Dissertation

submitted to the
Combined Faculty for Natural Sciences and Mathematics
of the Ruperto Carola Heidelberg University, Germany

in fulfillment of the requirements for the degree of
Doctor of Natural Sciences

put forward by

Dipl.-Inf. Sven Andreas Chris Markus Schatral
born in Heidelberg, Germany

Date of oral exam:

Design of Multi-Gigabit Network Interconnect Elements and Protocols for a Data Acquisition System in Radiation Environments

Advisor: Prof. Dr. Ulrich Brüning

Abstract

Modern High Energy Physics experiments (HEP) explore the fundamental nature of matter in more depth than ever before and thereby benefit greatly from the advances in the field of communication technology. The huge data volumes generated by the increasingly precise detector setups pose severe problems for the Data Acquisition Systems (DAQ), which are used to process and store this information. In addition, detector setups and their read-out electronics need to be synchronized precisely to allow a later correlation of experiment events accurately in time. Moreover, the substantial presence of charged particles from accelerator-generated beams results in strong ionizing radiation levels, which has a severe impact on the electronic systems.

This thesis recommends an architecture for unified network protocol IP cores with custom developed physical interfaces for the use of reliable data acquisition systems in strong radiation environments. Special configured serial bidirectional point-to-point interconnects are proposed to realize high speed data transmission, slow control access, synchronization and global clock distribution on unified links to reduce costs and to gain compact and efficient read-out setups. Special features are the developed radiation hardened functional units against single and multiple bit upsets, and the common interface for statistical error and diagnosis information, which integrates well into the protocol capabilities and eases the error handling in large experiment setups. Many innovative designs for several custom FPGA and ASIC platforms have been implemented and are described in detail. Special focus is placed on the physical layers and network interface elements from high-speed serial LVDS interconnects up to 20 Gb/s SSTL links in state-of-the-art process technology.

The developed IP cores are fully tested by an adapted verification environment for electronic design automation tools and also by live application. They are available in a global repository allowing a broad usage within further HEP experiments.

Zusammenfassung

Moderne Teilchenphysikexperimente erforschen die Zusammensetzung von Materie inzwischen aufwendiger als jemals zuvor und profitieren dabei stark vom Fortschritt in der Kommunikationstechnologie. Die immer höher auflösenden Detektoraufbauten stellen die Datenerfassungssysteme vor große Herausforderungen um die Informationen zu verarbeiten und zu speichern. Die Ausleseelektronik muss präzise synchronisiert werden um aufgenommene Ereignisse genau zu rekonstruieren und zeitlich einzuordnen. Außerdem führt die starke radioaktive Strahlung, welche durch die Teilchenbeschleuniger erzeugt wird zu ernsthaften Fehlfunktionen in den elektronischen Systemen.

Diese Arbeit beschreibt eine Architektur für ein vereinheitlichtes Netzwerkprotokoll mit angepasst entwickelten physikalischen Schnittstellen für die Verwendung in zuverlässigen Datenerfassungssystemen unter radioaktiver Strahlung. Speziell konfigurierte serielle bidirektionale Punkt-zu-Punkt-Verbindungen werden verwendet um Hochgeschwindigkeitsdatenübertragung, Kontrollzugriff, Synchronisierung und eine globale Taktverteilung in einem einzigen Kommunikationskanal zu realisieren. Damit können erhebliche Ressourcen eingespart werden und die Ausleseaufbauten werden deutlich effizienter und kompakter. Hervorzuheben sind außerdem die entwickelten strahlungstoleranten Funktionseinheiten um einzelne und mehrere Bitfehler abzufangen, und eine gemeinsame Diagnoseschnittstelle die sich in den Kontrollkanal des Protokolls integriert und eine einfache Fehlerbehandlung in großen Aufbauten ermöglicht. Viele innovative Entwicklungen für FPGA und ASIC Plattformen wurden umgesetzt und sind im Detail beschrieben. Besonderes Augenmerk liegt hier auf den physikalischen Netzwerkschnittstellen von LVDS Verbindungen bis zu 20 Gb/s SSTL Serialisierern in hochmodernen Fertigungstechnologien.

Die Entwicklungen sind umfangreich in einer angepassten Verifikationsumgebung für Softwarewerkzeuge der Entwurfsautomatisierung, sowie im Live-Einsatz getestet. Sie sind in einem zentralen Speicher frei verfügbar und bieten sich für eine universelle Verwendung in vielen Teilchenphysikexperimenten an.

Contents

Chapter 1

Introduction

1.1 Motivation

Our world is increasingly being inundated by physical devices with embedded network connectivity, which enables them to exchange data among each other. New digital sources and applications are constantly driving the need for a more powerful network interconnect technology, which allows a much faster use, processing, distribution and evaluation of data. It is obvious that also modern research in all fields of natural sciences benefits from this technological progress and innovation.

Nowadays large international groups of scientists build very complex setups with digital acquisition and processing systems and conduct tedious and expensive projects, which maybe lead to new discoveries; whereas in the past, fundamental findings about nature could be gained by single geniuses with their rather small but innovative experiments. Some experiments are even that complex that they can not be realized without extensive research in other fields of science like computer engineering. They aim at pushing boundaries like energy, speed and resolution to new limits, which creates difficult tasks for the data acquisition system such as e. g. processing and storing the huge amount of event data. Side effects of extreme conditions, like strong temperature variations and heavy ionizing radiation, create demanding challenges on the electronic devices, which would usually fail in such an environment. Thus, the technology of fast and reliable digital data acquisition and processing is a key tool for success in modern research. Especially in High Energy Physics (HEP) experiments the so-called front-end consists of several thousand sensor units to record complex events, while the back-end resources, which store the data, want to be kept to the minimum for the purpose of reducing costs as

much as possible. A network with this many-to-one communication pattern is actually demanding for usual network based technologies. Moreover, besides high bandwidth data transmission, efficient data aggregation, control capabilities and high reliability, a data acquisition system addresses requirements, like precise time synchronization to correlate events accurately in time. By now, more or less all sampling devices use serial instead of parallel interfaces for high bandwidth data transmission. Serial interconnects improve utilization because they require fewer pins and wires, allow higher data throughput and come with a variety of advantages, like flexibility in terms of distance, media type, noise immunity and performance. A drawback is that state-of-the-art serial interfaces in the range of multi-gigabit transmission are very complex and high-speed mixed-signal circuits with tight electrical specifications. Special solutions are particularly needed for the design, implementation and configuration of data acquisition networks with demands mentioned above.

This thesis has been written in the context of a modern and very complex particle accelerator experiment, which will be introduced in the next section by paying special attention to the demanding challenges on the data acquisition system.

1.2 The CBM experiment

The GSI Helmholtz Centre for Heavy Ion Research [9] performs basic and applied research in physics and related natural science disciplines since 1969. The new Facility for Antiproton and Ion Research (FAIR) [8] extends the existing GSI accelerator in Darmstadt, Germany, as depicted in fig. 1.1. Since the two superconducting FAIR synchrotrons (SIS100/300) are significantly larger than the existing SIS18, which has a circumference of about 200 meters compared to the 1100 meters of the SIS100/300, they will deliver proton beams of unprecedented intensities and energies up to 90 GeV [30]. The system of storage and cooler rings improves the quality and justifies expectations for highest beam intensities, brilliant beam quality, highest beam energies and highest beam power. Furthermore, a parallel operation of four scientific programs can be realized with the double ring concept. The variety of beams offers unique research opportunities in the field of nuclear structure physics, nuclear hadronic physics, plasma physics, biophysics and material research. One of these programs is the Compressed Baryonic Matter (CBM) experiment.

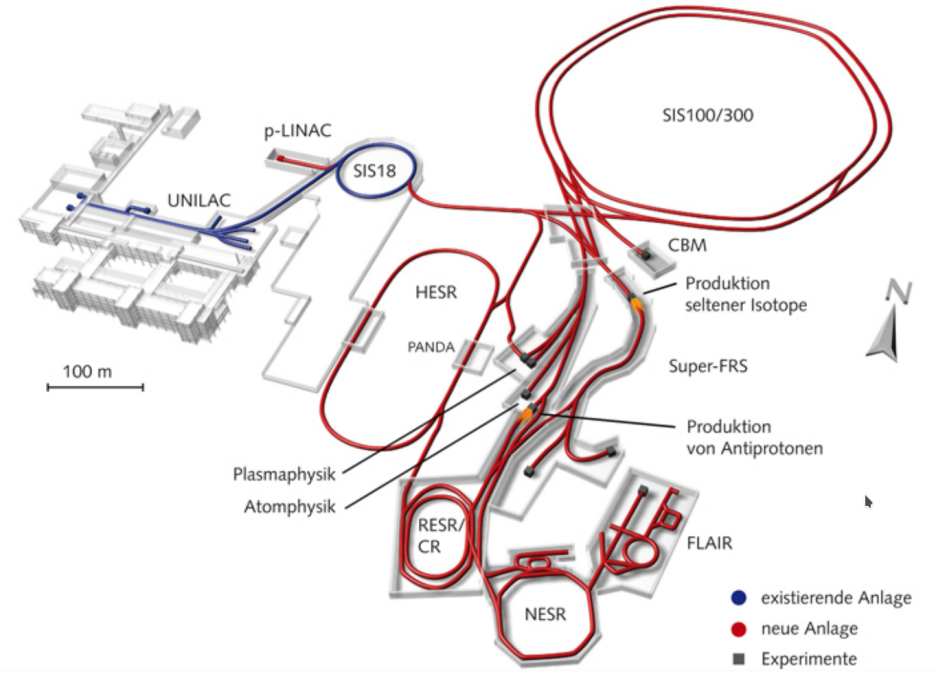


Figure 1.1: The GSI Darmstadt with the existing SIS18 accelerator is shown in blue on the left side and the new FAIR facilities and accelerator in red on the right side [8].

The FAIR accelerator mostly competes against the Large Hadron Collider (LHC) at the European Organization for Nuclear Research (CERN), Geneva, Switzerland [5]. The CBM equivalent heavy ion experiment at CERN is called ALICE, focusing on highest beam energies at high temperatures but comparatively moderate beam intensities. The characteristics of the FAIR synchrotrons allow the further exploration of the QCD (quantum chromodynamics) phase diagram of strongly interacting matter in the medium temperature area at high densities as depicted in fig. 1.2. It will present new findings on the transition from confined to deconfined matter, the critical point and theoretical new phases of matter like the quarkyonic matter or color superconductivity [30].

The CBM research program is a leading project in future high energy heavy ion research and requires considerable financial and human resources. The international collaboration consists of more than 400 scientists from 15 countries and the total costs amount to more than one billion euros. The main goal of the CBM experiment is to study effects in heavy-ion collisions, including their correlations and event-

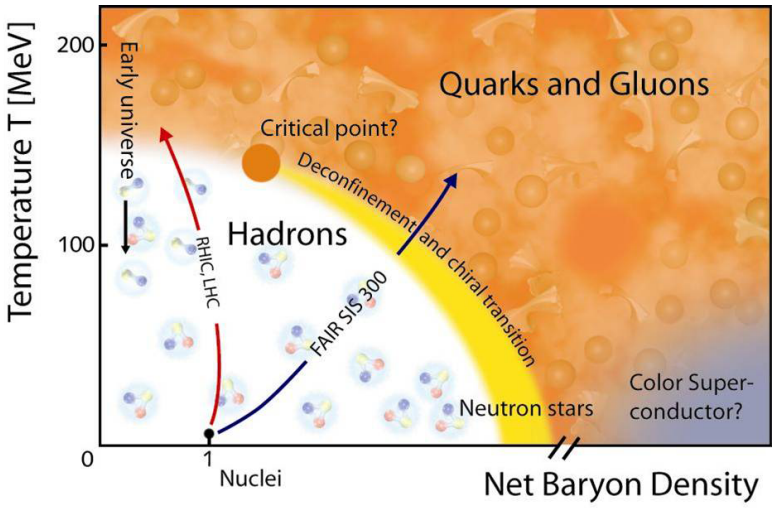


Figure 1.2: The QCD phase diagram and the working range of the FAIR accelerator [30].

by-event fluctuations to further investigate the quark-gluon plasma, which gives new information about the structure of baryonic matter. Regarding this, one demanding challenge is to identify hadrons and leptons of Au+Au collisions at extremely high reactions rates up to 10 MHz in order to reach the statistical needed amount of samples. The particle identification is done by a large detector setup, consisting of a combination of various detector systems, each using different detection technology. Thereby the tracks of all generated particles are exactly reconstructed, giving detailed information about their particular characteristics [90].

1.2.1 The CBM Detector Setup

The final CBM detector setup is planned to be located in a large cave at the new FAIR facilities. The whole detection system is highly granulated and consists of several different detectors, which use different detection technologies. Thereby the whole setup is arranged in layers behind a target, where heavy ions collide with other nucleus or organic probes, thereby creating new particles. The collision takes place in the strong field of a huge 140 tons super-conducting dipole magnet, which deflects the particles depending on their charge and mass. If particles cross

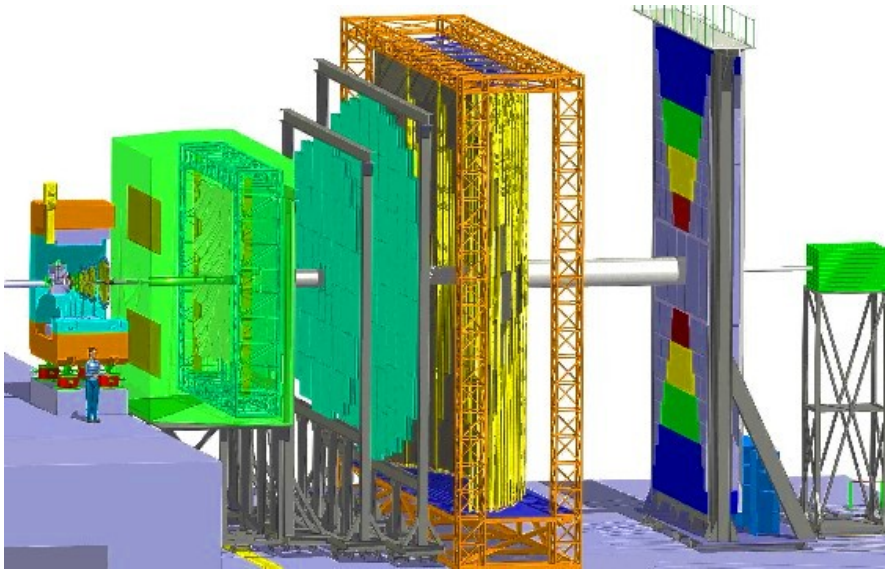


Figure 1.3: The CBM experiment setup in its electron configuration with the TRD and RICH detectors [8].

through the detection system, electronic signals are generated, which are digitized, processed and read out. According to their time of flight and characteristic energy loss, it can be differentiated between hadrons, muons, electrons and photons. The most challenging detector is the silicon tracking system (STS), which consists of low-mass silicon micro-strip detectors with two hybrid-pixel detector layers and is located directly behind the target; several hundred thousand segments are read out individually and the track reconstruction is possible over a wide momentum range from about 100 MeV to more than 10 GeV with a resolution around 1%. An additional Micro-Vertex Detector (MVD) with $\sigma = 3\mu m$ resolution can detect D mesons and is built-up with two layers of ultra-thin Monolithic Active Pixel Sensors (MAPS) to reduce multiple scattering. Due to the limited space in the magnet and a position close to the target, the electronics of the STS and MVD have heavy constraints on material budget, low noise analog circuits and radiation hardness [34]. The following Ring Imaging Cherenkov Detector (RICH) and the Transition Radiation Detector (TRD) are capable of detecting electrons with different technologies and in a wide momentum range [35]. As an alternative to the electron detector, a muon detection system – consisting of Muon Tracking Chambers (MuCh) – can be used. Depending on the beam energy and the mass of the mesons, the muons momentum is determined by its penetration depth.

Around 10 meters behind the target a Time-Of-Flight (TOF) measurement of charged hadrons is done with an array of Resistive Plate Chambers (RPC) of ceramic electrodes. The challenge is to achieve an accurate timing resolution of around 90 ps over the whole array with its 120 m^2 area. An electromagnetic Calorimeter (ECAL) gives information about photons and neutral particles and can be arranged in different ways and positioned freely behind the target. The Projectile Spectator Detector (PSD) is needed for the collision positioning by measuring the number of non-interacting nucleons [33].

1.2.2 The Data Acquisition System

In a usual data acquisition system a global trigger signal forces the recording of samples in interesting units and the data is stored in the front-end electronics until a decision is made, if the data should be rejected or saved for later analysis. The CBM experiment tries to detect more unusual events and therefore works with very high reaction rates. The detectors have to be extremely fast and require radiation hard electronic components. As about 10 million collisions take place within one second and several millions of segments have to be triggered at once, a conventional data acquisition system comes up against its limits. A trigger signature may be complex and requires also information from other detector sub-systems. It is not a feasible task for a setup in this dimension having a resolution of 100ps. Moreover, the storage of recorded information in electronic circuits, which are exposed to such heavy radioactive radiation, should be reduced as much as possible.

In the CBM read-out chain the DAQ uses a free-running approach and has therefore demanding requirements on all network components and the network protocol. The suggested network structure is depicted in fig. 1.4. The front-end electronics on the left side have to decide independently which information should be recorded. The detectors have an intelligent, self-triggering read-out logic, which asynchronously generates time-stamped data streams with a total magnitude of several terabit per second. This huge amount of data is transmitted to aggregating and processing boards and finally handled and saved by a high performance computer farm, where the streams from all sub-detectors are combined to whole events. This first level event selector (FLES) performs online software operations on the data in order to reduce it by a factor of 1000 compared to the raw data stream. This is done by very fast and precise event reconstruction algorithms using massive hardware

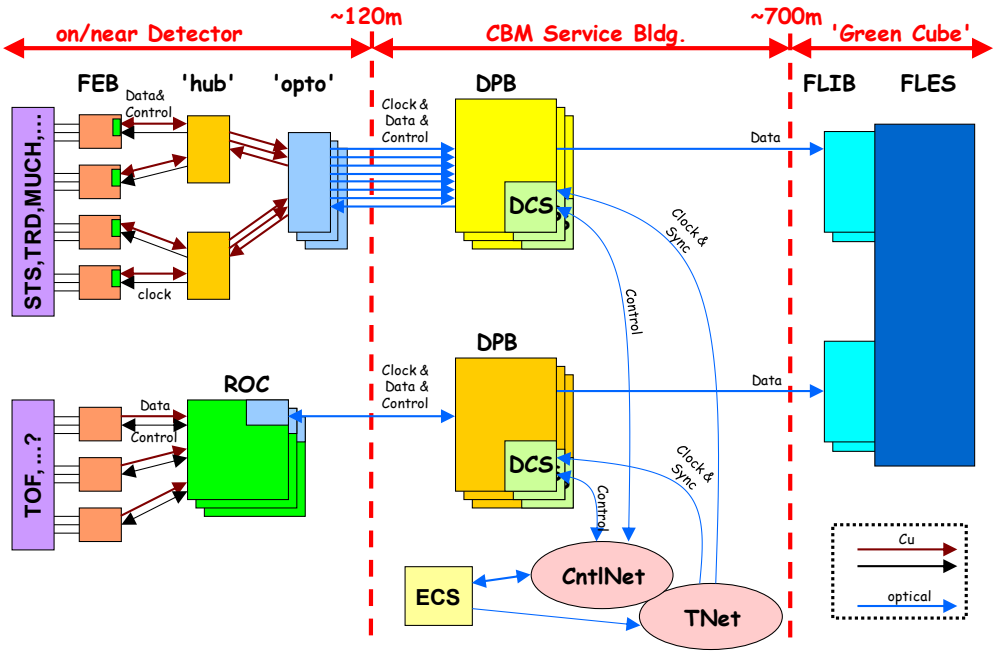


Figure 1.4: The CBM data flow within the new acquisition network structure of the CBM experiment [66].

parallelization, GPGPUs as accelerators and the selection of the most promising events [27]. Compared to other high-energy physics experiments, this concept is slightly different. The software allows very complex algorithms to be implemented and a more flexible adaption to enhanced needs than a hardware-based selection solution.

The free-running DAQ concept has also challenging demands on the network. As no global trigger signal is available, the read-out electronics require a precise time synchronization and therefore all clocks on nodes and leaves need to be derived from one master clock. The data is continuously streamed by event activity and not depending on network utilization. In addition, every device needs to be initialized, properly configured and integrated in the read-out chain before the experiment is executed. And due to the position of devices in the dipole magnet, heavy area constraints require dense interconnection solutions on copper cables as well as on optical fibers. This led to special requirements on the network like reliable high-speed network interconnects, a well-balanced hierarchy of elastic buffers and data bandwidth, efficient stream merging and a unified network protocol, which provides the features data transmission, device control,

initialization, and a global clock distribution and device synchronization on unified links. Moreover, the network needs to be highly configurable in terms of hierarchy levels and bandwidth, considering the particular read-out chain configurations of all sub-detectors. Many components are placed in a high radiation environment and the network protocol has to run absolutely reliable and fault tolerant without live-locks on different hardware devices like FPGAs and ASICs. Due to limited budgeting, preferably commercial off-the-shelf (COTS) parts are used but for the front-end detector components special solutions are required. A first prototype of the network protocol used in the CBM experiment has been presented in [44]. Although the development was very innovative and led to promising first results, many new challenges emerged within the upgrade of the data acquisition system. Especially the new requirement of radiation robustness against Single Event Upsets (SEUs), the need for designing custom multi-gigabit serial interfaces (The OMGT project) and the terms of precise time synchronization of the whole read-out setup (Interconnects with deterministic latency) lead to an accurate examination of the status quo and the planning of concepts and developments for the future to ensure a systematic continuation of the research program.

1.3 Conclusion

In the introductory chapter the reader has been introduced so far to the scope of this thesis and the context of the CBM experiment. Chapter 2 provides background information on ionizing radiation, the impact on electronic devices and presents some mitigation techniques for integrated circuits with relevance for the CBM experiment. In chapter 3 the state of the art, as well as advantages and disadvantages of the former development are discussed; also some comparable research implementations are briefly explained. Chapter 4 describes very detailed the concept and implementation of the new network protocol for the DAQ extension. Special features are the radiation hardened functional units and the interface for statistical error and diagnosis information. A verification and comparison to the old network implementation is also presented. Chapter 5 presents the network subcomponents of the whole data acquisition system, which uses the new CBMnet cores and describes every device in detail. The sophisticated synchronization algorithms to ensure deterministic latency and design structures for fault tolerance are explained and verified through simulation, as well as live application in

laboratory and beam time tests. Whereas chapter 4 describes the upper media layers of the protocol, chapter 6 goes more into detail about the physical layer implementation and proposes the architecture of a 4-tap FIR full-digital multi-gigabit transmitter, which has been developed, verified and manufactured. Chapter 7 describes the results of this thesis and concludes by providing a brief summary and giving an outlook on future work.

Chapter 2

Radiation Mitigation and Hardening

The final CBM experiment works with very high reaction rates, which results in a very strong radiation level in the environment around the target as well as in the deflection area behind it, and a still high radiation level inside the cave. Thus, the electronic devices of the detection system and the DAQ are heavily disturbed by radiation effects, which induce undesired behaviour in the integrated circuits of silicon devices and its analog circuits, logic and memory. The effects range from digital errors like bit flips in stored data and unintentional state changes in control logic to destructive errors due to an alteration process in the silicon. In case of a failure of a device, a reboot of the hardware can help to restore the working state but some effects might even result in a permanent device damage.

This chapter gives a rough overview about radiation effects on the specific FPGA and ASIC devices, which are used in the CBM experiment and further examines some mitigation techniques, which can be considered for the elaboration and improvement of the CBMnet protocol implementations. The radiation level expected in the CBM experiment is much higher compared to other special use cases like military or space operation. Therefore an evaluation with regard to the intended application is required.

2.1 Radiation Effects

The effects which occur when radiation traverses through matter are the basis used to determine and characterize particles with detection devices. Unfortunately, the radiation deposits energy in terms of i.a. nuclear reactions and inelastic collisions with the atomic electrons of the material, causing various undesired effects (nondestructive errors) or damage (destructive errors) in silicon devices. Radiation can be categorized as ionizing or non-ionizing, depending on the energy of the radiated particles. Regarding the effects in semiconductor devices, it can be distinguished between the following particles [48]:

- γ, X-rays and the higher range of ultraviolet light cause direct ionization if the carried energy is higher than 10eV. Due to their very small size, secondary effects on the lattice are rare.
- α, β-rays as well as protons and heavy ions are charged particles which cause direct ionization and, if they carry very high energy, a displacement, decay or excitation of the nucleus in the silicon can happen, which also causes indirect ionization.
- neutrons are uncharged particles which only cause a lattice displacement and further indirect ionization through decay

In case of direct ionization, the magnitude of disturbance, which is caused by a single charged particle, depends on the linear energy transfer (LET) and describes the amount of energy per unit length lost by an ion or photon traversing a material. The total amount of energy deposited and the number of electron-hole pairs generated further depends on the size of the ion and the density of the material. For example: A high LET is the type of radiation, which deposits a large amount of energy in a small distance, like heavy ions and alpha particles. In contrast, a low LET is the type of radiation, which deposits a less amount of energy along the track, like X-rays and γ-rays. The

The linear energy transfer is defined by

$$LET = \frac{dE}{dx} \cdot \frac{MeV}{cm} \tag{2.1}$$

where $\frac{dE}{dx}$ is the energy loss per distance due to Coulomb interactions, like ionization or excitation, or nuclear energy loss, like nuclear interactions, Cerenkov radiation, or emission of Bremsstrahlung. It is also common to define the LET with respect to the material density ρ. This results in the so called *mass stopping power* as the total energy lost per path length by a charged particle:

$$S = \frac{dE}{dx} \cdot \frac{1}{\rho} \cdot \frac{MeV \cdot cm^2}{g} \tag{2.2}$$

As an example [55]: A *stopping power* of $100MeV \cdot cm^2/mg$ through silicon generates electron-hole pairs with the charge per penetration depth $Q = 1pC/\mu m$. The final impact on a MosFET further depends on many factors, e. g. the exact position of the particle track relative to charge-sensitive parts of a transistor, size of depletion region, time of charge gathering, charge sharing, technology and process, so no calculation is given. But to get an impression, the gate capacitance of minimum-size transistors for feature sizes of 180 nm to 65 nm, lies in the sub-femtofarad order (gained by simulation).

Regarding the effects of radiation on semiconductor devices, it can be distinguished between aging effects, which are always destructive and alter the atomic structure of the device, and Single Event Effects (SEE), which can be destructive or non-destructive, but always have a spontaneous effect on the device like a bit flip or latch-up.

2.1.1 Aging Effects

If an integrated circuit is exposed to radiation, some gradual effects take place over time, altering its electronic behavior. The formation of electron-hole pairs directly depends on the energy deposited in the material. Due to electric fields in the semiconductor, they are further separated and can not recombine again. Electrons have a higher mobility and can leave the silicon-oxide in a transistor, holes are collected within the gate-oxide. This leads to alteration of the transistor's switching behavior, like threshold voltage shifts or increasing noise and leakage currents. This effect is called Total Ionizing Dose (TID), depending proportionally on the LET and particle flux, and is described with the SI-unit Gray (Gy) or the unit Rad, while 100rad = 1Gy. At least the threshold voltage shifts depend on the

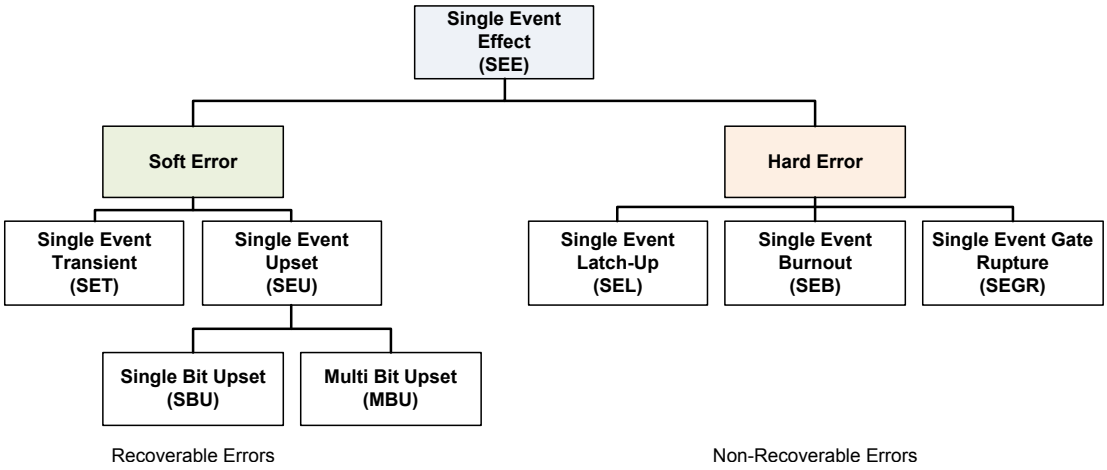


Figure 2.1: Overview over the Single Event Effects that can happen in digital semiconductor devices. Hard errors lead to damage, while soft errors only affect the logical function. Figure similar to [96].

size and thickness of the gate oxide and as process technology shrinks continuously, the effect is negligible for devices in 130nm and below [76]. Besides, the TID effects can also be repaired by annealing. If the device is not exposed to radiation, the electron-hole pairs try to recombine again. This process of de-trapping can be supported by heating the device.

A second effect is the displacement damage, which is caused by high energy particles, disturbing the lattice structure and doping of the semiconductor material. It can lead to a decreased electrical performance, and also to increased leakage currents in integrated circuits. Fortunately, the effect on MOS-FETs, compared to bipolar transistors, is rather small, as minor changes in the doping have no serious impact on the electric properties.

2.1.2 Single Event Effects

In contrast to the aging effects, SEEs happen spontaneous and mostly affect the digital parts of integrated circuits. As earlier mentioned they are caused by direct ionization from a penetrating charged particle or from created electrons as the result of sub-atomic collisions in the silicon or silicon dioxide. In the presence of an electric field, the charges are divided and a current is induced in the semiconductor

material. It can be distinguished between hard errors and soft errors, while hard errors cause lasting damage on a device. Soft errors only affect the logical function and can be cleared by a reboot at least.

- A **Single Event Latch-Up (SEL)** is a latch-up induced by a heavy ion or proton passing through the junctions of a parasitic PNPN structure, causing in a short. Without protection, the resulting current can lead to high temperatures and thermal destruction of the circuit. Moreover, in power MOSFETs the latch-up can lead to wrong output currents with harmful consequences. Especially bulk CMOS devices are highly susceptible. Nonetheless, in CMOS SOI processes the problem no longer appears, as the P/N structures have no direct contact to the substrate.

- A **Single Event Burnout (SEB)** mostly happens in power MOSFETs when a heavily ionizing particle causes a drain-source voltage, which is higher than the breakdown voltage of the parasitic structures. It also produces a high current which may destroy the device.

- A **Single Event Gate Rupture (SEGR)** is also common in power MOSFETs when a heavy ion causes a strong electric field in the gate region while a high voltage is applied. This also produces a high current which may destroy the device.

- A **Single Event Transient (SET)** is a charge injection in a combinatorial logic path, caused by the ionization of a high energy particle. This generates an asynchronous voltage/current spike, which propagates through the circuit. The pulse width of this glitch is typically around a few hundred picoseconds, depending on the particle energy, net capacity, drive strength and process. It should be mentioned, that in most cases a SET does not effect the logical correctness. It can only grow to a noticeable SEU when it has any impact on the logical function of the combinatorial elements, like an injected transient pulse in the net of a NAND will have no effect while the other input stays zero. Further the pulse needs to be sampled by a subsequent Flip-Flop, and therefore needs to exceed the critical amount of charge and meet its setup- and hold-times correctly. Obviously, SETs are a more severe problem in high-frequency circuits, where the probability of sampling is increased and node capacities are very small. Especially in technology nodes beyond 90 nm SET induced soft errors become an increased issue [17].

- A **Single Event Upset (SEU)** is a state change in a memory element, like a Flip-Flop, Latch or SRAM, resulting in a static error, caused by the ionization of a high-energy particle. But, not every induced current pulse leads to a bit-flip, since in CMOS logic always one part of the coupled logic is already turned on and therefore only the OFF transistors are sensitive (see also 2.2.2). Further, if a temporal SET is sampled in a memory unit, it is also referred to as a SEU, most likely as a Single Bit Upset (SBU). In case of an SEU in the configuration memory in an FPGA, which contains the routing information, random errors in the logical circuits and unpredictable behaviour are the results. It can also happen, that the energy injected by particle is so large, that more than one bit-flip is caused. If the charge is spread over several sensitive areas in the circuit and alters their logical states in parallel, one talks of a Multi Bit Upset (MBU). Due to the continuous downscaling in manufacturing processes, this effect occurs more often nowadays. A study, which presents on-orbit results with a Virtex-4 FPGA in a spacecraft came to the result, that 6,37% of the SEUs were MBUs (up to 6bit) [75].

Although they do not lead to permanent mechanical damage, soft errors in integrated circuits are meanwhile a huge concern because they influence the correct function of the system much more than all other hard reliability mechanism (gate oxide breakdown, metal electromigration, etc.) combined. Compared to them, the appearance of soft errors is minimum 250 times more likely [17].

