \pm 4.3 \cdot 10^{-14}$
0,1,2,3,4	$4.9 \cdot 10^{-15} \pm 1.7 \cdot 10^{-15}$
0,1,2,3,5	$2.0 \cdot 10^{-14} \pm 2.0 \cdot 10^{-14}$

Table A.1: BER rates for selected setups with multiple receiving channels performed at 6.4 Gbps with 50 m cables. For all channels the same random number generator with the same seed were used, hence all the data are synchronous.

A.4 MuPix4 Emulator

The MuPix4 is a prototype chip for the final MuPix chip of the Mu3e pixel subdetector. It is used to develop a telescope CITE-LENNERT. For telescope debuging and to test LVDS links between two FPGAs an FPGA based emulator was produced.

The firmware written for the Stratix V development board uses the HSMA and HSMB ports to simulate the data of a MuPix4 chip read-out by boards developed by DIRK-CITE. Four chips can be simulated at the same time on one single FPGA. The emulator follows the same protocol as the pixel sensor and generates randomly distributed hits with slightly disordered Gray code timestamps. In addition, the control of single address bits of single channels provides a good debug tool.

The read-out protocol works as described below:

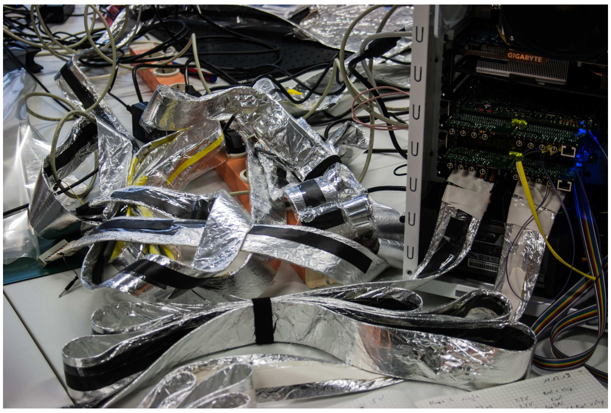


Figure A.5: MuPix telescope read-out FPGAs, they receive the emulators data via the parallel cables shielded by aluminum foil in the picture.

1. Wait for LDPIX (load pixel)
2. Wait for LDCOL (load column)
3. Rise PRIOUT (indicates data is loaded)
4. Wait for RDCOL (read data)
5. Send hit data, go to step 4 rowNo times, else set PRIOUT to 0
6. Go to setp 2 colNo times
7. set PRIOUT to 0, start over

The developed emulator can also be used to set up a whole readout chain.

A.5 SantaLuz Crosstalk Measurments

Cross Talk [%]	Measured Channel							
transmitting ch	0	1	2	3	4	5	6	7
0	0.4	0.1	0.2	0.0	0.0	0.0	0.3	0.0
1	0.1	2.4	1.3	0.4	0.1	0.4	0.0	0.0
2	0.0	2.1	0.3	2.1	1.0	0.4	0.0	0.0
3	0.0	0.0	8.7	0.1	1.4	0.6	0.1	0.0
4	0.0	0.0	3.2	3.3	0.6	1.5	0.7	0.0
5	0.0	0.0	0.0	2.2	1.4	0.6	1.2	0.3
6	0.0	0.0	0.0	0.2	2.1	0.4	0.4	0.8
7	0.0	0.0	0.0	0.0	0.2	0.3	0.1	0.0
total	0.5	4.8	13.9	8.3	6.9	4.4	2.5	1.1

Table A.2: Cross Talk in receiving channels introduced via the transmitting channels with untuned Stratix V transmitter. The measurements are performed at 8 Gbit/s.

Cross Talk [%]	Measured Channel							
transmitting ch	0	1	2	3	4	5	6	7
0	0.4	0.0	0.2	0.2	0.2	0.0	0.0	0.0
1	0.0	2.9	1.8	0.5	0.2	0.0	0.0	0.0
2	0.0	1.8	0.4	2.4	1.0	0.0	0.0	0.0
3	0.0	0.0	6.0	0.4	1.2	0.3	0.0	0.0
4	0.0	0.0	4.2	3.3	0.4	1.0	0.8	0.0
5	0.0	0.0	0.2	1.8	1.0	0.0	1.2	0.3
6	0.0	0.0	0.2	0.4	2.8	0.2	0.4	0.5
7	0.0	0.0	0.2	0.4	0.2	0.0	0.0	0.0
total	0.4	4.6	13.1	9.3	7.0	1.5	2.3	0.8

Table A.3: Cross Talk in receiving channels introduced via the transmitting channels with tuned Stratix V transmitter. The measurements are performed at 8 Gbit/s.

