

Bachelor's Thesis

Testing And Programming Of A Data Aquisition Component For The New ATLAS Pixel Detector Front-End Electronics

prepared by

Alrik Stegmaier

from Rostock

at the II. Physikalischen Institut

Thesis number: II.Physik-UniGö-BSc-2012/08
Thesis period: 16th April 2012 until 23rd July 2012
First referee: Priv.Do. Dr. Jörn Große-Knetter
Second referee: Prof. Dr. Arnulf Quadt

Abstract

Due to the high LHC luminosity, the current ATLAS Pixel Detector experiences radiation damages. To provide a good detector performance in the next years of operation with even increased LHC luminosity the Pixel Detector requires an update. Part of this upgrade is the Insertable B-Layer (IBL) project which is planned to be installed in 2013. This upgrade also makes a new readout-chain of the detector necessary. One part of this chain is the electrical Back Of Crate (eBOC) card which is an alternative to the optical BOC for test setups. Within this thesis, an existing prototype of the eBOC was improved and further tested. Moreover an Error-checker was developed to verify a proper signal handling of the eBOC on startup.

Keywords: Physics, Bachelor thesis, ATLAS Pixel Detector, IBL-Upgrade, readout-chain, eBOC, VHDL-Programming

Contents

1. Introduction	1
2. The Pixel Detector of ATLAS	5
2.1. The ATLAS Experiment	5
2.1.1. The Inner Detector	6
2.1.2. The Outer Detector	6
2.2. The Pixel Detector	7
2.3. The readout-chain of the Pixel Detector	9
2.4. IBL-Upgrade of the Pixel Detector	11
2.5. The readout-chain in the laboratory	12
3. The electrical Back of Crate Card	15
3.1. Description of the hardware	15
3.1.1. The electrical layout of the board	15
3.1.2. The FPGAs	17
3.2. Programming of the FPGAs	19
3.2.1. The VHDL-language	19
3.2.2. The FPGA firmware	20
4. Upgrades of the eBOC	23
4.1. Production and testing of a new prototype-board	23
4.2. Changes to the hardware	25
4.3. Changes to the FPGA firmware	26
4.3.1. The Error-checker	27
4.4. Tests and validation	28
4.4.1. Testing-tools	28
4.4.2. Tests with the ROD	28
4.4.3. Tests with the MCC-Emulator	29
4.4.4. Stress-testing the eBOC	29

Contents

5. Summary and outlook	31
5.1. Summary	31
5.2. Outlook	31
A. Appendix	33

Nomenclature

Acronym	Meaning
ATLAS	A Toroidal LHC Apparatus
BOC	Back-Of-Crate card
CERN	European Organization for Nuclear Research
CMS	Compact Muon Solenoid
DDR-SDRAM	Double Data Rate Synchronous Dynamic Random Access Memory
DFF	D-type Flip-Flop
eBOC	electronic BOC
FA	Full Adder
FE	Front-End chip
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
IBL	Insertable B-Layer
I/O	Input/Output
LED	Light-Emitting Diode
LHC	Large Hadron Collider
LUT	Look-Up Table
LVDS	Low Voltage Differential Signaling
MCC	Module Control Chip
Mux	Multiplexer
PECL	Positive Emitter-Coupled Logic
RAM	Random-Access Memory
ROB	Read-Out Buffer
ROD	Read-Out Driver
ROS	Read-Out System
RTL	Resistor-Transistor Logic
SBC	Single Board Computer
SCT	Semiconductor Tracker

Nomenclature

Acronym	Meaning
TIM	Timing, Trigger and Control Interface Module
TOT	Time Over Threshold
TRT	Transition Radiation Tracker
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VME	Versa Module Eurocard

1. Introduction

The world is governed by four fundamental interactions. These fundamental interactions are the gravitational, electromagnetic, weak and strong interaction. So far, only the electromagnetic, weak and strong interaction can be described by one theory. This is the so called Standard Model of particle physics. Gravity cannot be described by any quantum theory. The particles described by this theory are shown in Figure 1.1.

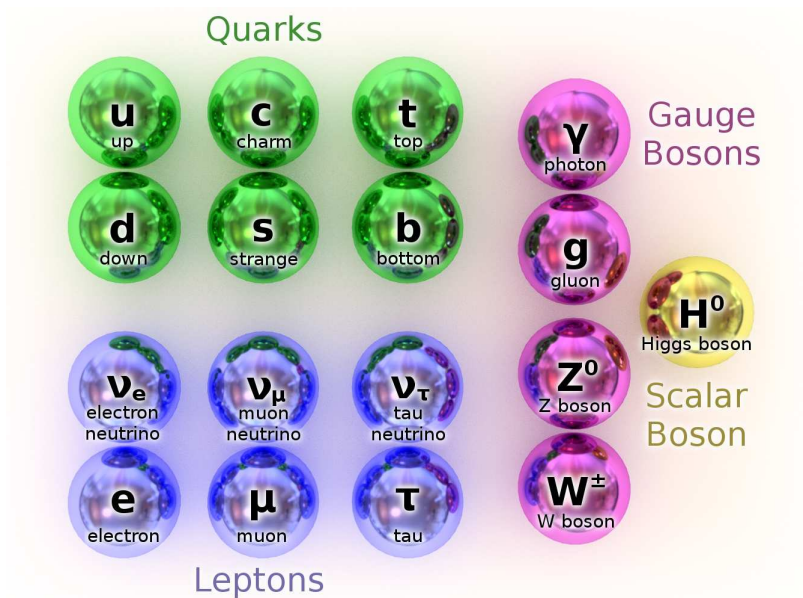


Figure 1.1.: The elementary particles of the Standard Model. Not shown are the anti-particles of the quarks and leptons and the different colour charges of the gluon and quarks.

The theory is able to produce very accurate predictions but it is still incomplete. Some of the problems of the Standard Model are that it does not describe gravity and has circa 20 parameters which can only be obtained experimentally [1]. Furthermore, no explanation for neutrino oscillations and their non-zero mass can be derived from the Standard Model. Another problem is the absence of particles which could account for dark matter and dark energy. Other models were developed to improve these shortcomings but all of them remain unproven until today [2, 3].

1. Introduction

To increase our understanding of nature and improve physics theories, experiments are performed. One way to gain insight into nature is to collide particles with high energies in a particle collider and analyse the decaying particles. The currently largest and most powerful collider is the Large Hadron Collider (LHC) located at CERN, in Geneva, Switzerland. It is an underground ring-accelerator with a circumference of about 27 km. In 2012, the center of mass energy in this particle collider reached 8 TeV [4, 5]. The luminosity produced at the collider was sufficient to find a candidate for the last elementary particle predicted by the Standard Model, the higgs-boson [5, 6]. In the future, the accelerator will be used to measure its properties and search for new physics beyond the Standard Model. For this precise task two multipurpose detectors are installed at the LHC: the ATLAS-detector and the CMS-detector. Those detectors are highly complex and require sophisticated readout-electronics. This bachelor thesis will focus on the ATLAS experiment.

After several years of LHC operation, the radiation damage took its toll on the sensitive detector material of the innermost detection-layers of the ATLAS experiment. This leads to a decrease in resolution and makes a detector upgrade necessary. In 2013, the LHC will have a long shutdown for upgrades. After the shutdown, the LHC is scheduled to run with an increased center of mass energy of 14 TeV and an increased luminosity (increase from $\mathcal{L} \approx 6 \times 10^{33} \text{ cm}^{-2}\text{s}^{-1}$ to $\mathcal{L} \approx 10^{34} \text{ cm}^{-2}\text{s}^{-1}$). The ATLAS detector will have to be upgraded to handle this additional stress. During the upgrades, a new detection layer will be inserted into the inner detector. This new detection layer with new sensors will guarantee continued high resolution and provide a backup for further degradation of the inner detector due to radiation. An upgrade of the readout-electronics is also required to cope with the expected higher data event rates and to be able to work with the new sensors in the new detection layer. Currently, the data is sent to the readout system outside the detector without any digital encoding. This will change after the upgrade and requires the development of readout-electronics that are able to handle the new encoding. To test the new upgrades of the inner detector prior to their installation, the complete readout-chain of the innermost detector of the ATLAS experiment has to be recreated in the laboratory [7].

This thesis will give an overview of the ATLAS experiment and the readout-chain in Chapter 2. Section 2.1 explains the ATLAS experiment in more detail. The Pixel Detector will be further described in Section 2.2. The readout-chain in the Pixel Detector will be explained in Section 2.3 and in Section 2.4 a closer look at the planned upgrades will be given. The readout-chain for laboratory testing will be explained in Section 2.5. Additionally, the eBOC, a component of the laboratory readout-chain, will be introduced

in Chapter 3. Upgrades and tests of this board will be described in Chapter 4. Finally an outlook for future work will be presented in Chapter 5.