2.2 Soft Errors

From a designers perspective, soft errors are far more important during the development and implementation of a digital design than hard errors or aging effects. Most of the mitigation techniques, like *Redundancy*, needs to be considered during the early design phase to avoid extraordinary time and effort later. The sensitivity of data and control path in a digital design is application dependent and a meaningful quantification of errors is necessary to estimate the level of mitigation required to meet the designers expectations. In a huge setup like the CBM experiment not only a valuation on device level is sufficient but a system task. See also 4.2.5. As earlier mentioned, SETs only become a problem in digital circuits if they are sampled in a memory element. Therefore, mainly SEUs are responsible for soft errors.

2.2.1 Fault Declaration

Not every soft error in semiconductor devices affects the data or control path in that way, a permanent fault is produced. More precisely, it depends on the type of hardware, the type of logic and the state of execution. The correct classification of hardware is crucial as the successful operation depends on the selection and implementation of particular mitigation techniques. Obviously, the further examination is related to the challenges of DAQ networks.

As a first step, a rating of components is important about how much influence they have on the running design and in which operational state (e. g. initialization vs. in service). For a digital design the determination of modules which can lead to a dead or live lock in the system is useful. Bit flips in the control path are far more critical than bit flips in the data path, as without the appropriate control signals they are not even noticed. By the way, payload can very easily protected with checksums. A modified state vector of a Finite State Machine (FSM) will most likely always lead to faulty system behavior, as all output signals of the module trigger further processes. In contrast, a modified value at the data input of a FIFO, as long as no write enable is assigned, will have no effect as the wrong value is overwritten within the next clock cycle.

Hence it can be distinguished between the following types

- **Data path faults**, only affect the record, which is processed at that moment but can lead to wrong interpretation results in data processing stages.
- **Control path faults**, changing the behavior of state machines, memory elements like FIFOs, or pipeline stages. This can result in unpredictable state transitions or wrong read and write operations in memories or registers. At worst, a dead lock occurs and the logic needs to be reset. In both cases data loss is very likely.
- **Configuration faults**, which can lead to wrong settings in an ASIC like modified values in a register file changing the calibration of analog circuits in the chip. In an FPGA, faults on the configuration memory will result in faulty connections of configurable logic and therefore unpredictable system behavior.

As the declared errors above have diverse impact on the DAQ system, for every fault type a reasonable adapted mitigation technique should be used.

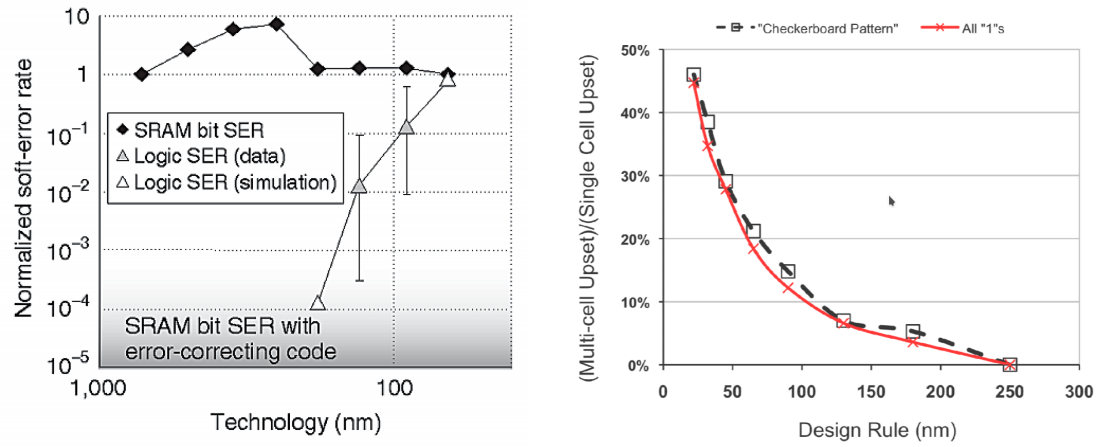
2.2.2 SER, FIT and Cross Section

The rate at which a device or system encounters soft errors is called the Soft Error Rate (SER). It is typically expressed as the number of Failures In Time (FIT), while one FIT is equivalent to one failure in one billion hours of device operation. Because in nowadays technology nodes a SER of 50.000 FIT in the memory of a chip under terrestrial conditions will only result in one soft error every two years, assuming a 24/7 running, this might be sufficient for standard customer scenarios. For high reliability applications and where many chips are used in parallel the failure rate can drastically increase. Obviously, a variation of environmental conditions, like heavy irradiation, raises the SER strongly.

Modern mixed signal ASICs consist of digital logic, analog circuits and various additional area consuming structures, like metal fill or decap cells. Thus, not all area in a semiconductor device is sensitive to soft errors. Furthermore, in a digital CMOS logic the active area is always only 50 %, as one part of the logic is already ON respectively OFF. For a specific device and a known particle the probability that a soft error occurs is given by the cross section. It proportionally depends on the size of the particle and the effective area the device offers to the particle. For a known device cross section σ and particle flux Φ the soft error rate can be calculated easily:

$$R_{SE} = \Phi \cdot \sigma \tag{2.3}$$

with the unit $1/s$. For particles, the cross section rises from neutrons, over protons to heavy ions and for a particular particle type it also rises with increasing energy until a saturation. Also the device cross section depends on technology, manufacturing process and, of course, on the designers layout. For Xilinx FPGA devices cross sections for several particles can be found. For ASICs, the cross section needs to be estimated with simulations or beam tests.



(a) SER comparison of SRAM and logic obtained from silicon characterization and simulation [17].

(b) Increased probability of MBUs in SRAM cells with decreasing feature size, taken from [92].

Figure 2.2: SEUs trends with decreasing technology feature sizes.

2.3 Radiation Impact

All sub-micron semiconductor devices suffer from SEEs to some degree, while the impact on single-user commercial hardware is very application dependent and more or less negligible. In the CBM experiment, where the hardware is exposed to very intense irradiation and larger setups of components run in parallel, the constructional differences and error-proneness of FPGAs and ASICs needs to be carefully considered. Despite the fact that they are built-up with nearly the same elements, some architectural characteristics make ASICs less sensitive to soft errors as their logic interconnections are hard-wired and no configuration storing is necessary. Also, differences regarding aging effects exists.

2.3.1 Elements

Sequential logic elements like latches and flip-flops are built-up nearly similar to SRAM cells, as they use cross-coupled inverters as storage. But usually they are more resistant against disturbance because often multiple transistors drive a node and they are capable of driving higher currents for the subsequent logic than SRAMs, which are highly optimized for less area consumption. Nonetheless, the

SER sensitivity of sequential and combinatorial logic circuits is also increasing with scaling and for a technology process of 65 nm and below, it reaches the same order as depicted in Fig. 2.2a. The SRAM per-bit SER could be held steady although the feature size is decreasing and the total amount of SRAM bits per device has increased. Certainly, for both elements the amount of charge necessary to alter the state depends on the node capacity, operating voltage and drive strength of the transistors. A bit flip in a data or control path does not necessarily lead to an error. It rather depends on the functional state, like a circuit can do something system critical or just standby. A bit flip in a memory unit definitely leads to a permanent error if the cell is occupied. With decreasing feature size also an upset in multiple cells caused by a single induced charge is more likely as is depicted in 2.2b. The actual differences along this trend also depend on cell layout and how charge is collected and shared. Thus, MBUs should be considered in high reliability applications.

The impact of soft errors on oscillator circuits, like DLLs or PLLs, which are also widely used in integrated circuits, should also be observed. Experimental results from [73] have shown, that VCO and charge pump are the most sensitive components. Although their sensitivity against induced SETs in the feedback path increases proportional with the frequency, the final impact on the output signal in lock mode is ambivalent. Strikes in the charge pump of a 400 MHz PLL result in slightly varying duty cycle and increased jitter with up to 1 ns in rare cases, but no missing output pulses or speed variations were observed. However, it should be mentioned that, depending on the system, distorted clock pulses may affect sensitive circuit elements and result in data-transfer corruption across entire distribution networks [51].

2.3.2 FPGAs

The basic concept of an FPGA is that any boolean function can be mapped in a truth-table and therefore any algorithm can be represented with a combination of Look-Up Tables (LUT), Flip-Flops (FF) and Multiplexers (MUX). These elements are grouped in Configurable Logic Blocks (CLB). LUTs act to replace traditional gates like they are used in ASIC designs. All used Xilinx FPGAs are i. a. built-up with CLBs, Block RAMs (BRAM), DSPs, Clock Management Tiles (CMT) and special circuits for the I/O region, like Multi Gigabit Transceiver (MGT) or other

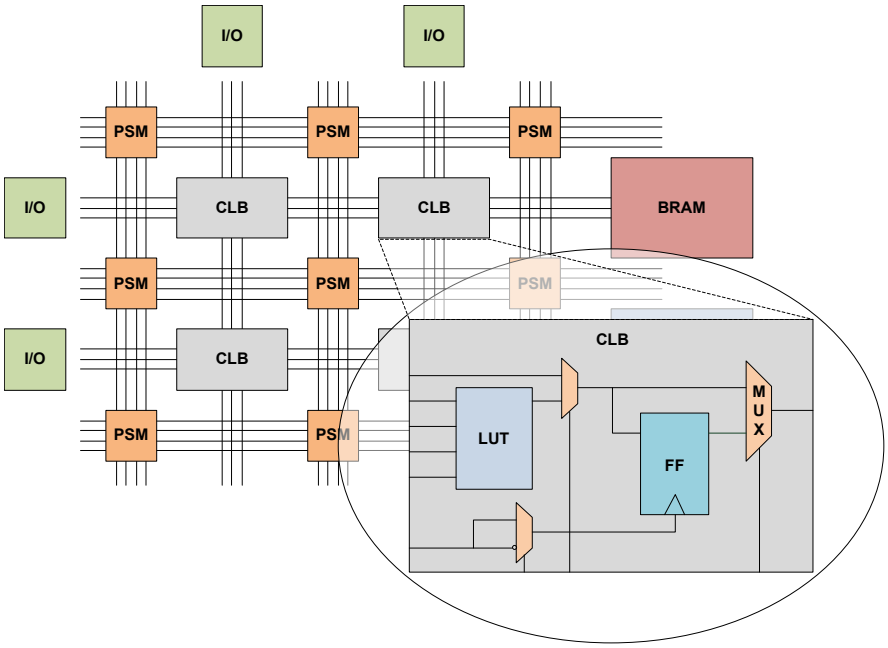


Figure 2.3: Very simplified view of the composition of a Xilinx FPGA with CLBs and PSMs. LUTs represent the combinational logic of the design, while FFs store values. With the PSMs the logic is connected depending on the bitfile loaded in the FPGA.

hard IP cores. These elements are dynamically interconnected with Programmable Switch Matrices (PSM) in a hierarchical manner, so that they fulfill the designers implementation of a function or algorithm. Thus, the final wiring depends on the firmware loaded into the FPGA. CLBs and PSMs are made up with SRAM cells because they make the best offer for a timing-efficient and speed-optimized design. As an alternative, Flash memory based FPGAs more recently have been proposed to be less sensitive to SEUs, due to the high capacity of the floating gates. Nonetheless, they strongly suffer from aging effects (no possible re-programming after a TID of 2.5 krad) and also hard errors are causing a problem. In the past, Flash-based FPGAs were not available in sizes and speeds that are comparable to those of SRAM-based FPGAs, and are therefore not foreseen for the final CBM experiment setup [29].

Unfortunately, SRAM based FPGAs are susceptible to SEUs. In contrast to the values dynamically stored in FFs, the configuration written to the PSMs and LUTs is static and does not change during operation mode. An SEU induced

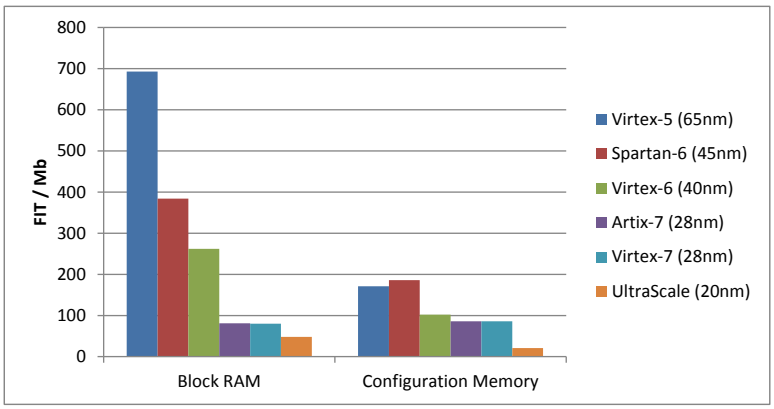


Figure 2.4: The soft error rate in FIT/Mb at nominal VDD and temperature. Data taken from [4].

error in these units will lead to random system behavior, like wrong re-wiring in combinational and sequential logic, configuration changes in FPGA cores like clock managers and global reset, and thereby lead to a locked system. But there are two aspects reducing the impact of SEUs on FPGA configuration memory. First, the SRAM cells used here are more robust designed and have higher capacitance than SRAM cells used for general-purpose memory, which are rather optimized for speed and area. Actually, the probability for a bit flip in the configuration memory compared to the whole SEU rate, is around 10%, as only a small amount of configuration cells is used in a conventional design. Second, Xilinx FPGAs in general are designed to have an inherently low susceptibility to SEUs [3]. Especially with the latest UltraScale/UltraScale+ devices, Xilinx reaches up to tripled resilience with techniques spanning process, layout, circuit and device architecture. Actually, Xilinx devices at 65 nm have shown nominal rates on the order of below 200 FIT/Mb for configuration memories and below 700 FIT/Mb for Block RAM [4].

It is also possible to read the configuration back from the memory and compare it against the original bitfile to check for arisen errors. A technique called *Scrubbing* (see 2.4.5) allows the periodically reprogramming or partially reconfiguration of the FPGA, while still operational. Moreover, there are also developments on autonomous repair cells for FPGAs [69].

2.3.3 ASICs

In the CBM experiment mainly mixed-signal ASICs are used. The digital logic is either implemented with standard cell libraries or, like in the SPADIC, a home made standard cell library is used where separated substrate contacts are possible which allows for a better isolation between bulk material and digital signal transitions [13]. But always at least a full-custom physical layer implementation is required for the I/O interface where also digital elements are developed individually to reach speed and drive strength. One big advantage of ASCIs over FPGAs is the independence from a configuration memory. As their connections are hard-wired, no programmable logic is necessary and this significantly reduces the effective error rate. Regarding the effects of radiation, the digital part suffers far more from SEEs than from TID effects, but the latter have more impact on the analog part. For the relevant process technologies from 65 nm to 180 nm, aging effects can lead to increased leakage currents and threshold shifts. Further, NMOS compared to PMOS transistors show a higher susceptibility to these effects due to the different electron-hole mobility in the silicon-dioxide [12]. However, they can be mitigated due to special layout design techniques like the use of guard rings, separated potentials, transistors with round gates, variations in wafer process and additional well contacts. The possibility of hard errors like latch-ups can also be mitigated with special processes, like silicon on insulator (SOI). Besides, as earlier mentioned, when not exposed to radiation, a self-annealing process in the silicon-dioxide takes place, revoking TID effects.

With the decreasing feature size in ASIC manufacturing, SEEs have become more concern because of lower capacitance, lower operating voltages and increased clock speeds. Severe SETs are more likely and can easily be latched by subsequent logic. Also SEUs in storage elements, like flip-flops, registers or memories, can occur, resulting in functional errors in the digital logic [31]. For ASIC designs in technologies from 40 nm to 90 nm, rates from 1000 to over 5000 FIT per million logic gates are expected and a similar SER for memory cells [49]. Finally, there is no golden rule for ASIC development. Hardening techniques must always be chosen considering process and design options, and also applied differently across all circuit structures because the overall sensitivity will be most likely limited by the reliability of the most sensitive component.

CBM Detector	TID per year	Hadron Flux Φ
STS	8 Mrad	$1 \cdot 10^6 \frac{1}{s \cdot cm^2}$
TRD	10 krad	$5 \cdot 10^4 \frac{1}{s \cdot cm^2}$
PSD	1 Mrad	$4 \cdot 10^6 \frac{1}{s \cdot cm^2}$
Cave	20 rad	$100 \frac{1}{s \cdot cm^2}$

Table 2.1: FLUKA Monte-Carlo simulation results with a 35 GeV Au beam on a 250 μm Au foil target with 1 % interaction rate. TID and particle flux are given for the CBM STS and TRD detectors as well as for the PSD. Values taken from [89].

2.3.4 Relevance for CBM

Regarding the relevance of radiation tolerant read-out electronics in the CBM experiment, the demands on the STS detector DAQ system exceed the needs for the TRD because the devices are located closest to the beam and the targets, where the collisions takes place. Therefore the examination is only done for the worst operating conditions. In the final read-out chain, for the STS and TRD detectors only ASICs will be used. Nonetheless, FPGAs are used for prototyping and are widely used in smaller beam test setups. Although the final radiation intensity and operating time are still not known in this moment, the expected radiation dose per year will reach up to 8 Mrad in worst case for the STS detector system (See table 2.1), which will have no severe influence on the FPGAs and ASICs, as for the used technologies from 65 nm to 180 nm a complete self-annealing of TID effects within a few days without irradiation is expected. FPGAs can be operated reliably up to a total dose of 300 krad and for ASICs the limit is even higher. Studies on TID effects with a dedicated test ASIC showed its complete regeneration within a few days without beam and at room temperature, after exposition to an ionizing dose of 2.4 Mrad [50]. Therefore TID effects on the used devices are negligible for the CBM experiment and will not be observed further in this thesis.

The impact of hard errors is strongly reduced by the selection of an SOI process,

avoiding parasitic NPNP junctions. Nevertheless, the earlier mentioned layout techniques should be used at least for power MOSFETs, since later discovered errors can only be fixed by an expensive chip re-layout and fabrication.

Before the relevance of SEEs on the DAQ network is discussed, a brief analysis of the earlier mentioned findings, regarding the different SER per bit of FPGAs and ASICs, is necessary. Although Xilinx did put a lot of effort in the development and evaluation of robust cell design ever since, it is still questionable, that FPGAs show better results by factor ten compared to ASICs. Obviously, in an FPGA design only a small amount of the whole CLB capabilities is used to implement a function. Thus, the many times better error rates only compensate that in an FPGA a much larger number of transistors is necessary to implement the equivalent ASIC algorithm. As FPAGs are built-up with SRAMs and the SER of SRAMs and logic lies within the same order, a similar FIT amount has been supposed for the further examination, as the effective sensitive area for one functional unit stays the same in both devices, wether the layout is very dense like in an ASIC or rather relaxed like in an FPGA. The key goal is to estimate a FIT rate for the final read-out chain with ASICs by running tests with prototype FPGAs. In verification tests, the fact that FPGAs compared to ASICs are more vulnerable to SEUs, because they need a configuration memory, is compensated with blind scrubbing, where the genuine firmware is periodically written to the configuration memory to correct arisen errors.

Regarding soft errors, the calculation of the final error rate in the whole read-out chain is extremely complex, as many factors, e. g. the gradient of the particle flux or the actual cross section of sensitive area (regarding devices, wiring and positioning), come into play. With good cause, only a rough estimation, like in [53], is given:

- The used SyoresCore3, as prototype for the HUB ASIC, is equipped with a Xilinx Spartan 6 SLX150T FPGA with $33.9 \cdot 10^6$ configuration bits in total [99]. Thereof $28.3 \cdot 10^6$ bit configuration memory and $4.82 \cdot 10^6$ bit of BRAM memory [98].

- The beam intensity is given with minimum needed 10^6 particles per second for the detectors, up to maximum 10^9 particles per second useful for electronic tests [88].

- A CBMnet V2.0 link core with a four lane LVDS master occupies 14,5 % of the FPGA configuration memory and 6 % of the BRAM memory, disregarding globally used units like clocking.
- The neutron cross section per bit of the Spartan 6 configuration memory is given with $1.0 \cdot 10^{-14}\ \frac{cm^2}{bit}$ and for the BRAMs with $2.2 \cdot 10^{-14}\ \frac{cm^2}{bit}$ [4].
- The amount of CBMnet cores required is equal to the number of STS-XYTER chips, which are in an order of 15 000 for the final setup [41].

The resulting average bit cross section for the four lane CBMnet core is

$$\begin{aligned} \sigma_{bit} &= \sigma_{conf} + \sigma_{BRAM} && (2.4)\\ &= \frac{28.3}{33.9} \cdot 0.145 \cdot 10^{-14}\ \frac{cm^2}{bit} + \frac{4.82}{33.9} \cdot 0.06 \cdot 2.2 \cdot 10^{-14}\ \frac{cm^2}{bit} && (2.5)\\ &= 0.14 \cdot 10^{-14}\ \frac{cm^2}{bit} && (2.6) \end{aligned}$$

and the average core cross section

$$\begin{aligned} \sigma_{Core} &= N_{bits} \cdot \sigma_{bit} && (2.7)\\ &= 0.14 \cdot 10^{-14}\ \frac{cm^2}{bit} \cdot 33.9 \cdot 10^6\ bit && (2.8)\\ &= 4.75 \cdot 10^{-8}\ cm^2 && (2.9) \end{aligned}$$

As earlier mentioned, around 10 % of the SEUs happen in the PSMs altering the interconnects of the FPGA. Therefore, as an estimation for the error rate in an equivalent ASIC, only 90 % of the errors are supposed. With the particle flux and the formula from 2.3, the SER per core and for the small part of the chain, running 15 000 cores in parallel, can be calculated. The results are given in table 2.2.

Obviously, this analysis does not take MBUs into account. Moreover, no estimation can be made if an error occurs in a functional unit which is crucial for the core operation. Nonetheless, this rough estimation is useful, when the necessity of hardening techniques is discussed for the CBMnet protocol in later chapter.

Particle flux Φ	TTF / Core	TTF / Chains	SER / Core	SER / Chains
$10^6 \frac{1}{s \cdot cm^2}$	428 s	28.5 ms	0.0023 $\frac{1}{s}$	35.1 $\frac{1}{s}$
$10^7 \frac{1}{s \cdot cm^2}$	42.8 s	2.85 ms	0.023 $\frac{1}{s}$	351 $\frac{1}{s}$
$10^8 \frac{1}{s \cdot cm^2}$	4.28 s	285 μs	0.23 $\frac{1}{s}$	3510 $\frac{1}{s}$
$10^9 \frac{1}{s \cdot cm^2}$	0.428 s	28.5 μs	2.3 $\frac{1}{s}$	35100 $\frac{1}{s}$

Table 2.2: Time To Failure (TTF) and SER of single and multiple cores for various beam intensities. It should be mentioned, that the TTF means the time until errors occur in the core or within the chains. In this example, several chains run in parallel, which are independently operating. Thus, an error in one chain does not affect others, neither the whole tree is blocked due to an error.

2.4 SEU Mitigation and Hardening Techniques

The reasonable selection of technology and production process of a semiconductor device is important to decrease the susceptibility to irradiation later. Further, layout techniques or improvements within the backend flow are also used to reduce the occurrence of SEUs, but these solutions can seldom reduce the SER by more than five times [17]. Besides, there are a lot of hardening techniques for the development of digital CMOS designs, like Error Detection And Correction (EDAC) methods, which always work with adding redundancy. They can either be implemented using securing methods on the bit or word level, like single error correction double error detection (SECDED), or redundancy on modular or system level. Global control path and state checking mechanisms, like watchdogs, can monitor sequences on higher levels, to avoid live or dead locks. The combination of these methods allows the development of runtime reconfigurable modules with a concurrent error detection mechanism [74]. In all cases additional circuitry is required. In systems with very high reliability demands on the data transmission, the correction of corrupted data is mandatory. The implementation of redundancy on the link or higher layers needs to be carefully considered, as it can have also negative impact on the error tolerance. The idea is to duplicate critical components or functions to increase the reliability of the system. This can be done

	Hardware Redundancy	Information Redundancy	Time Redundancy
Method	Modular Redundancy	EDAC	Multiple Execution
Example	TMR	CRC / FEC	ARQ (Retransmission)
Level	Logic / Module / Device	Word / Message / Record	Sequences
Layer	Physical / Link / Network	Physical / Link	Link to Application
Costs	Min. 200% Resources	Up to 50% Bandwidth	High Control Logic Overhead
Latency	Voting Stage	Appending / Checking Stage	Multiple Execution Times
Maintenance	Observation and Repair	None	Observating System / User
Implementation	Moderate	Easy	Complex
Reliability	Depending on Maintenance	Depending on Method	Depending on Implementation
Error Proneness	Moderate	Low	High
Worst Case	Faulty Behaviour	Useless Data	Livelock / Deadlock

Table 2.3: Rough overview about forms of redundancy and their possible implementation, as well as advantages and disadvantages.

on several hierarchy levels, like system, device, module or logic. While hardware redundancy, such as Triple Modular Redundancy (TMR), is mainly implemented in the control unit, from very fine grain level to very coarse level of whole devices, information redundancy techniques, like Error Detection And Correction (EDAC), are rather used for data. With time redundancy, a multiple execution of the same operation is performed on higher layers (e. g. application layer) to determine if the result is correct or a retransmission is issued in case data have been marked as faulty. Obviously, one has to be careful, as the decision logic, whether a result or transmission is right or wrong, can also be prone to errors. Depending on the method, the resulting demands for additional resources can be very high and the more complex system may lead to less instead of higher reliability.

2.4.1 Hardware Redundancy

In case of TMR, a module or system is triplicated and performs the same operation simultaneously. If one of the three systems fails, a majority voting decides which result is taken for the single output and thereby correcting the error. This method can mask any single fault and many combinations of multiple faults but there are several limitations and side effects. An error in the voter will always lead to a wrong

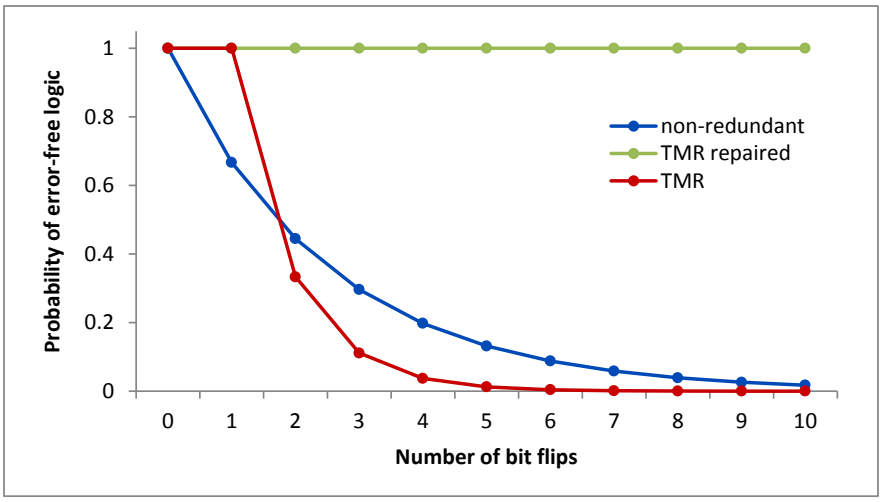


Figure 2.5: Comparison of non-redundant logic and TMR logic, regarding the probability to have an error-free system after several bit flips. In case the TMR logic is repaired after every bit flip, the system stays completely error-free. Figure similar to [53].

majority decision and an error in the subsequent system. An alternative is to shove the voter to the next level or to enlarge the system, which is triplicated. This, at least, has the advantage that resources are saved compared to fine grain TMR, as the voter logic is necessary only once. But this does not always make sense, as the possibility depends on the device or logic and chances for two simultaneous errors are increased. Additional resource demands may also require more I/O pins, which are rather limited. As the output of the triplicated logic is compared periodically, the method will not work over clock domain crossings. Synchronization stages have no deterministic timing behavior and therefore a result may be delayed due to different sampling. TMR can also not protect against MBUs as two wrong results in the triplicated module will mislead the voter. Last and most important fact is, that TMR can have lower reliability than non-redundant logic if several bit-flips occur and the logic has not been corrected meanwhile. This is true, if bit-flips lead to persistent faults, altering the state of the logic. In case of an error in the data path, it will be masked and corrected by TMR. After the first upset, the probability of a second fault is increased due to the larger occupied area and more used resources. Assuming that the TMR design uses three times the resources of the non-redundant design, the probability of a fault in the latter is always 33 % compared to TMR logic. After a first fault, the probability is still

33 % in the non-redundant logic, but in contrast, for the TMR logic it is 67 %. This behavior leads to the probability of a functional system, depending on the number of bit flips n compared to the method, which has been examined in [53] and can be described with

$$p_{functional}^{(n)} = \prod_{i=1}^{n} \left(1 - p_{error}^{(i)}\right) = \begin{cases} \left(\frac{1}{3}\right)^{n-1} & \text{TMR design} \\ \left(\frac{2}{3}\right)^{n} & \text{non-redundant design} \end{cases} \tag{2.10}$$

The comparison of non-redundant logic and TMR logic, regarding the probability to have an error-free system after several bit-flips, is depicted in fig. 2.5. Hence, TMR needs repair with a higher repetition rate, than errors are expected for the whole TMR logic, to improve system reliability. Otherwise the method is even more worse than the renouncement of redundancy. This likely means to fix the current state of the logic, which also produces an unscheduled transition and causes further maintenance, e. g. resynchronization. In case of an FPGA, it may also be necessary to fix the configuration memory. If the bit-flip occurs directly on the data processed within the TMR logic, the error will be masked anyway and repairing is unnecessary. Regarding the latency added by TMR, only the voting stage adds delay.

2.4.2 Information Redundancy

To improve the reliability of digital data transmissions, redundancy can also be added on lower layers. Checksums, parity bits or Cyclic Redundancy Checks (CRC) are error-detecting codes on the link or physical layer and work by adding a hash to a message, which is generated through polynomial long division with the message or a lookup table. Alteration in the message will lead to a different hash and thereby offer inequality. If no data correction mechanism is available, a checksum can only inform that the data have been corrupted, but not in which magnitude. Thus, data will be useless, but the detection of errors happened is highly secured. Depending on the length of the checksum, and the granularity the checksum is employed, the bandwidth is respectively reduced due to message overhead. As an alternative to only detecting errors by CRCs, a Forward Error Correction (FEC) can be used to secure a message with an error-correcting code (ECC). The included redundancy allows a processing stage the detection and

correction of a limited number of bit faults, depending on the length of message and ECC. Therefore, overhead highly depends on the correction depth favored. In general, there is not much to say, as the principle is very common, but a short examination regarding DAQ systems is necessary. In all cases the impact of radiation is decreasing with every hierarchical device away from the beam, so it is obvious that the checksum or parity bits should be added to recorded data as early as possible to hedge errors in-between. The hash generation must be verified and observed very accurately, as wrong hashes can not be corrected anymore and may lead to live locks in later processing stages. For example, every correction mechanism, like a retransmission, will result in an infinite loop as as the CRC check will always fail, no matter how many times the data have been sent again.

2.4.3 Time Redundancy

In case a FEC is not applicable or not sufficient, reliability can be ensured by retransmitting faulty data. This method is also known as Automatic Repeat reQuest (ARQ) and comes along with a huge amount of complex sequence logic. ARQ works with positive or negative acknowledgements and timeouts to control the data flow and avoid deadlocks. Transmitter and receiver always have to exchange information about their current network state to ensure the correct data will be retransmitted. As some messages may be shorter than the delay than the communication channel, the transmitter may have sent several messages until notified about damaged data. Thus, all messages need to be stored until their proper receipt has been acknowledged from the subsequent stage and a unique identification of messages must be guaranteed with framing and control characters. Storing of messages can be done in a ring-buffer, where the filling level is well-regulated by different pointers. A retransmission caused by damaged data is depicted in fig. 2.6. A second constraint is that the timeout check needs to be set regarding the round-trip time, which may vary through different cable lengths or different physical layer implementations. As the transmission over high-speed serial links, under normal conditions, has a heavily increased statistical error rate compared to the SER inside integrated circuits, a retransmission mechanism is reasonable to correct faulty data. Only corrupted framing and acknowledgement messages needs to be handled.

In case the integrated circuit suffers excessively from soft errors due to strong

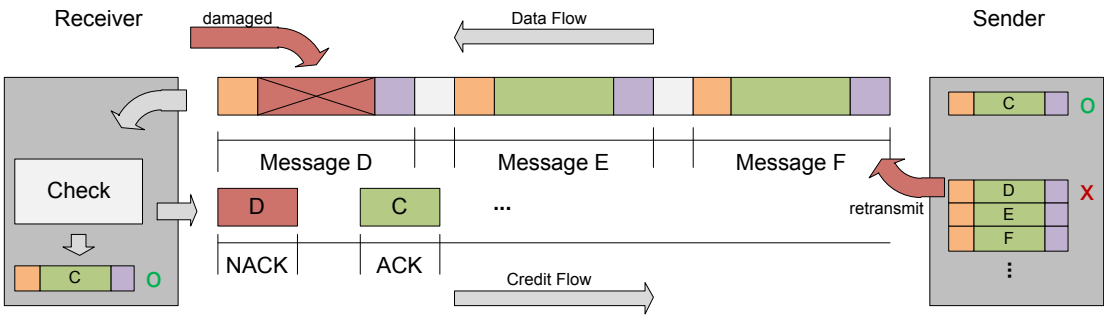


Figure 2.6: Example of the data flow between two endpoints using retransmission for data correction. Sent messages need to be stored at the transmitter side and several messages may be sent until positive or negative acknowledgement. Soft errors in the inner-module control state (e. g. full or empty signals, pointers, ...) can lead to data loss or lock.

irradiation, the use of retransmission logic needs to be carefully considered. The sequence control FSMs are error-prone and a bit-flip may lead to a stuck network logic. The ring buffer memory, usually built-up with SRAM, and the pointers need to be secured to avoid data loss. Wrong pointer values may overwrite former entries or compound messages. As two link partners are involved, every communication is handled over an unreliable link and has a high latency between the endpoints control logic. In case of an error or timeout, the other endpoint needs to be informed, but further errors in the communication can not be precluded. Summarizing, reliable retransmission as error control method in an environment with high SERs is not an easy task and in case of very high SERs, the system is prone to deadlocks. It should also be mentioned, that in case the message or CRC is corrupted before written to the transmission buffer, the transmitted data will always result in a negative acknowledgement at the receiver side and an infinite loop of retransmissions.