Cross Talk [%]	Measured Channel							
receiving ch	0	1	2	3	4	5	6	7
0	-	-	0.0	0.0	0.0	0.1	0.0	0.0
1	-	-	-	0.3	0.3	0.2	0.3	0.2
2	0.2	-	-	-	0.0	0.0	0.0	0.0
3	0.3	0.3	-	-	-	0.0	0.0	0.0
4	0.0	0.3	0.3	-	-	-	0.1	0.0
5	0.0	0.0	0.2	0.3	-	-	1.0	0.1
6	0.0	0.0	0.0	0.2	0.3	1.2	-	1.2
7	0.0	0.0	0.1	0.0	0.3	0.2	0.3	-

Table A.4: Cross Talk in receiving channels introduced via the receiving channels with Avago optical transmitter AFBR-57D7APZ. The measurements are performed at 8 Gbit/s and normalized to the 500 mV output signal stated in the specifications.

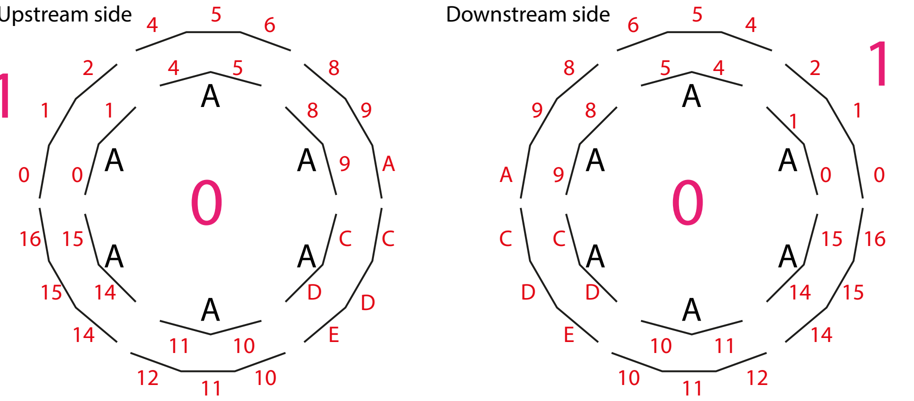
A.6 MuPix Address Scheme

Mu3e Phase I
Pixel Readout Scheme

v1 2.10.2013 NB
v2 21.10.2013 NB
v3 24.2.2014 NB

All addresses in hexadicemal notation

Vertex Layers



Upstream side
Downstream side
1
0
A
0 1 2 4 5 8 9 A C D 10 11 14 15
0 1 2 4 5 6 8 9 A C D E 10 11 12 14 15 16

Upstream → z → Downstream

Chip	1	2	3	4	5	6
Link	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2

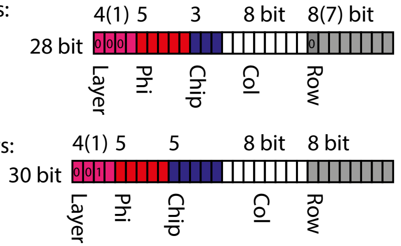
FPGA Types:

A: 5x3 Chips with 3 Links:	45 Links	12 FPGAs
D: 4x9 Chips with 1 Link:	36 Links	26 FPGAs