2. The Pixel Detector of ATLAS

2.1. The ATLAS Experiment

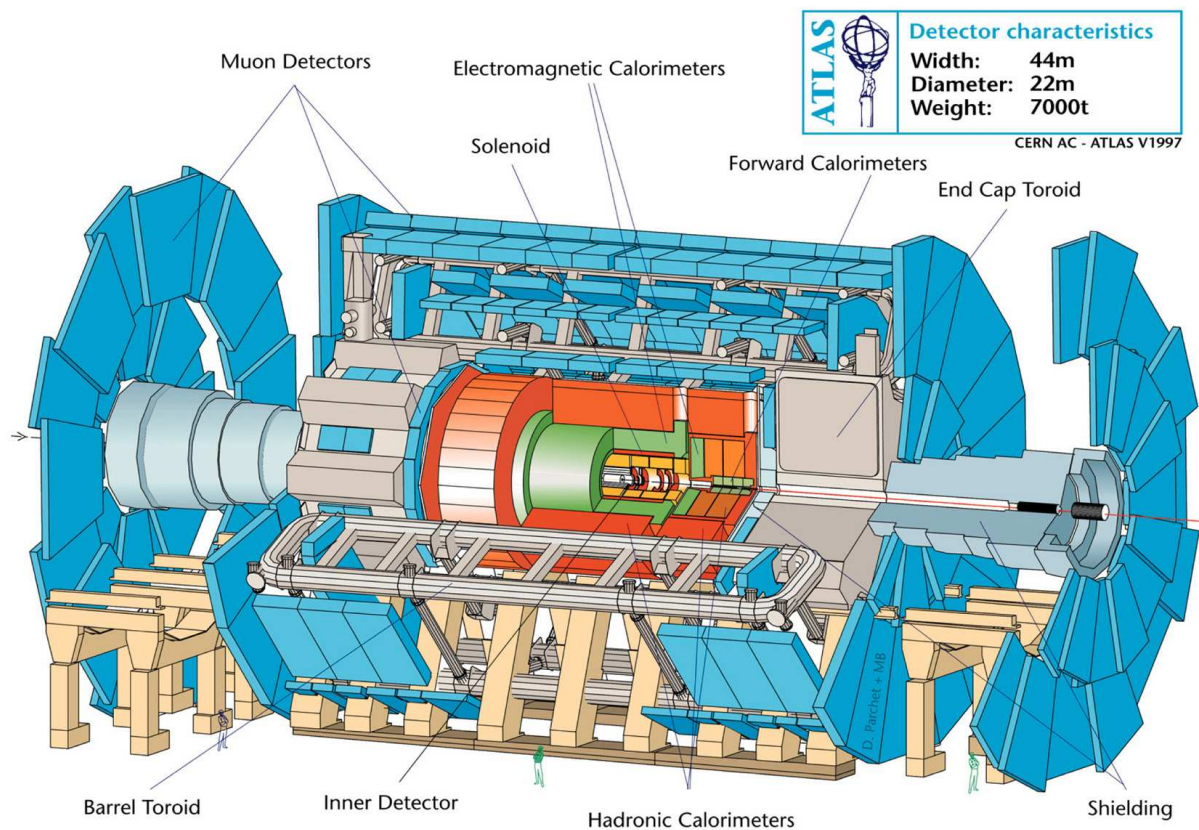


Figure 2.1.: A schematic view of the ALTAS-detector of the LHC¹.

The ATLAS experiment is a modern multipurpose detector with multiple layers of several subdetectors. The subdetectors are specialized on particular measurements to increase the performance of the complete detector. The layout of the whole detector is an onion-model and is commonly used for particle detectors. The shape of the detector can be described as being cylindrical with detection layers at the barrel and the end-caps. Additionally, specialized forward detectors far from the collision point are used for

¹cern.ch

2. The Pixel Detector of ATLAS

measurement of elastic scattering and luminosity of the beam particles. Both protons and lead ions can be used as beam particles. The ATLAS experiment is currently the biggest multipurpose detector in the world with overall length of 44 m and a diameter of 22 m [4].

2.1.1. The Inner Detector

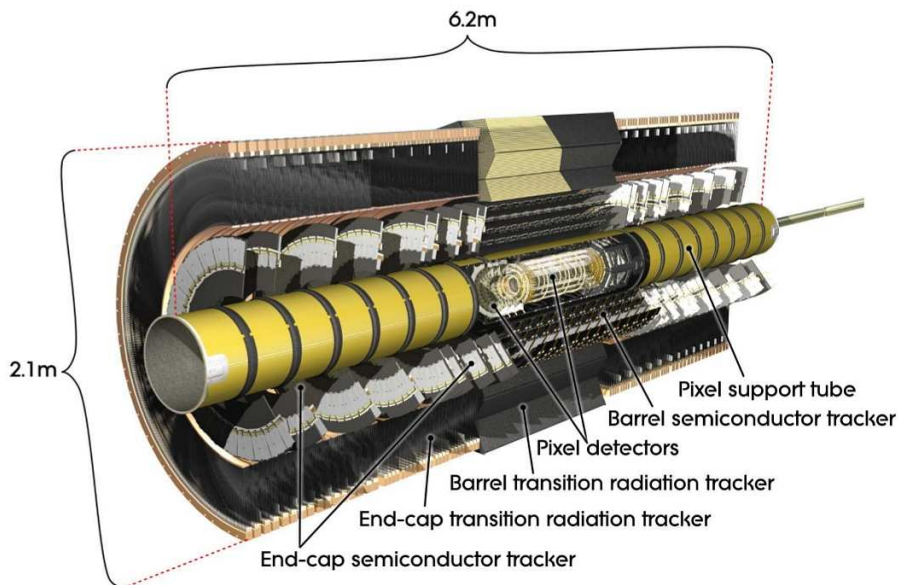


Figure 2.2.: A schematic view of the inner detector of the ATLAS experiment [8].

The innermost layers of the ATLAS experiment are used for tracking charged particles and particle identification. They consist of the Pixel Detector (see Section 2.2) very close to the collision point, the Semiconductor Tracker (SCT) and the Transition Radiation Tracker (TRT). The Pixel Detector and the SCT are both silicon based semiconductor detectors and have a high spatial resolution. The TRT is used to identify charged particles by the transition radiation they produce inside the subdetector. The inner detector also uses a large magnetic field of 2 T strength to make a reconstruction of the momentum of the passing charged particles possible. This magnetic field is created by the solenoid magnet inside the detector [4].

2.1.2. The Outer Detector

The outer detector consists of the calorimeters and the muon detectors. Calorimeters are used for measuring the energies of the particles by stopping them. There are two types of

calorimeters in the ATLAS experiment: the electromagnetic and the hadronic calorimeter. The electromagnetic calorimeter is closer to the collision point and is specialized in measuring the energies of electromagnetically interacting particles such as photons and charged particles. The hadronic calorimeter is designed to stop the remaining hadrons that were able to get through the electromagnetic calorimeter.

The outermost subdetector of the ATLAS experiment is the muon detector. The outer detector uses a non-homogeneous magnetic field with a strength of about 0.5 T for measurements of muon momenta. It is generated by a large toroid magnet. Its non-homogeneous field was mostly chosen due to cost reduction as it does not decrease the detector performance [4].

2.2. The Pixel Detector

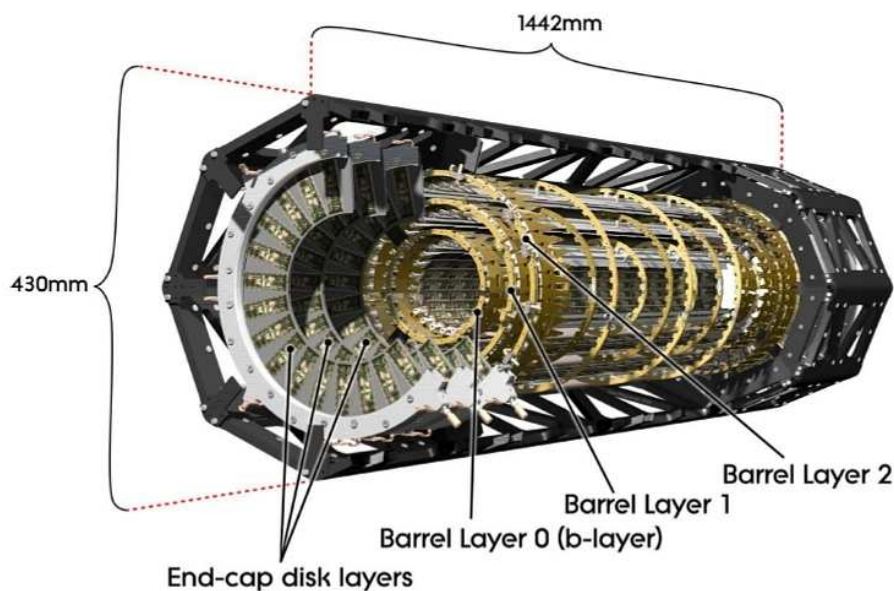


Figure 2.3.: A schematic view of the Pixel Detector of the ATLAS experiment [8].

The Pixel Detector of the ATLAS experiment is a silicon based semiconductor detector that is used to reconstruct the tracks of charged particles with a high degree of precision. A good spatial resolution is necessary for a precise momentum reconstruction. For this the magnetic field inside the inner detector can be used. Another important task of the Pixel Detector is the reconstruction of primary and secondary vertices since more than one collision is happening per proton-proton interaction inside the detector. The detected particles have to be linked to the collision they originated in.

The Pixel Detector consists of 1744 modules that are arranged in three barrel layers

2. The Pixel Detector of ATLAS

and two end-caps. Figure 2.3 shows a schematic view of the Pixel Detector. The three module layers are called b-layer (or layer 0), layer 1 and layer 2. The b-layer has its name from the importance of this layer in the process of b-tagging². The composition of the layers is shown in Table 2.1. The modules in each layer are mounted on mechanical support and cooling structures called staves in the barrel region and disk sectors in the end-caps. 13 modules are mounted on each stave and all the staves in the barrel region look identical while the disk sectors of the end-caps contain six modules each [8].

Layer Number	Mean Radius [mm]	Number of Staves	Number of Modules	Number of Channels	Active Area [m ²]
0	50.5	22	286	13,178,880	0.28
1	88.5	38	494	22,763,520	0.49
2	122.5	52	676	31,150,080	0.67
Total		112	1456	67,092,480	1.45

Disk Number	Mean z [mm]	Number of Sectors	Number of Modules	Number of Channels	Active Area [m ²]
0	495	8	48	2,211,840	0.0475
1	580	8	48	2,211,840	0.0475
2	650	8	48	2,211,840	0.0475
Total one endcap		24	144	6,635,520	0.14
Total both endcaps		48	288	13,271,040	0.28

Table 2.1.: Composition of the detection layers and their distance from the beam axis [8].