2.4.4 Control Flow and Monitoring

Compared to microprocessors, where the running of an application is very expensive in terms of computation time, and many interim results are saved to and read from memory, the processing of messages in a network node is insofar easier, as the messages contain all their context. A processing of a message in a network

unit can be seen as a short finite timeframe. After this time frame the network unit starts over, not depending on a context. This enables the possibility to encapsulate the processing and determine checkpoints, from which the execution can be restarted after a functional error. Likewise techniques are called lockstep or rollback recovery and are well-known from microprocessors, where the context is saved every checkpoint to have a error-free copy of all information required to restore the processors state [10]. In a network protocol the restoring of a clean state of the control path is very important, as otherwise deadlocks can occur and the whole system gets stuck [25]. To avoid this, two demands have to be met. The logic needs a well-defined procedure, how to go ahead with data left in the unit after a local rollback and how to be reintegrated in the flow, because subsequent stages also have to deal with the unusual state change. Secondly, special purpose hardware modules, known as watchdogs, should be used to detect erroneous logic or system behavior. They monitor the control flow execution and perform consistency checks. In case of a malfunction or timeout they trigger the rollback to a checkpoint and the reintegration.

2.4.5 Scrubbing

As earlier mentioned, the drawback of SRAM-based FPGAs is that SEUs can onset in the configuration memory, resulting in wrong re-wiring of elements and may lead to random system behavior. Scrubbing is a technique to detect errors by reading and comparing the whole or partial memory against a genuine bitfile. After bit errors were detected the corrected data are written back to the FPGA. To detect for arisen errors, the read back strategy compares the corrupted firmware against the original one. If only a refresh of the configuration cells is favored, blind scrubbing can be used, by just overwriting the former configuration. While blind scrubbing is more than two times faster compared to read back, the latter gives valuable information about configuration alteration. Scrubbing can be performed on different granularity from only refreshing a module, or frame, to the whole device. If done internally, the scrubber logic, which issues the interchange to and from the memory, can use the Internal Configuration Access Port (ICAP), which needs much less time to repair. If done externally, an additional microprocessor or ASIC is required and read and write commands are done over e. g. a JTAG interface. Xilinx offers with the Soft Error Mitigation (SEM) Core (available since the Series 6 devices) a complete framework for SEU detection, correction and

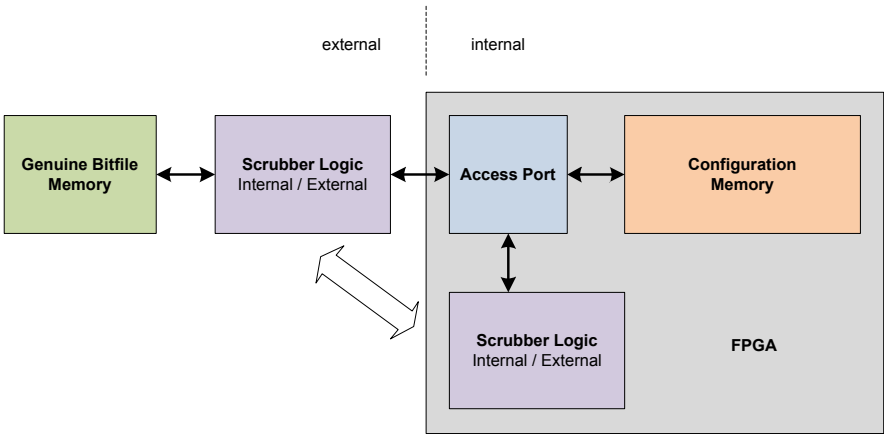


Figure 2.7: FPGA configuration scrubbing with blind or read back strategy. The scrubber logic can be implemented within the FPGA fabric or provided by an external device.

classification. The Xilinx ICAP of the FPGA is used for internal scrubbing, where no external hardware is required and the IP core also provides error injection feature to evaluate the mitigation capabilities [7]. Another group within the CBM experiment, which is closely involved in FPGA radiation mitigation techniques, is the Infrastruktur und Rechnersysteme in der Informationsverarbeitung (IRI) at the Goethe University Frankfurt am Main. Their latest publications on FPGA scrubbing can be found in [53] and [29].

2.5 Analysis and Verification

To verify implemented methods against single or multiple bit errors, fault injection tests can be done, altering logic bits in particular modules. A first analysis can be done with verification and simulation tools. Within the development of network components a software framework has been set up to emulate SEUs, induced by ionizing radiation. The software uses TCL commands to change values in selected nets for a given time and a given intensity (see 4.2.10). A second way to emulate the impact of SEUs on the design is to use FPGAs with readback scrubbing. While scrubbing is usually used to repair the corrupted FPGA configuration memory, for SEU emulation a modified firmware with a bit error is loaded in the FPGA. Although this method emulates the effects of radiation on the configuration memory

very accurate, the impact on the dynamic memory is disregarded, which makes it inappropriate regarding ASIC design verification. Further, the statistically relevant results are hard to reach, as for SBU tests only one bit per run should be altered in the firmware and every injection via JTAG needs several seconds. At last it should be mentioned, that debugging in an FPGA is very limited and usually only possible for one component. Finally, in-beam tests can give a validation that the design works under real radiation conditions, as fault injection tests alone can not cover all possible impact on the design.

2.6 Conclusion

The different kinds of radiation have negative impact on semiconductor devices and can lead to soft errors as well as device damage. Fortunately, aging effects are no longer a severe problem in sub 100 nm processes. Even after exposition to very intense irradiation, FPGAs and ASICs benefit from the self-annealing process at room temperature. Hard errors like latch-ups need to be avoided by reasonable process selection and appropriate hardware structures. While the probability of SEUs induced by terrestrial radiation on integrated circuits increases with the decreasing process feature size, it is still in a magnitude, where the FIT rates are acceptable for single-user applications, for the CBM experiment the expected effects of radiation are so heavy, that effective mitigation and hardening techniques had to be elaborated. Care must be taken when using redundancy techniques, as they can also decrease system performance and reliability. To implement a reliable DAQ network, a detailed rating of components is important about how much influence they have on the running system and how errors can be mitigated. Regarding the CBMnet development, the use of hardening techniques is described in chapter 4.

Chapter 3

Design Space

In this chapter an overview is given about state of the art DAQ networks and read-out solutions, which fulfill requirements from similar projects, like the GBTx, used in different experiment read-out setups of the LHC at CERN, or the TRBnet used for the HADES DAQ. Furthermore, a detailed description about the current state of the CBM DAQ setup and network protocol implementation is given. Since the use of the network protocol should be extended to several new FPGA and ASIC devices, fulfilling new tasks and being placed in an even higher radiation environment, reliability is a main claim regarding new implementations. Complex setups needs to be supported by the new ASIC devices and the need of several custom physical layer cores for application specific purposes arose. Thus, the advantages and disadvantages of the current network protocol are evaluated accurately in this chapter and bring up key aspects for the research contribution of this thesis.

3.1 DAQ Requirements

A Data Acquisition (DAQ) system is a collection of hardware and software to measure or control the physical characteristics of an event in the real world and the interconnection network is one of the key features, as it has to collect and process the output from all the instruments to reconstruct the physical process. Because of the many-to-one communication pattern, usually the evaluation of network topologies leads to the selection of a direct network with a tree structure. This scheme is actually demanding for network based technologies like Ethernet and TCP/IP, because it easily leads to overloading of buffers and switches. The

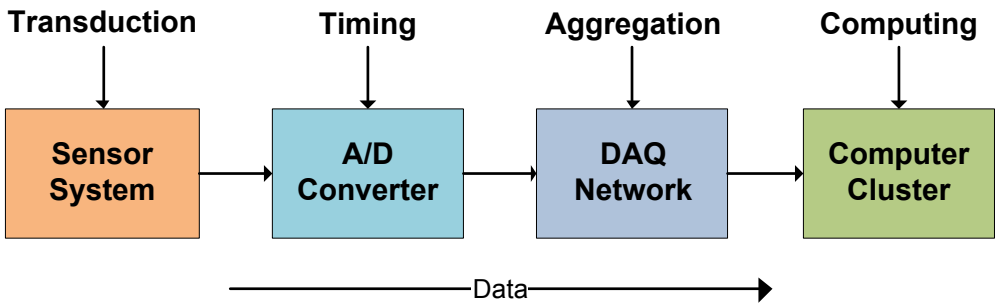


Figure 3.1: The chain of a digital data acquisition system.

main requirements that the interconnection network usually has to address are focused on

- Bandwidth
- Latency / Synchronization
- Reliability
- Scalability
- Congestion notification

As the sampling process should not depend on the network utilization, the network has to provide enough bandwidth to handle the instantaneous rates of multiple parallel data flows. For CBM, there are two communication patterns, the data traffic and the control/synchronization traffic. The data is only streamed from the front-end devices, which record and digitize the signals, to the back-end, which stores the data for later analysis on a computer cluster. Mainly in reverse direction, control and synchronization traffic is streamed from the back-end or from an intermediate network node to the front-ends. Each stage in the network aggregates data to reduce the number of links and therefore, the link speed increases gradually. As there is almost identical logic used for the network communication in several devices, preferably common, or at least unified hardware is used to reduce costs and simplify configuration and maintenance. The network must also be reliable to minimise the risk of an unrecoverable data loss. However, for CBM also a little data loss is acceptable (See section 3.4.2).

3.2 DAQ Solutions and Protocols

Earlier evaluation of DAQ solutions and other network timing protocols showed, that no solution could directly fit the CBM experiment requirements [44]. The standard timing protocols, like the Network Time Protocol (NTPv4) or the Precision Time Protocol (PTP) have flexible capabilities for the synchronization of an application layer on various physical implementations. Unfortunately, the timing resolution is very hardware-dependent and with respect to the CBM experiment, it is too little to reach the requirements of a few hundred nanoseconds. Well standardized telecommunication networks, like the mostly compatible Synchronous Optical Network (SONET) and the Synchronous Digital Hierarchy (SDH) are widely used solutions and convince with their modular and flexible network structure, high reliability and low costs. The synchronous network setting achieved with high-quality clocks over asynchronous transmission modes is a preferred solution to avoid special synchronization hardware and the need for bit stuffing. However, the fixed framing format does not support specific user formats very well and limits the possibilities of balancing network traffic. Another solution used for data acquisition purposes and detector instrumentation is the open source project White Rabbit (WR), which aims on fast and reliable data transmission and a timing deterministic control system with a combination of synchronous Ethernet and the PTP. Because of the use and compatibility with Ethernet, WR components are highly available, easy to handle and rather cheap, but bandwidth and throughput are limited to standard Ethernet speeds. Hence, it is not a good solution for a hierarchical network with intense data aggregation and increasing data flow speed. Besides, none of the above solutions delivers any protection against the effects of ionizing radiation on hardware and therefore, they are at least inappropriate for CBM in their original development state.

3.2.1 The TRB Network

A promising development for DAQ systems is the network of the Trigger and Read-out Board of the Hadron and Di-Electron Spectrometer (HADES) experiment running at the GSI in Darmstadt, called Trbnet [59]. The event rate is up to 100 kHz and the expected data rate is in the order of 400 MByte/s in peak. The TrbNet supports two different FPGA boards. The Trigger and Read-out Board (TRB), which provides 128 channels for Time-To-Digital-Converter (TDC) boards

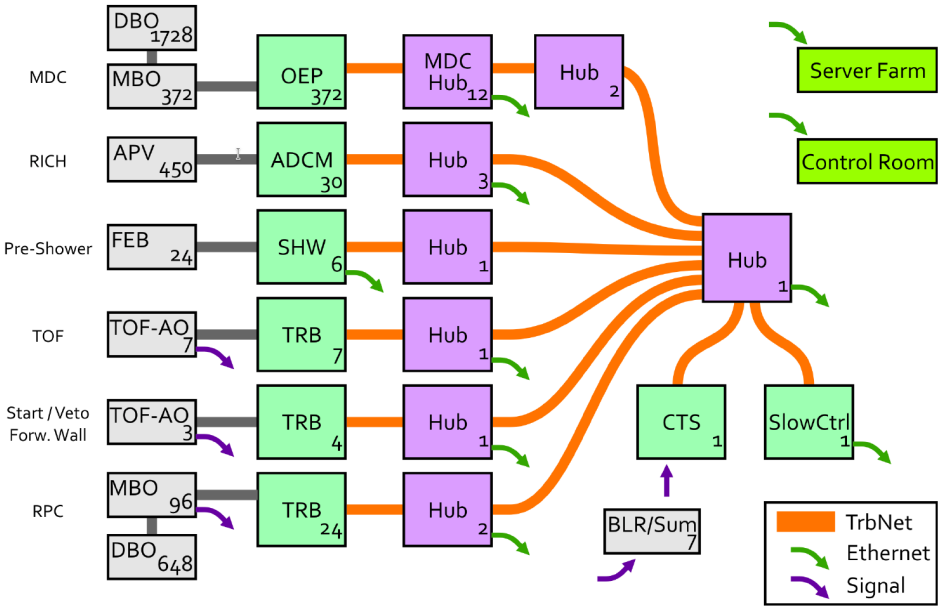


Figure 3.2: The HADES DAQ network setup. Front-end boards and Hubs are connected via optical links running the TrbNet protocol. Servers are connected over Gigabit Ethernet [60]

with a resolution around 30ps to take data from all timing relevant detectors of the HADES setup and the hub board, which is equipped with 20 optical links with a particular data rate of more than 3 Gbit/s and can also serve as a bridge to off-the-shelf computers with an Gigabit Ethernet interface. The TrbNet provides three virtual channels for trigger distribution, data transfer and monitoring and slow control in one common network protocol. Only clock and synchronization are distributed with a separate tree. The trigger messages are sent with the highest priority and have a very low transport latency of less than 3us through the whole network to all devices. The application provides data of 32 bit words, which are divided in packets of 80bit. The transfer layer adds a 16bit header, containing network addresses of sender and receiver, the length of the packet and information about the type of content to each 64bit of payload in case of a data packet or to one of several network control types, like a network header or a termination word. Sent data is also secured with a checksum and can be retransmitted if corrupted during transmission. Although the TRBnet measured performance meets or exceeds the requirements for the HADES experiment, for the challenging CBM detectors like the STS or TRD the data bandwidth is still too small, regarding an event rate

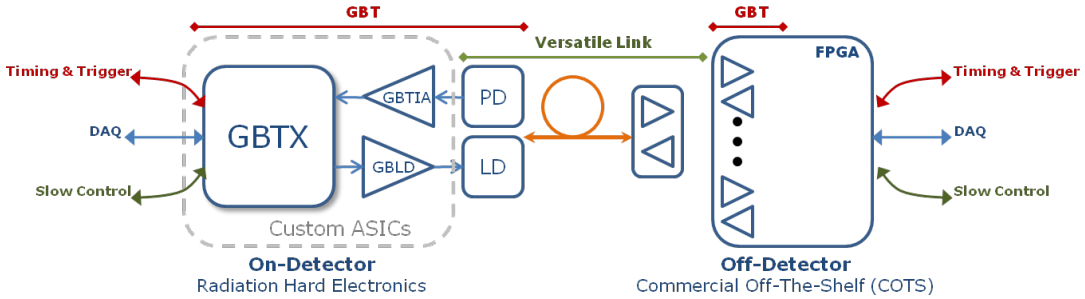


Figure 3.3: The architecture of the GBT chipset, consisting of four different ASICs, the GBTX, the GBTIA, the GBLD and the additional GBT-SCA [63].

up to 10MHz and a data rate of 2 TByte/s. The additional 8B/10B coding used for the serial transmission lanes reduces the final utilization below 60%. At that point in time the development of the CBMnet started, the TrbNet gave a nice impression of a reliable and well-adapted custom-built read-out network. Although the former limitations of an additionally required distribution network for clock and synchronization, as well as the limited bandwidth were not sufficient for CBM, meanwhile there have been foreseen some improvements. The increase of the link speed and available bandwidth, reduction of the protocol overhead, an option for a trigger-less system and the global clock distribution as integral part of the protocol [58]. Also the operating of the system in an environment with electromagnetic noise and radiation looks promising [60].

3.2.2 The GBT Project

The GigaBit Transceiver (GBT) [65] is a chipset development, used within the DAQ of the future upgrade of the LHC accelerator at CERN. Due to the Super LHC (SLHC) upgrade the amount of data rises by a factor of ten, while radiation levels in the front-end area also increase. The GBT has been invented to support a high speed data transmission up to 4.8 Gb/s between the detector read-out boards and the counting room, while also providing control and trigger data on the same link. Therefore the architecture is split into four different ASICs, namely the GBTX, for the serial communication with the front-end devices over the GBT protocol, the GBTIA, a transimpedance amplifier with an optical receiver, the GBLD, a laser driver for optical interconnects, and the GBT-SCA [21], a

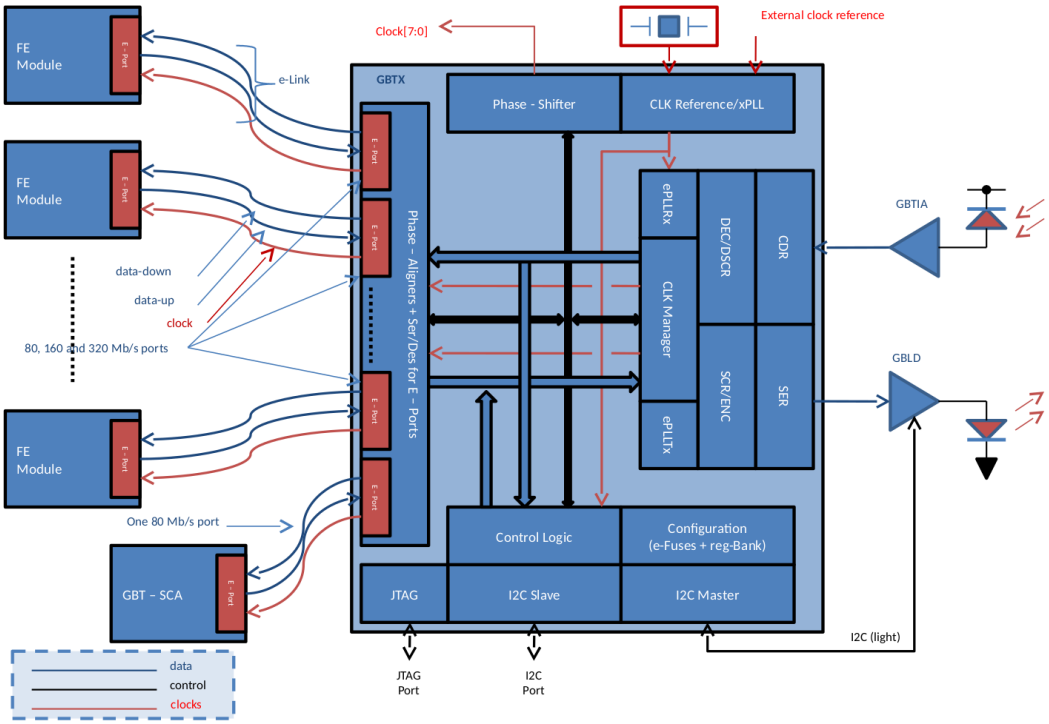


Figure 3.4: The structure of the GBTX ASIC with the 40 E-links to the front-end devices on the left side, and the 4.8 Gb/s transceiver connected to the laser drivers on the right [63].

dedicated ASIC for slow control. The architecture of the GBT is depicted in fig. 3.3. Similar to CBMnet the front-end implementation of the GBT protocol needs to be implemented on custom made ASICs, while in the counting room COTS components, like FPGAs can be used to run the protocol, since no radiation impact on electronics happens there [54]. In the front-end area, the SLHC leads to radiation levels of 10^{16} n/cm^2 and 100 Mrad over the whole experiment lifetime [64]. Thus, radiation robustness against bit upsets is ensured by Forward Error Correction (FEC) and Triple Modular Redundancy in low-speed and high-speed functions. Further, the ASIC is manufactured in a 130 nm technology to benefit from the inherent resistance to Total Ionizing Dose (TID) effects (See also 2.1.1).

The GBTX ASIC is mainly divided into two parts. On the back-end side the GBTX is composed of a 4.8 Gb/s serial transmitter and receiver, a CDR and a phase shifter, while the sending and receiving of one GBT frame is done over three

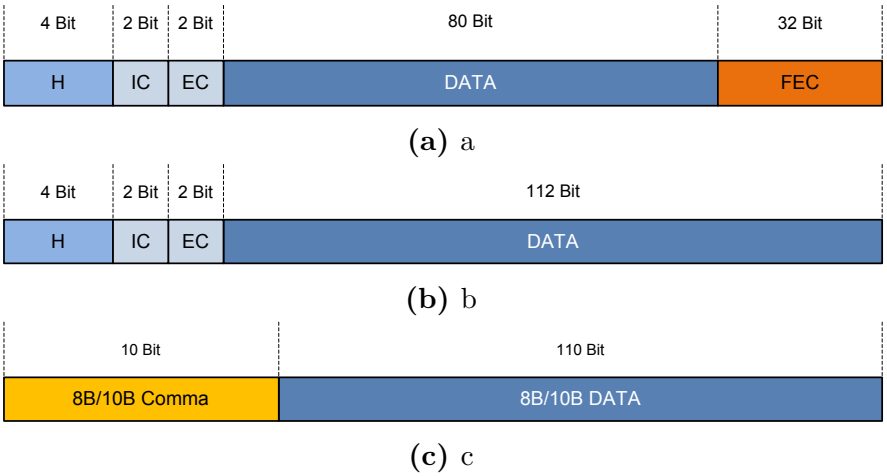


Figure 3.5: The different formats of the GBT frame, depending on the running mode. FEC, Wide Bus, or 8B/10B is selectable.

shift registers and a 3:1 high-speed multiplexer to reduce the amount of circuitry running at full speed. More information on the SerDes can be found in [62]. On the front-end side, one GBTX provides 40 bi-directional serial E-links running at 40 MHz, and providing a bandwidth of 80 Mb/s each, using SLVS drivers with DDR transmission. Every E-link consists of three pairs of differential wires for a clock, a data signal in front-end direction, and a data signal in back-end direction. Several of these links can be grouped together to increase the bandwidth for a single front-end device. For these devices, the GBT project proposes E-link port adapter macros which can be used for the integration in the front-end ASICs.

The GBT can operate in three different modes, which slightly changes the format of the GBT frame, the mapping of front-end groups and the I/O. However, the GBT frame has always a fixed size of 120 bit, like depicted in fig. 3.5. In the GBT mode (Fig. 3.5a) symmetric up and down links are used, and the GBT frame consists of a 4 bit header (H), followed by a 2 bit internal control field (IC) and a 2 bit external control field, 80 bit data (D), and 32 bit forward error correction (FEC). Thereby up to 16 consecutive bit errors can be corrected afterwards. When using the other two modes, the different frame formats apply only for the uplink, while the downlink always uses the GBT format. In the Wide Bus mode (Fig. 3.5b) the format is nearly the same but no FEC is used, which results in a higher effective user bandwidth, while error correction capabilities are disabled. In addition, the uplink data are scrambled to reach DC balance. In the 8B/10B mode (Fig. 3.5c)

the frame starts with a 10 bit comma character for frame synchronization, while the remaining 110 bit carry user data [63].

The GBT chipset provides data transmission, timing trigger and control (TTC) and slow control (SC) options on electrical and optical links and with the E-link port adapter macros and interesting feature of reusability within front-end ASICs is presented. But some architectural concepts show significant disadvantages compared to CBMnet. Since every E-link of the GBTX consists of 6 pins, this results in 240 pins for all 40 front-end links, while the CBMnet HUB ASIC provides five front-end links with 1 Gb/s each, resulting in 40 pins in total. An analysis and evaluation of the GBT protocol already has been made in [44] and [86], concluding that within the CBM DAQ more efficient front-end data transmission is needed, significantly more bandwidth, as well as different synchronization mechanisms, while less radiation robustness is required. Especially due to limited space inside the super-conducting dipole magnet, a very dense interconnect solution and efficient data aggregation is mandatory.

3.3 CBMnet State of the Art

The first shot of the CBMnet [45] was the development of the network protocol, used on three different FPGA devices: The Readout Controller Board (ROC), the Data Combiner Board (DCB) and the Active Buffer Board (ABB). The ROC is connecting several Front-End Boards (FEB) and collects the generated data from the detector ASICs. In the ROC, the data is packed into CBMnet packets and subsequently sent to an arbitrary number of hierarchical levels of DCBs over optical links. One DCB can connect up to five ROCs and aggregates the data packets onto one optical link to the ABB, which represents the interface between the CBMnet network devices and the computer cluster farm. On the physical layer, all interconnects were implemented using the Xilinx Multi Gigabit Transceivers (MGT) for serial data transmission. To achieve electrical DC balance and a guaranteed number of state changes within the serial signal for reliable clock recovery, 8B/10B encoding is used. The network protocol combines three different traffic classes, which have their own virtual channel, on one link. Data Transport Messages (DTM) for recorded data from the read-out ASICs, Detector Control Messages (DCM) to access slow control interfaces, local configuration or the registerfile of a particular device, and Deterministic Latency Messages

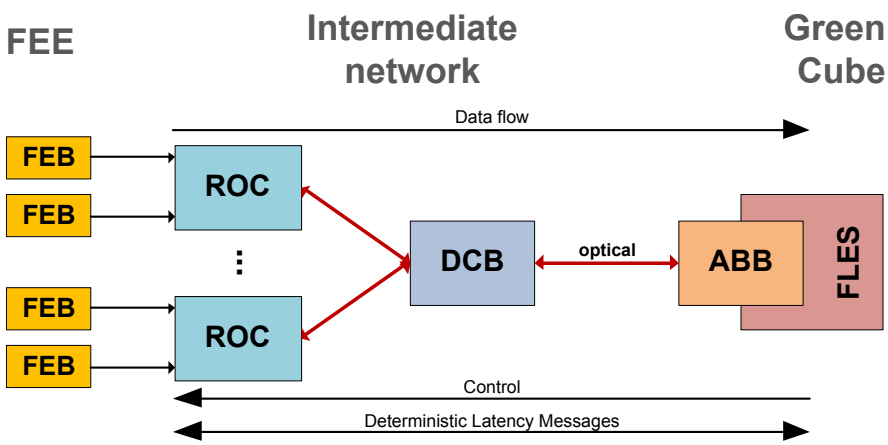


Figure 3.6: The CBM network structure with the three message streams: data, control and synchronization as depicted in [42].

(DLM), which are short 16 bit messages, having a deterministic latency through the network from the root to all leafs. On the link DTMs and DCMs are secured by a 16 bit CRC, and a link-level retransmission mechanism provides reliability in case that bit errors occur.

Summing up, the former state of the art was a network protocol with the following features [44]:

- Communication over one optical link supporting
  Data Transport Messages (DTM),
  Detector Control Messages (DCM) and
  Deterministic Latency Messages (DLM)
- Optimized data utilization about 91 %
  (about 73 % considering 8B/10B encoding)
- Retransmission for control and data messages
- Clock recovery and synchronization
- Deterministic link latency
- Implementation on three different FPGA devices

At that point in time the writers work did start, an enlargement of the CBMnet protocol to the front-end ASICs for the read-out chains of the STS and TRD

detectors was planned to improve the early stage data aggregation and timing distribution. With the integration of the CBMnet in the then submitted SPADIC ASIC [14] and the STSXYTER ASIC [36], many new challenges came into existence, e. g. the development of full custom physical layer components, like I/O cells, high-speed serial interfaces, clocking and timing units tailored to the particular needs of the CBM DAQ. As no FPGA hard blocks were available, new concepts how to synchronize ASICs with limited I/O capabilities within the read-out chain needed to be elaborated. As a long term goal, the development of the CBM HUB ASIC had been projected, which is a special developed network ASIC for combining the data of multiple types of front-end electronic detector ASICs from LVDS to optical links with different, flexible routing structures and delivering all the synchronization and timing capabilities for CBM, which are currently implemented in the FPGA devices. In this context, the development of a full-custom multi-gigabit Serializer/Deserializer (SerDes) in cooperation with the Indian Institute of Technology Kharagpur, India, was planned. Besides, the former implementation of the CBMnet has had some disadvantages, which will be examined in the following section.

3.4 Disadvantages

The first design implementations of the CBMnet protocol were only done as a development for FPGA use. They were used in demonstrators, laboratory setups, beam time simulations, and after all, they were operated far away from the beam and therefore radiation impact was unlikely and the expected soft error rates were very low. Until then, no serious examination has been made regarding error handling and radiation impact. Besides, reliability considerations always needs to be made with respect to the data transferred, and also congestion handling and data rejection in case of overload need to be elaborated. As the CBMnet protocol was gradually used in more devices, the distribution of firmware and debugging of designs became increasingly difficult and more time consuming.

3.4.1 Radiation Impact

With the two front-end ASICs and the ROC3 as prototype for the HUB ASIC, the first devices using CBMnet have been developed, which are located in an

environment very close to the beam. Thus, the demands on the network protocol increased and a more extensive elaboration of the impact of radiation on the designs had become necessary. In simulations and test, especially the retransmission feature led to problems, as it comes with a huge amount of complexity. In case of induced soft errors the runtime until hang-up was heavily decreased, because only on-link error handling had been implemented already, but also in the on-chip logic soft errors can happen. Further, the appended checksum on data and control messages is added very tardy between the link layer and the physical layer, removed by the subsequent receiver again, and thus only applied for the serial transmission. During on-device processing, combining, storing and buffering of messages, no protection is enabled and thereby corruption might not have been noticed at all.

Although it might be possible to reset the logic every time it gots stuck in a single device test, in beam time setups with more than thousand devices, the main part of the DAQ would be disabled all the time and recorded data would be lost. Therefore, the retransmission has to be designed in a way, that either all logic parts are kept in a valid state or the logic can reinitialize itself in case of an error. For example, if a message is damaged before written to the retransmission buffer, the whole channel gets blocked due to an infinite loop of retransmissions.

Anyway, it is an important design approach to minimize the amount of logic in areas with high expected SER, as redundancy always increases costs (see 2.4). Especially, if one part of the DAQ is less exposed to radiation impact, the main complexity and error handling should be located there. And with a complex packet indexing and storing method, the exchange of control and credit information at which point the retransmission should start, over also error-prone links, the approach of total reliability is hard to reach. Especially the use should be considered, if total reliability is not needed at all, as will be shown below.

3.4.2 Reliability

The CBMnet implementations claimed to be nearly 100 % reliable, offering bit error detection for the serial transmission and a retransmission feature for data and control [43] [47]. But as mentioned earlier, total reliability in systems with frequently occurring bit errors in the on-chip logic is hard to reach and actually, the CBM DAQ does not require this reliability at all. Now, a closer look on the traffic within the DAQ is required. First, the detectors can not provide 100 %

efficiency anyway, due to dead times after a hit or insensitive area. Hence, also the read-out logic and network do not need to operate fully reliable and very little loss is acceptable for the data path within limits [67]. Regarding the data path message content, it can be distinguished between recorded hit data or context information. Context messages contain a timing value, like epoch markers, and if a context message is lost, it will lead to wrong interpretation of subsequent hit messages, as events can not be assigned within the time structure correctly. Earlier, with every new value of an epoch counter, an epoch marker message has been sent, the loss of which can severely disturb the data evaluation. Since modifications have been made in the defined message format, the timing information is added to the header of every CBMnet packet. Therefore, timing information is redundant and does not have to be secured separately. But obviously, damaged data messages need to be detected and flagged. Besides, the CBMnet also provides control access and timing distribution, and especially the accurate service of the latter is system critical, these traffic classes need to be secured.

In a free-running DAQ, data volume is event driven and in case of congestion over a longer period, e. g. due to a lock in the retransmission control logic, data have to be dropped anyway. Thus, regarding the reliability, it is much more important, that a continuous data flow is guaranteed.

3.4.3 Debugging

In the final DAQ all devices will run nearly autonomously, as a particular maintenance of several thousand devices is not possible. But in case of lost messages, control timeouts or broken links, one can only restart whole devices and reinitialize the chain manually. Misaligned synchronization or data congestion will be completely ignored. In case of bad wiring or other sources of interference, resulting in a high SER, statistics about interconnection quality could be extremely helpful to improve the setup. To gather this information, some independently running modules are necessary and a common interface to access and evaluate the measured values. Moreover, compared to FPGAs, where malfunction can relatively easy be determined by an integrated logic analyzer, like Xilinx ChipScope, custom-made ASICs do not offer debug capabilities for internal circuits out of the box.