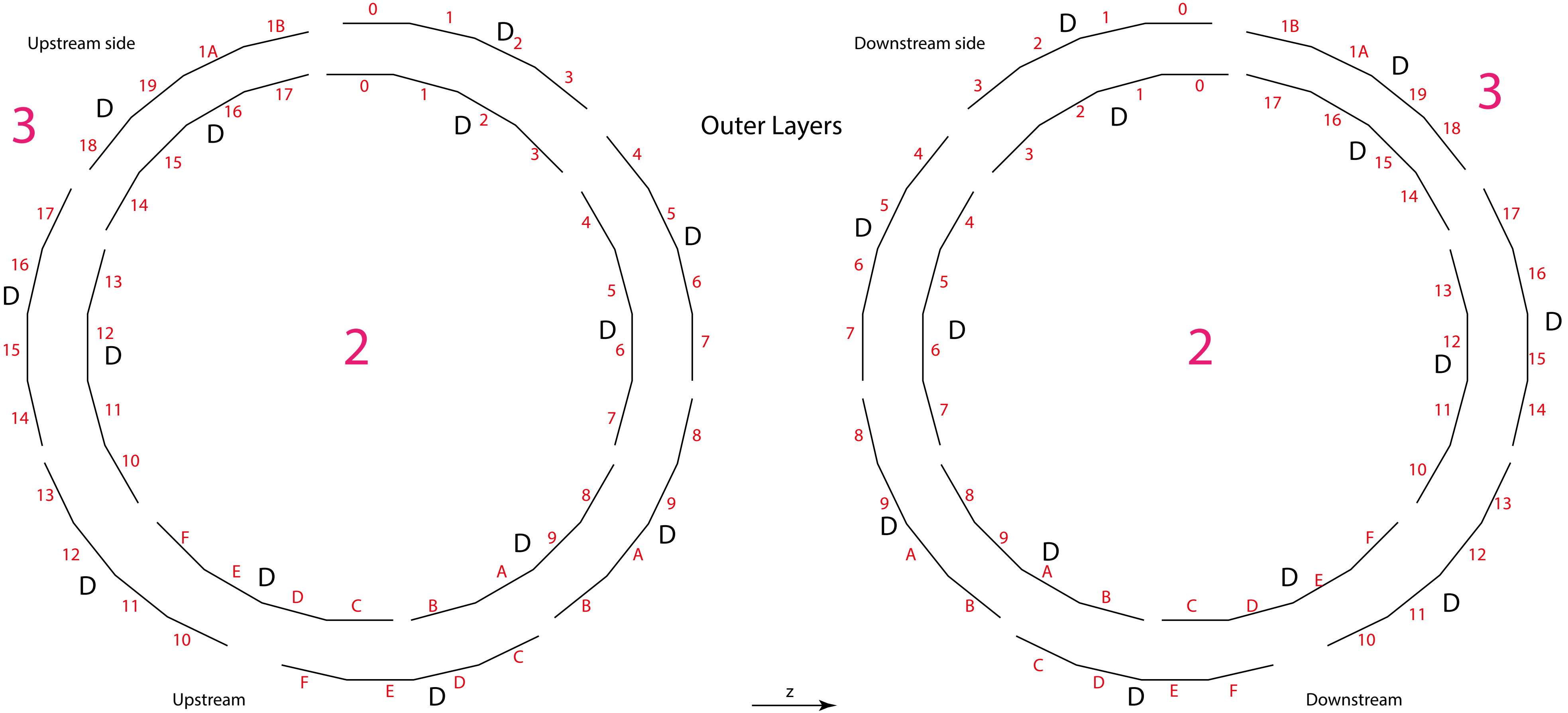
Address Scheme:

Inner Layers:

Outer Layers:



Outer Layers



Upstream side
Downstream side
3
2
D
0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12 13 14 15 16 17 18 19 1A 1B

Upstream → z → Downstream

Chip	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
Link	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

List of Figures

List of Tables

Bibliography

[1] CMS Collaboration. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Physics Letters B*, 716, 2012.

[2] Barry R. Holstein John F. Donoghue, Eugene Golowich. *Dynamics of the Standard Model.* Cambridge : Cambridge University Press, 1992.

[3] C.P. Burgess and Guy D. Moore. *The standard model : a primer.* Cambridge : Cambridge University Press, 2007.

[4] Wikimedia Commons. Standard model of elementary particles, 2014. [Online; accessed 10-februar-2014].

[5] Y. Fukuda et al. Evidence for oscillation of atmospheric neutrinos. *Phys. Rev. Lett.*, 81:1562–1567, 1998.

[6] Q. R. Ahmad et al. Measurement of the charged current interactions produced by B-8 solar neutrinos at the Sudbury Neutrino Observatory. *Phys. Rev. Lett.*, 87:071301, 2001.

[7] K. Eguchi et al. First results from KamLAND: Evidence for reactor antineutrino disappearance. *Phys. Rev. Lett.*, 90:021802, 2003.

[8] K. Nakamu and S.T. Petcov (DPG). Update: Neutrino mass, mixing, and oscillations. *DPG*, May 2013.

[9] Laurent Canetti and Mikhail Shaposhnikov. The ν MSM and muon to electron conversion experiments. *Hyperfine Interact.*, 214(1-3):5–11, 2013.

[10] L. Maiani S.L. Glashow, J. Iliopoulos. Weak interactions with lepton–hadron symmetry. *Physical Review D*, 2, 1970.