Each module consists of a sensor, 16 Front-End (FE) chips and a Module Control Chip (MCC). The FE-I3 is the current model of the Front-End chip inside the detector. All modules are functionally identical. The sensors consists of $50 \times 400 \mu\text{m}^2$ and $50 \times 600 \mu\text{m}^2$ sized pixels and have 46,080 channels each. The sensors have a n- and a p-doped layer of silicon placed on top of each other. This np-junction creates a zone with no free charge carriers so that when a voltage is applied no current passes through this zone. A voltage is applied to increase the effect. If a charged particle passes through the sensor, a certain amount of free charge carriers is generated that moves to the respective electrodes. This produces a current that can be amplified and measured. Detector-noise is reduced by cooling the individual modules to $-10 \text{ }^\circ\text{C}$. The segmented electrodes on

²B-tagging is the process of marking the decay products of a b-quark. This is possible since the b-quark does not decay as fast as other heavy quarks and can therefore travel a short distance from its point of origin. Reconstructing the tracks of the decay-products precisely is of key importance at this process.

one side enable a position measurement of the event. The FE-chips are used to digitize the signal and send it to the MCC. For this task each FE-chip has 2880 individual charge sensitive analog circuits with a digital readout. This means there is one electronic cell per pixel. Each analog circuit includes a preamplifier, a second amplifier and a differential discriminator. The charge that was collected on each pixel is integrated by the preamplifier by using a feedback capacitor which is constantly discharged by a feedback current. In the discriminator, the signal is compared to a threshold value. This determines whether the signal is recognized as a logical ‘1’ or ‘0’. The threshold can be adjusted to improve the detector performance. The actually measured quantity is not the collected charge in the detector but the time over threshold (TOT). This time is proportional to the collected charge [8].

The MCC combines the event data from the module’s FE-chips, serializes the data and sends the data with a bandwidth of 40 Mbit/s to the Opto-Board where the data is converted to an optical signal to send it to the off-detector part. More details of the readout-chain will be discussed in Section 2.3. The detector has a resolution of about 12 μm in the $R - \varphi$ -direction and 100 μm in the Z -direction. Its efficiency is about 97% [8].

The Pixel Detector is built from very lightweight materials to reduce the scattering-cross-section of the material for traversing particles, specifically multi-scattering. This also reduces the fraction of scattering events with a high energy transfer, which would cause a problem for accurate energy and momentum measurement of the particles. These are reasons why carbon was chosen to be used in the frame of the Pixel Detector. Thus the Pixel Detector being 1.4 m long and having a radius of 22 cm weighs only 4.4 kg.

2.3. The readout-chain of the Pixel Detector

An overview of the electric wiring and optical connections of the readout-chain of the Pixel Detector is given in Figure 2.4. It consists of the Module, the Opto-Board, the BOC, ROD, SBC, TIM and the ROS. An overview of the Module was already given in Section 2.2. Data from the Modules is only sent after receiving a trigger. The MCC is also responsible for configuring the individual FE-chips. Both the off-detector analysis and the Modules have one counter that increases its value once every 25 ns and is used as a clock. This counter can be used to determine the 25 ns window in which an event happens. Another counter is increased with the trigger and reappears in the event data from the Modules. With this counter the data can be linked to the individual trigger. Both counters have to be synchronized via specific command signals from the ROD [8].

In the Opto-Board the data-stream is converted into an optical signal. This is necessary

2. The Pixel Detector of ATLAS

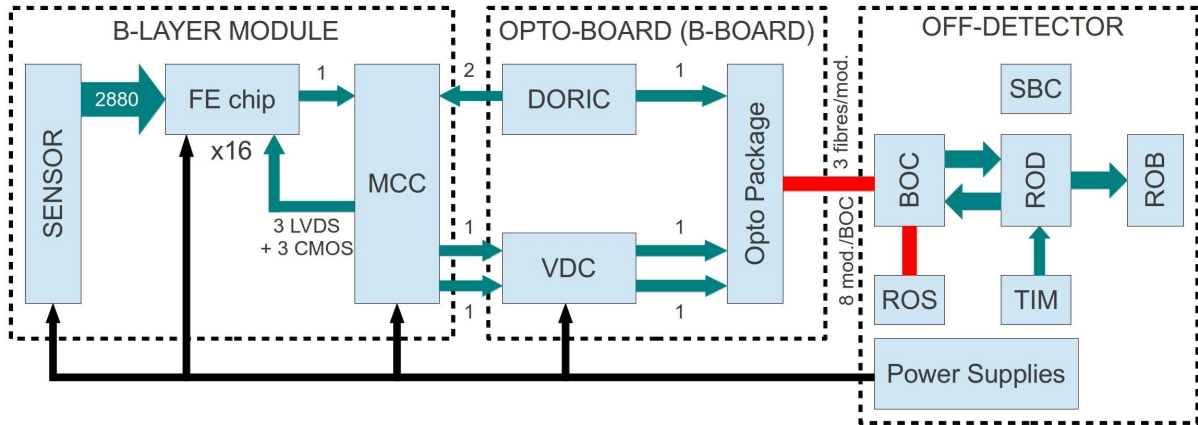


Figure 2.4.: A schematic view of the readout-chain of the Pixel Detector.

since the distance between the detector and the off-detector readout-chain is about 80 m. The data from the Opto-Board is converted back to an electric signal inside the BOC. The BOC is also used to generate the clock for the Modules and the ROD. Each BOC is responsible for eight Modules but is only connected to a single ROD. The ROD-BOC pair is placed inside a larger support structure called VME-crate. Up to 16 of these pairs can be put into one crate. In the crate the ROD and the BOC are connected via three 160 pin connectors. On the ROD an event can be built and sent via the S-Link connectors on the BOC to the Read-Out System (ROS). Histogramming and data analysis functionality can also be enabled on the ROD. They also offer analysis-capability. The information gained this way can be used during data taking for monitoring purposes. It can also be used during calibration for determining and adjusting the thresholds of the FE-chips for data taking. The Timing, Trigger and Control Interface Module (TIM) is the system that sends the triggers and initializes the production of trigger sequences in the ROD. The TIM also handles the generation of configuration commands for the Modules on the ROD and can be used to control the readout-chain. One Single Board Computer (SBC) per crate can be used for communication between the RODs and a PC. This is mainly used for configuration and scans. The final readout of the data from the ROD is done by sending it through an S-Links connection on the BOC to the Data-Aquisition (ATLAS-DAQ) [8].

Not all data that is collected inside the Pixel Detector will be used for analysis. Only data that was triggered is sent to the readout system. This is due to the fact that each event inside the detector produces about 1-2 MB of data. With 40 million bunchcrossings per second where collisions can occur and about 23 collisions per bunchcrossing, this would produce about 920-1840 TB of data per second. No computer system could handle this large amount of data properly. This is why several triggers are introduced that sort out data that is not important for analysis. There are three trigger levels. The first one

is created by having a preselection of events by easily determinable detector signatures, such as the presence of a muon. This level 1 trigger is hardware based and uses mostly information from the calorimeters and the muon system. The level 2 and level 3 triggers are software based and use information from all the detectors. This trigger system reduces the data to about 200 recorded events per second. This amount of data can be handled with modern computers and is used in the analysis [9, 10].

2.4. IBL-Upgrade of the Pixel Detector

As mentioned in Chapter 1, the luminosity of the LHC will be increased from $\mathcal{L} \approx 6 \times 10^{33} \text{ cm}^{-2}\text{s}^{-1}$ to $\mathcal{L} \approx 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ in 2013. Together with the increase of the center of mass energy from 8 TeV to 14 TeV, more events of higher energy are expected inside the inner detector. This means the event data needs to be sent out of the detector at a higher rate. Currently, all the data from the FE-I3-Modules of the Pixel Detector is sent out at 40 MBit/s, 80 MBit/s and 2×80 MBit/s. This will be increased to 160 MBit/s in order to cope with the larger amount of data that will be produced. Another aspect that affects detector performance is the fact that the active part of the Pixel Detector is subjected to a large amount of fluence. This fluence causes damage even though the Modules of the Pixel Detector are radiation hard. Only comparatively few pixels of the detector have actually been damaged by radiation so far but their number is expected to rise. A larger loss of pixels is caused by the temperature changes. As mentioned in Section 2.2 the Modules are cooled down to $-10 \text{ }^\circ\text{C}$ to reduce detector noise. When the cooling is switched off the detector will warm up to room temperature. This causes thermal stress in the connections from the sensors to the read-out electronics on the Modules. These connections are called bump-bonds. Decreased quality of connectivity is one of the main reasons for losing pixels and therefore for reduced detector performance [7].

To solve this issue a new detection layer with an average distance of only 33 mm to the beam axis will be introduced in 2013. The new detection layer adds redundancy to the Pixel Detector. This redundancy protects the Pixel Detector against a loss of resolution when more and more pixels of the older layers of the Pixel Detector start to fail. The upgrade is called Insertable B-Layer Upgrade or IBL-Upgrade. It is one of the upgrades to prepare the detector for running another few years with higher beam luminosity and energy. This additional layer introduces a newly developed sensor and read-out electronics. One Module in the Insertable B-Layer will then only contain 2 FE-chips that are able to reach 26,880 pixels that are smaller than the ones already installed in the Pixel Detector. Another design for the new Modules has only 1 FE-Chip per Module. The smaller pixels

2. The Pixel Detector of ATLAS

are necessary since the same spatial resolution has to be reached in the new innermost layer as the outer layers while being closer to the beam axis. The smaller distance from the beam axis also requires new technology that is more resistant to radiation. One of the measures to increase radiation resistance is to save the configuration for the FE-chips three times and then choose the majority of these values.

The active area of the Modules will be increased from about 75% to about 90% on the modules. The active area is the area on a chip that is sensitive to particles. New FE-chips are also employed. They are called FE-I4 and already have all functionality of the old MCCs. This means that no MCCs are required in the new detector layer. Another important aspect is the introduction of 8B/10B encoding for digital signals. This encoding reduces the chance of an undiscovered bitflip during detector read-out [7].

All these upgrades also require the rest of the read-out chain to be updated. This includes the production of a new BOC that is able to decode the digital encoding. The ROD only works with data that has a rate of 40 MBit/s. It is also important to notice that therefore the data transfer rate needs to be slowed down from 160 MBit/s to 40 MBit/s per connection-pin to the ROD. This reduction in data-rate is achieved by splitting the single-line 160 MBit/s signal to four lines with 40 MBit/s on the BOC. The new FE-I4 also has a different command structure than the FE-I3 and therefore upgrades to the ROD are required [8].

2.5. The readout-chain in the laboratory

A readout-chain in the laboratory is required to test the various hardware upgrades during the process of development. Undiscovered hardware errors would cause a lot of problems in the detector and therefore thorough testing of all the hardware is required. Using a readout-chain in the Laboratory also enables the various universities to test their hardware at the place of development. If all the testing in the laboratory is done, the new hardware can be submitted to production.