3.4.4 Existing Code Base

In the beginning of the development it could not have been estimated completely how the final needs would be. For example, CBMnet was originally conceived as a network, which allows traffic between the end nodes of the front-end electronics (FEE). With this feature, an end node might trigger a readout in another node to get data from channels where the peak amplitude is below normal readout threshold. Meanwhile this problem has been solved differently from physics side but existing implementation still has some capabilities of a full-blown network protocol, which consumes resources unnecessarily. In addition, some developments just were done for prototype reasons and due to time and human resource limitations, a lot of functionality still had to be implemented. Within the protocol revision, some first adaptions of the CBMnet to the new needs did not lead to success, as the internal hierarchy and module structure was not capable for a parametrized implementation, which could be run in all CBMnet devices with different configurations.

3.4.5 Resource Limitation

In the new front-end ASICs, unbalanced links have been added to reduce the fan-out of pins and increase the data throughput in back end direction. Especially the retransmission feature needs buffer memory for every lane and lots of control logic. Thus, a four lane LVDS link core occupies around 15 % of the FPGA resources of the largest Xilinx Spartan-6 device. As at least four FEBs should be connected, the resulting capabilities are not sufficient for any redundancy implementation.

3.4.6 Code Development

Through experiment enhancements and modifications, and also experience from live application, the requests from the detector working groups on the implementations, protocol and plug-ins, did constantly change. Because of the brought-up code base limitations, this led to an iterative development process, which tried to offer new features, but in the end, it lacked of an overall specification for the final needs. On the way to the final setup, several different prototype boards were used for the read-out chain development and test, but not all devices could be upgraded at

once due to limited availability of new hardware platforms and secondly, working solutions were still important and used for beam times. Hence, it was common practice continuing the use of older devices, which sadly caused compatibility problems and support outlay. Within a DAQ design, many components have to work together, sharing the resources, and need unified interfaces to communicate properly. In the beginning, code bases were located locally and only partially shared, which complicated the distribution of updates. As the read-out chain has different setups for the various detectors, an FPGA device might need several firmwares, which all require their own individual configuration in terms of number of interconnects, bandwidth, system speed, clock recovery and synchronization. For some devices also additional plugins are needed, which are connected to the CBMnet logic blocks. For FPGA devices the generation of firmware depends on the latest code, but also configuration, constraints and project files. With the release of a new version of the Xilinx ISE and Vivado tools the cores had to be updated or at least be verified again. If errors did occur in service of the designs, it was complicated to distinguish if it is a bug, a tool problem or misuse by application. Thus, a common repository and built system would also be beneficial.

3.5 Conclusion

Besides the very universal timing protocols and widely used solutions, which are inexpensive and well-standardized but offer no hardware and physical layer implementations, some serious solutions are used in similar experiments. While having a closer look on the details, unfortunately no solution could fulfill the demands of the CBM DAQ. Although some developments showed interesting progress, like the TRBNet, either speed, area, reliability or synchronization constraints limited their application. Even the very promising GBTx resigned as read-out ASIC for the STS detector, as the space in the dipole magnet is very limited and a denser interconnect solution is necessary. The DAQ network needs to run on several different devices and needs therefore custom physical layer implementations, which have to provide a unified interface for the link layer for accurate synchronization and deterministic timing behavior. With the current CBMnet implementation, a reliable network protocol for data acquisition under normal conditions has been developed for single point-to-point FPGA interconnects. But through extensions and modifications of the DAQ system, many new challenges

has to be met, particularly regarding more complex setups with the new ASIC devices. Hence, it became necessary to rework the CBMnet cores to the current needs of the data acquisition. The development of the improved network protocol and its implementation on several designs is described in the next chapter.

Chapter 4

The CBMnet Protocol Upgrade

With the extension of the former CBM DAQ setup new tasks and requirements arose. As already discussed in chapter 3, the CBMnet implementations needed to be revised, extended, and the integration on several new designs had to be done. In this chapter the network extension is described and a detailed elaboration is given about the new implementation of the upgraded network protocol version 3.0. The focus was on developing a library of modules for all implementation layers, which needs less resources and delivers more reliability. Therefore, the whole logic of the generic IP cores has been rewritten from scratch to meet the structural and functional demands for a radiation tolerant implementation. Also additional plug-ins, providing the generic CBMnet link port interface, have been developed to fulfill the demands of a particular design. Finally, the new implementations are compared to the former CBMnet modules.

4.1 CBM Network Extension

The topology of the CBM experiment detector setup, with a particularly large number of multiple different sub-detectors on the front-end side and a high performance computer farm as data sink on the back-end side, led to the selection of a hierarchical arranged network, structured as a tree [11]. The communication between the network devices works using copper or optical point-to-point interconnects as they deliver the fastest possible connection and are most suitable for synchronization scenarios. As the devices may work in various locations with different potentials, all connections need to be AC-coupled, and therefore serial

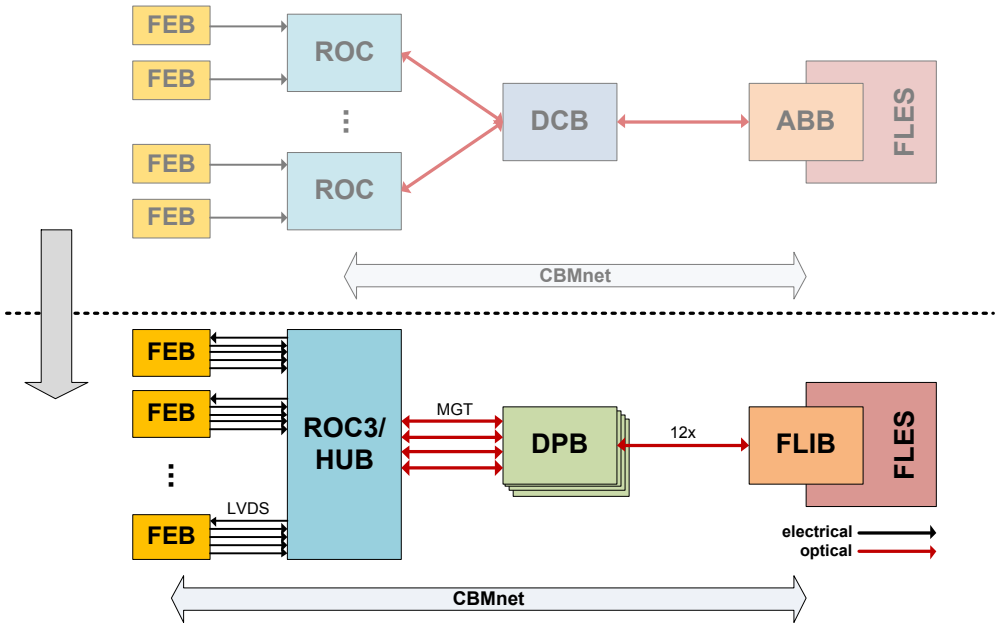


Figure 4.1: The planned CBM network structure of an example read-out chain compared to the old setup. CBMnet is directly integrated in the FEB ASICs providing data rates up to 2 Gbit/s per chip. The ROC3/HUB ASIC enhances the early stage data aggregation.

links must ensure DC balancing. The data flow is unidirectional from the front-end electronics through several data processing and combiner boards to a high performance computer cluster, the First Level Event Selector (FLES), while slow control and synchronization are distributed in opposite direction mainly. In the read-out chain many different types of devices are present and the final build-up depends on the detector type and the beam time scenario. Furthermore, different designs use the same hardware, like the ROC3 design, as a prototype for the HUB ASIC, and the DPB, both use the custom-made SysCore3 FPGA development platform [28].

Usually, the detectors are connected directly to an ADC ASIC on the front-end boards (FEB), which manages the read-out and digitization of the electronic signals, like the SPADIC or STSXYTER ASIC. The specialty of these ASICs is, that they support the free-running DAQ design with a self-triggering hit detection logic and therefore the network traffic is not controlled by an external trigger but

rather depends on the decision logic, the calibration of the analog circuits and the physical event. Thus, the read-out ASICs have to take decisions individually, if the recorded data should be saved and processed. In case that the network protocol is already integrated, like in the two ASICs, the signal information is directly packed in CBMnet messages and sent over the network links. Since the ASICs are manufactured in a 180 nm technology process, the speed of the digital LVDS I/O cells is limited to around 400 MHz, to ensure a successful transmission with copper wires, like common HDMI cables, over several meters in spite of signal attenuation. Through speed constraints from the Spartan-6 FPGA, regarding the serial I/O interface, the frequency has been set to 250 MHz or 500 Mbit/s using Double Data Rate (DDR). To increase the bandwidth for the data path anyway, the ASICs have unbalanced links with also two or four lanes in back-end direction. Alternatively, an additional Read-Out Controller (ROC) board is necessary, which delivers a large number of Time to Digital (TDC) input channels on one side and interfaces the network directly with optical links on the other side. They are currently equipped with small form-factor plugs (SFP). A data processing stage, also using the CBMnet, is the Data Processing Board (DPB), firstly located in the DAQ for the TRD detector. It provides feature extraction algorithms to select and extract relevant event data, background suppression and physical relevance checks to limit the outgoing bandwidth of all detectors to 1 TB/s [24].

The tree structure of the network is composed by using hubs as data concentrators. Either the FPGA device, the Data Combiner Board (DCB) is used, which can combine up to five optical links, or the HUB ASIC, delivering different configurations of high-speed LVDS interconnects for connecting multiple front-end electronic detector ASICs on the front-end side and Multi Gigabit (MGT) links on the back-end side with a link speed-up of a factor of up to ten. The last stage in the read-out tree is the FLES Interface Board (FLIB) which is upgrading the earlier used Active Buffer Board (ABB) and delivers up to 12 optical CBMnet links to interface with the Green Cube over PCIe Gen2. The incoming CBMnet messages are preprocessed in hardware and packed into microslice containers for subsequent software processing [23].

The very high reaction rate of up to 10 MHz in the experiment sets hard requirements for nearly all components. Especially the STS detectors have demanding requirements on the read-out system. They generate data rates of 2 Gbit/s per chip and several thousands of front-end chips are mounted inside the dipole magnet, which results in the need of very dense interconnect solutions. As some devices are

also located in close proximity to the targets, they are exposed to heavy ionizing radiation and therefore special radiation hardening design techniques are necessary. In a nutshell, the network protocol has to comply with the following challenges:

- Different network traffic
- Various types of hardware
- Limited space for hardware
- Autonomy of nearly all components
- Potential separation
- Radiation tolerance

In the following section, a brief description of the CBMnet protocol is given and subsequent, a detailed overview of the new features of the CBMnet version 3.0 is presented.

4.2 CBMnet Version 3.0

The new CBMnet protocol implementation is a complete Data Acquisition (DAQ) network solution, usable within all stages of the read-out chain, providing the required data throughput, control and timing capabilities for the CBM experiment on unified point-to-point interconnects, supporting synchronization over source-, and self-synchronous communication. It was designed to provide a basic set of functionality, which is used in every device and a set of optional CBMnet or external plug-ins, which can be added in certain network devices to cover further tasks, like on-device auxiliary modules, USB interfaces or local register files. The IP cores are highly configurable in terms of speed, bandwidth, number of interconnects and hierarchy, which makes the protocol perfectly suitable for several FPGA and ASIC devices.

CBMnet simply works on a tree topology and has to aggregate data from the leaves and shove them to the root. In the other direction clock and synchronization are distributed from the source to all end points. Therefore a CBMnet link has always at least one master end and one slave end. The master controls the link initialization and initializes the node at the slave end after the link is established.

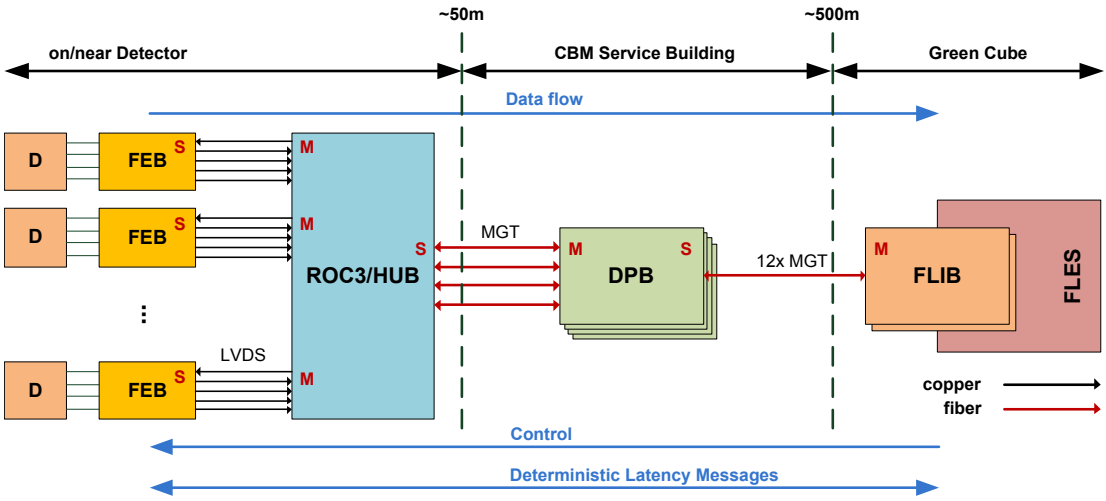


Figure 4.2: Extended CBM DAQ system with master/slave link indication.

The structure is depicted in fig. 4.2. The initialization procedure and service run autonomously, to guarantee as less maintenance as possible for larger beam time setups. With a special developed diagnostics interface, common statistical information can be gathered and in case of punctual accumulation of errors, a selective control of individual devices is always possible.

As already mentioned in section 3.4, CBMnet has had several disadvantages in detail, which were well needed to be improved. Superficially the implementation of the version 3.0 provides similar features, and main settings, like the packet format, have been preserved, because they are either sufficient or they are needed for compatibility reasons. The user interface stayed nearly untouched, as the subdivision in three traffic classes is still reasonable. The 8b/10b coding of service characters, data, control and synchronization messages has been retained, as still DC balancing is required, because of AC-coupled links, and frequent signal transitions are needed for clock data recovery (CDR) circuits in serial receivers. Besides, the sophisticated assignment of special characters assures error detection in packet framing.

Having said that, nearly all parts have been improved or adapted to the current needs. The whole logic of the physical layers, link layer and network layer implementations has been rewritten from scratch to get as simple and lightweight code as possible to reduce costs. All current demands have been taken into account. The new-organized module structure and radiation tolerant implementation improved

the use in several designs. As earlier mentioned already, even for same hardware devices, a lot of different firmwares are needed by the various detector groups. The designs just slightly differ in configuration and a lot of logic is identical. The new CBMnet implementation greatly simplifies the use of the cores in several designs, without further adaption of hardware modules. To share modifications, new features and bug-fixes as soon as possible, an automated build system has been set up which generates the firmware of all CBMnet designs. It requires the clear structure, unique parameters and well defined generic modules to work properly. Furthermore, with the version 3.0 there have been made a lot of adaptions to make the protocol fulfill the needs by achieving higher utilization.

Regarding preserved settings of the former CBMnet protocol and the new implementations, details are given below.

4.2.1 Traffic Classes

The four services the CBMnet protocol has to deliver are high throughput data transmission, slow control access for devices and detector, system-wide clock distribution and precise time synchronization, ensuring a deterministic behavior also over link reinitialization. The requirements of these services are somehow heterogeneous, as e. g. synchronization messages need almost no bandwidth but a deterministic broadcast distribution, while slow control messages do not require both at all. Therefore, the services also have different demands on processing priority. As a fixed framing format or content decoding can not deliver the required utilization and flexibility, a virtual channel solution has been selected, providing three traffic classes for data, control and synchronization. The clock distribution is either guaranteed by CDR circuits or an additional clock signal line [44]. The three traffic classes are

- Data Transport Messages (DTM)
  They are used to deliver hit-data and epoch-markers from the detector readout digital logic to the backend. As the other traffic types are mainly sent during the initialization and rather rarely for control purposes, data messages benefit from the high bandwidth of the link and achieve a utilization up to 100 %.

- Detector Control Messages (DCM)
  Slow control messages are issued from the backend or an intermediate device in front-end direction and are used to access a specific subsequent device to modify or readout its register file, or to access external plug-ins which are interfacing with local detector electronics. For example, when the read-out chain is set-up, every device gets its unique identifier in the network address space via broadcast slow control messages.

- Deterministic Latency Messages (DLM)
  The third traffic class provides single-word-width messages, used for synchronization. These DLMs are like special control characters because they are the smallest possible processing unit in the network and therefore processed within one clock cycle internally. As there are only 16 different types of DLMs defined, but mapped to 16 bit, even protection against MBUs is assured. This is absolutely essential, as DLMs are responsible for a consistent timing framework and an error due to lost or corrupted messages might not be detected anyhow differently. DLMs trigger timing-related functions in the network, such as epoch-marker synchronization in the read-out ASICs. As their name implies, their transit times through the tree from the root to the leaves have to have deterministic latency. Reproducible round-trip times are ensured by the Priority Request Insertion [22] method, which can insert DLMs at any required time, also if the media access is already active. The part of DCM or DTM processed at that moment will be delayed by one clock cycle, but merged later at the receiver side again. DLMs set hard requirements on the physical layer, as all word alignment and delay fine tuning takes place there.

Regarding the chronological sequence, DLMs and DCMs are mainly sent during the initialization of the read-out tree. Afterwards, DLMs are only sent periodically to check for consistent timing settings, and control messages only sent on user demand. The main traffic is created by DTMs due to the self-triggered front-end electronics.

4.2.2 Communication Layers

The CBMnet IP is implemented for maximum flexibility and reusability of components. Thus a well-defined categorization in layers, similar to the definition of the

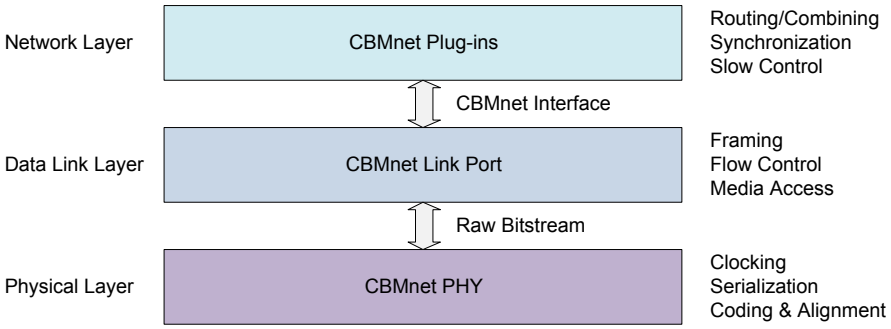


Figure 4.3: The CBMnet layers and their tasks similar to the Open Systems Interconnection model (OSI).

Open System Interconnect (OSI) reference model, has been made [100].

The physical layer defines the transmission and reception of raw bit streams over serial full duplex links and interfaces with the above link layer. It is responsible for the line coding of all traffic, with special characters for message framing and error detection, clock distribution and recovery as well as word and bit alignment, to ensure correct sampling and timing adjustments. The physical layer implementation is very important, as within the final read-out tree, devices can only be accessed over network links from the back-end (besides of additional debug interfaces for single devices). Therefore, intelligent initialization procedures in the lower layers are necessary, so that a reliable connection is set up autonomously.

The data link layer consists of the link port, which defines the frames of which messages are sent between CBMnet devices, handles the media access to the physical layer, and is therefore the core module, which is present in every CBMnet device. Further tasks are the flow control, stream merging, buffering and credit management. It provides the generic interface (see 4.2.4), which allows the flexible cascading of plug-ins. As no flow control is implemented in the physical layer, the link port sends *IDLE* characters, in case no messages are delivered. Regarding the flow control, the particular receiver is monitoring the allocation of the data and slow control input buffer space. If a traffic holdup occurs in subsequent stages and the receiver can not consume further packets, a congestion control stops the transmission of new messages at the transmitter side by sending WAIT packets periodically. As up to four parallel lanes in one direction are supported, four different WAIT characters exist. In case of absent WAIT characters over three periods, the transmitter continues sending of messages. This ensures, that also in

case of corrupted control characters, no messages will be sent, overwhelming the receiver. Moreover, it has been resigned on a complex credit based flow control or handshake, like in the former CBMnet protocol, to eliminate the risk of a complete lock, due to wrong interpretation.

The network layer is not directly integrated in the core protocol, as not all stages in the read-out tree need the same capabilities. Also, the CBM DAQ system is not arranged like a full blown network, where every single unit needs to communicate with any other device, but as a tree. Further, demands depend on network traffic, like no routing for data packets is necessary. Therefore, addressing, routing and combining of messages are realized by additional plug-ins, providing the CBMnet interface.

4.2.3 Framing and Packet Format

The smallest data unit within the protocol is a PHysical transfer digIT (PHIT), consisting of 16 bit data plus 2 control bits, to indicate the type of the PHIT, like payload, framing or service characters. While DLMs have the fixed length of one PHIT, and are the smallest possible processing unit in CBMnet, the DTMs and DCMs packet format length is flexible for two reasons. First, as CBMnet needs to support various sub-detectors, and all of them have their own container format which stores the recorded data. If the data from several channels are combined, a message can contain more or less information, depending on the number of channels read out. Second, on top of the link layer runs the FEET optics control protocol, using the CBMnet slow control channel. It provides PUT/GET commands, based on so called *multi oper lists* and a software stack, called *rocutil* to issue the commands from a computer [87].

Adjacent to DLMs, which are already protected against bit errors, a 16 bit checksum is appended to DCMs and DTMs to detect corrupted messages. Both types can carry payload from 8 to 64 Byte and are framed by start and end delimiters as depicted in fig 4.4. A full description of framing and control characters, and their 8b/10b representation, can be found in the repository in the *cbm_lp_defines.h* file in the includes directory. This Start Of Packet (SOP) or Start Of Slow Control (SOSC) header defines the packet type and is coded with special 8b/10b K characters to be safely detected in the bit stream. The same applies for the termination word of a packet, called End of Packet (EOP, respectively EOP_C).

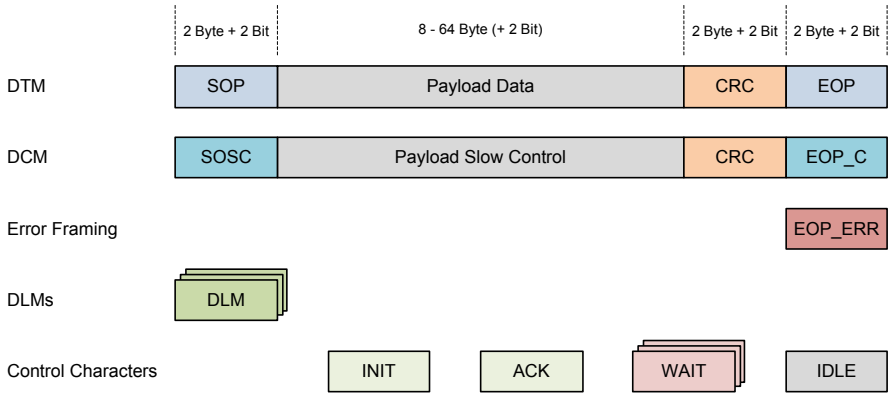


Figure 4.4: The CBMnet packet framing and control characters. The end delimiter of corrupted messages will be replaced.

In case a mismatch between payload and checksum is detected, the termination word will be changed to EOP_ERR to identify corrupted packets.

As DTMs are always created at the front-end side and sent to a processing and storing stage at the back-end side, the data are not modified in between. To ensure, that corrupted messages can always be detected, the appended CRC is not removed until the message finally arrives. Similar to the DTMs, DCMs are secured until they arrive at the destined device.

4.2.4 Interfaces

There are three important interfaces regarding the CBMnet implementation.

The physical layer is connected with the link port over two 18 bit interfaces. One for outgoing, and one for incoming data. A *PHY_ready* wire signals the successful initialization of the SerDes and coding layer. There is no further interface synchronization necessary, as the link port handles all the media access, and both layers usually run with the same clock. Only between the receiver in back-end direction and the link port, a synchronization stage is added, as this is the point where the synchronization loop gets closed.

The link port, as well as CBMnet components and plug-ins provide a generic interface, as depicted in fig. 4.5, to ensure a flexible use in all devices for various applications, which is similar to the former CBMnet. It delivers synchronous inputs

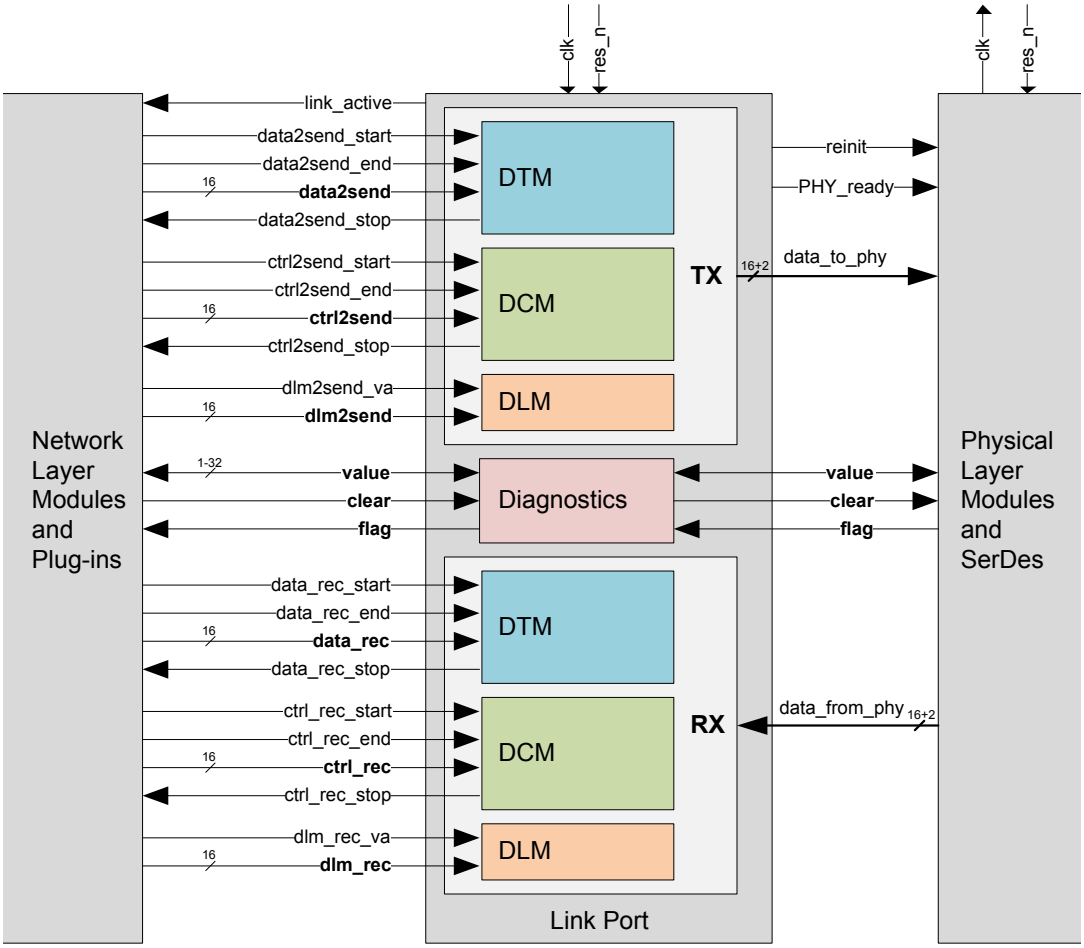


Figure 4.5: The CBMnet link port interface between the network layer modules and plug-ins, and the physical layer. Each traffic class has their own start/stop interface for sending and receiving DTMs, DCMs and DLMs.

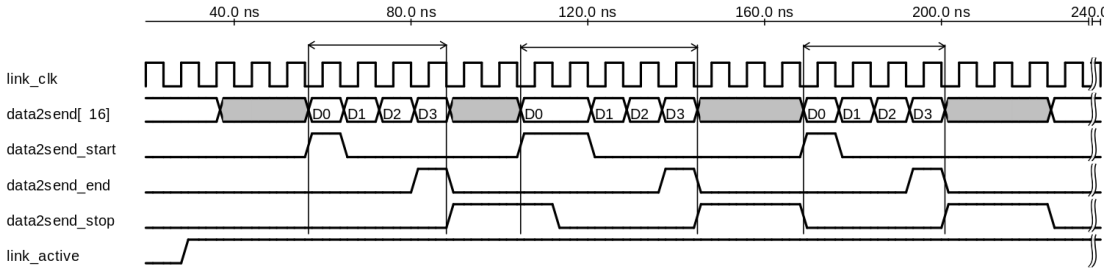


Figure 4.6: Three examples for sending a data packet with start/stop signalling over the CBMnet link port interface. Procedure for sending slow control packets is analog.

and outputs for the three traffic classes data, slow control and synchronization. If the initialization procedure has succeeded and deterministic latency is ensured, the *link_active* signal is assigned. The flow control for DTMs and DCMs is managed by common valid and stop signaling. A start and end signal indicate the beginning and tail of every message and these signals are active exactly for one clock cycle. In case the logic wants to assign a start while a stop signal is asserted, the start signal stays active until the stop is released and the first word of data is accepted. An example how data is send over the CBMnet link port interface is depicted in fig. 4.6. The interface width for data and slow control payload is always 16 bit and a message must contain of 8 to 64 byte. As DLMs are timing critical and need to have deterministic latency, an insertion is possible at all times through a valid signal, unless the link is inactive. DLMs are control characters, which are sent within one clock cycle, thus the valid signal is assigned for just one pulse.

The third interface is the CBMnet diagnostics interface, which will be described in 4.2.9.

4.2.5 Reliability Decisions

As already mentioned, the devices of the CBM DAQ system are exposed to heavy ionizing radiation (see 2.3.4). The wish for total error-free hardware, partially located in an environment very close to the collision zone seems practically utopian. But reliability has to be ensured to a certain point, which leads to the discussion which faults are tolerable and which faults are critical. Not in every case it is necessary to deliver a radiation hardened system, but rather a radiation tolerant

system like in [26]. In chapter 2 a particular declaration of fault types has been made already and the impact of radiation has been discussed. In chapter 3 a detailed overview about the disadvantages of the former CBMnet protocol implementation has been given and an explanation, that total reliability is not required at all, as the detectors have finite efficiency anyway. This led to the following design decisions with respect to the new CBMnet implementation.

Regarding the data path, reliability is not necessary until the data loss stays reasonably small. The highest SER is expected for the STS detector, with an SEU every 428 s for a four-lane CBMnet LVDS core. Within this period, the raw bit throughput is 107 GByte for the whole interface. A rough calculation of data loss is given in table 4.1. The real payload is only a fraction of a CBMnet packet, and the relative value depends on the packet length and the line coding. It is assumed, that one SEU does not damage more than one packet. Thus, the data loss has to be considered in relation to the absolute payload transferred within the mean time between failures. Calculation shows, even in case the link is fully utilized, the resulting data loss is very negligible. Obviously, the read-out chain consists of more devices than one CBMnet core. Also other components and plug-ins are operated. But even multiplied by a factor of hundred or thousand, data loss does still not matter at all, compared to the efficiency of the detectors. Hence, there is no need for redundancy to be implemented in the protocol. Only a checksum is required to detect corrupted messages. Regarding slow control messages, their reliability is ensured with timing redundancy on a higher layer. If inequality between the payload and the checksum is detected, the slow control operation is not being executed and a *NACK* is sent back to the origin. In the same manner, a unsuccessful CRC indicates, that the value read back from a device may be wrong. Hence, the PUT/GET command has to be performed again. As slow control messages appear very infrequently, chances are small, that a message is corrupted at all.

The reliability of the control path in a module is far more important than an error-free data path. SEUs in the control path lead to wrong behavior of control logic, like FSMs, FIFOs and thereby higher network layer functions, like routing, addressing and traffic control. Therefore, soft errors in the control path have to be prevented or corrected within a very short time frame, so that only very little data loss occurs. Furthermore, errors in the control path need to be detected immediately, as otherwise the whole network is locked and at worst, data have to be dropped or can not be recorded anymore, due to congestion. The reasonable

	Byte	Content	Payload normalized	8b/10b efficiency	Payload effective	Payload GByte	Data loss
Short Packet	8	Payload	57.1 %	80 %	45.7 %	48.9	0.0000000163 %
	6	Framing					
Long Packet	64	Payload	91.4 %	80%	73.1 %	78.3	0.0000000817 %
	6	Framing					
Buffer purge	512	Payload	100 %	-	100 %	48.9	0.000001047 %
Link lock	-	-	-	-	-	-	0.00117 %

Table 4.1: Expected calculated data loss of detector data due to an SEU every 428 seconds in the data path or control patch of a four-lane LVDS CBMnet core running at 2GBit/s. Additionally, the data drop during a link loss is estimated.

buffer size of the implemented SRAMs in one link port lane is at least 256 x 16 bit, because all FPGAs provide enough resources of these BRAM sizes. Also in ASICs this is a reasonable SRAM size and ensures flexible storage of CBMnet packets in case of stopped data flow. Assuming a long message effective payload like in table 4.1 and a link speed of 2.5 Gbit/s, buffers will be occupied after 1.12 μs. Thus, the control path needs to be corrected latest by the end of this period to avoid dropping of messages. Obviously, a buffer with 256 entries depth can consume the payload of eight long CBMnet packets. Thus, in case of an error in the control path, which leads to a purge of the whole buffer, the data loss compared to an error in the data path is actually increased by a factor of up to around ten, but still does not come into effect. The factor slightly differs, as the buffer logic stores framing characters parallel to the data. Thus, no memory storage is completely utilized by payload.