[11] A. Blondel et al. Research Proposal for an Experiment to Search for the Decay $\mu \to eee$. *ArXiv e-prints*, January 2013.

[12] W. J. Marciano, T. Mori, and J. M. Roney. Charged Lepton Flavor Violation Experiments. *Ann.Rev.Nucl.Part.Sci.*, 58:315–341, 2008.

[13] U. Bellgardt et al. Search for the Decay $\mu^+ \to e^+e^+e^-$. *Nucl.Phys.*, B299:1, 1988.

[14] W. Ootani. New results from MEG experiment. pages 475–480, 2013.

[15] J.L. Hewett, H. Weerts, K.S. Babu, J. Butler, B. Casey, et al. Planning the Future of U.S. Particle Physics (Snowmass 2013): Chapter 2: Intensity Frontier. 2014.

[16] K. Nakamura et al. Review of particle physics. *J. Phys.*, G37:075021, 2010.

[17] I. Perić et al. High-voltage pixel detectors in commercial CMOS technologies for ATLAS, CLIC and Mu3e experiments. *Nucl.Instrum.Meth.*, A731:131–136, 2013.

[18] A.-K. Perrevoort. Characterisation of high voltage monolithic active pixel sensors for the mu3e experiment. Master's thesis, Heidelberg University, 2012.

[19] H. Augustin. Charakterisierung von HV-MAPS. Master's thesis, Heidelberg University, 2012.

[20] A. Damyanova. Development of a Scintillating Fibre Tracker/Time-of-Flight Detector with SiPM Readout for the Mu3e Experiment at PSI. Master's thesis, Geneva University, 2013.

[21] Carlo Licciulli. [präzise zeitmessung für das mu3e-experiment. Master's thesis.

[22] Patrick Eckert. PhD thesis, Kirchhoff Institut für Physik. Personal contact.

[23] Lukas Huxold. Cooling of the Mu3e Pixel Detector, 2014.

[24] Fabian Förster. Master's thesis, University of Heidelberg, 2014. Personal contact.

[25] S. Bachmann et al. The proposed trigger-less TBit/s readout for the Mu3e experiment. *Jinst*, (9), January 2014.

[26] Niklaus Berger. Read-out simulation. In *Mu3e internal notes*, November 2013.

[27] Gianfranco Cariolaro. *Unified signal theory.* London : Springer, 2011.

[28] John David Jackson. *Classical Electrodynamics.* John Wiley & Sons, Inc., New Jersey, USA, 3 edition, 1991.

[29] *Return-to-zero coding in optical intersatellite links.* Technische Universität Wien, 2003.

[30] R Forster. Manchester encoding: Opposing definitions resolved. *Engineering Science & Education Journal*, 200.

[31] Jianfeng Zhang, P.V. Holm-Nielsen Nan Chi, C. Peucheret, and P. Jeppesen. 10 Gbit/s Menchester-encoded FSK-labelled optical signal transmission linkS. *Electronics Letters Online*, (20030769), April 2003.

[32] Dennis Derickson; Marcus Müller. *Digital Communications Test and Measurement: High-Speed Physical Layer Characterization.* Prentice Hall, December 2007.

[33] P. A. Franaszek A. X. Widmer. A dc-balanced, partitioned-block, 8b/10b transmission code. *IBM Journal of Research and Development*, 27(5):440, 1983.

[34] Byte oriented dc balanced (0,4) 8b/10b partitioned block transmission code, 1984.

[35] Patent: 64b/66b decoding, for packetized serial data, 2001.

[36] Bijan Raahemi. *ERROR CORRECTION ON 64/66 BIT ENCODED LINKS*. University of Ottawa, May 2005.

[37] Robert Melancton Metcalfe. *Packet Communication*. Cambridge : Massachusetts Inst. of Tech., 1973.

[38] IEEE: Ethernet 802.3.

[39] G. Held. *Ethernet Networks: Design, Implementation, Operation, Management*. Wiley professional computing. Wiley, 2003.

[40] IEEE: 10GE 802.3e.

[41] IEEE-802.3 Clause 49, 64b/66b.

[42] Amendment: Media Access Control (MAC) Parameters, Phzsical LLayer, and Management Parameters for 10 Gbps Operation.

[43] System Packet Interface Level 4 (SPI-4) Phase 2 Revision 1: OC-192 System Interface for Physical and Link Layer Devices.

[44] Interlaken protocol definition, October 7, 2008.

[45] Understanding data eye diagram methodology for analyzing high speed digital signals. *ON Semiconductors*, AND9075/D.