In the laboratory, no long distances between the modules and the crate exist. Together with the fact that optical connections are difficult to set up, none of these connections are used in the laboratory. This means that no Opto-Boards need to be used. The BOC is therefore exchanged by a fully electronic version: the electronic BOC (eBOC). An eBOC was already developed for the FE-I3. With the FE-I4 chip a new prototype needs to be developed. For testing purposes the triggers of the TIM can also be generated on the ROD.

For the development of the FE-I4, the MCC-Emulator was developed. Figure 2.6 shows

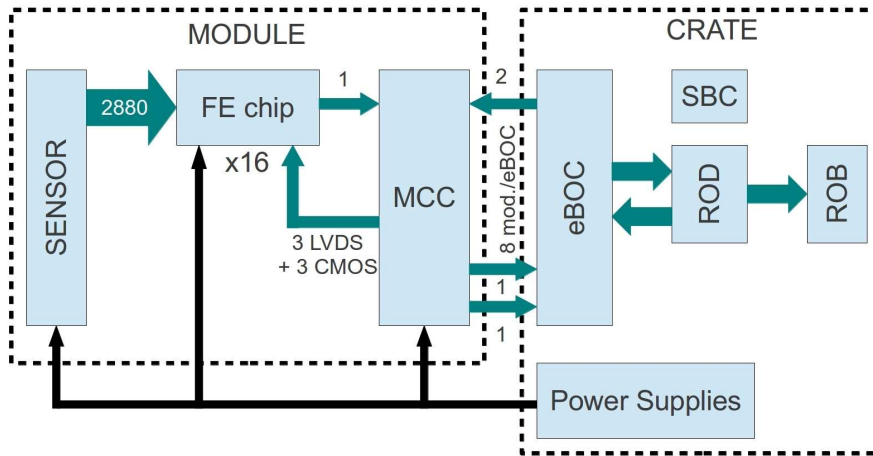


Figure 2.5.: Layout of the readout-chain in the laboratory.

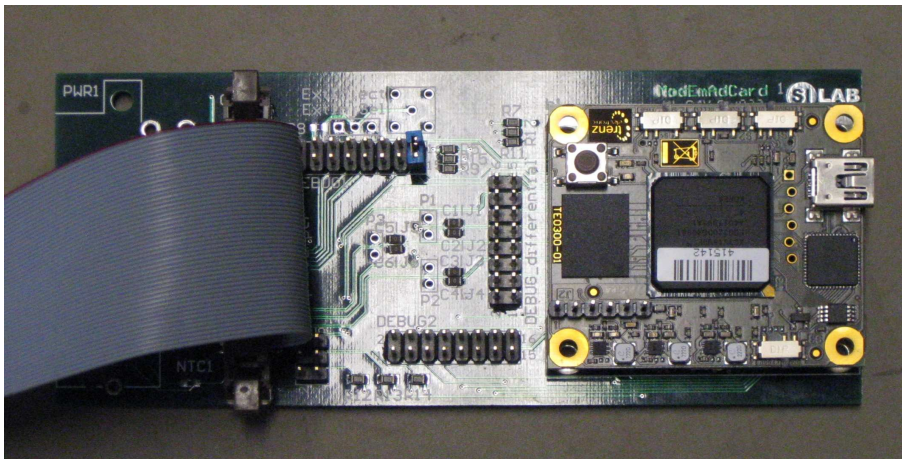


Figure 2.6.: A picture of the MCC-Emulator.

the MCC-Emulator. The ribbon cable on the left of the picture is used to connect the adapter board of the Emulator to the eBOC. On top of the adapter board is the MCC-Emulator with its FPGA. This is a small board that is able to simulate a module. It is currently configured to simulate a Module with FE-I3 chips that uses 160 MBit/s data-rate and 8B/10B encoding. That enables testing of the decoding algorithms of the readout-chain without adopting it to the new command structure of the FE-I4 chips. The data rate of the MCC-Emulator can also be switched to 40 MBit/s and 80 MBit/s and in each speed mode the board can be configured to transmit an unencoded signal. The unencoded signal can be used for testing purposes. This is useful for reliability testing and discussed in detail in Subsection 4.4.3.

3. The electrical Back of Crate Card

As mentioned in Section 2.5, there is a prototype eBOC for the development and testing of the FE-I3 chip. However, this board is not able to cope with the data-rates and digital encoding of the FE-I4 chip. Therefore a new prototype needed to be developed with which testing of the FE-I4 readout-chain is possible in the laboratory. This development was started in 2010 with the bachelor thesis of Benjamin von Ardenne [11] and led to the construction of one prototype in 2011 and another one in this thesis in 2012. Continued improvements of the prototypes are necessary as the design matures. The design of the board will be described in this chapter.

3.1. Description of the hardware

3.1.1. The electrical layout of the board

On the left side of the board there are two connectors with 160 pins each for communication with the SBC and the ROD via the crate. The bottom one is also used for voltage supply and transmission of a 40 MHz clock from the eBOC to the ROD. The two supply voltages are +3.3 V and +5.0 V while the only purpose of the +5.0 V line is to drive a PECL driver for transmitting the clock signal. PECL stands for Positive Emitter-Coupled Logic and describes a standard for transmitting bit streams as a differential signal. While the ROD is capable to generate its own clock, the eBOC has the capability to drive the ROD with the eBOC clock to ensure proper communication. This feature can be deactivated on the eBOC by removing a jumper switch. In the center of the board there is an oscillator that creates the 40 MHz clock that drives the system and a transformer that creates another supply voltage of +1.2 V for both FPGAs. The FPGAs will be discussed in Subsection 3.1.2. The transformer can be switched off by removing a jumper bridge. This is useful for removing the firmware of the FPGAs on the board. The right side of the eBOC hosts the connectors for the modules. With these connectors a supply voltage is connected to the Modules, a clock and the commands from the ROD are transmitted and data can be received from the modules. While all the data connections to the modules

3. The electrical Back of Crate Card

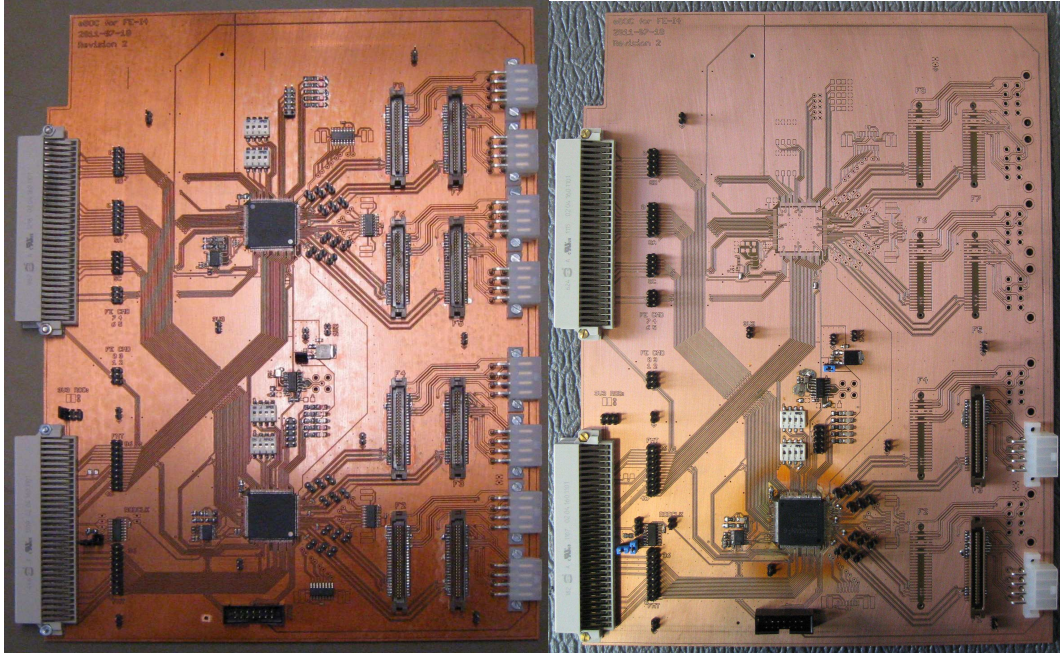


Figure 3.1.: A picture of the two prototypes of the eBOC. The new one is on the left and the old one is shown on the right. This picture of the old prototype was taken before the damaged southern FPGA was removed and a new FPGA on the northbridge was placed on the board.

are designed as low voltage differential signals (LVDS) only the clock is transmitted to the ROD as a differential signal with a voltage difference of +350 mV as a logical '1' and a difference of -350 mV as a logical '0'. All other data lines use 0V as a logical '0' and +3.3 V as a logical '1'.

There are also several pins on the board for testing. Every data-line on the board can be probed via an additional pin on the board. All the potential surfaces on the board also carry pins for testing. Together with a multimeter or oscilloscope the operation of the board can be tested while the eBOC is performing its normal tasks. While this does not interfere with any of the communication of the eBOC with the rest of the readout-chain it has to be noted that the probe of the oscilloscope has an internal resistivity. This causes reflections of the transmitted signals on differential lines that show up on the oscilloscope as additional oscillations.

As seen in Figure 3.1 there are two prototypes for the eBOC. The old prototype uses only one FPGA and was constructed by Johannes Agricola for a bachelor thesis [12]. It was originally equipped with one FPGA on the southbridge. This FPGA took damage when it was connected to a +5.0 V potential and was thus removed from the board by my predecessor Johannes Agricola [12]. To replace its functionality a new FPGA was mounted on the northbridge. Unfortunately the board took damage in the process of

removing the FPGA though most of its functionality remained unaffected. Details can be found in the bachelor thesis of Johannes Agricola [12] or in Chapter 4.

Each FPGA on the eBOC has access to five debugging ports and five LEDs. The LEDs can be used to display the status of the Error-checker that is discussed in Section 4.3.1. The debugging channels are data lines that can be used to communicate internal errors to the user and send data into the FPGAs.