Finally, most severe errors are configuration faults, as they can lead to wrong synchronization settings, alteration of electrical calibration or shutdown of a device. Unfortunately the detection of wrong timing settings is very complicated and if reliability is not ensured, a corrupted time-stamp counter would not be noticed at all, resulting in wrong event interpretation. The configuration settings of the high-speed serial transceivers are actually the most sensitive part within the CBMnet protocol. Wrong tuning or calibration, as well as alignment, can disturb the transceiver's operation in a way, that a whole link reinitialization is

necessary. While the initialization of the serial transceiver is done within 1 μs, the initialization of the physical coding sublayer, word-alignment and deterministic latency adjustments can last up to 5 ms. Within this time, recorded data have to be dropped, as soon as buffers are occupied. Moreover, in case of a link breakdown, the whole sub-tree needs to be resynchronized. Therefore nearly all recorded data have to be dropped, since they can not correlated correctly. Depending on the location of the device within the read-out chain, data loss can happen to a very large extend. Therefore, configuration settings need to be protected accurately against corruption. When running the CBMnet cores in an FPGA design, scrubbing has to be used to secure the FPGA configuration.

4.2.6 SEU hardened Implementation

As already explained in chapter 2, the influence of ionizing radiation on semiconductor integrated circuits can be subdivided in Single Event Effects (SEE) and Total Ionizing Dose effects (TID). Regarding SEEs, only SEU hardening has been generally implemented in the protocol. Other physical hardening and mitigation techniques are only applied to a particular full custom core, implemented in a CBMnet design, or for ASIC devices, e. g. the Universal Core Library (UCL), developed by the Chair of Circuit Design, Heidelberg University, has been used. This is a special home-made standard cell library, providing, amongst others, the unusual feature of separated substrate contacts [13]. Regarding SEUs, the protocol implementation of the digital part has been developed using several techniques. The main goal is to put the logic in a functional state again quickly, so that data loss happens as little as possible.

In the link port, SEUs can force a state machine entering dead states. Therefore all CBMnet modules have a "clean logic" which can be triggered to continue operation from a checkpoint. An example module with the implemented hardening techniques is depicted in fig. 4.7. The state machines are designed in a way, so that all dead states or forbidden transitions will trigger the clean logic. Thereby, all outgoing and incoming transfer of the module is stalled for a moment, before regular operation is pursued. As it can happen, that messages have been read or written to a buffer at this moment, these fragments are shifted out, until the next start of a packet is detected. Within the data flow, checkpoints can be determined after every whole CBMnet packet, from which the execution of network operation

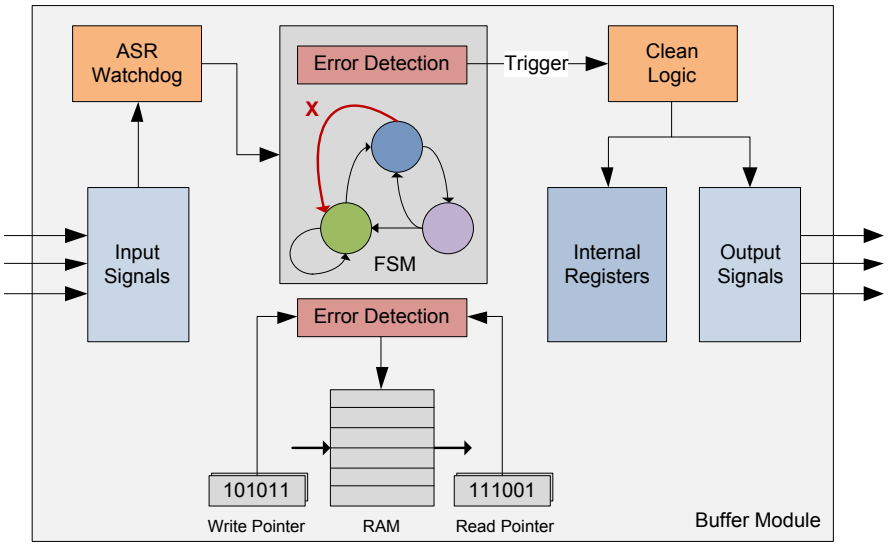


Figure 4.7: Implementation of ASR watchdog, clean logic and pointer error detection in an example CBMnet buffer module.

can be restarted after a functional error. These frequent checkpoints ensure, that not more than 64 Byte have to be dropped due to an error in the control path logic. An Automated State Recover (ASR) watchdog is enabled with every new SOP, disarmed with either the successful receipt, or armed after a 64 Byte timeout. The subsequent logic of course can handle also not assignable PHITs. They will be dropped until a framing character is detected again. Some smaller buffers are built-up with register-based FIFOs, where every stage stores the context information for the particular content. Thus, errors can only affect single PHITs. Pointers of storage elements, like RAM-based FIFOs, are implemented twice to detect errors. In the rare case of a mismatch between the pointers, the storage element has to be cleared. In case of wrong arming of clean logic or the ASR watchdog, unfortunately data will get lost, but a reliable ongoing operation is always guaranteed.

In the particular PHY, a control path logic is only active during link initialization. Meanwhile, special sequences of 8b/10b characters are exchanged over the serial interface to align incoming data. The initialization is completed with a double-check handshake. Afterwards, no state machines are used anymore, but only pipeline stages built-up with flip-flops, and data is streamed directly through the serial channel. SEUs induced here can only result in single or multiple errors in the

raw bit stream. Of course the breakdown of the link has to be avoided at all costs, as it can have the most impact on data loss. Therefore a Self Repairing TMR (SR-TMR) module is used to secure important functions, like the link initialization FSM. With this redundancy, at least the impact of SEUs will be mitigated. The built-up and principle of this SR-TMR will be described in 4.2.11.

As explained earlier, the main configuration and timing adjustments of a design are the most important settings regarding reliability. Hence they need to be secured properly to assure a reliable operation and consistent timing. Configuration is stored in the particular on-device register file (RF), which is generated by an EDA tool, developed within a thesis at the Computer Architecture Group (CAG) Heidelberg University [40]. The EDA tool reads XML files, where a user defines register, their width and access permissions, and creates the respective verilog HDL, which can directly be used in a design. As the register file has a generic read/write write interface, an RF connect module has been developed (see 4.4.3) to parse the PUT/GET commands from the FEET optics protocol. To ensure reliability, and avoid a single point of failure, the RF and all configuration logic is implemented with TMR and the sophisticated SR-TMR logic. All timing relevant logic, like epoch counters and DLM receive logic are implemented twice. Therefore a detection of wrong timing settings can be assured. As DLMs are sent periodically to reset the counters and guarantee consistency, a refresh can be done in case of a detected error. Of course this implementation leads to a strong enlargement of logic and an increased need of resources, but it is only necessary for the critical control points of a design.

4.2.7 Clock Distribution and Time Synchronization

The system-wide clock distribution in CBMnet is a main feature. It is essential to operate a synchronous DAQ-network to achieve deterministic link latency, which has also to be deterministic over a link reinitialization. But for the examination of a consistent timing framework, many factors have to be considered: Physical detector delays, analog circuit delays, digital processing delays, and link delays depending on the cable length and physical implementation. The CBMnet link port and PHY can only handle the digital processing delays in the devices and over links. Thus, for the experiments a consistent voltage and temperature environment is necessary to avoid delay drifts. To achieve deterministic behavior of the network,

all clocks are derived from one master clock, which will be used for the transmit links in the direction from the root to the leaves. For the Multi Gigabit Transceiver (MGT) links the clock is recovered in the receiver, and then divided to synchronize the local PLL for the system clock. As this clock is also used as reference clock for the transmitter in back-end direction it has to have an appropriate quality. Xilinx FPGA devices require a maximum RMS jitter of 40ps but the recovered clock jitter is usually around 90ps RMS. Therefore a COTS jitter cleaner device is added on the printed circuit board (PCB) or an additional mezzanine card, which cleans the clock and feeds it back into the FPGA. This principle has been described earlier in [42]. In case of a link hang-up, the device at the slave end uses its free-running PLL to continue operation until the recovered clock is available again. For the LVDS links of the FEE devices, the clock is sent over a particular clock wire and directly used as main clock for the ASIC logic. The parallel data synchronization to the word-clock is either done with a barrel shifter, and the latency added is respected during round trip time measurements, or a deterministic initialization procedure is used, which is described for the FLIB design in 5.2. The delay adjustment of the serial transmit stream is done with clock clips and clock inversion. An example of the initialization mechanism with the ROC3 design is described in chapter 5.3.

4.2.8 Reset

The reset and initialization of the read-out chain is a very critical function, which must allow to restore a working state of the whole tree and also parts of it and merge them into the running system. As a link or functional unit hang-up can occur in rare cases due to multi-bit errors caused by radiation, it has to be possible to reset an interconnect, a specific device or logic section. All CBMnet links in the tree are AC-coupled. No additional reset-signal is available. Several types of resets are available.

The hard-reset or power-on reset, usually issued from the board to reset the whole FPGA or ASIC with its logic, memories and interconnects. All data and configuration will be lost. It is also possible to trigger a hard-reset by sending two reset characters within a defined time frame. The first disarms the reset-protection, the second issues the hard-reset.

The soft-reset, issued by a state machine to reset a specific functional unit in the design. Configuration settings and status of subsequent units of the device are preserved. The soft-reset is issued at any time an error in the control path of a module is detected and therefore may cancel a buffer-read operation or a packet transmission. Since the data buffers of the logic may be in an undefined condition after the soft-reset, the FSM passes into a cleaning state. Remaining data fractions will be dropped until the next framing of a message is detected. Afterwards normal operation mode continues.

The link-reset which can be triggered by the physical layer itself, from the link master over the link with special reset characters or from the control logic of the design, e.g. the register file. The physical layer logic block, running with a local clock, always observes the current status and quality of the link. If one link partner detects too many errors in the coding, it is likely that the bit-stream is out of alignment. After a timeout the link init logic starts to re-initialize the link. The detection of an init character at the opposite party of the link will also issue the link reset but in case of a misalignment of the parallel data it may not be recognized. Although in this case the link logic should detect coding errors but ultimately it is also possible to reset a dead link. The master can send a unique sequence of 16 non 8b/10b characters, which will not appear in normal operation mode but go through the AC coupling anyway. Moreover, the watchdog is independent from the coding layer, so that this sequence will also be detected if the word alignment is displaced. This sequence triggers a watchdog, which is directly connected to the receiver input registers and thereby generate a local reinitialization. Especially for the more complex gigabit transceiver links of an interconnect slave, it is important to have a local clock running to maintain the logic, as missing input data and transitions will cause the clock data recovery (CDR) to lose its lock.

4.2.9 Diagnostic Interface

The final read-out chain will run with several thousand devices at the same time. Therefore it is very useful to get statistical and particular status information from the single devices about link quality, configuration and network status. Especially to get information about which part of the logic or periphery is mainly sensitive to

hang-ups or decreases the link throughput. Amongst others, with the diagnostics interface it is possible to get information about

- eye opening of an incoming data stream of a particular lane
- sample point of the data in the serial stream
- latency of an interconnect to the subsequent device
- errors in the CRC checksum or 8b/10b coding
- number of start-ups of the particular physical layer
- number of soft resets of a functional unit or a state machine
- link status
- dropped messages in case of congestion

The information is stored in a local register file and can be read out via control messages. It is also possible to modify values like the sample point for example. The generic diagnostic interface of every unit characteristic delivers three signals. The first offers a 16 bit output signal, which counts for events or errors, or saves an attribute value. The second signal is a one-bit flag output, which is enabled, if the 16 bit signal is unequal zero, hence at least one event has happened. The third signal is a clear input, which sets the 16 bit signal back to zero. With this interface it is very easy to gather information from devices in the read-out chain, where an increased number of suspicious events is happening. An additional plug-in can be connected which autonomously generates control messages with diagnostic information. In the past some problems were caused by bad copper cables which led to an increased bit error rate. But also for the normal operation mode disturbed by SEUs, it has been very useful to localize a dead lock in the system to rework the implementation.

4.2.10 Verification Tools

The CBMnet cores also provide tools for test and verification. In case of a hang-up during beam times it is very difficult to distinguish between a bug or a radiation-caused error. Hence, it is important to verify the CBMnet implementation and components for their logical correctness and tolerance against soft errors in

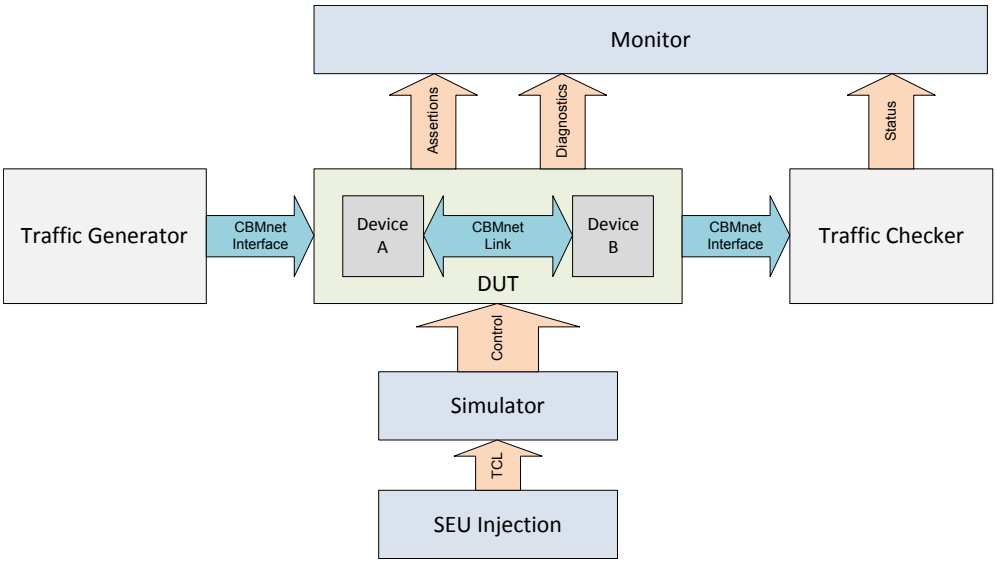


Figure 4.8: The verification scheme of the FIS tool using TCL commands to control the simulator for SEU injection.

advance. For the functional simulation, as well as for the FPGA design tests, special developed traffic generator modules allow a transaction based verification, while only arrived data will be checked. These traffic generator modules create example detector data, track the runtime of DLMs through the network and verify the payload of packets transmitted. All packets contain a consecutive number, so dropped packets can be easily identified. Besides, data flow can be controlled very detailed to reproduce different load and congestion scenarios. These modules are completely written in Verilog HDL and therefore allow the verification of a whole read-out chain either in simulation or hardware devices. For the simulation, the number of simulated devices is not limited, but runtime may increase. A first version of link testers already has been implemented for the first shot of the CBMnet [42].

As already mentioned in 2.5, fault injection tests in FPGAs are very time consuming and debug capabilities are limited. To gather information about the reliability of the CBMnet implementation, an SEU Fault Injection Simulator tool (FIS) has been developed for the Cadence Incisive simulator NC Sim. The verification scheme of the FIS tool is depicted in fig. 4.8. It contains the traffic generator modules, a monitor, written in System Verilog with interfaces to the DUT and the Traffic Checker, and a SEU Injection script. This script uses TCL commands to

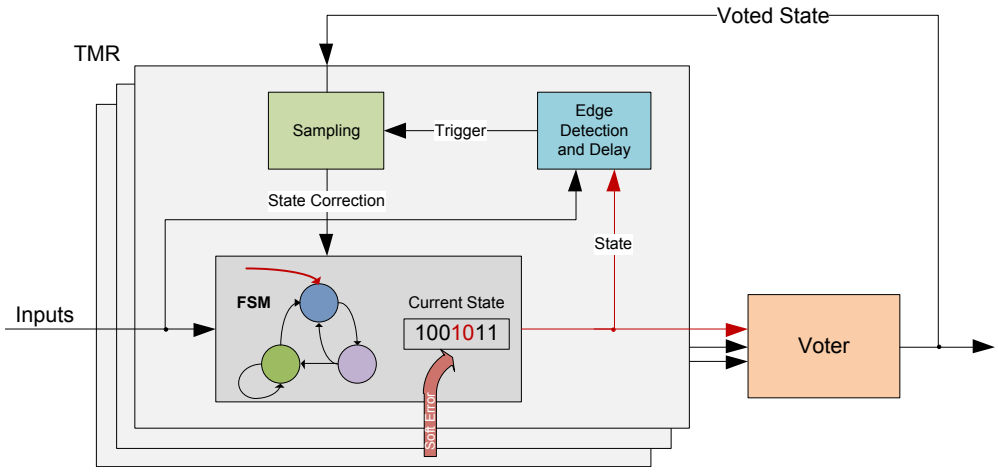


Figure 4.9: The SR-TMR method described for a finite state machine. Autonomous self repair is done with a delayed sampling of the voted state.

control the simulator and to retrieve signal and netlist information below a defined scope. As the naming of all CBMnet building blocks is clearly defined, e. g. all link port modules start with cn_lp_*, a grouping of signals can be made. Within this groups, faults can be injected with a given magnitude, emulating the occurrence of SBUs or MBUs simultaneously, and a defined or random injection period. The FIS tool also allows a assertion based verification, because for one thing diagnostics information is assessed, and on the other hand System Verilog assertions within the state machines are evaluated. More information on verification techniques can be found in [20].

Certainly, it should be considered, that the verification cannot only be done in simulation, because the fault injection tests can not represent the full spread of SEU effects. It should also be mentioned, that since 2014, also Cadence offers with the Incisive Functional Safety Simulator a fault injection tool to verify the ability of the designs to handle unexpected events [2].

4.2.11 Self Repairing TMR

The self repairing TMR method can be used to protect functional units, which are not timing critical, like slow control or configuration. As usually done when using

TMR, the logic is triplicated and the output values are merged to a single result using a majority voting. Thereby errors in one part of the logic are masked. As explained in 2.4.1, TMR needs repair before the next soft error occurs, otherwise the probability of errors is even increased. An example, how the SR-TMR works for a finite state machine, is depicted in fig. 4.9, but of course also other binary values of any length can be secured. In case the current state of the FSM is corrupted and the logic either gets stuck or the FSM resets itself, due to wrong state transitions, it will be masked by the majority voting and the voted valid state is present at the output. The signal change at the output of the damaged FSM will be detected and a delayed sampling of the voted state is triggered thereby. This sampling forces a hyper transition into the correct state of the FSM. The sampling has to be delayed for two reasons. First, in case of a valid state change, the delay has to be at least one clock cycle to avoid an infinite correction loop and the suppression of state changes at all. Second, the delay has to be at least two clock cycles to prevent overwrites of former states by the sample logic. In case of frequent state changes the edge detection will delay the trigger repeatedly also by alteration of the inputs. Therefore, it is only possible to correct the erroneous FSM while it is idle for more than two clock cycles. But as the expected SEU rate is at the order of seconds, a delayed correction of even microseconds is negligible. Autonomous self repairing works fully reliable but there are small limitations. It is always assumed, that erroneous input changes are not permanent. This means that also other configuration logic needs to be corrected anyway. Further, the input values must not depend from the same erroneous source. For example, in case a slow control action is performed due to a received message, there have to be also three independent receivers.

Unfortunately there is no out of the box solution for SR-TMR. Every implementation has to be manually adapted at least for a particular functional unit. This principle works also very well for values stored in the register file, but comes with a large overhead of additional logic. Fortunately there is not much CBMnet logic which has to be protected this way. Only the operation critical configuration is secured.

4.3 CBMnet Generic Cores Implementation

The CBMnet cores are used in several read-out designs, so they need many parameters and configuration options to serve all the use cases. The link end in back-end direction of an interconnect is always set as master, which enables the functionality to control the link and the initialization of the subsequent tree. If the core is configured as slave, and the link is active, it is fully dependent of the master's control commands. The cores can provide FPGA designs with up to 16 symmetrical transceivers, or unbalanced links with up to four lanes in back-end direction, and just one lane towards the FEEs. The link port handles the framing, buffering, arbitration of the virtual channels, error detection and flow-control. Some designs only act as a data transmitter, while others combine data to a link with higher transmission speed. All designs need capabilities for routing control messages. In front-end direction, from the root to the leaves, the tree is clocked fully synchronously with deterministic latency for DLMs. In back-end direction some synchronization stages are required because of clock phase shifts. The generic cores only enable the device resources needed for the particular operation and the resulting logic is a very slim and lightweight implementation. All internal interfaces have been equaled as much as possible to increase module reusability, and soft error handling has been implemented all over.

4.3.1 CBMnet Link Port

The CBMnet link port modules (depicted in fig. 4.10), are responsible for control flow management and packet generation of the three traffic classes, and handle the media access to the physical layer. The link port topmodule (LP_TOP) can be divided in four main parts: the link port transmitter (LP_TX), the link port receiver (LP_RX), the link port initialization (LP_INIT) and the diagnostics interface (LP_DIAGNOSTICS). The modules are built-up with very similar structure to ease modifications and code extensions. The link port control module (LP_CONTROL) manages the signal distribution between the different lane configurations. The input, respective output, of a transmit or receive lane to the higher layer is controlled with the CBMnet start/stop interface (fig. 4.5). Each traffic class has its own buffer stage to prevent inter-channel locks. Only DLMs have a fixed-length transmit path to ensure deterministic latency, and arbitration is guaranteed by the Priority Request Insertion (RPI). The LP_TX_OUT serves

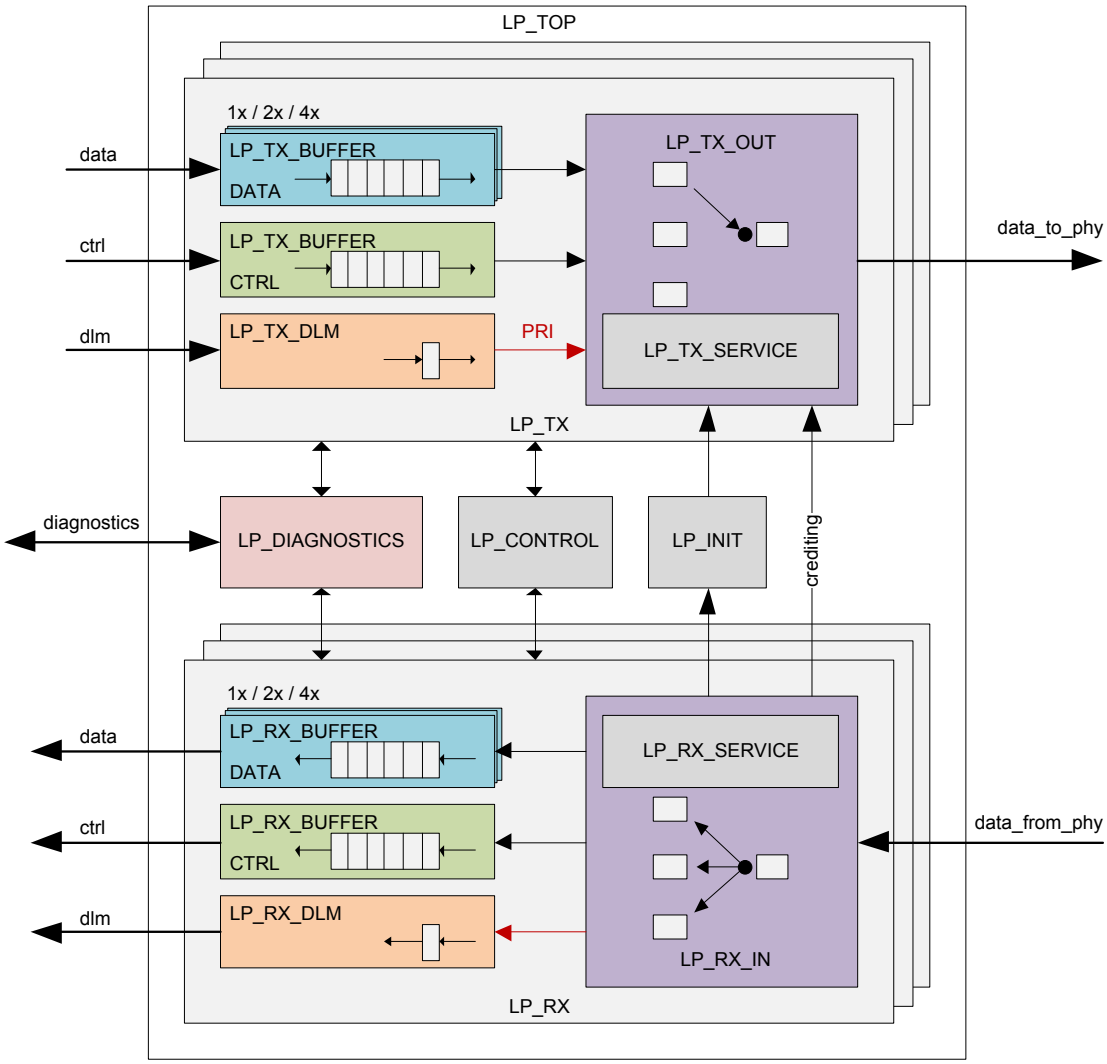


Figure 4.10: The generic link port modules provide the interface for the three traffic classes and control the media access to the physical layer and are highly configurable in terms of bandwidth. In case higher bandwidth is needed, the data channel can be replicated to either 2x or 4x configuration. The diagnostics module offers control and debug capabilities interfacing with the local configuration register file.

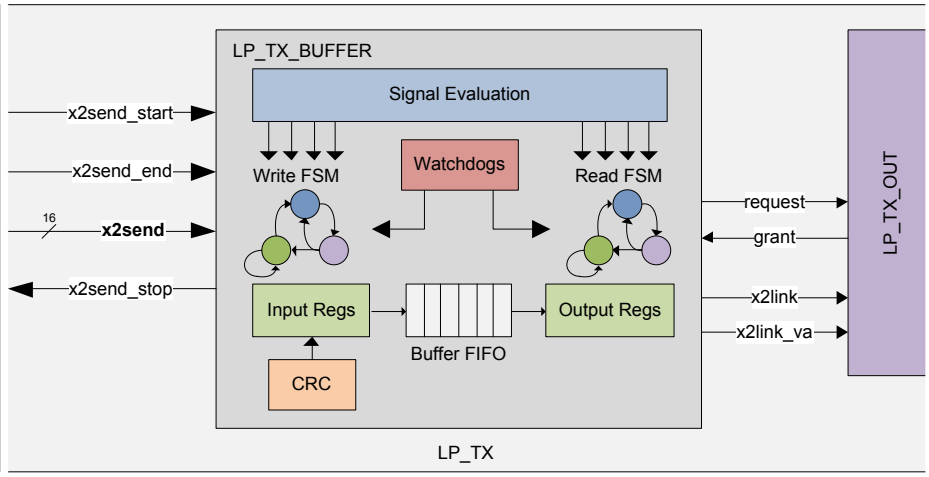


Figure 4.11: The link port transmit buffer, how it is used in the data and control path. To encapsulate units within the module, single FSMs are used for write and read.

as media access control by assigning one traffic class onto the link until the message is finished. Respectively, the LP_RX_IN manages virtual channel distribution, error detection and flow control. In case a DLM has to be sent while a DTM or DCM transmission is processed, the DLM is inserted within the next clock cycle and the data or control packet is thereby delayed. In the LP_RX_IN module of the subsequent receiver, the DLM is extracted from the stream and the data or control packet is merged again. To support unbalanced links, the number of transmit and receive lanes can be set by parameters independently. All other logic is fully configurable and enables the hardware structures needed for the respective operational mode. The LP_TX_SERVICE module issues the sending of control characters to manage the link initialization and flow control. When using unbalanced link, only higher data bandwidth is desired. Therefore, the link port is designed in a way, that only the data channel is replicated to either 2x or 4x configuration.

The LP_TX_BUFFER module, which is used in the data and control path is depicted in fig. 4.11. As this module is the main storage of data within the network protocol, it has been designed with special care to prevent locks and data loss due to soft errors. Every logic unit is encapsulated to reduce impact in case of malfunction. The earlier mentioned hardening techniques are used to continue operation in any case. ASR-Watchdogs observe the data flow and

reset respective FSMs, when armed. The clean logic performs a proper rollback to a functional state, when triggered by one of the FSMs. A CRC is appended as soon as possible to allow corruption detection later, especially to protect the message while stored in an SRAM FIFO. If the Buffer FIFO is not empty, and a full message has been written into it, the read FSM starts generating a packet. A request is created to the LP_TX_OUT module, and as the link is active a grant is asserted back to the FSM for one clock cycle to continue packet sending out of the buffer. As long as the message is delivered, the valid signal stays active. Similar to the LP_TX_BUFFER module, the LP_RX_BUFFER module is designed. If configured as an intermediate design, the CRC is only checked and stays appended. Both buffers are implemented using a speculative FIFO, which allows a speculative shift-in or shift-out to decide later, if the value should be stored at all. This is necessary to prevent that fragments of corrupted packets remain within the data path.

4.3.2 CBMnet PHY

The physical layer provides the electrical interconnect with the transport medium on one hand and interfaces with the link port on the other. Internally, it contains of an individually selected number of transmit (TX) and receive (RX) lanes, the lane clocking, and also provides a diagnostic interface. The PHY_CONTROL module manages control signal distribution between the lanes, as they need to share information regarding initialization status for example, in case of unbalanced links. Every lane is divided in a Physical Coding Sublayer (PCS), which performs the character coding, runs the main initialization routine, and a hardware-dependent Serializer/Deserializer (SerDes) implementation.

The main initialization routine is responsible for the word alignment of incoming data, because (as usual regarding serial interconnects) data can start anywhere in the serial bit stream. If valid characters are not detected until a timeout, the whole bit stream is delayed with a barrel shifter until proper alignment is achieved. As receiver and transmitter may run with different reference clocks, synchronization of the control signals is necessary. A detailed description of the initialization routine is given in 5.3.1. The barrel shifter and initialization FSMs are protected with SR-TMR, as soft errors in these modules would disturb the link function and

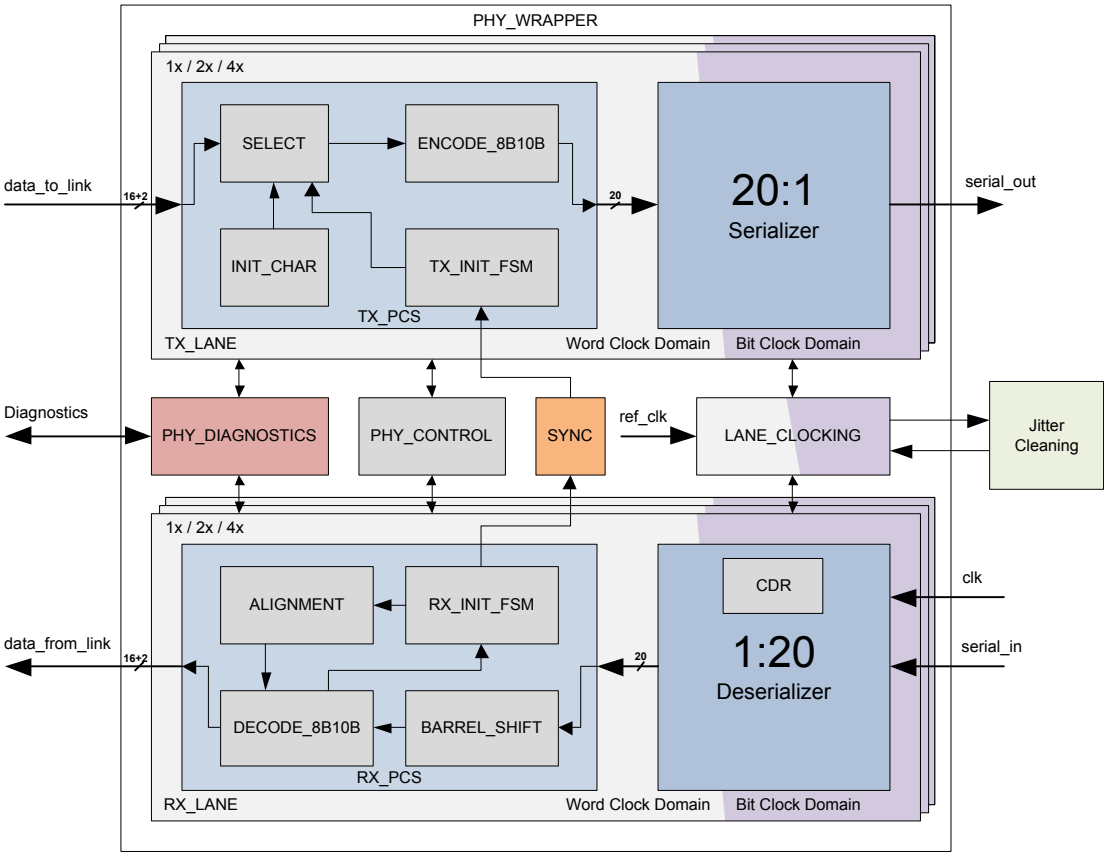


Figure 4.12: The generic CBMnet PHY wrapper with unbalanced link support. Every lane is divided in a high-speed bit clock domain, and a low speed word clock domain. The SerDes implementation depends on the particular device hardware and may require jitter cleaning of the recovered link clock.

a complete reinitialization would be necessary. Nonetheless, all other modules are self-cleaning in case of an error to guarantee a continuing operation.