[46] WAVECREST Corporation. *Relationship Between Eye Diagram and Bathtub Curves*, 2003.

[47] Agilent Technologies. *Measuring Jitter in Difital Systems Application Note*, 2013.

[48] Martin Stigge, Henryk Ploetz, Wolf Mueller, and Jens-Peter Redlich. Reversing crc - theory and practice. *HU Berlin Public Report*, 2006.

[49] MIT 6.02. *LECTURE 7: Detecting Bit Errors*. 2010.

[50] Philip Koopman and ridib Chakravarty. The International Conference on Dependable Systems and Networks. 2004.

[51] Philip Koopman. The International Conference on Dependable Systems and Networks. In *32-Bit Cyclic Redundancy Codes for Internet Applications*, 2002.

[52] Justin S. Davis Robert B. Reese (Robert Bryan). *Finite state machine datapath design, optimization, and implementation*. 2008.

[53] D.J. McElroy. High density floating gate electrically programmable rom, January 15 1980. US Patent 4,184,207.

[54] Nikolaos I. Margaris. *Theory of the Non-linear Analog Phase Locked Loop (Lecture Notes in Control and Information Sciences)*. Springer, 2004.

[55] Tom A. D. Riley, Miles A. Copeland, and Tad A. Kwasniewski. Delta-sigma modulation in fractional-n frequency synthesis. *IEEE Jurnal of Solid State Circuits*, 28(5):553–559, May 1993.

[56] Altera. *Stratix V Device Handbook Volume 2: Transceivers*, 2013.05.06 edition, May 2013.

[57] Makoto Murase. Linear feedback shift register, 1992.

[58] Tim Molteno Roy Ward. *Table of Linear Feedback Shift Registers.* Department of Physics, University of Otago, Box 56, Dunedin, New Zealand, October 2007.

[59] Xilinx. *Efficient Shift Registers, LFSR Counters, and Long Pseudo-Random Sequence Generators*, 1.1 edition, July 1996. XAPP 052.

[60] Megan Dewar and Brett Stevens. *Ordering Block Designs: Gray Codes, Universal Cycles and Configuration Orderings (CMS Books in Mathematics).* Springer, 2012.

[61] *FPGA-based Implementation of Signal Processing Systems.* Wiley, 2008.

[62] J. Rose, A. E. Gamal, , and A. Sangiovanni-Vincentelli. Architecture of field-programmable gate arrays. *Proceedings of the IEEE*, 81(7):1013 – 1029, 1993.

[63] *Altera Homepage*, January 2014.

[64] Altera. *Reducing Power Consumption and Increasing Bandwidth on 28-nm FPGAs*, wp-01148-2.0 edition, March 2012.

[65] Altera. *Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices*, 2014.01.10 edition, January 2014.

[66] *Altera Homepage: DSP*, February 2014.

[67] Altera. *White Paper: Enabling High-Performance DSP Applications with Stratix V Variable-Precision DSP Blocks*, 1.1 edition, May 2011.

[68] Altera. *Embedded Memory Blocks in Stratix V Devices*, 2013.05.06 edition, May 2013.

[69] Altera. *Stratix V Device Overview*, 2014.01.10 edition, January 2014.

[70] Altera. *High Speed Mezzanine Card (HSMC)*, 1.7 edition, June 2009.

[71] QSFP+ 10 Gb/s 4X Pluggable Transceiver Solution (QSFP10). *SFF Committee Specification* , 2013.

[72] Altera. *Stratix V GX FPGA Development Board Reference Manual*, 1.3 edition, July 2012.

[73] Helmholtz Alliance. Ha-101 physics at the terascale. *Annual Report 2011*, 2012.

[74] Christian Färber. PhD thesis, 2013. Personal contact.

[75] C. Grzesik. Fast Optical Readout for the Mu3e Experiment, 2014.

[76] INF-8074i Specification for SFP (Small Formfactor Pluggable) Transceiver, May 2001.

[77] SFF-8431 Specification for SFP+, July 2009.

[78] TrioFlex. *Online Catalog*, January 2014.

[79] Avago Technologies. *AFBR-57D7APZ Data Sheet*, January 2013.

[80] HQDP Series 228.6 mm Cable Length Signal Integrity Performance. Technical report, Samtec.

[81] Molex. *Datasheet: 106410-1003*, September 2013.

[82] Molex. *Online: QSFP+ Interconnect*, January 2013.