3.1.2. The FPGAs

Two Spartan-6 FPGAs are mounted on the board whereas each is responsible for four Modules. Spartan-6 FPGAs are a series of low cost FPGAs from the vendor Xilinx. An FPGA is a programmable logic component that enables the user to generate arbitrary integrated logic functionality.

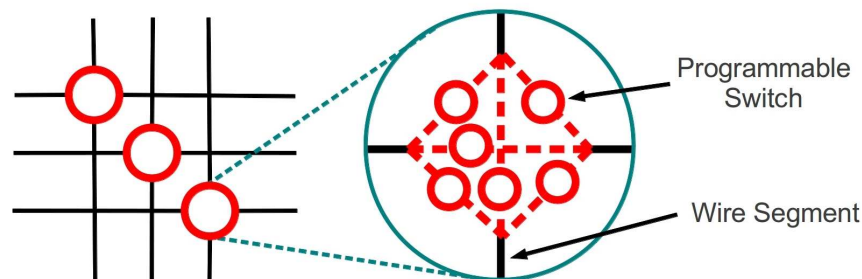


Figure 3.2.: Schematic representation of switch box topology on a modern FPGA.

FPGAs consist of configurable logic blocks that can be flexibly wired together. For this purpose a net of wires between the logic blocks exists that can be linked together via switches as shown in Figure 3.2. This also enables the connection of I/O pads to the logic blocks. A special wiring net for transportation of clock signals exists since clocks need to be synchronized all over the FPGA to guarantee proper communication. There are also dedicated logic blocks for generating all the required clock signals from one external clock source [13].

The logic blocks of a modern FPGAs consists of a few logic cells that in turn consist of 4-input look up tables (LUTs) and a full adder (FA), a D-type flip-flop (DFF) and several muxes. Figure 3.3 shows a simple logic cell of a modern FPGA. Some vendors recently started to replace the 4-input LUTs by 6-input LUTs to increase performance. Theses LUTs consist of a register that is able to save a specific output for each input combination that is provided in the four (or six) input channels. In normal mode the output of the LUTs is combined in a multiplexer (mux) while alternatively also adding

3. The electrical Back of Crate Card

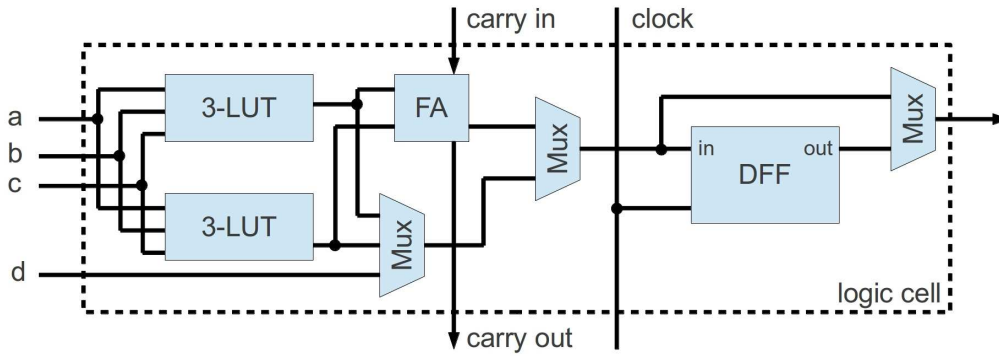


Figure 3.3.: Diagram of logic cell on a modern FPGA.

the signals in the FA is possible. A mux is a logic circuit that has at least two data-lines as an input, at least one selection line and less outputs than inputs. The selection line selects which inputs are transmitted to the output lines. The FA is also able to use carry. For this the carry of another logic cell can be transmitted to the FA and the FA can also transmit the carry created to another logic cell. The flip-flop is then used in conjunction with the clock-signal to delay the output until it can be sent synchronously to the clock. A flip-flop is a circuit that can have two states that can be used to store information. In a real logic cell not all the components that are described as being separate here need to be truly separated. For instance: The FA can be realized in part or completely as LUTs to increase performance or save space [13].

Different logic blocks with specialized structure are usually employed on a modern FPGA together with dedicated logic cores that can perform certain tasks a lot faster than the configurable logic on the FPGA. One logic core that is on a lot of FPGAs is a multiplier. Multiplications are often needed in signal processing, one of the main tasks of FPGAs. The configurable logic blocks of modern FPGAs are usually also specialized for certain tasks. There are blocks which are able to address memory faster than others or act as large registers. Other blocks perform arithmetic operations fast while another type of logic block may be specialized in addressing dedicated logic cores efficiently. Another commonly used block is RAM. The RAM is used to store data similar to a register. On modern FPGAs DDR-SDRAM is employed that enables the transfer of 2 bits per RAM clock cycle. Using 2 clocks for the RAM is therefore necessary. The phase difference of the two clocks needs to be 180° . This is done by using an inverted or delayed clock as the second clock. On Spartan-6 FPGAs there is also dedicated RAM that is used to store bits for output called ODDR2 [14].

3.2. Programming of the FPGAs

The development of the FPGA firmware is done by using a Hardware Description Language (HDL). The HDL will be translated into Resistor-Transistor Logic (RTL) for programming an FPGA. The RTL-design can then be mapped to the resources that will be available on the hardware. Several optimization runs are performed to ensure maximal performance of the design on the hardware. Complex analysis of the expected pathlengths of the design will then be carried out to warn the user about routing problems on the hardware. If all tests pass a bitstream will be generated. The bitstream can be sent over special programming pins into the FPGA where the logic circuits will be generated accordingly. This includes the programming of the switches and LUTs.

Most FPGAs do not have an internal permanent memory. This means the programming will be lost after an interruption of the power supply of the FPGA. It is possible to link the FPGA to a permanent memory such as a flash memory that reprograms the FPGA once the power is restored. Some vendors produce special flash memory exactly for this task.

3.2.1. The VHDL-language

VHDL stands for Very High Speed Integrated Circuit Hardware Description Language. It is specialized on describing the hardware on a more abstract level than pure RTL hardware description. VHDL is only one of two HDLs commonly used for this task. The other one is called Verilog. VHDL was developed as a standard by the US Department of Defense to document the functionality of the integrated circuits that were supplied by different vendors. In contrast to other high level programming languages such as *C++* or *Java*, VHDL does not generate a bytecode that is used as a chain of commands that will be carried out in the hardware but rather a set of instructions for generating connections between logic blocks. A VHDL code consists of two parts: a part describing an entity and a part describing the architecture of the entity.

An entity defines the ports that the VHDL code will be using. It also describes the connection standards used with these ports and whether they will be used as inputs, outputs or both. The architecture of the entity describes the operations that will be performed. Due to the nature of VHDL all operations inside the architecture part of the VHDL code will be carried out simultaneously in the hardware. For sequential operations processes can be used. They are initialized inside the architecture part of the code as well. Another important concept is the usage of signals in an FPGA program. Signals can be used to communicate values inside the architecture part for example between processes or

3. The electrical Back of Crate Card

to or from I/O pins. They consist of single bits or arrays of bits. If the signal is containing an array of bits they can be indexed and addressed like in the *C*-programming language by using an index. Arrays can be indexed in two ways: in ascending or descending order. This accounts for the hardware practice to send the lowest order bit first in some applications [15, 16].

3.2.2. The FPGA firmware

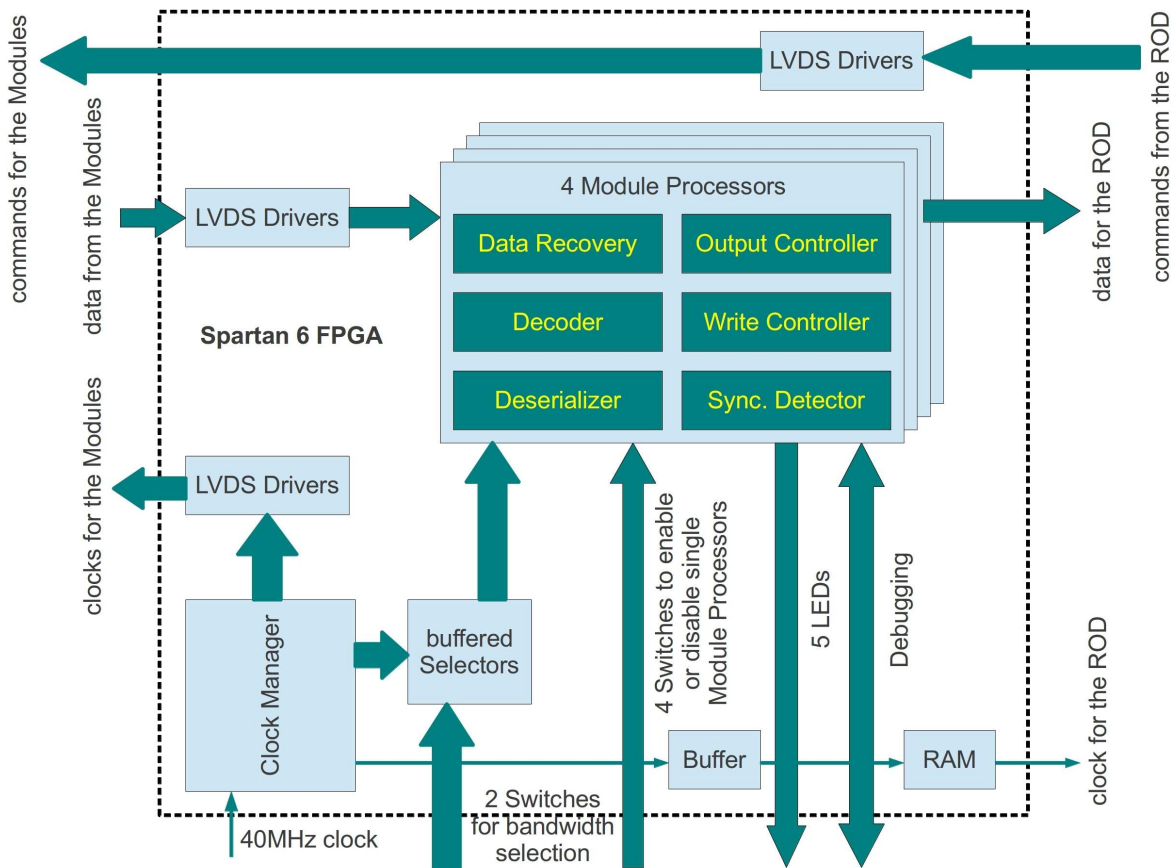


Figure 3.4.: A schematic view of the FPGA firmware.