The SerDes modules are currently available in four different implementation. It can be distinguished between high-speed LVDS SerDes cores, which operate with a bit clock frequency of 500 Mbit/s per lane and are available for FPGA and ASIC use, and Multi-Gigabit Transceiver (MGT) cores, which operate between 2.5 Gbit/s and 5 Gbit/s, and are also available for several FPGA devices and the HUB ASIC. A synchronous read-out tree is always ensured by global clock distribution. Therefore, every SerDes has either a dedicated clock line, where the received clock is directly used for the whole device, or the clock is recovered from the link by a Clock Data Recovery (CDR) circuit, by using signal transitions to feed a PLL.

4.4 CBMnet Plug-ins

The network layer functionality is only needed in some devices and is therefore not part of the generic CBMnet modules, but can be additionally added as plug-ins. Depending on the design, also further features are required. All plug-ins are accessed over the earlier mentioned PUT/GET commands using the CBMnet control channel. Therefore control message routing is required in every design. Depending on the number of used plug-ins, the address space is partitioned. Compared to the former CBMnet implementation, where a full-blown and complex HTAX crossbar has been used, the new implementations are much easier implemented and radiation tolerant, resulting in decreasing resource consumption. Moreover, the functionality has been divided into two different plug-ins, as no full network switching capabilities are required. Some of the important plug-ins are described below.

4.4.1 Data Combiner

Efficient data aggregation is needed within the CBMnet and is provided by the data combiner. It is built-up generic with simple logic and can be parametrized to combine any user-defined number of links. Compared to the HTAX crossbar it supports the common CBMnet interface, thus no message translation is necessary.

A round robin arbiter ensures even message bundling to one single output. Error detection and ASR-watchdog functionality is also implemented to ensure reliable operation. The data combiner can be used for the data as well as for the control path.

4.4.2 Control Router

To access single devices within the read-out tree, also network routing capabilities are required. They are enabled by feeding the outgoing control path of every link port module through the control router plug-in. The destination address of a message is read out and the message is routed to the particular logic either within the design, or is forwarded to another device over CBMnet links. For initialization purposes and distributing network addresses, also broadcast operation is possible. Similar to the data combiner, the control router is optimized for less area consumption, radiation hardened and built-up with simple logic. It can also be parametrized to support a user defined address space and any number of endpoints.

4.4.3 Register File Connect Module

The final storage of configuration values within a design is the generated register file. It provides a generic interface with a simple read/write signaling. The earlier mentioned FEET optics protocol uses the CBMnet control channel and also provides a software stack to issue commands from a computer. The PUT/GET commands allow the execution of up to four simultaneous operations to set or read multiple values. To access the device register file, a command translation is necessary. This is done in three steps. First, the multi-oper list is stored in the local memory and the header is evaluated. It gives information about source and destination and contains the number of commands issued. Second, every single command is translated and propagated to the register file interface. If the access was successful, an acknowledge message is generated for the particular command and stored in the transmit buffer. Otherwise a NACK is appended. This step will be repeated for every PUT/GET command. Last, the stored acknowledgements are packed in an answer message and send back via the CBMnet control channel. In case a message is damaged or a control operation is disturbed, the value will

not be written to the register file. It is ensured, that all outgoing control signals are not persistent, to allow any self repair necessary for subsequent modules.

4.4.4 External plug-ins

Beside the CBMnet plug-ins, also external logic is used by the several detector working groups. Especially in the front-end devices, also trigger, signal generation and measurements are done for debug and analysis purposes. Therefore an auxiliary module enables the access to general purpose input outputs (GPIOs) of the FPGA. Further, to read out example data from a detector or access the register file of a front-end device to adjust analog calibration, always several devices of the read-out chain are necessary. Either a ABB or FLIB is needed as host interface for a computer. To ease the service of single parts of the read-out chain, a USB glue logic has been developed. It provides a low cost solution for data read out without additional requirement for special hardware and can be used with every commercial of-the-shelf computer with USB 2.0 support running a Linux operating system. Both plug-ins have been developed by the IRI at the Goethe University Frankfurt am Main. Also several other plug-ins are currently available.

4.4.5 Automated Build System

Within the CBM collaboration, and its working groups, many designs are developed together and therefore code bases have to be shared. But redundancy always creates serious problems if frequent synchronization is not guaranteed. Especially debug scenarios are considerably complicated if no common code base is used. To improve this situation, a common CBM soft repository, using an Apache Subversion (SVN) revision control system, has been established, and the CBMnet cores have been divided into a reasonable structure. Further, a global automated build system has been launched, and all designs have been added to an automated build flow by adapting a makefile for the Xilinx ISE design flow, provided by Dirk Hutter, FIAS Goethe University, Frankfurt am Main. The CBMnet directory structure is divided into

- **Unified Building Blocks**
  Containing all device independent HDL modules, e. g. like DLM and timing,

sync buffer, network layer functions, as well as generic link core and PCS logic.

- **FPGA Cores**
  Delivering a device dependent configuration for every interconnect, partially using the FPGA hard macros. Cores are available for Virtex-4, Virtex-5, Spartan-6, Kintex-7.

- **Link Designs**
  This directory is also subdivided into subfolders containing wrapper modules for a design-specific implementation of an FPGA core. Depending on the needed bandwidth, also several cores are generated in parallel.

- **Global Designs**
  This folder contains all designs, which are assembled of different link designs for back and front-end links, network, control, configuration logic, and plug-ins. For every device, several design configuration and constraint files exist.

- **Verification**
  This directory contains all test and debug logic to perform verification of single link designs, and also global designs. Test pattern generation and fault-injection are supported.

As ASIC developments are independent processes, compared to the frequently regenerated FPGA designs, and the distribution of external IP is restricted, they were only shared within the respective design group in a repository with limited access. For every submission, a code snapshot has been taken from the CBM soft repository.

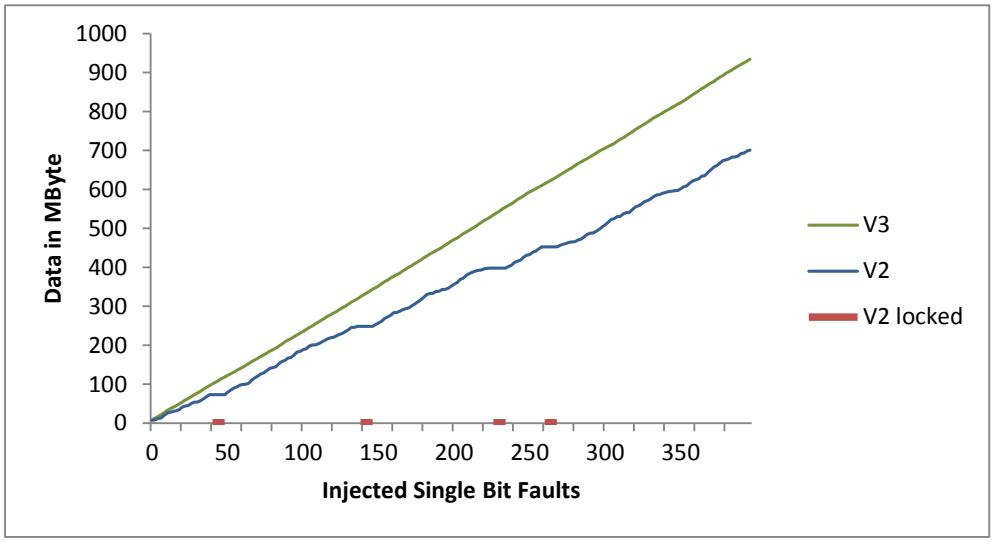
Usually FPGA designs are built within the Xilinx ISE Design Suite, but all commands can also be executed manually. To support the autonomous built of whole designs, the structure mentioned above is also used within the design flow. Every FPGA core contains an adapted makefile, which generates the HDL for the particular FPGA hard macro. A device config file contains information about the specific hardware platform. All link designs are generated using configuration scripts, which contain design parameters, like number of lanes, number of front-end links, type of front-end devices, etc. For each of the CBMnet sub building blocks a partial project file is delivered by the designer. It lists all files, that are

relevant for the building of the sub core, and allows to generate full project files on-the-fly during the automated build process. The global makefile first generates all necessary IP cores. Further, the link designs are synthesized and netlists are built. These netlists are read in and composed in a Xilinx Native Generic Database (NGD) file. The mapping, place and route, as well as the bitfile generation take place, considering the particular design constraints. With every new commit to the CBM soft repository a new built of all designs is triggered. Thereby, the latest firmware is always directly available, all logs can be analyzed afterwards using a web interface, and it is much easier to reconstruct, which code modification caused problems in a design.

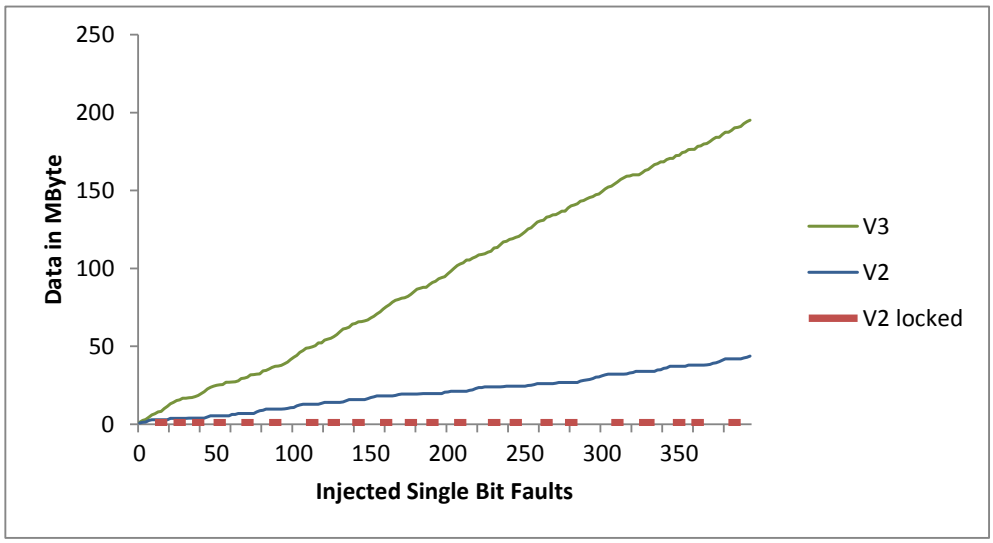
4.5 Evaluation

To compare the two CBMnet core implementations against each other, an evaluation regarding reliability and costs has been made. Surprisingly, the expected calculated data loss is still very small for all scenarios depicted in table 4.1, but actually the calculation is only valid for a fully utilized link. The CBM DAQ is free-running and data volume appearance may be very event dependent. Moreover other data corruption beyond the CBMnet protocol should be regarded, and for larger designs like the HUB ASIC even more impact on the design is expected. A final estimation should be taken from live application, as no information about ASIC cross sections are available yet for the 65 nm process. It should be mentioned again, that the CBM experiment works with very high data rates, since interesting events are very rare. Therefore, the data loss within the transportation has to be reasonably small. However, the rough estimation provides a realistic impression of consequences regarding soft errors in the data path, control path, and configuration settings.

To compare the reliability of the new CBMnet cores against the earlier implementation, extensive fault injection test simulations with a read-out chain setup have been made, consisting of four front-end devices in four-lane configuration with pattern generators, the ROC3 as data concentrator and the FLIB as data sink. The front-end emulators generate variable length data packets, as much as the links can handle to ensure a full utilization. Tests have been made, injecting faults separately for the data and for the control path. Overall 400 single bit faults were injected random every 5 to 15 ms only in the ROC3 design. In case



(a) Fault injection in data path.



(b) Fault injection in control path.

Figure 4.13: Results of single bit fault injection tests versus received data, comparing the CBMnet version 2 against the new version 3. Even for errors only in the data path, the version 2 gets locked. Regarding differences in the control path, the version 2 can only reach 23 % of the data throughput and the design had to be reset several times.

no more data was received for more than 50 ms, a manual reset of the device had to be done. The results are presented in fig. 4.13. It can be recognized, that even for errors happening only in the data path (a), the version 2 has a heavily decreased data throughput. While the version 3 delivers a smooth data stream and only single packets have to be dropped due to corruption, the version 2 triggers many retransmissions, which stall the whole data transmission of a lane for a few microseconds. In four cases also a manual reset had to be done due to corrupted data in retransmit buffers. Also the version 2 only reaches 75 % of the data throughput of version 3. Regarding faults injected in the control path (b), for the version 3 design the total amount of data received is only around 21 % compared to the first test, but still the design runs reliable and all control logic self cleaning mechanisms work properly. For the version 2 control path, the data throughput is decreased to only 7 % compared to the former data path fault injection tests. Finally, comparing the core implementations against each other, the version 2 only receives 23 % of the data of the version 3.

A presentation comparing fault injection tests in the configuration settings between the two core implementations has been resigned, since in the version 2 around every seventh injected fault lead to a hangup and the design had to be reset, while the version 3 runs fully reliable for all protected signals.

Summing up, the version 3 implementations show a heavily increased reliability regarding induced soft errors. Although a data throughput decrease of around 80 % seems very heavy, this results are owed the intense injection of errors. As presented in chapter 2, the expected SER during the final experiment is considerably smaller by several orders of magnitude.

Regarding costs, the area and resource consumption of the new link port and PHY implementation was reduced to around 83 %, while the configuration overhead increased to 324 % due to redundancy added. But as the important configuration settings only occupy around 4 % of the example design, the final resource consumption could be reduced to 96 % of the earlier implementation, but now providing many new features and high reliability.

4.6 Conclusion

The CBM detector setup extension led to a lot of design challenges and new developments, which have been examined and described in this chapter. The whole code base has been revised, completely new written, and thereby adapted to the new demands of the CBM DAQ. Furthermore special care had been taken to a generic implementation to allow a flexible use of the CBMnet IP cores in any device with any configuration in terms of speed, bandwidth, hierarchy and interconnect type. All HDL is available well-structured in the CBM soft repository, and all designs are generated fully automated, while it is always ensured, that the latest code base snapshot has been used. Regarding reliability and resource consumption, both could be improved significantly. Especially radiation hardened functional units have been proven, using the sophisticated ASR watchdog logic and the SR-TMR method. In larger setups, users benefit from gathering statistical error and system information, using the new generic diagnostic interface. Since the testing of large read-out setups in beam times is very difficult and time consuming, and debug capabilities are limited, a SEU fault injection simulator tool has been developed for the Cadence simulation environment to allow an assertion-based verification of the whole read-out chain. In the following chapter, final design implementations, using the new CBMnet cores, are presented.

Chapter 5

CBM Design Implementations

This chapter gives an overview of current CBMnet implementations, while using the generic link port, physical layer modules, and plug-ins from chapter 4. As these building blocks are highly configurable, they just had to be integrated in the particular design, using the respective parameters. On one hand, there are the FPGA designs, like the FLIB, DPB, and Syscore3, which make intense use of the Xilinx integrated hard macros. These have to be configured in a special way, or with sophisticated algorithms to assure fault tolerance and deterministic latency. Since also for the same hardware platform designs with different settings (e. g. link speed, bandwidth etc.) are needed, parameters and configuration options are implemented in the whole design structure and make flow of the automated build system (4.4.5). On the other hand, there are the ASIC designs, like the SPADIC, STSXYTER, and the HUB ASIC, which necessitated the development of special alignment logic, clocking structures, I/O interfaces, and in case of the HUB ASIC, also the development of a full custom multi gigabit transceiver. The relative physical layer implementation is extended by design-specific calibration logic, which is described in detail, and the synchronization concepts are presented to gain the special feature of deterministic latency through the whole read-out chain.

5.1 Front-End ASICs

For the particular STS and TRD detector read-out chain, the CBMnet has been integrated directly in the front-end ASIC to extend the protocol use [46]. The demands for an increased data bandwidth by a remaining number of I/Os could

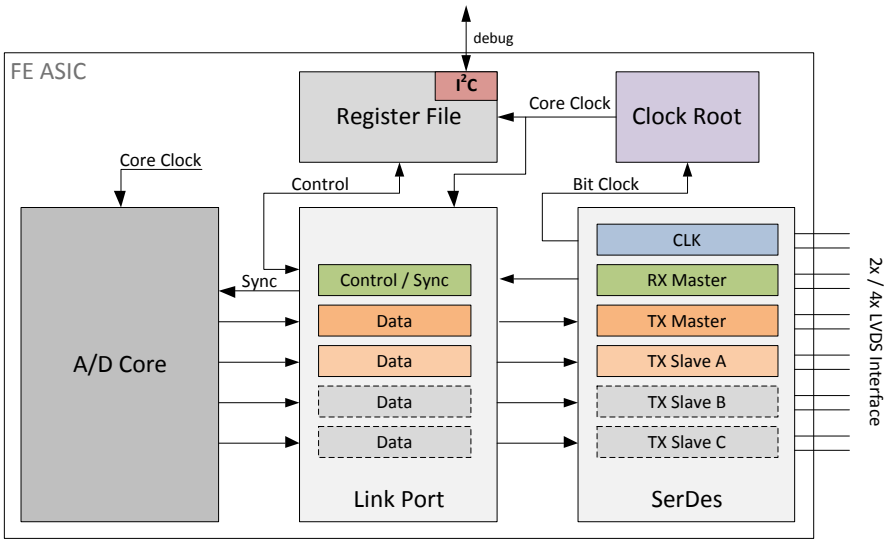
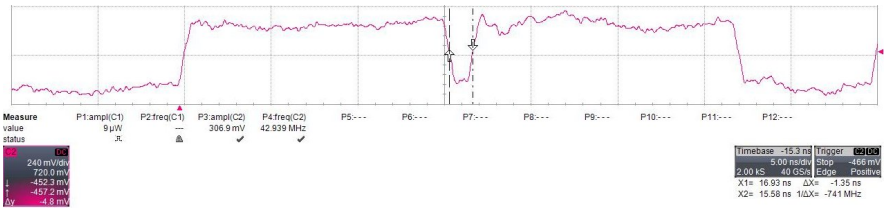
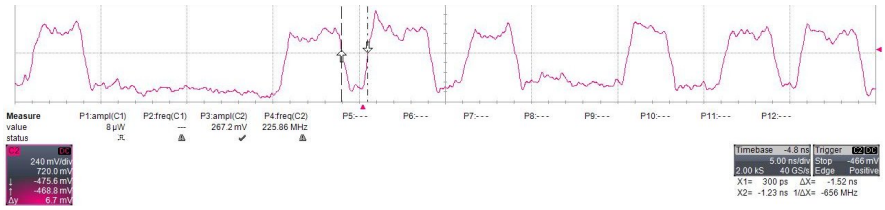


Figure 5.1: Generic view of a front-end ASIC design containing the CBMnet link modules, an special developed SerDes implementation and the configuration register file. The TX slaves shown grayed are only available in the STSXYTER design.

only be fulfilled by developing special CBMnet serial interfaces. Moreover, the CBMnet provides accurate synchronization features for the digitizing part and therefore an external trigger circuitry can be waived. The submission of the SPADIC ASIC has been done in cooperation with the research group of the Chair of Circuit Design, Heidelberg University, and the STSXYTER ASIC within a collaboration with the AGH University of Science and Technology, Krakow, Poland. Both chips have a quite similar built-up regarding the CBMnet link part. They both use the same clocking scheme, contain a configuration register file and a CBMnet back-end link implementation with unbalanced transmission lines using serial LVDS interfaces. The only difference is, that the STSXYTER can read out more channels from its detector and therefore requires a higher bandwidth. Thus, the unbalanced link implementation provides two lanes in back-end direction for the SPADIC and four lanes for the STSXYTER. The link frequency of the LVDS interface has been set to 250 MHz per lane using double data rate (DDR), resulting in a bandwidth in back-end direction of 1 Gbit/s for the SPADIC and 2 Gbit/s for the STSXYTER. A generic view of the link implementation in a FE ASIC is depicted in fig. 5.1.



(a) Data stream with 50 MHz bit clock.



(b) Data stream with 250 MHz bit clock.

Figure 5.2: Output of the SPADIC serializer with different bit clock speeds. In subfig. (a), the indicated glitch only corrupts a small piece of area between the two ones sent. In subfig. (b), the serializer runs with full speed bit clock. Obviously, it is difficult to distinguish between the glitch and the subsequent sent zero value.

5.1.1 LVDS Link Interface, Problems and History

The link implemented in the FE ASICs contains the same generic link port used in the FPGA designs. It is configured for a additional data throughput by adding one or three data channels in back-end direction. The front-end ASICs represent the leafs in the CBMnet based read-out tree. Besides the additional I^2C interface for debug purposes, all access can only be done over the CBMnet LVDS link. Therefore the initialization logic within the SerDes has been kept very simple and all complex synchronization logic is located in the ROC3 or HUB ASIC instead.

To waive all complex clocking circuitry, like a clock data recovery (CDR), a bit clock is directly sent to the FE ASIC over a dedicated differential clock wire. Inside the ASIC it is used to feed the clock root, which provides the bit clock for the SerDes, and further divided clocks for the word clock domains and core logic. Therefore the whole digital logic is fully dependent of the external clock signal. If clock gating is used, the ASIC's logic will just "freeze" for a moment. This feature is also used for alignment and latency adjustments and will be described in 5.3.1.

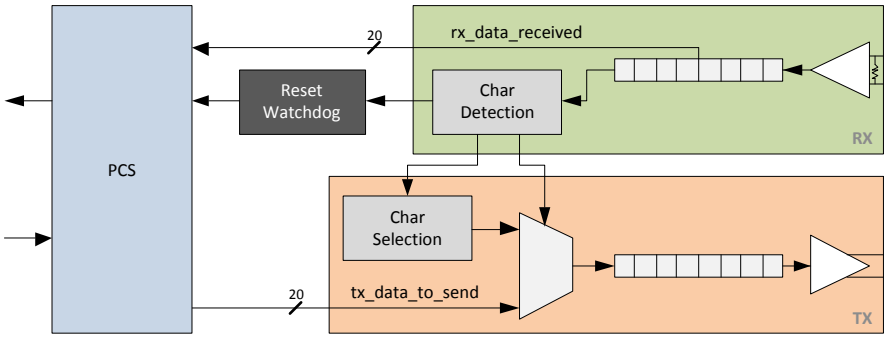


Figure 5.3: Simple SerDes implementation in the FE ASICs, which gains full control over the link from the subsequent ROC device.

In the first version of the SPADIC ASIC also a simple semi-custom design of a serializer and deserializer has been used. Although the SerDes logic was already ASIC proven, due to inappropriate constraints in the design flow, the two clocks, which sampled the outgoing serial registers had a not negligible skew. This led to a glitch in the serial bit stream of around 1.3 ns, independent from the clock frequency. Since the link speed is 250 Mhz, double data rate is used, and the resulting UI width is 2 ns, the sampling of the serial data from the SPADIC was heavily disturbed, because around 70 % of the signal were corrupted. Finally the transmission frequency of the ASIC had to be reduced to 175 Mhz. For the subsequent submitted STSXYTER ASIC the backend flow has been elaborated and a correct synthesis has been done. Thus, the full functionality is available for the STSXYTER ASIC [77].

5.1.2 SerDes Implementation

For the ASIC submissions, new physical layer modules had to be developed to prove the concept of deterministic latency and ease the read-out by using CBMnet directly on the front-end devices. To gain full control over the link from ROC side, a very simple SerDes and PCS is implemented in the FE ASICs like depicted in fig. 5.3. All complex initialization logic, extensive state machines and alignment units have been dropped. The LVDS-SerDes circuit is not explained here in detail, but further information on SerDes design and implementation is given in chapter 6. This led to a complex initialization logic within the ROC design, which executes the word alignment by also balancing the propagation time of synchronization

messages to ensure equal deterministic latency from the ROC3 to all connected FE ASICs. Within the elaboration of the CBMnet physical layer implementations, the LVDS link cores have been made radiation tolerant and the initialization routine was adapted properly to the demands to save resources. The serializer and deserializer are implemented as simple 20 bit shift chain, where parallel data is read from or loaded into. Since only the bit and word clock are available, no complex multiplexing circuit has been used. This gained further advantages, like an easy alignment independent reset of the link, which is necessary, since the only possibility to fix a dead link in the final detector setup is via the CBMnet link to the subsequent ROC.

For the successful initialization of a FE ASIC with a subsequent device, several requirements have to be met and are subdivided in four initialization steps.

- **Reset**
  A watchdog logic looks for the unique reset character 0xFFC00 in the serial bit stream, which does not appear in the regular 8b/10b coding, but can be transmitted through AC coupling anyway, and has enough hamming distance to any K or D character, so that no link reset can be triggered erroneously even due to multi bit upsets. After the reset the logic can detect four different patterns, and sends the respective pattern back to the ROC:

  1. *undefined* ▷ 0xAAAAA
  2. 0xAAAAA ▷ 0xCCCCC
  3. Align Char ▷ Align Char
  4. ACK Char ▷ Regular Link Operation

- **Bit Alignment of Data**
  The phase of incoming data in relation to the clock signal edges has to be shifted to a point, so that data can be sampled safely in the receiver. Therefore, a clock pattern can be recognized as either 0xAAAAA or 0x55555 in the receiver and the transmitter responds with the patterns listed above. With these simple bit sequence, the ROC can easily detect the state of the FE ASIC by only using a 3 bit input register and no word alignment is required.

- **Word Alignment of Data**
  The FE ASIC contains no barrel shifter to avoid hidden phases. The word alignment is done by shifting the bit stream on ROC side. As soon as the align character (K28.5 + K28.3) is detected, the SerDes output switches to transmission mode and can only be stalled again by the the detection of the reset character. Now the transmitter also sends the align character and the ROC will adjust the word alignment on its receiver side.

- **Regular Link Operation** The align character is sent until the ROC has finished the alignment on receive side and confirms the successful initialization with an ACK character, which is only sent one-time and again is replied by the FE ASIC. Afterwards the deterministic latency initialization is done by the ROC.

The phase of incoming data with respect to the clock stays constant, also when using cables with different length, since clock and data signals are delayed by the same amount. Thus, eye measurement and sampling adjustment on ASIC side have to be done only once and can be set fix for all process, voltage and temperature (PVT) corners and cables. In back-end direction only the data stream is sent to the ROC and no clock signal is appended. Thus, the phase of incoming data depends on the cable length and an eye measurement and phase alignment is required. The whole initialization mechanism is described in 5.3.1.

5.2 FLIB and DPB Link

The FLES Interface Board (FLIB) acts as the bridge between the CBMnet part of the read-out chain and the First Level Event Selection (FLES), which is a high performance computer cluster for live event reconstruction to reduce the amount of data for final storage. The front-end side of the design provides up to 16 serial interfaces which are bundled to one PCIe host interface. The design provides a link speed to the DPB of up to 10 Gbit/s per link. The current implementation is done for a HighTech Global Kintex-7 PCI Express development board with the external Faster Technology FM-S18 Octal SFP transceiver FPGA Mezzanine Card (FMC). The Data Processing Board (DPB) is responsible for link speed-up and detector data feature extraction, and acts as a intermediate device between

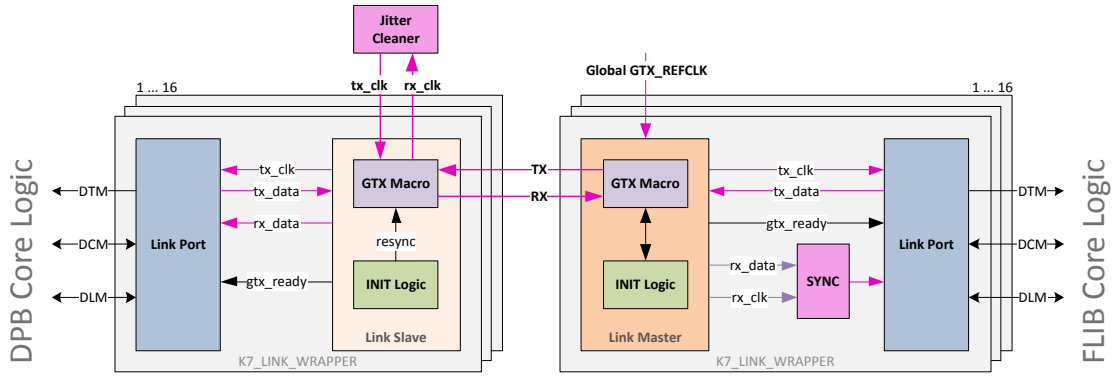


Figure 5.4: The link between the FLIB and the DPB with assured deterministic latency for all lanes is possible by using one global reference clock, which is recovered from the link master and used as transmit clock again.

ROC3/HUB ASIC and the FLIB. It also uses a Kintex-7 FPGA and can therefore use the same link implementation [80].

The FLIB and DPB link, which is depicted in fig. 5.4, uses the generic CBMnet link port, which can be used unlimited in parallel service. For the serial interfaces the Kintex-7 built-in GTX primitives are used and configured manually to ensure a deterministic timing behavior on all used lanes. As one FLIB represents the root of a read-out tree, it has to fulfill special demands on clock distribution and synchronization. Therefore, the clocks of all transmitters have to be derived from one single oscillator. On DPB side, the received clock needs to be recovered and further used as transmit clock back to the FLIB. Since the quality of the recovered clock is too low for direct use, jitter cleaning is required. Furthermore, to also ensure deterministic latency over reset or power cycle, all internal delay stages in the Xilinx GTX core for comma character detect and alignment, as well as buffering have to be bypassed. The native interface width has to be used and data need to be manually tapped at the correct position to prevent uncertain delay. The derived parallel clock by the CDR has to be aligned to the parallel data, but a barrel shifter for comma alignment must not be used in a slave receiver, since it results in a hidden delay, which can not be detected by the transmitter, which has to equal all transmit latencies. To fix this problem, another method is used. Until the receiver uses the recovered clock, the internal oscillator is running free, and even when set to the same frequency, no two crystals run absolutely even. Thus,

the phase relation between the two clocks is not fixed and the word alignment will be different after every resynchronization of the CDR. Assuming this, the correct deterministic word alignment can achieved after a finite number of resets. This method is also used in other CBMnet devices [42]. The CDR clock from the link slave is also used as reference for its transmitter to ensure a synchronous transmission back to the link master. For the clock jitter cleaning is required to gain signal quality. The data received by the master need to be synchronized, respectively phase aligned to the transmit clock for further processing. The latency is adjusted by DLM plugins, which measure the round-trip time and calculate the particular delay.

5.3 ROC3 Prototype

The Read-Out Controller in the third version (ROC3) supports also CBMnet using front-end LVDS interconnects and acts as a prototype for the HUB ASIC. It collects data from several connected FEBs and combines them to one back-end link to the DPB. Therefore it is responsible for data aggregation, detector slow control access, link speedup, and synchronization of the subsequent FEBs. A design block diagram of the ROC3 is depicted in fig. 5.5. It uses the new improved version 3 of the special developed universal SysCore architecture, which allows the CBM collaboration to prototype FEE, or evaluate ROC and DPB design. It is equipped with a Spartan-6 FPGA and provides many other interfaces, like FMC HPC connectors, USB, optical SFP connectors and many other components [28]. The ROC3 design toplevel is reasonably divided in three parts. The back-end link, using the Xilinx GTP transceiver, is configured for synchronous transmission by using the recovered clock for the transmitter, similar to the GTX link in the FLIB/DPB. The current implementation provides a 2.5 Gbit/s link. If the back-end link can not be used because no host interface board, ABB or FLIB, is available, also the USB core can be used to collect data, provide control messages and local synchronization. Because of the buffer scheme, the USB protocol does not allow any fixed latency settings.