[83] Robert Czerwinski Dariusz Kania. *Finite state machine logic synthesis for complex programmable logic devices.* 2013.

[84] Ransom Stephens. PRBS31: Slower, costlier, worse. *EDN Network*, 2012.

[85] Ilya Narsky. Estimation of upper limits using a poisson statistic. *arXiv:hep-ex/9904025*, page 24, 1999.

[86] T. Bayes. *Facsimiles of Two Papers by Bayes.* Hafner Publ. Co., 1963.

[87] P.S. Laplace. *Philosophical Essay on Probabilities.* Dover Publications, 1951.

[88] Altera. *Floating-Point Megafunction*, 7.0 edition, November 2012.

[89] DuPoint. *DuPoint Kapton: General Specifications*, March 2012.

[90] DuPoint. *DuPoint Kapton: Summery of Properties.*

[91] X.-F. Navick, M. Carty, G. Chapellier, M.and Chardin, C. Goldbach, R. Granelli, S. Hervé, and M. et al. Karolak. Fabrication of ultra-low radioactivity detector holders for Edelweiss-II. *NIM*, 520:189–192, 2004.

[92] Universal Laser System. *PL6MW Platform.*

[93] M. Baranek J. Bartl. Emissivity of aluminium and its importance for radiometric measurement. *Measurement of Physical Quantities*, 4:31–36, 2004.

[94] Avago Technologies. *MicroPOD AFBR-77D4SZ, AFBR-78D4SZ*, March 2013.

[95] Cisco. *Cisco 40GBASE QSFP Modules*, 2014.

[96] Altera. *Dynamic Reconfiguration in Stratix V Devices*, 2013.05.06 edition, May 2013.

[97] Altera. *Extending Transceiver Leadership at 28 nm*, 2.1 edition, October 2012.

[98] Altera. *Using the On-Chip Signal Quality Monitoring Circuitry (EyeQ) Feature in Stratix IV Transceivers*, 1.2 edition, September 2010.

[99] Altera. *Debugging Transceiver Links*, 13.1.0 edition, November 2013.

[100] Altera. *Avalon Streaming Data Pattern Generator and Checker Cores*, v9.1.1 edition, January 2010.

Acknowledgements

I would like to express my thanks to everyone that accompanied me during my time in the Mu3e group in Heidelberg. It was always a pleasure to work in such a atmosphere.

First of all, I would like to thank Dr. Niklaus Berger who made it possible to carry out a thesis in Heidelbergd and for his support during the whole time. My special thanks goes to Dr. Dirk Wiedner for his help, support and discussions.

I thank Carsten Grzesik with whom I carried out most of the work. Numerous joint discussions have brought us a long way.

Last but not least, I would like to thank Prof. Dr. Günther Dissertori who supported the idea of this ETH thesis at the University of Heidelberg from the beginning.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Eigenständigkeitserklärung

Die unterzeichnete Eigenständigkeitserklärung ist Bestandteil jeder während des Studiums verfassten Semester-, Bachelor- und Master-Arbeit oder anderen Abschlussarbeit (auch der jeweils elektronischen Version).

Die Dozentinnen und Dozenten können auch für andere bei ihnen verfasste schriftliche Arbeiten eine Eigenständigkeitserklärung verlangen.

Ich bestätige, die vorliegende Arbeit selbständig und in eigenen Worten verfasst zu haben. Davon ausgenommen sind sprachliche und inhaltliche Korrekturvorschläge durch die Betreuer und Betreuerinnen der Arbeit.

Titel der Arbeit (in Druckschrift):

Fast Optical Readout of the Mu3e Pixel Detector

Verfasst von (in Druckschrift):

Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich.

Name(n):	Vorname(n):
Corrodi	Simon

Ich bestätige mit meiner Unterschrift:

- Ich habe keine im Merkblatt „Zitier Knigge" beschriebene Form des Plagiats begangen.
- Ich habe alle Methoden, Daten und Arbeitsabläufe wahrheitsgetreu dokumentiert.
- Ich habe keine Daten manipuliert.
- Ich habe alle Personen erwähnt, welche die Arbeit wesentlich unterstützt haben.

Ich nehme zur Kenntnis, dass die Arbeit mit elektronischen Hilfsmitteln auf Plagiate überprüft werden kann.

Ort, Datum	Unterschrift(en)
Zurich, 1.4.2014	S.

Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich. Durch die Unterschriften bürgen sie gemeinsam für den gesamten Inhalt dieser schriftlichen Arbeit.