Both Spartan-6 FPGAs are programmed with the same bitstream and therefore perform the same tasks. As shown in Figure 3.4, each FPGA contains a clock manager, four Module Processors, an array of switches, several LVDS drivers, an ODDR2 block and clock buffers. The clock manager is used to generate the different 40 MHz and 160 MHz clocks used in the FPGA. The Module Processors contain subcomponents for data recovery, synchronization, deserialization of the incoming data, decoding of 8B/10B encoded data, a write controller and an output controller. Data recovery samples the incoming

bitstream at a high frequency to improve reconstruction efficiency and reconstructs the bitstream. Synchronization detects the synchronization sequences that are sent in between packages and is able to align the 8B/10B decoder with the data packages. Deserialization uses registers to deserialize the incoming data for processing. This is necessary since packages are only decoded as a whole and therefore the complete package must be available to the decoder before processing can begin. The 8B/10B decoder decodes the incoming packages while the write controller and output controller handle the output of the decoded data.

The switches are used to select the bandwidth mode and connect or disconnect the different Modules electrically. The LVDS drivers generate Low Voltage Differential Signals. This is a signal standard to efficiently communicate electrical binary signals at a high speed and with low power consumption. Instead of using one line for communication and having a logical '0' be 0V and logical '1' be +3.3 V two lines are employed. Both lines have a potential of +3.3 V that is inverted between the lines. Logical symbols will then be transported as a difference of the two potentials. The ODDR2 is used to buffer the clocks and as discussed in Subsection 3.1.2 clock buffers are used to route clock signals through the clocking net instead of the data net.

Since we use Spartan-6 FPGAs of the vendor Xilinx, their ISE Design Suite was used to simulate the designs and program the FPGAs. Although a timing simulation is employed in the Design Suite all designs had to be tested on the hardware again. This is due to the fact that the simulation is not able to predict the signal delays on the hardware accurately enough.

4. Upgrades of the eBOC

4.1. Production and testing of a new prototype-board

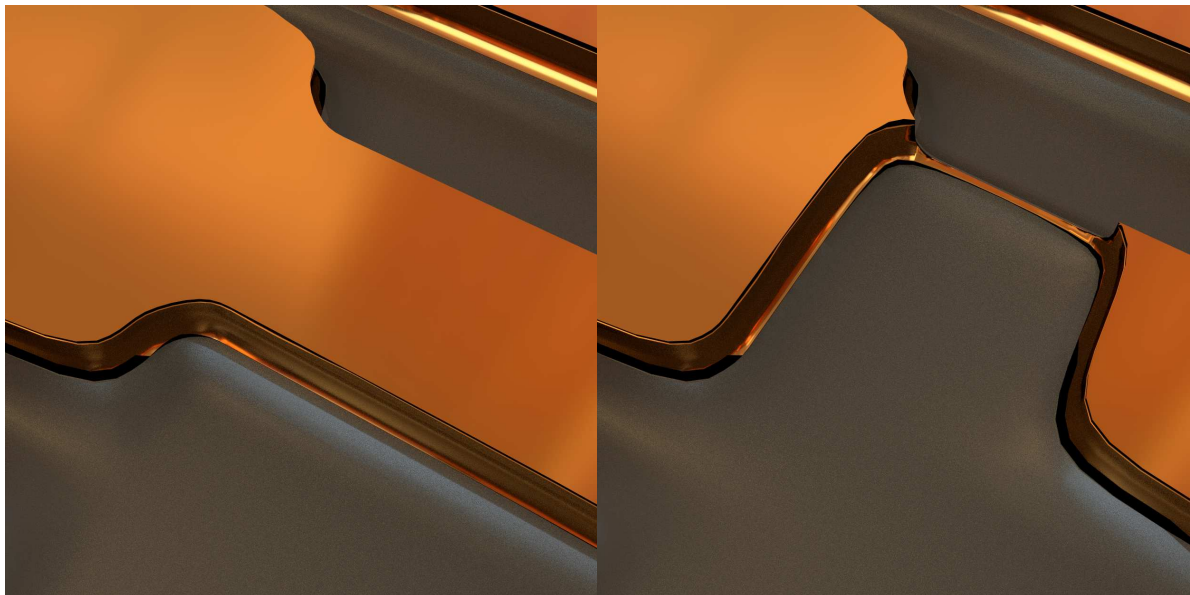


Figure 4.1.: A picture of the milling error. On the left side is a picture of the correct milling and on the right side of the actual milling.

A new prototype board was created by using the same schematics as the prototype board explained in Chapter 3. After milling the board all data lines had to be checked first by eye and then with a multimeter to ensure the quality of the milling process. This is also an important task since errors in the milling that will not be discovered might damage the parts that are later placed on the board. Some of the data lines will also be covered by the electric components that are placed on the board. Therefore no repairs of milling errors in certain parts of the board without removing electric components are possible later on. Even though all tests were passed successfully one minor milling error was not detected. A schematic view of the error can be found in Figure 4.1. One data line was almost cut by the milling process leaving only a small copper bridge for electrical connection. This error

4. Upgrades of the eBOC

caused the flash memory of the southern FPGA to become unprogrammable. At some point the faulty connection decreased the conductivity of the line due to oxidation and programming of the FPGAs got impossible. Speculating that the southern flash memory caused the problem it was first tried to bypass it by removing it. After desoldering the memory the milling error was discovered and repaired. Placing a new flash memory on the spot of the old one turned out to be complicated since three connection pads of the flash memory got damaged while removing the flash memory. With the help of a microscope all repairs were conducted successfully. Testing confirmed the success of the repair.

As testing of the old eBOC progressed during production of the new one changes to the hardware of both boards became necessary. All those changes were tested on the existing prototype and then employed on the new prototype and tested again. The changes to the hardware will be discussed in Section 4.2. After equipping the new prototype with all the necessary electrical components the testing of all connections was performed. This means that all soldering points were checked by eye, microscope and then with a multimeter to confirm proper connections. This revealed a shortage between the ground potential and a +3.3 V potential that was repaired by removing a resistor. No replacement of the removed resistor was required due to the fact that the resistor was used to connect some unused power lines to the supply voltage. After this small repair, a supply voltage was applied to the board and all the potentials on the board were checked. During these tests, it was observed that the power consumption of the board seemed to be constant but different for each time it was switched on. Due to the low order of this change in power consumption (less than 10%) testing had been continued.

A program was written to test both of the FPGAs individually. For this the FPGAs were programmed to react to a signal that could be provided to any of the input pins by sending an answer on a specific output pin. This was done to confirm that all input and output pins could be addressed individually and reliably. After this test, a new program was written that required the FPGA to read data from its RAM, perform arithmetic operations with it, write it back into the RAM and compare the results with a solution that was provided to the FPGA. This test was done to confirm normal operation of the RAM, slices and clock manager of the FPGA. All these tests were conducted successfully on both FPGAs and therefore proved that they were not damaged during their placement on the board.

The PECL driver of the board was not working properly. In an attempt to repair the driver, the ground potential was connected to a +5.0 V potential. This caused the FPGAs to receive negative supply voltages. FPGAs are very sensitive to negative voltages and therefore received damage [13]. Time intensive repairs are required to make the new

prototype operational again. However, a replacement PECL driver has been installed on the board already.

4.2. Changes to the hardware

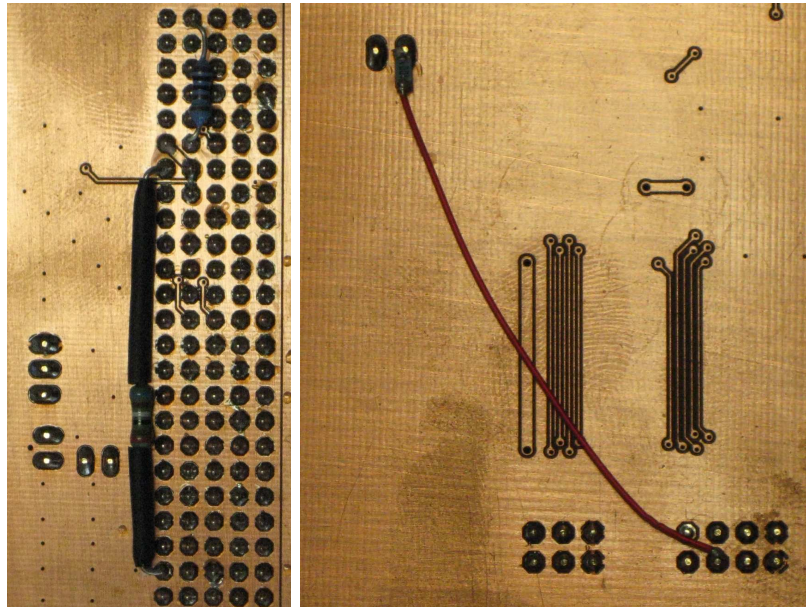


Figure 4.2.: Pictures of some of the hardware changes of the eBOC prototype.

Comparison of the electric layout of the board with the layout of an eBOC produced in former ROD-BOC development revealed differences in the electrical layout of the connections. This led to the discovery of a flaw in the new eBOC design where a +5.0 V potential was connected to a ground potential. The +5.0 V potential is generated by a transformer on the ROD. Shortening this potential on the eBOC to the ground potential could have damaged or destroyed the transformer on the ROD. However, since a modern chopper/high frequency transformer is employed on the ROD for generating the required potential no damage to the ROD or eBOC was done.