Received slow control messages are routed to the respective front-end link or design internally, based on the routing table. In opposite direction, control messages are combined and sent towards the back-end. The ROC3 core design runs with a 125 MHz clock, while the front-end links are only clocked with 25 MHz. This

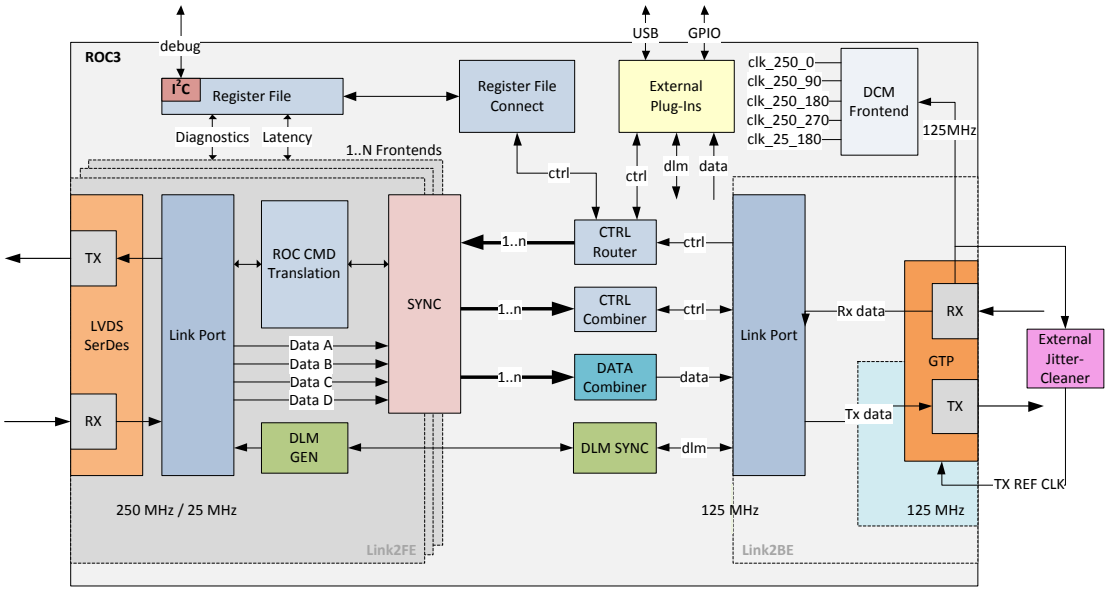


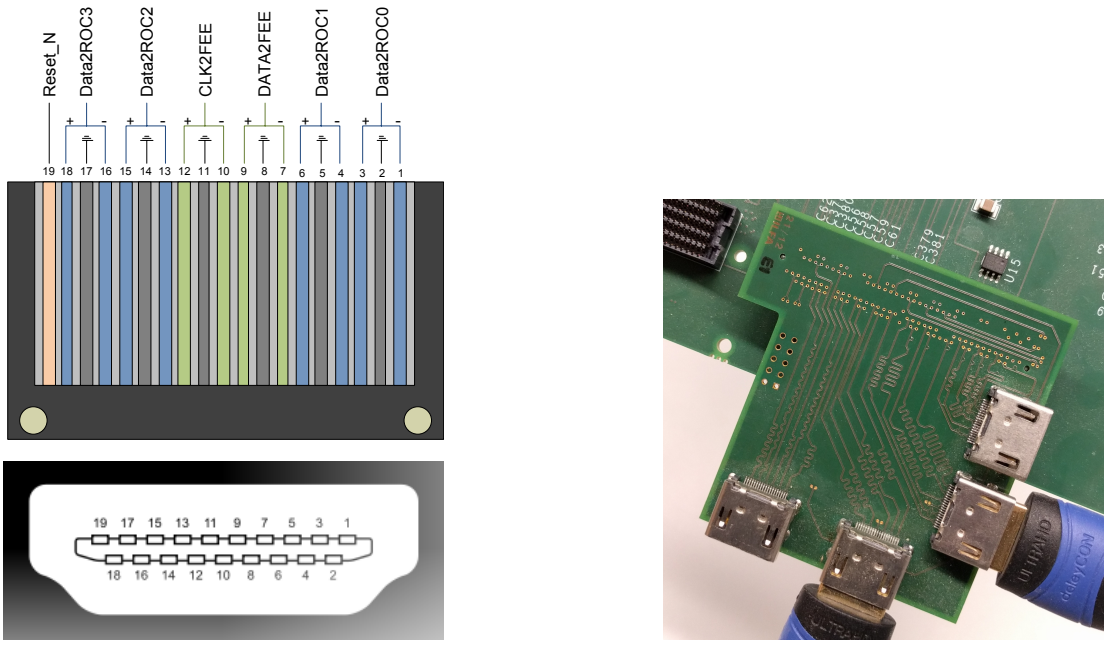
Figure 5.5: The ROC3 design based on the SysCore3 development platform. Several LVDS front-end links are combined to one back-end link to the DPB. All CBMnet features are supported.

results in a link-speedup of factor five. Received data from the front-end are at first synchronized to the 125 MHz clock domain, and then combined. DLMs are also at first synchronized to the slower clock domain and then reflected, since the synchronization adjustments regarding all ROCs are happening in the subsequent device. The fast-to-slow sync is done by a handshake, while the slow-to-fast sync is done by sampling and edge detection. For deterministic latency reasons the phase relation between the two clocks is fix and DLMs are only allowed to send within a coarse timing grid at the back-end.

5.3.1 LVDS Front-end Link

The connection betwenn FEBs and ROC is done using standard HDMI cables, since they offer a low-cost solution for serial transmission and provide six differential wire pairs. To connect the FEBs to the ROC a PCB with four HDMI plugs for the SysCore3 onboard FMC was fabricated [84]. The wiring is depicted in fig. 5.6a.

The ROC3 digital design contains up to four unbalanced front-end links with up



(a) The wiring of the HDMI connector for the unbalanced link from ROC to the FEBs.

(b) The developed HDMI FMC for the SysCore3 board providing four HDMI plugs.

Figure 5.6: HDMI usage for front-end connections.

to four lanes in back-end direction each. This provides high bandwidth streaming with up to 2 Gbit/s per link. One link consists of a special developed full-digital LVDS SerDes circuit, using the Xilinx built-in SelectIO DDR capabilities. Since the input SerDes (ISERDES) and output SerDes (OSERDES), which would ease the development of a serial interface, only offer DDR feature with source-synchronous systems, they could not be used, but a manual design approach had to be implemented. A first simulation has been made in [85].

The structure of the transmitter and receiver are depicted in fig. 5.7. Regarding the transmitter, data and clock can be individually delayed bit-wise by the Delay TX module. Also clock gating is possible using the Xilinx ODDR primitives. The receiver also contains an IODELAY primitive to shift data with up to 256 taps per clock period for eye measurement, managed by the IODelay Control module. The final values depends on the speed grade of the FPGA and target frequency, and vary between 16 ps and 53 ps.

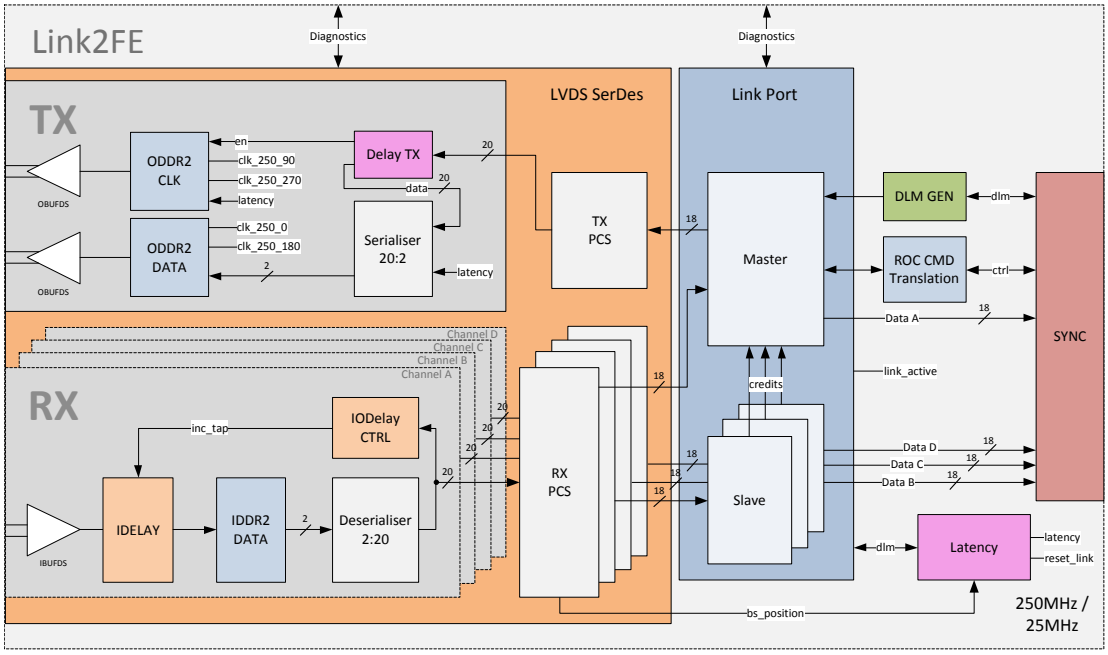


Figure 5.7: The ROC3 LVDS front-end link design using the Xilinx built-in SelectIO DDR capabilities. Deterministic latency is ensured for all connected FE ASICs.

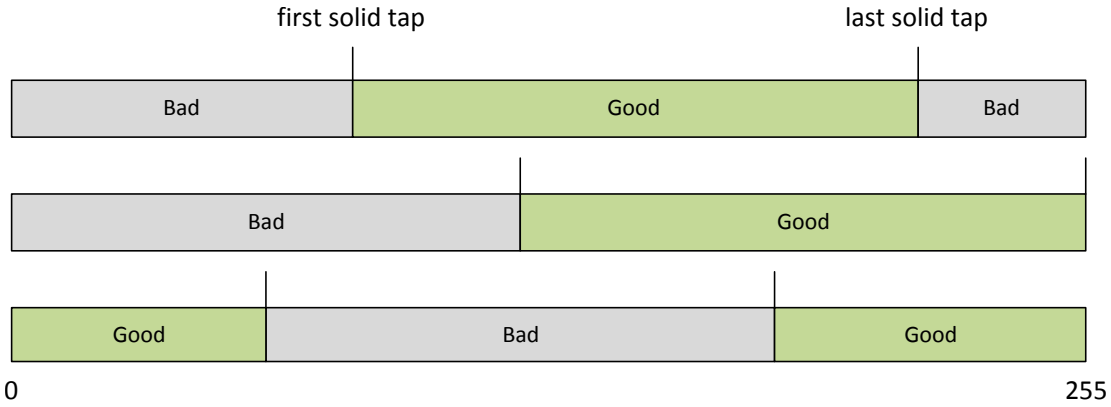


Figure 5.8: Types of good sampling ranges of the incoming bit stream.

To ensure deterministic latency to bit clock level when sending messages to the FEBs, all sequential logic has a path with fixed delay to the SelectIO primitives. Final sampling of the outgoing clock and data inside the I/O cells assures consistent results, independent of placing and routing. Only negligible PVT variations apply. An initialization algorithm, to ensure simultaneous and deterministic arrival of synchronization messages for LVDS ASIC and FPGA interconnects, has been developed and is described in more detail in [78]. It is assumed, that the phase relation between outgoing clock and data has been set to a fix value, determined by a former dedicated test phase. Furthermore, the maximum cable length difference supported is five meters. The algorithm consists of the following steps:

- **Bit alignment of incoming data**
  The asynchronous data stream from the FE ASIC is aligned to the bit clock by first running an eye measurement. Therefore a random bit pattern is sent to the FE ASIC, which will sent back a clock pattern. The incoming data stream is increasingly delayed by the IODelay primitive and the good and bad tap values identified.

  The median for the best sampling position is calculated and adjusted with

$$tap_{median} = first\_solid\_tap + \frac{last\_solid\_tap - first\_solid\_tap}{2} \quad (5.1)$$

- **Word alignment** The serial bit stream is aligned to the word clock in the FE ASIC, by bit-wise data shifting. Afterwards the incoming data stream in the ROC is aligned to the word clock with a barrel shifter.

- **Latency adjustment** The round-trip times are quantified by sending DLM0 over every link, which will be reflected in the particular FE ASIC logic, and sent back to the ROC. With these values and the internal barrel shifter positions, the additional individual delay is calculated. Now, the respective outgoing clock signals and data streams from the ROC are adjusted by clock gating, inversion, and data bit delay like depicted in fig. 5.9. Finally, the word clock phases in all FE ASICs differs by less than 2 ns.

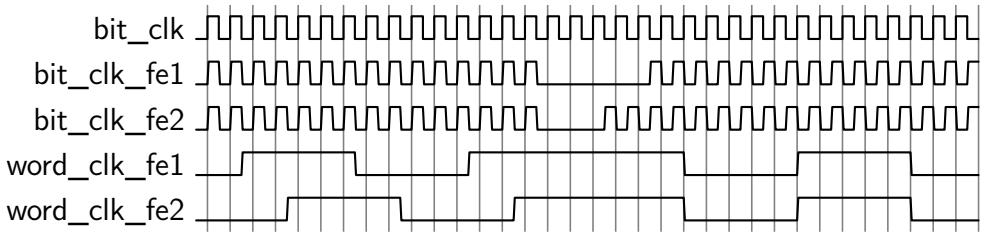


Figure 5.9: Clock gating of the particular outgoing front-end ASIC bit clock until all word clocks are phase aligned.

The method has been tested in a simulation with 24 front-end ASICs, while each simulated wire has an increased cable length, resulting in a signal runtime difference of 1.9 ns, which is below the maximum allowed deviation. Checking the variation after delay adjustment algorithm, all word clock phases allow deterministic and simultaneous receipt of DLMs on bit-clock level. To verify the correct circuit behavior also in hardware devices, a test setup, consisting of a SysCore3 and two Xilinx Evaluation Boards SP605, connected with different length HDMI cables, has been used to emulate eight FE ASIC devices in different wiring scenarios. As for the SysCore3 and the two SP605 the HDMI FMC was used, eight FE devices were emulated simultaneously. To proof the concept, every emulated FE device contains a time-stamp counter, incrementing on every rising and falling edge of the bit-clock. The test triggering process has been implemented like it is used in the final beam setup. After device reset the time-stamp counter is initialized with a random value to ensure diversity. Consecutively, the link initialization and latency adjustment algorithm is executed and a DLM is sent to reset the time-stamp counter. An external asynchronous pulsed signal, which is distributed to all FE devices, is used to capture a snapshot of the current counter value and generates a DTM with a front-end device identifier and the time stamp snapshot. The evaluation of gathered data showed is depicted in fig. 5.10. All time stamps correlated to one event just differ by one counter tick due to sample uncertainty.

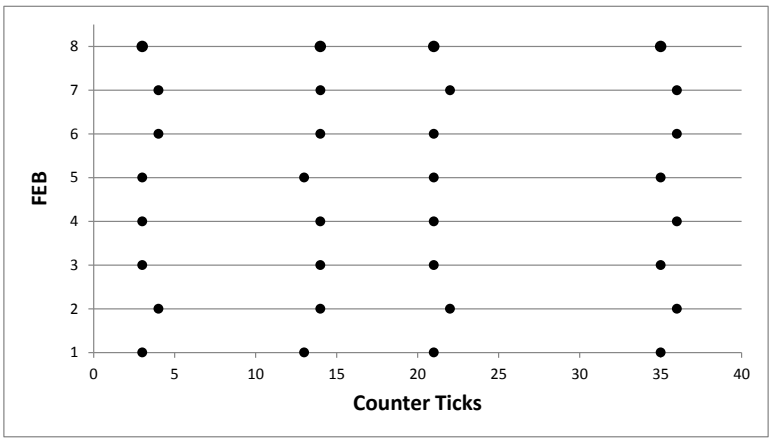


Figure 5.10: Sampled time stamp counter values of all emulated FEBs, triggered by an external asynchronous pulse.

But all sampled values are recorded with a resolution of bit-clock level, which further only depends on voltage and temperature variations.

5.4 HUB ASIC

The integration of the CBMnet in the front-end ASICs was successful in so far, that they deliver a much more efficient solution to aggregate data and provide synchronization than the earlier ROCs, using TDC interfaces and an external trigger network [81]. Based on current information, for the STS and TRD detectors several ten thousand front-end ASICs will be required. Even with a very efficient and lightweight implementation of front-end interconnects in an FPGA read-out controller, several thousand of them would be needed to ensure enough connectivity. But actually, even if there are some advantages when using FPGAs within the DAQ, like the flexibility of later firmware upgrades, there are some reasonable arguments against it. First, to ensure the required bandwidth and data throughput, FPGAs with sufficient I/O capabilities and fabric resources have to be used. As already mentioned earlier, with a four FEB design, the largest Spartan-6 device was nearly fully used and a lot of effort had to be done to meet timing. Faster FPGAs are more expensive, and especially in large quantities they can counterbalance the non-recurring costs for a full custom development and the lower price per piece of an ASIC fabrication. Second, a very dense interconnect solution was required,

as all concentrator devices have to be placed inside the CBM magnet. The space constraints were so hard, that no FPGA could fit the demands, regarding area and costs. E. g. for the TRD read-out special FEBs have been designed, carrying between 4 and 10 SPADIC chips [39]. Further, also a very dense optical solution had been developed by the working group [97]. Third, within the CBM magnet highest radiation impact is expected on electronic devices. Especially regarding TID effects, ASICs have a much higher radiation tolerance than FPGAs. Besides, if high SEU rates are expected, the FPGA configuration needs to be secured by frequent scrubbing. Due to additional necessary hardware and control circuitry, this scheme can not be realized in the front-end area.

Thus, within the CBM read-out chain for the STS and TRD detectors, the HUB ASIC development had been projected to extend the early stage data aggregation [82]. It replaces the particular Read-Out Controller (ROC) between the FEBs and the DPB. Therefore, it has to deliver high-speed serial interfaces on the front-end side, four multi-gigabit transceivers on the back-end side, and flexible routing and combining structures in between. Moreover, as the configuration logic in the front-end ASICs have been kept very simple, and all timing capabilities, like clock distribution, synchronization control and latency adjustments, are sourced out to the next stage FPGA ROCs, they had to be implemented in the HUB ASIC [79]. A large part of the network logic could be re-used from the ROC3 prototype, but especially the full custom parts needed to be developed. While the circuit design of the low-speed LVDS interface could also be done easily within the CAG project group, for the multi-gigabit transceiver there have been no expertise then, nor sufficient financing from the CBM project, to realize such a design. Thus, the development was planned in cooperation with a working group of the Indian Institute of Technology Kharagpur (IITKGP), India, who already had wide experience regarding analog VLSI design [93].

5.4.1 Top Level

The HUB ASIC focuses on aggregating data from the FEEs to the back-end link and providing slow control and clock distribution and synchronization features in opposite direction. Besides, a link speed-up is accomplished by combining up to ten high-speed LVDS links with up to 2 Gbit/s bandwidth to the four-lane multi-gigabit transceiver with 5 Gbit/s per lane. There were some general constraints

regarding the final implementation:

- Submission in TSMC 65 nm low power process
- Planned die size: 3420 μm x 3420 μm
- About 5-7 Watt power consumption
- Dynamic front-end link configuration between 2x and 4x
- Use of recovered clock from master back-end link as system and transmit clock
- Power-down feature of unused lanes

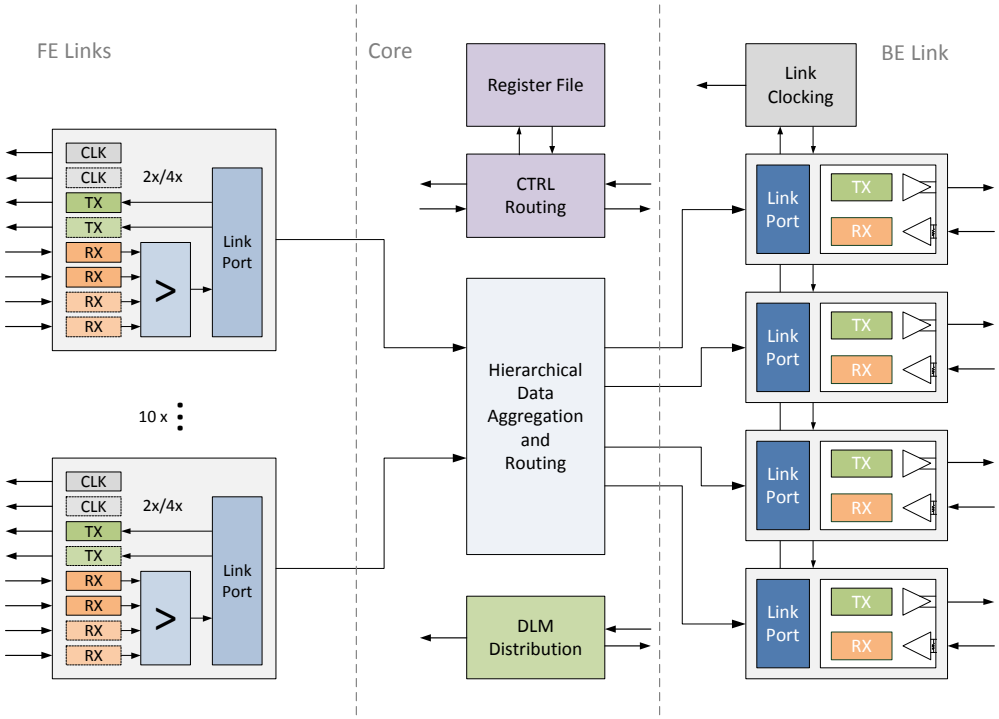


Figure 5.11: Simplified top level of the HUB ASIC. Intelligent data aggregation considering load balance from the LVDS front-end links to the multi-gigabit back-end link.

To gain a clear modular and highly configurable design structure, the top level is reasonably divided into three main parts, which are depicted in fig. 5.11. The LVDS front-end links, where every link is usable in either 2x or 4x lane

configuration. The SerDes runs with the same speed like the front-end ASICs, resulting in a link bandwidth of 2 Gbit/s, with either one STSXYTER or two SPADICs. As the front-end links only run with a word-clock speed of 25 MHz, data are combined and synchronized to the core domain, with its higher clock speed. Second, the core logic, which runs at 250 MHz, where data are aggregated in a hierarchical stage combiner and assigned to the high-speed back-end link, considering load balancing. This intelligent feature ensures, that in case one link is inactive, data will not be dropped due to congestion. Also the necessary CBMnet plug-ins are used for the synchronization, routing for the slow control path, as well as the global configuration register file. Further, a lot of additional test and configuration logic has been implemented. Third, the back-end link with its four lane multi-gigabit transceiver. While the whole digital back-end link implementation has been done by the CAG working group, the full custom 5 Gbit/s SerDes development has been done by the IITKGP group. As the core logic part uses nearly the same modules and synchronization concepts also employed in the ROC3 design, it will not be examined in more detail further. The back-end design flow of the HUB ASIC has been examined in the context of a master thesis [86].

5.4.2 Front-End Links

The front-end links in the HUB ASIC are implemented to allow flexible configuration for different FEBs. Since only a small amount of slow control and synchronization messages is streamed in front-end direction, unbalanced links, providing multiple lanes for data transmission in back-end direction, are supported to increase read-out bandwidth. One lane transmits serial data over an LVDS interface with 250 MHz using DDR mode, which corresponds to 500 Mbit/s. Every link tile contains two outgoing clocking units, two outgoing transmit units, and four incoming receive lanes. Therefore the operation of two unbalanced 2x links is possible or of one unbalanced 4x link, resulting in a total bandwidth of 2 Gbit/s. The clock is provided by a dedicated line to the FEB, and no additional clock data recovery circuit is required. Synchronization of the FEBs is done similar to the implementation in the ROC3 design, but as no hard macros are available for the ASIC, the implementation of a phase alignment logic and a front-end SerDes had to be done. The SerDes is realized by using a counter, which is running from zero to the word width W, and reset after reaching the word width. With every

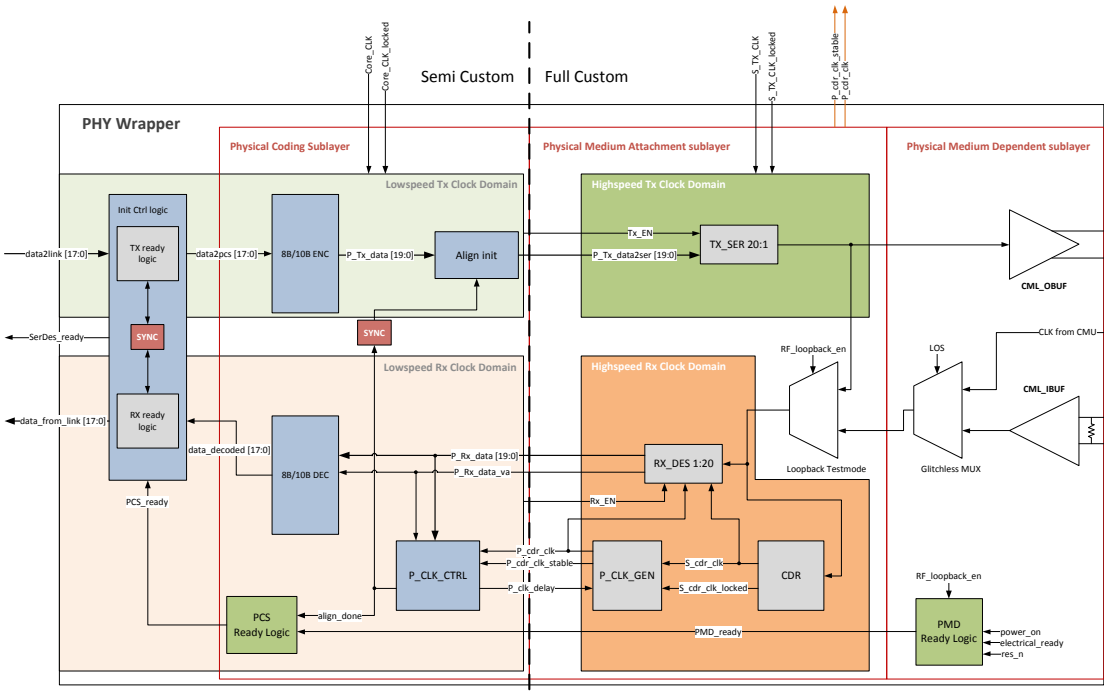


Figure 5.12: Integration of the full custom multi-gigabit SerDes in the CBMnet PHY. An eye measurement circuit evaluates the best sampling position, and the P__CLK__GEN unit provides a clock delay feature, to align the word clock on the parallel data.

rising or falling edge, bits with the index of the current counter value are assigned to a final multiplexer, which is switched with the full clock speed and provides the DDR data stream. The deserializer module is implemented equivalently and converts the serial DDR data stream to 20-bit words. The phase alignment logic is implemented as a chain of eight delay buffer elements, with a propagation delay of 0.26 ns each, and is fully built-up with digital standard cells. An elaboration of the development and description of these circuits can also be found in [86].

5.4.3 Back-End Link

The multi-gigabit back-end link of the HUB ASIC is used to connect the subtree of the read-out chain to the subsequent DPB. It consists of four independent lanes and the link clocking unit, while every lane is built-up of a link port and

a PHY. The structure of the PHY is depicted in fig. 5.12. Since in the FPGA prototype of the HUB ASIC a Xilinx GTP hard macro is used, for the HUB ASIC the whole SerDes and PHY had to be developed. As SerDes is meant the whole full-custom block in the PHY designed by the IITKGP team. It consists of the physical medium attachment sublayer (PMA) and the physical medium dependent sublayer (PMD) and contains the serializer/deserializer circuit, Clock Data Recovery (CDR), Phase Locked Loop (PLL) and I/O buffers as well as the circuits for quality of signal measurement, amplification and equalization. The further logic in the PHY, like the physical coding sublayer (PCS) and initialization logic will is a semi-custom design.

The transmitter as well as the receiver interface contains three groups of signals: data, clocking, and configuration. The transmitter reads parallel data with every rising edge of the master word clock and serializes the data using double data rate (DDR) transmission. The data output of the receiver is defined to have a width of 20 bits and the data should be clocked out with the positive edge of P_cdr_clk. More details on the full custom SerDes implementation can be found in [71] and [18].

For the receiver also an eye measurement circuit is implemented. The basic idea is having alternate paths besides the main deserialization path with independently controllable sampling positions. By sweeping the sampling position of these alternate paths it is possible to find the edges of the data eye. Therefore the output of these secondary paths is compared to the primary path and whenever the outputs do not match, one can assume that this position is outside of the data eye. Whenever the outputs of primary and secondary path match, the sampling point is inside the eye. By tracking the sampling positions and their output, the eye's size can be determined and passed on to the control logic.

5.4.4 Clocking and Synchronization

To achieve deterministic link latency, it is important to select one lane as master lane, which will provide the clock for the whole design. Hence, the receive part runs synchronously with the core clock and it is assured, that a DLM received from the master lane has no variable delay, while passing the HUB ASIC. To select one master lane, all parallel word clocks from the receive part of the particular lane are conducted to a clock distribution unit, like depicted in fig. 5.13. The

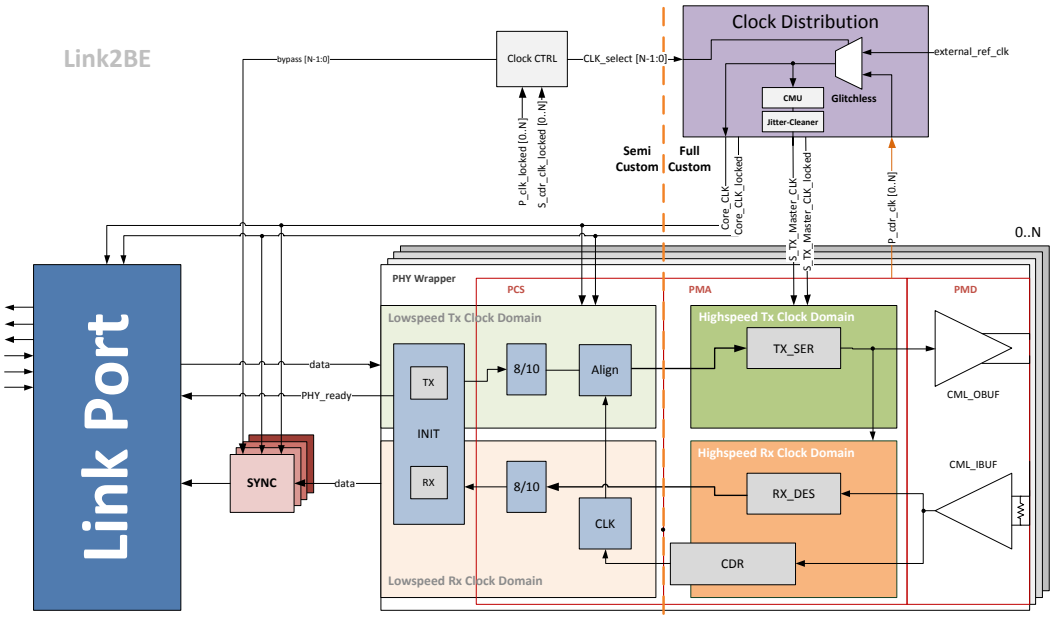


Figure 5.13: Clocking scheme within the HUB ASIC. One lane is set as master and its recovered clock is used as main clock for the whole design. DLMs are only sent using the master lane.

corresponding locked signals indicate if the CDR has locked on the serial stream properly. If there is a Loss Of Signal (LOS) on one of the receive lanes, the respective locked signal switches to zero again. The clock control block selects via one-hot-coding one clock, which will be jitter-cleaned and used as master clock, for the core and for all transmit paths. Additionally, for the transmitters, the master clock is multiplied to reach the high-speed frequency by the Clock Multiplier Unit (CMU). In case one lane fails, maybe due to bit upsets, the clock distribution unit switches to another master lane, where also the control characters are transmitted. By default, and if all CDRs has no lock, the clock distribution provides the external reference clock to the core and all transceivers. The switching between the input clocks in the clock distribution is done by a special developed circuit, and always happens glitch-less, to allow a straightforward switch without disturbance.

As already mentioned, a parallel clock P_cdr_clk is derived from the high-speed serial receive clock, which is recovered from the link. Like in the Xilinx multi-gigabit transceivers, and also in the high-speed LVDS interconnects, it is important, that the word clock is aligned is done with respect to the data - rather than the other

way around. This ensures deterministic delay for every transmitted word, even in case of reinitialization. A clock skip mechanism enables the possibility to delay the next transition of the parallel clock by one bit clock. Therefore, the P_clk_delay signal is pulled high for one parallel clock period. The P_cdr_clk_stable wire signals as soon as the parallel clock has locked again. After P_cdr_clk_stable is high, the duty cycle of the P_cdr_clk has to be 50 %. Each lane of the link will deliver its own parallel word clock. The synchronization of the data to the system clock is done in the digital part.

5.5 Design Use and Tests

For all designs, simulations have been set up using the FIS verification environment from 4.2.10 to proof the implementations against SEUs and validate the synchronization concepts. Further, all designs have been tested successfully in FPGA devices. Regarding the two FE ASICs, the STSXYTER is able to deliver the full speed of 2 Gbit/s, while the SPADIC ASIC link frequency had to be reduced. Thus only 700 Mbit/s are available in the current version, compared to the 1000 Mbit/s planned. To also gain experience from live application, a design implementation of the CBMnet link port and PHY has been intensively tested directly in a beam tests in the COSY accelerator at the Forschungszentrum Jülich, which is very convenient to carry out a test to measure single event effects. While scrubbing was required to mitigate the impact of SEUs on the configuration memory, the implementation of the CBMnet version 3.0 in a beam with $5 \cdot 10^6$ $\cdot$ protons $\cdot$ s^{-1} $\cdot$ cm^{-2} approved its reliability while running for several hours without hang-up. This test finally confirmed the successful application of SEU handling concepts for complex network logic in an FPGA.

Regarding the HUB ASIC, a first prototype has been designed, implemented, and submitted in the planned TSMC 65 nm LP process [83]. The manufactured chip is depicted in fig. 5.14. First laboratory tests also verified the implementation of front-end links in the 65 nm TSMC process. Unfortunately, the multi-gigabit SerDes in the first submission of the prototype could not be put into operation due to clock signal disturbance in the high frequency block. Although a close cooperation with the IITKGP during the bring-up phase tried to solve the problem, a final solution of the working group from India is still outstanding.