Three resistors have been added to the board to account for the difference in electrical layout. This means that some pins on the southern 160 pin connector that were not used on the eBOC were interconnected via resistors and a new potential has been applied to one pin on the northern 160 pin connector. The single pin belonged to a data-line of the FPGA and was reserved for transmission of a busy signal of the eBOC. Since the eBOC FPGA design does not communicate the status of the eBOC to the ROD this line was not used on the FPGAs. The schematics of the ROD revealed that a potential on this pin was required to notify the ROD that the eBOC was not busy. To protect the FPGAs on

4. Upgrades of the eBOC

the board the data line was cut and a potential of the correct strength was applied. The potential was achieved by using a resistor and the +3.3 V potential of the board. The right picture of Figure 4.2 shows this modification. That potential caused the ROD to communicate correctly with the eBOC and receive the clock the eBOC was sending. This also means the problems Johannes Agricola had with the board have been fixed.

All changes to the hardware were carefully cross checked with all available schematics of the ROD and eBOC to ensure that no harm could be done to the ROD or eBOC by including them. They were also tested on the older prototype before including them in the new prototype where they were tested again.

4.3. Changes to the FPGA firmware

In the firmware of the first prototype, the differential clock signal for the Modules was routed like a normal signal to the I/O pins which lead to problems with the timing constraints during the synthesis process. This is due to the fact that data signals use datapaths in the FPGA that are not optimized for transportation of high speed clock signals. Clock skew can be the result. It describes the phenomenon that two or more originally synchronous clock signals might become unsynchronous after routing them through the chip. To avoid the resulting error message in the development environment the clock signal was tagged as a data signal. No negative effect of this design practice was observed during testing with only one connected module but it might have caused problems with several connected modules and was therefore altered.

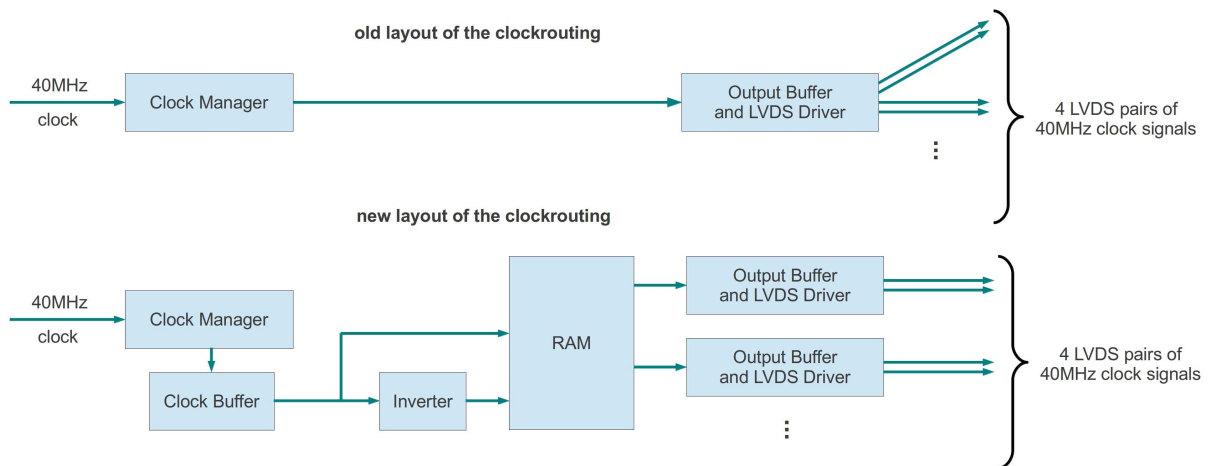


Figure 4.3.: A comparison between the old and the new routing of the clocks for the modules.

To solve this issue and fulfill the design standards for clock signals clock buffers and

ODDR2 were utilized. Clock buffers are entities on the FPGA that route clock signal through the dedicated timing network on the FPGA. Sending the clock signal through the timing network and then the ODDR2 ensures that the clock remains synchronous to the commands that are sent to the Modules and to each other. If hardware is used in the future to simulate several Modules on a single FPGA this will simplify the programming of the hardware [17].

The eBOC was also designed to cope with two different 80 MBit/s data rates (80 MBit/s and 2×80 MBit/s). These rates were originally considered to be used besides the new 160 MBit/s data rate. The option to switch to this unused data rate was removed.

4.3.1. The Error-checker

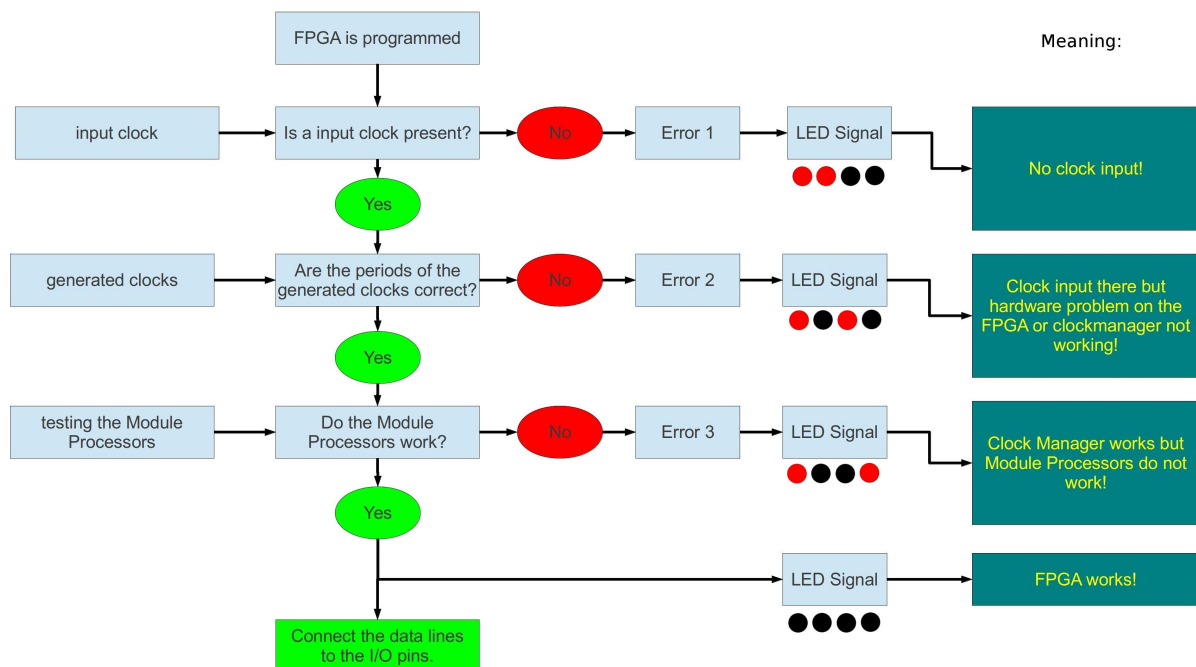


Figure 4.4.: A flow diagram of the states of the Error-checker.

One design that was specifically developed for this bachelor thesis is the Error-checker. It combines several processes to check the FPGA's functionality. It was programmed to be modular and easily adjustable. See Appendix A for an abbreviated version of the Error-checker. It works by having several states that are communicated via signals between the different processes of the Error-checker. When the board initializes none of the data carrying ports are connected inside the FPGA. This ensures that no data streams are sent or received when the FPGA has a defect. Figure 4.4 shows the states of the Error-checker. At the initialization an error state is set which symbolizes a missing

4. Upgrades of the eBOC

clock. If four clock periods are observed the status is changed to the state that shows an error in the clock manager. In this case, the periods of the output clocks of the clock manager are compared to each other. If they match the desired ratio the Error-checker assumes a state that describes a problem in the Module Processors. To test the Module Processors a prerecorded bit sequence that has all the properties of a real encoded event is sent to the Module Processors. The sequence will be decoded and compared to another prerecorded sequence. If both match the status of the Error-checker reports absence of hardware errors. This state will also enable all the data connections to the I/O pins on the FPGA. Since all of the operations were tested by this series of tests the FPGA should always work correctly after passing them.

To show the error states, each Error-checker state is linked to a unique LED sequence. This allows for an easy detection of FPGA problems on the board by looking at it.

4.4. Tests and validation

4.4.1. Testing-tools

As the first prototype was developed with a now obsolete software environment, former testing tools could not be used anymore. Instead of time consuming upgrades of the software a hardware solution was chosen. This led to the development of the Comparator. The Comparator is a piece of VHDL code that can record a trigger and repeat it with a configurable frequency. It also has the ability to compare two signals and count how often the signals are identical and how often they are not identical. The results of this count are saved in two VHDL signals that can both be sent out as a bitstream. The Comparator can also handle different data rates of the two signals and save the last few signals that were compared.

4.4.2. Tests with the ROD

For testing purposes the eBOC was connected to the ROD inside the VME crate. The configuration software from the laboratory PC was used to verify a functional ROD with the eBOC plugged into the backside of the crate. This was only the case after all hardware modifications of the eBOC had been completed. Before the changes an LED would light up on the ROD after sending a certain amount of data from the ROD to the eBOC that indicated that the ROD was busy. This was due to the fact that the ROD thought no eBOC was connected. It therefore stored all the data in an output buffer until the buffer was full which was indicated by the LED.

4.4.3. Tests with the MCC-Emulator

As discussed in Section 2.5 the MCC-Emulator can be used to simulate a Module. This is discussed in detail in the bachelor thesis of Benjamin von Ardenne [11]. The board sends out 8B/10B encoded events at a choosable rate of 40 MBit/s, 80 MBit/s or 160 MBit/s once a trigger is sent to the Emulator. To enable testing of the decoding operation of the eBOC the event is also sent out at a debug pin without any encoding. This non encoded stream is linked during testing operation to the Comparator that was introduced in Subsection 4.4.1 via one debug pin on the eBOC. The Comparator can then compare the non encoded and the just decoded data stream to measure decoding errors.

Something similar was already done by Johannes Agricola and Benjamin von Ardenne by reading out the registers on the ROD and comparing the decoded bit-sequences with the unencoded sequences sent through the debug pin on the MCC-Emulator [11, 12]. Their solution relied on the software environment and was therefore unusable without complex modifications as was discussed in Subsection 4.4.1. Developing the Comparator ensured testing independent of the software environment and avoided the software problems.

Successful decoding of two events every 50 μ s was sustained for at least an hour by using the Comparator to repeat the triggers accordingly.

4.4.4. Stress-testing the eBOC

It was planned that the eBOC decodes fake data from the MCC-Emulator with varying time differences between triggers to determine the maximum trigger rate of both the old and the new code. According to a description of the ATLAS experiment a maximum trigger frequency of 75 kHz (upgradable to 100 kHz) is expected [18]. Unfortunately both boards were unoperational and require lengthy repairs at the end of the bachelor project. This means that no long term stress testing of the eBOC hardware and VHDL design of the FPGAs is possible. The stress test would determine the maximum trigger rate at which events can be decoded and check if this rate decreases significantly when hardware resources are reserved for the Error-checker. All of the values are of course worse than the values obtainable in proper eBOC operation since the Comparator that is used for testing requires FPGA resources itself.

5. Summary and outlook

5.1. Summary

The IBL-Upgrade of the Pixel Detector of the ATLAS experiment in 2013 requires testing of the new hardware in the laboratory. For this a readout-chain in the laboratory has to be recreated. The eBOC is used as an alternative to the BOC in the laboratory for testing. The development of the eBOC that was created by Benjamin von Ardenne and Johannes Agricola has been continued in this thesis.

A new prototype board has been created and tested. Changes to the hardware were made to fix problems with the ROD-eBOC-communication. Continued development of the FPGA programming lead to the introduction of an Error-checker that tests the FPGA for errors at startup and an onboard testing ability for decoding errors was developed for laboratory tests. One of the workarounds of the FPGA code was removed and replaced by a solution that fulfills the design standards for clock routing on FPGAs. However, lengthy testing of the new features could not be employed. Both prototypes ceased to operate properly due to various hardware problems. Time intensive repairs are required and the most probable causes for the problems are pointed out in Section 4.1 and 5.2.

5.2. Outlook

Continued development should lead to an exchange of the current LEDs on the board for low power LEDs because there is strong evidence that the boards cease operating properly after driving the LEDs for a long time. Another important change for the future prototypes is a protection against negative voltage. Since ROD, Crate and FE-chips or MCC-emulator are directly connected to the eBOC and are not explicitly protected against negative voltage. Both the ROD and the MCC-emulator have FPGAs that are very sensitive to negative voltage. Additionally, the eBOC also has two FPGAs and a possible reason for the failure of the newly built prototype could have been negative voltages at the FPGAs or flash storage as described in Section 4.1.

The Error-checker could also be further developed. It could be equipped with the

5. Summary and outlook

ability to detect problems in the readout-chain such as unresponsive Modules or incorrect speed selection on the eBOC. These features would be very unique and could inspire development and testing of similar systems for more complex tasks.

Continued testing of the performance of the system is also necessary. Especially testing the maximum speed at which data can be decoded. This would give a benchmark for the performance of the employed VHDL-code. It is also necessary to implement a multiplexer (Mux) into the Module Processors. The eBOC is designed to multiplex one channel with a bandwidth of 160 MBit/s into four channels with 40 MBit/s bandwidth per channel. Currently only one 40 MBit/s data line is used to transmit decoded data while the rest of them transmit encoded data for testing purposes.

A. Appendix

```
1  ----- Header -----
library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
5 library UNISIM;
use UNISIM.VComponents.all;
7
----- Entity -----
9 entity errorchecker is
generic (clock_ratio: integer :=4);    -- describes the desired clock-
ratio
11 port(clock_in      : in std_logic;    -- external clock
generated_clocks: in std_logic_vector (0 to 1);
13 stream_out       : out std_logic;    -- to the input of the Module
Processor
stream_in        : in std_logic;    -- from the output of the Module
Processor
15 leds            : out std_logic_vector (0 to 3);
connect_lines    : out std_logic);    -- when '1' all I/O pins are
connected
17 end errorchecker;
19 ----- Architecture -----
architecture Behavioral of errorchecker is
21 signal state1, state2, state3, state4 : std_logic := '0';
23 -- example event and expected response
signal test_event      : std_logic_vector (0 to 10) := "1011011000";
25 signal expected_response : std_logic_vector (0 to 8) := "11011010";
begin
27 leds <= "1100";    -- setting the leds to error 1: "no
clock"
process (clock_in)    -- testing if a clock is present
29 variable count      : integer range 0 to 4 := 0;
begin
```

A. Appendix

```
31  if rising_edge(clock_in) and state1 = '0' then
    if count = 4 then
33      state1 <= '1';           -- test is passed
      leds <= "1010";         -- setting the leds to error 2: "wrong
                                clock_ratio"
35    else
      count := count+1;
37    end if;
    end if;
39 end process;

41 process (generated_clocks)           -- testing the clock periods
    variable count1: integer range 0 to 4 := 0;
43    variable count2: integer range 0 to 4*clock_ratio := 0;
    begin
45      if state1 = '1' and state2 = '0' then
          if rising_edge(generated_clocks(0)) then
47              count1 := count1+1;
          if rising_edge(generated_clocks(1)) then
49              if count2 = 4*clock_ratio then
                  if count1 = clock_ratio then
51                      state2 <= '1';   -- test passed
                      leds <= "1001";  -- setting leds to error 3: "problem with
                                          Module Processor"
53                  else
                      state2 <= '1';
55                      state3 <= '1';   -- test not passed
                      state4 <= '1';
57                  end if;
                  else
59                      count2 := count2+1;
                      end if;
61              end if;
          end if;
63      end if;
    end process;

65 process (generated_clocks(0))       -- sending out test event
67    variable count: integer range 0 to 11 := 0;
    begin
69      if state2 = '1' and state3 = '0' then
          if rising_edge(generated_clocks(0)) then
71              if count = 11 then
                  stream_out <= '0';
```

```

73     state3 <= '1';
    else
75         stream_out <= test_event(count);
        count := count+1;
77     end if;
    end if;
79 end if;
end process;
81
83 process (generated_clocks(1), stream_in)  -- comparing answer from the
    Module processor
variable dump: std_logic_vector (0 to 8) := (others => '0');
85 variable count: integer range 0 to 9 := 0;
variable start: boolean := false;
87 begin
if state3 = '1' and state4 = '0' then
89     if rising_edge(generated_clocks(1)) then
        if stream_in = '1' then
91             start := true;
        end if;
93
        if start then
95             if count = 9 then
                if dump = expected_response then
97                 connect_lines <= '1';           -- connecting I/O pins
                    state4 <= '1';                 -- test passed
99                 leds <= "0000";                 -- error leds are turned off
                else
101                 state4 <= '1';
                end if;
103             else
                dump(count) := stream_in;         -- writing stream to dump-file
105                 count := count+1;
                end if;
107             end if;
        end if;
109 end if;
end process;
111
end Behavioral;

```

VHDL-Code for generating a simplified version of the Error-checker.

Bibliography

- [1] C. D. Froggatt, H. B. Nielsen, *Trying to understand the Standard Model parameters*, Surveys High Energ.Phys. 18 pages 55–75 (2003), hep-ph/0308144v2
- [2] R. Oerter, *The Theory of Almost Everything: The Standard Model, the Unsung Triumph of Modern Physics*, Plume (2006)
- [3] J. D. Lykken, *Beyond the Standard Model*, CERN Yellow Report CERN-2010-002 pages 101–109
- [4] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, Institute Of Physics Publishing And SISSA (2008)
- [5] ATLAS Collaboration, *Observation of an Excess of Events in the Search for the Standard Model Higgs Boson in the $H \rightarrow WW^{(*)} \rightarrow l\nu l\nu$ Channel with the ATLAS Detector*, ATLAS-CONF-2012-098 pages 1–28 (2012)
- [6] CMS Collaboration, *Observation of a new boson with a mass near 125 GeV*, CMS-PAS-HIG-12-020 (2012)
- [7] F. Djama, *Overview of the ATLAS Insertable B-Layer (IBL) Project*, ATL-INDET-SLIDE-2012-430 pages 1–27
- [8] G. A. et al, *ATLAS pixel detector electronics and sensors*, Institute Of Physics Publishing And SISSA (2008)
- [9] C. Gabaldon, *Performance of the ATLAS Trigger System*, Institute Of Physics Publishing And SISSA (2011)
- [10] D. Froidevaux, P. Sphicas, *General-Purpose Detectors For The Large Hadron Collider*, Annu. Rev. Nucl. Part. Sci. pages 375–440 (2006)
- [11] B. von Ardenne, *Development of new data transmission methods for the read-out system of the ATLAS Pixel Detector*, Bachelor's Thesis (2010), II.Physik-UniGö-Bach2010/06

Bibliography

- [12] J. Agricola, *Development And Programming Of A Data Aquisition Component For The New ATLAS Pixel Detector Front-End Electronics*, Bachelor's Thesis (2011), II.Physik-UniGö-BSc-2011/10
- [13] A. K. Sharma, *Programmable Logic Handbook: PLDs, CPLDs and FPGAs*, Mcgraw-Hill (1998)
- [14] Xilinx, *Spartan-6 FPGA Configuration User Guide* (2012)
- [15] P. J. Ashenden, *The Designer's Guide to VHDL*, Morgan Kaufmann Publishers, third edition (2008)
- [16] J. Reichardt, B. Schwarz, *VHDL-Synthese*, Oldenburg Verlag, forth edition (2007)
- [17] Xilinx, *Spartan-6 FPGA Clocking Resources User Guide* (2011)
- [18] ATLAS Level-1 Trigger Group, *Level-1 Trigger, Technical Design Report*, ATLAS TDR-12 (1998)

Acknowledgements

I would like to thank Prof. Dr. Arnulf Quadt and Priv.Do. Dr. Jörn Große-Knetter for offering this bachelor thesis. I also would like to thank Nina Krieger for spending countless hours helping me with the hardware, software and all kinds of questions while writing her PhD thesis. Further thanks go to the electronics workshop of the second institute for physics for helping me during my thesis with good advice, the production of the new prototype and solving various problems with the boards.

Thanks also go to my family and friends for supporting and bearing me during the writing of my thesis.

Erklärung nach §13(8) der Prüfungsordnung für den Bachelor-Studiengang Physik und den Master-Studiengang Physik an der Universität Göttingen:

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe.

Darüberhinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, im Rahmen einer nichtbestanden Prüfung an dieser oder einer anderen Hochschule eingereicht wurde.

Göttingen, den July 23, 2012

(Alrik Stegmaier)