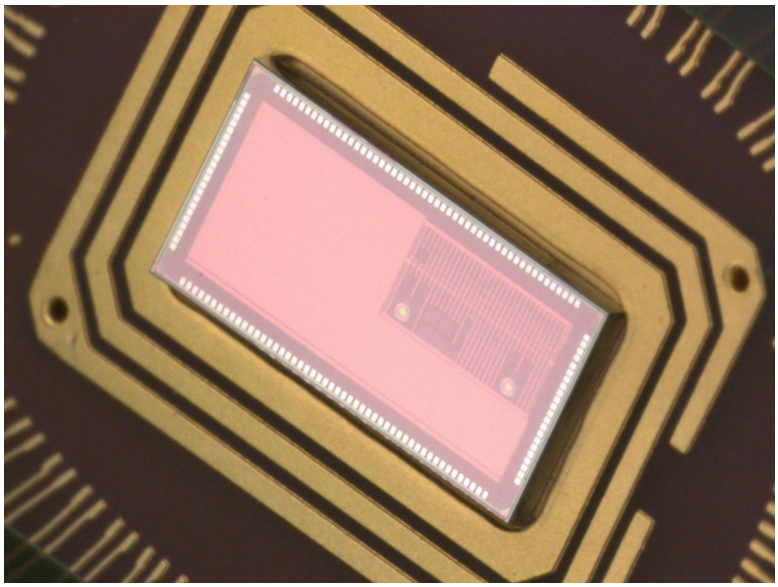


Figure 5.14: The manufactured HUB ASIC chip mounted on a PCB. The full custom physical layer block in the upper right corner can be clearly identified.

5.6 Conclusion

All CBMnet FPGA and ASIC designs have been frequently used in laboratory setups and beam times. The front-end ASICs with integrated CBMnet ease the synchronization of the detectors and the DAQ benefits from the high data bandwidth. Compared to implementations with the old CBMnet protocol, which did not support any SEU handling, the new generic cores improved the reliability of all designs and deliver all required features for the final experiment setup. Otherwise a meaningful operation of the DAQ system would not have been possible in the future. For every design, it has been shown how a initialization with synchronization and deterministic latency is assured in detail, like synchronous GTX links in the FLIB design using sophisticated initialization routines to achieve fix delay, synchronization of high-speed front-end links using Xilinx IOSelect capabilities, or special developed circuits, like in the HUB ASIC, and how DLMs are synchronized between clock domains with different speed. Finally, for the core logic of the designs, the easy integration of internal and external plug-ins has been described.

Since the back-end link SerDes in the HUB ASIC could not be taken into service successfully from the IITKGP, the decision was taken to build up competence for mixed-signal multi-gigabit SerDes design within the CAG group. The so called openMGT project is described in chapter 6.

Chapter 6

Multi-Gigabit Transmitter

While in the previous chapters only full-custom high-speed interconnects have been described, and multi-gigabit links only have been used in the context of FPGA designs and by use of external IP, this chapter gives a more detailed view on state-of-the-art multi-gigabit designs of serial interconnects, more precisely on a 20 Gbit/s SSTL transmitter. In the context of the Serializer/Deserializer development in a team of the Computer Architecture Group, the OpenMGT framework has been founded to ease the conception and verification of mixed-signal designs [68]. Regarding the transmitter, challenges are the very small time bases, signal integrity and impedance matching. A potent equalization implementation reduces data dependent jitter, which is a very important parameter to decrease the bit error rate. In the following the architectural decisions, design implementation and verification techniques are presented.

6.1 Serial I/O Challenge

A Serializer/Deserializer (SerDes) is a pair of two blocks, where the transmitter converts parallel data into a serial data stream and the receiver converts the serial stream back into parallel data. As serial interconnects usually use DDR transmission mode, in this context the definition Unit Interval (UI) means one bit time, which is equal to $^1/_{datarate}$. In the previous chapters serial interfaces and interconnects have been used intensively for network communication, but always like digital interfaces. In case of FPGA designs, built-in hard macros or dedicated serial I/O capabilities have been used, where in fact a lot of sophisticated analog circuits do their work, but the real transmission magic happens in the

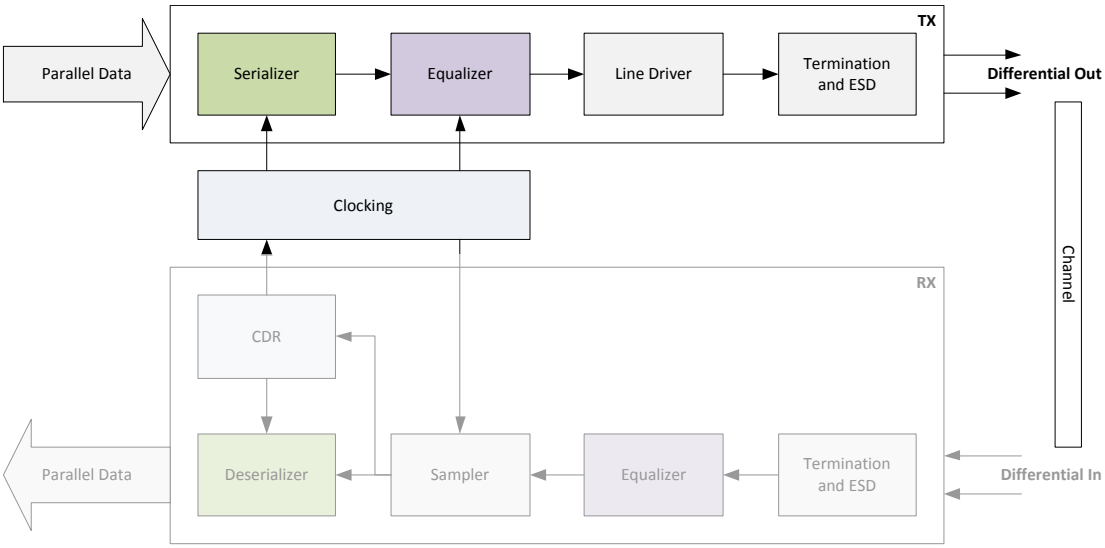


Figure 6.1: Structural built-up of a SerDes with transmitter and receiver equalization and Clock Data Recovery (CDR) circuit.

background. After successful configuration, the designer only has to deal with the digital parallel interface. For the ASIC devices, digital serializer and deserializer modules have been used, supplemented with a differential I/O buffer chain to gain the driver strength needed for off-chip signaling. Depending on the transmission channel and speed of the sent data, these simple set-ups are sufficient, but in case of multi-gigabit transmission many of challenges arise.

To overcome the speed limitation caused by skew between clock and data, the clock signal is embedded in the data stream, which requires for a Clock Data Recovery (CDR) circuit on receiver side and frequent data signal transitions in the stream to synchronize the CDR. Therefore a SerDes consists of the following functional blocks, which are depicted in fig. 6.1.

At higher frequencies the waveform is also more severely distorted by high-frequency losses, leading to attenuation and an increased Bit Error Rate (BER). This requires for equalization at transmitter, but also at receiver side to improve signal integrity and transmission reliability. Furthermore, one of the important parameters is the jitter performance of a SerDes, since the signal quality, and therefore BER, is directly affected by jitter. To reduce output jitter many design aspects must be considered, but special care must be given to reference clocks and power supply. With the smaller manufacturing sizes also the susceptibility to ElectroStatic

Discharge (ESD) damage increased. To protect the sensitive I/O amplifiers from ESD related failures, diode circuits are necessary, which put a lot of capacitive load on the line driver. Therefore the ESD circuit also needs compensation to extend the transmission bandwidth again. Regarding the final layout implementation, also many analog effects come into play, like skew and crosstalk between data and clock traces, capacitive coupling or attenuation.

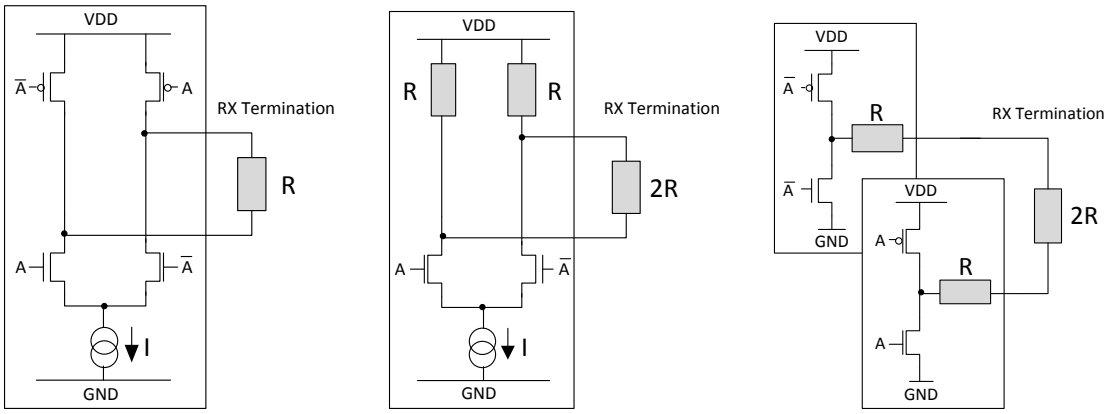
To sum up, a multi-gigabit SerDes implementation is quite more than a shift register, and to address all these topics, complex analog and digital circuits have to be developed and verified, to provide the performance also under process, voltage, and temperature (PVT) variations, for different transmission channels, and different speeds.

6.2 Architecture

While the bit rate is directly dependent of the serializer built-up and technology, the error ratio is determined by the combination of transmitter, receiver and the transmission channel in-between. Source-Series Terminated (SST) logic transmitters have been recently preferred because they deliver better power consumption properties and are compatible with many termination standards [72]. A design challenge of this type of line drivers compared to Current Mode Logic (CML) is to provide equalization and source impedance matching over process and temperature variations. Regarding the use in data acquisition systems with synchronization capabilities, a fixed latency data path is mandatory. In this section the architecture of the transmitter development is described.

6.2.1 Line Driver

As serial data rates increase and UI-widths shrink, phenomenon of signal attenuation and distortion became more considerably. In single-ended signaling a single voltage value is assigned on the wire from the transmitter and the receiver compares it with a set value or reference voltage in relation to a common potential to determine the received value. Obviously this wiring is susceptible to noise, crosstalk and electromagnetic interference. In differential signaling information is transmitted as difference between two voltages on a pair of wires, which means the

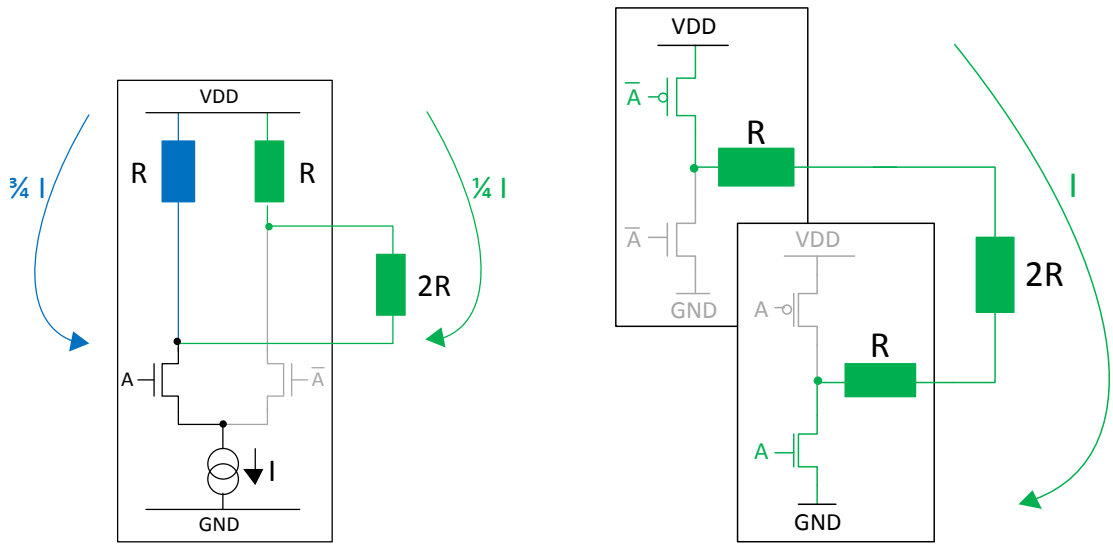


(a) LVDS driver. **(b)** CML driver. **(c)** SSTL driver.

Figure 6.2: Built-up comparison of three line drivers. While an LVDS driver speed is limited early by minimum rise times, CML and SSTL drivers can operate at much higher frequencies.

two wires are driven with complementary signals. Thereby the receiver compares two values to each other and responds to the difference between the two wires. Added noise or interference is highly rejected and therefore differential signaling allows for reliable low voltage (and thus low power) data transmission rates up to Gbit/s, and is defined in e. g. the technical interface standard for Low-Voltage Differential Signaling (LVDS) [52]. This standard specifies electrical characteristics of an interconnect with a differential line driver and a current source on transmitter side, while on receiver side the current flow generates a voltage over a termination resistor at the high impedance receiver inputs. Different logic values are created by enabling one of the two trails in the driver and thereby switching the direction of the current flow across the receiver resistor. As source current and termination resistor are nominally defined, voltage drop is determined by Ohm's law. A simplified diagram of an LVDS driver is depicted in fig. 6.2a.

As it comes to much higher data rates over several tens of gigabit per second, an LVDS driver comes to its limit, as the electronic switching characteristics allow a minimum rise and fall time of 260 ps and the rise and fall times must not extend one half of the transmitted signals UI. A modification of the LVDS driver is Current Mode Logic (CML), depicted in fig. 6.2b. It is built up with two 50 Ω parallel resistors, to bias the output transistors always in the active region, which allows very fast switching compared to LVDS or CMOS logic, where



(a) CML driver. **(b)** SSTL driver.

Figure 6.3: Comparison of CML and SSTL driver regarding the necessary current for achieving the same voltage swing at the receiver termination resistor.

transistors operate in saturation. Moreover, only NFETs are used, which show faster switching characteristics. A CML driver is enabled permanently, because one of the two branches is always active. This steady current flow on one hand gains a lower power supply noise, but on the other hand it results in a high static power consumption, independently of the signaling rate. While CML drivers were preferably used in the past, because they are fast and easy to integrate, meanwhile more energy-efficient circuits, like voltage mode drivers, are used [32]. This Source Series Terminated Logic (SSTL) driver (fig. 6.2c) can be implemented with standard CMOS logic and benefits from a power consumption proportional to the signaling rate. Moreover, for an ideal SSTL driver compared to a CML driver, only 25 % of the current are necessary to generate the same voltage level at the receiver, when using differential signaling. An analysis in fig. 6.3a and 6.3b shows, that because of the parallel resistors in the CML driver, only 25 % of the total current are used to generate a voltage signal over the termination resistor, while in the SST driver, the whole current is used [95]. Because of this circuitry, an SST driver is a more flexible solution for different standards and allows different termination options and is able to support a large range of termination voltages [57]. But beside many pros, there are also some disadvantages. Like depicted in fig.

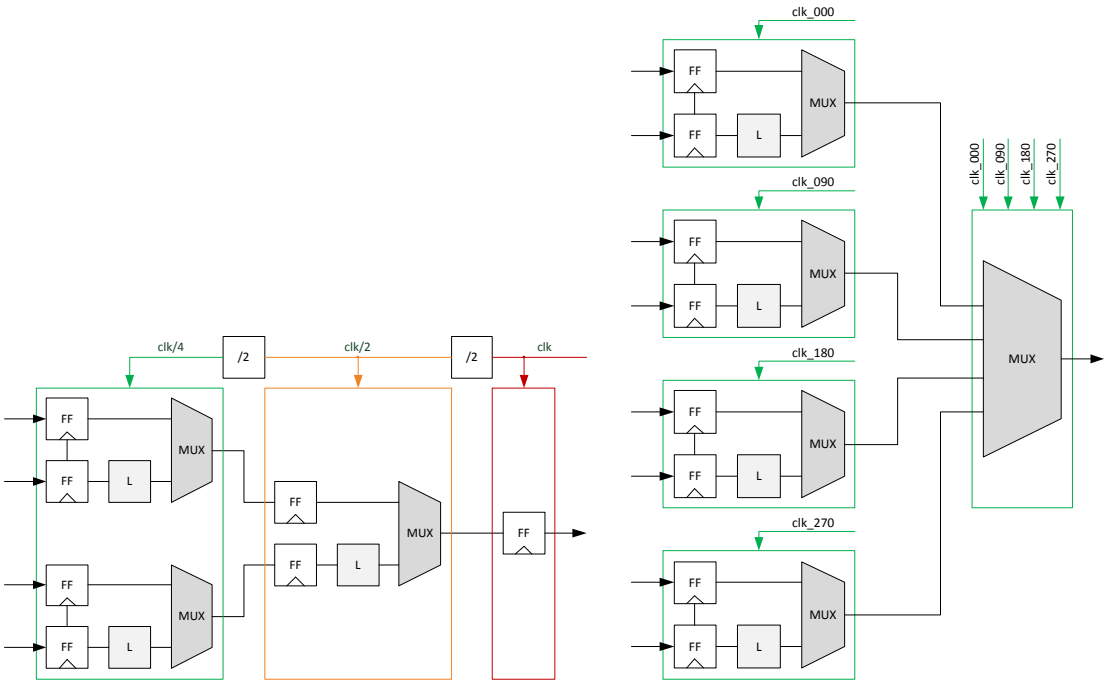
6.2c, the SST driver is only pseudo-differential and built-up with two independent segments, which are wired with opposite input signals.

Usually serial data are transmitted over wire with non-return-to-zero (NRZ) signaling, which means, that only binary values are available. Since various distortions arise at very high frequencies, like skin effect, dielectric losses and reflections, four-level Pulse Amplitude Modulation (PAM4) signaling became very attractive, as it only needs to operate at the half frequency compared to NRZ, while delivering the same data rate. On the other hand, tight linearity constraints have to be met to minimize the signal distortion when using four levels for sampling. Further, while the bandwidth is doubled, the Signal-to-Noise Ratio (SNR) is degraded to 1/3, compared to NRZ [16]. Thus, the application may be interesting for particular transmission channels, where a better performance can be achieved by using PAM4 than NRZ.

6.2.2 Serializer

In multi-gigabit designs, the serializer, which converts the parallel data into a serial stream, is always built-up with a hierarchical multiplexer tree of cascaded 2:1 stages, since a counter-based implementation would require to much logic and limit the maximum data rate. Thereby one stage performs selection and retiming of two bits, and multiplexes the output to the next stage, while doubling the rate. A figure of merit to describe a serializer's performance is the bit rate. It is directly dependent on the switching speed of the used technology and therefore the timing characteristics of combinational and sequential logic.

In a full-rate architecture the last flip-flop has to run with the desired signaling rate, which requires a very fast full-custom retiming cell and sets tough demands on the clock generation and distribution. The full-rate architecture is depicted in fig. 6.4a. Multiplexing circuits can be a bandwidth bottleneck if setup and hold times can not be met. Therefore, full-rate designs stress technology limits and the performance is only achieved at the expense of a higher power consumption [56]. An alternative can be half-rate or quarter rate architectures, where the last stage only consists of a 2:1 or 4:1 multiplexer, and no retiming is done (See fig. 6.4b). While both consume the same CV^2f power ideally, the latter relaxes timing constraints in the serializer, since clocks have to run with only a fourth of the full rate frequency. Compared to a full rate serializer they deliver better performance



(a) Full rate. **(b)** Quarter rate.

Figure 6.4: Built-up comparison of full rate and quarter rate architecture.

at lower power, while having a higher tolerance to PVT variations [37]. But these implementations raise other new challenges. As no final retiming is done, and the output multiplexer is controlled by quadrature clocks, duty cycle distortion will directly affect output signal integrity and increase deterministic jitter. Therefore they set hard requirements on the clock distribution and phase matching. With a higher fan-input MUX also the rise times extend due to more load capacity at the output node.

6.2.3 Feed Forward Equalizer

A second figure of merit to evaluate a link's performance is the Bit Error Ratio (BER), which is specified for a given SerDes and used transmission channel, and describes the number of bit errors divided by the total number of bits transmitted. To give an estimation about the expected BER, the signal quality needs to be evaluated, which can be expressed e. g. with an eye diagram, where the signal is repetitively laid several times on top of each other within one UI. A second diagram to express the link's performance is the bathtub curve, which gives the BER as function of the horizontal sampling position. Depending on the signal quality, there is a larger or smaller space in the center between two crossing points, where no transition has passed, called the eye. The larger that eye, the better the signal quality, since it can be clearly distinguished between two values. The size of the eye is negatively affected in horizontal orientation by high jitter, and vertically by a low signal-to-noise ratio. While voltage noise can lead to sampling errors when the signal vertically passes the logic threshold, jitter may lead to a horizontal shift and a signal transition across the sampling point. The sources of noise and jitter are in part deterministic and random, and can be combined by convolution to the Probability Density Function (PDF), which gives the probability that a transition varies around its expected position by a certain value. Hence, the final BER calculation underlies statistical processes.

On transmitter side, the following deterministic phenomena have an impact on the signal quality, because they all contribute to total jitter:

- **Noise**
  Variations or noise on the clock signal, ground or power supply are directly modulated to the output signal, called sinusoidal jitter, because of its form.

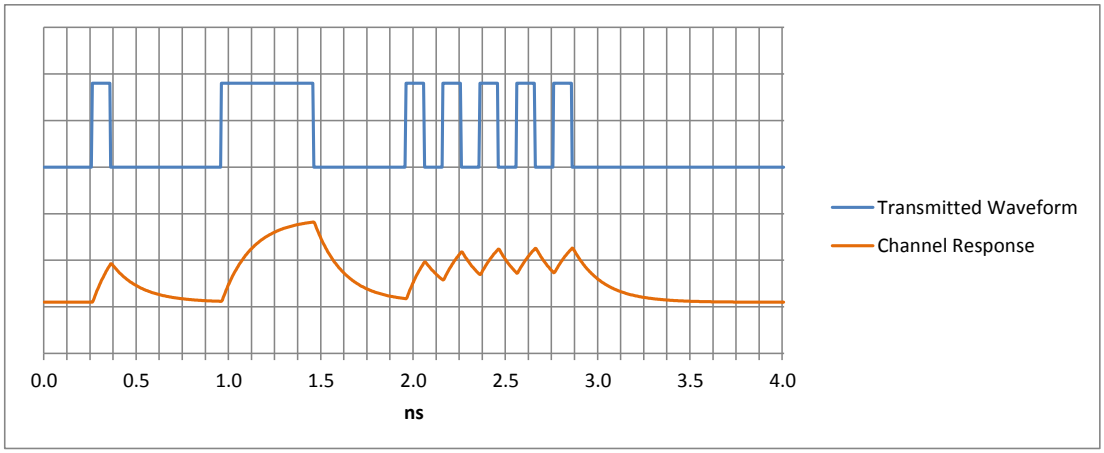


Figure 6.5: Channel response of a ideal transmitted waveform. The low-pass characteristics of the transmission channel will lead to ISI.

Since noise is a physical phenomena, it can only be reduced by layout techniques or environmental conditions.

- **Crosstalk**
  If high-speed data are sent on multiple adjacent lanes a transition coupling can lead to amplitude distortion, resulting in data dependent or random jitter at a very low statistical weight but with a high standard deviation [61]. Crosstalk can also be reduced by careful layout design and spatial considerations.

- **Duty-Cycle Distortion (DCD)**
  DCD can happen on the data as well as on the clock signal. On one hand induced by a reference clock which has a duty-cycle not equal to 50 %, on the other hand by changing switching characteristics of digital logic over PVT corners. Therefore a special duty-cycle correction circuit is used.

- **Inter-Symbol Interference (ISI)**
  Since data rates increase and transients have very short rise times, the low pass characteristics of the transmission channel came more and more into play, resulting in noticeable inter-symbol interference. The channel attenuation will blur the signal, which then interferes with subsequent symbols. The phenomena of pulse spreading is depicted in fig. 6.5, where different signal types of an ideal transmitted waveform are distorted by the

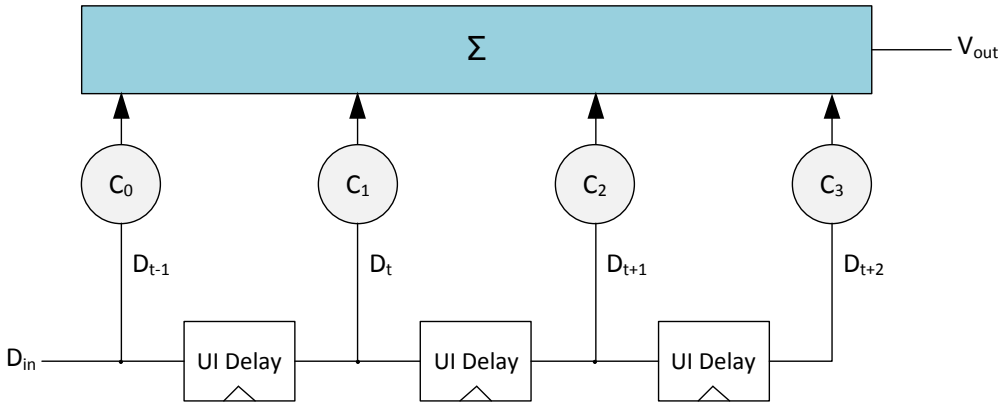
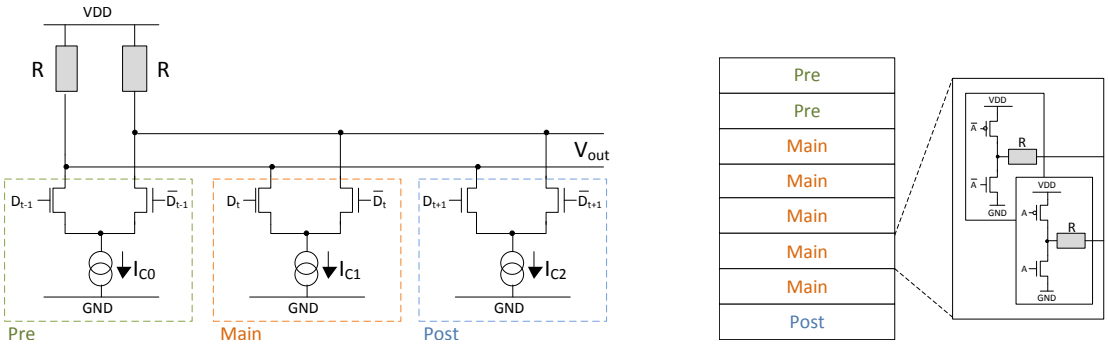


Figure 6.6: Example of a 4-tap feed forward equalizer using delayed and different weighted versions of the input signal.

low-pass transmission channel. For a clock pattern the resulting waveform of summed Single Bit Responses (SBR) is different compared to the channel response of one pulse, sent after many zero bits. Thus, ISI is data dependent and the distortion of one bit is deterministic for a defined pattern. It depends on the preceding and subsequent bits, and therefore also on the line coding, since it limits the maximum sequence length of consecutive equal symbols, like 8b/10b coding. Since ISI is a linear and time-invariant effect, an inverse filter can be added to the transmitter using pre-emphasis, to compensate the channel characteristics and mitigate ISI.

Since noise and crosstalk rather depend on layout implementation than on design techniques, the main architectural approach lies on DCD and ISI mitigation. The duty cycle balancing, which has been examined within the clock architecture development, is out of the scope of this thesis, but can be found in [68].

Regarding ISI, equalization can be used to effectively open the eye, but at decreasing overall signal swing. In a transmitter a Feed Forward Equalizer (FFE) is typically employed to compensate variations in the SBR and improve signal quality. It utilizes a Finite Impulse Response (FIR) filter, which summarizes delayed versions of the input signal with a series of different tap weights, like depicted in fig. 6.6. Thereby the emphasis of the signal is done by boosting the higher frequency content and decreasing the lower frequency content, while the transmitted power is always kept constant.



(a) CML FFE. **(b)** SSTL FFE.

Figure 6.7: Simplified diagrams of a 3-tap FFE built-up with CML or SSTL drivers.

The output signal for a 4-tap FFE is described by

$$V_{out} = C_0 \cdot D_{t-1} + C_1 \cdot D_t + C_2 \cdot D_{t+1} + C_3 \cdot D_{t+2} \tag{6.1}$$

while

$$\mid C_0 \mid + \mid C_1 \mid + \mid C_2 \mid + \mid C_3 \mid = 1 \tag{6.2}$$

with

$$C_0 \leq 0 \text{ and } C_2 \leq 0 \tag{6.3}$$

where D_x is the particular delayed digital input signal bit and V_{out} a nearly continuous value. The input data signals are called pre cursor (D_{t-1}), main cursor (D_t) and post cursor (D_{t+1}, D_{t+2}). The equalization coefficients C_n, determining the tap weight, should allow a fine grain configuration to achieve the desired equalization resolution. If there are no limitations regarding the allocation of tap weights, a 4-tap FFE can also be configured for PAM4 encoding.

If it comes to the implementation with the line driver, there are noticeable differences between CML and SSTL, like depicted in fig. 6.7. A realization of an FEE with current mode logic is quite simple and can be built-up with switched current sources, where the adjusted current in a tail represents the particular tap coefficient. The advantage is, that the impedance does not vary with the equalizer settings, and only delayed signals have to be assigned to the circuit. The downside is, that the whole circuit has to run with the full rate. In comparison, a FFE with source series terminated logic has to be built-up with several driver segments, which are individually assigned to a particular cursor, as a function of the equalizer

settings. In a SST driver equalization can only be realized with a large number of individually assignable driver segments, while the total impedance is the sum of all resistors connected in parallel. Therefore, in case of a CML driver, resistor process variations just need to be adjusted at a single point. For a SSTL driver every segment needs to be terminated by itself and must allow resistor tuning. To gain a higher emphasis resolution, segments can be weighted binary and need to be calibrated individually for every group of the same weight, while assuming no process gradient within one TX lane.

6.2.4 Segmentation and Impedance

To achieve a reasonable FIR resolution, the number of segments should not be too small. For example the PCI Express specification requests at least a resolution of 24 gradations for a transmitter [91]. NVLink even requires higher resolution of 36 gradations [70]. For PAM4 coding, also a limitless assignment of segments to cursors has to be possible. But, a higher resolution does not necessarily improve FFE capabilities and signal quality, but can be detrimental. With a rising number of segments, even more complex switching matrices are needed and wiring and clock distribution overheads increase rapidly. With binary weighting of segments, this problem can be solved, but other problems arise. The type of segment, and thereby its contribution to the total signal, is determined by its maximum current and thereby its output impedance. A segments impedance is the sum of the drivers output resistance and the termination resistor in series. The drivers output resistance has to be tunable in a wide range to compensate process and temperature variations of its own MOSFETs and the polysilicon resistor. Since all segments are wired in parallel, total impedance can only be modified by altering the number of resistors in parallel or by altering resistor values. The first can be done either by disabling whole segments, i. e. turning off both FETs within the driver, resulting in a high-Z state of the segment and thereby increasing the output resistance. The latter can be done by adding switched resistors in series. Case one is not the preferred solution, since FIR settings rely on the number of segments. If this number differs for process and temperature variations, FIR settings always need to be adapted.

To ease the design of different types of segments, they are built-up identically regarding transistor numbers and sizes, but shared resistors are used to determine

the maximum current a segment can deliver. This means, that the resistor in series is assigned to its individual segment, and all segments are connected behind the resistor. For a 4-tap FFE all segment types need to be available four times to ensure the fine adjustment for every cursor. This will lead to segments with widely varying impedance, while the tuning range of the output driver nearly stays the same. Fig. 6.8 depicts the variation of output impedance as a function of the tuning vector for several segment types (S1, S2, S4, S8). Because of the binary weighting, individual segment impedance of S2 is half the size of S1, while S4 is half the size of S2 and so on. For very high segment resistance, tuning is harder to achieve, since the absolute series resistor process and temperature variation increases. To achieve a high FIR resolution but also with a reasonably achievable tuning range, a resolution of 44 gradations has been selected and is realized by binary selection of 8 4X, 4 2X, and 4 1X segments. This leads to resistor values fulfilling the following equation:

$$8 \cdot R4 + 4 \cdot R2 + 4 \cdot R1 = 50\Omega \tag{6.4}$$

While resistances caused by additional circuitry are neglected. Regarding the impedance, this means, that these segments not only have to be tuned for the total impedance, but every segment needs to be tuned to its individual resistance value to prevent non-linear impedance and therefore non-linear FIR characteristics.

The impedance tuning is done within the driver, where stacked FETs are added above and below the output stage. To minimize wiring overhead also a binary weighting of the stacked transistors can be favored. If linear weighted resistors are added in parallel, the tuning curve shows the usual 1/R characteristics like depicted in fig. 6.8. As an alternative stacked transistors can be weighted counterbalancing the 1/R characteristics with a unary coding, but since resistance has to increase exponential not only FETs can be used but also polysilicon resistors have to be added.

6.2.5 Bandwidth Extension

Since with the ESD structures at the pad a lot of capacity is added to the output node, the low-pass characteristics of the transmission channel increase. A usual technique for compensation is the integration of a passive coil circuit at the output

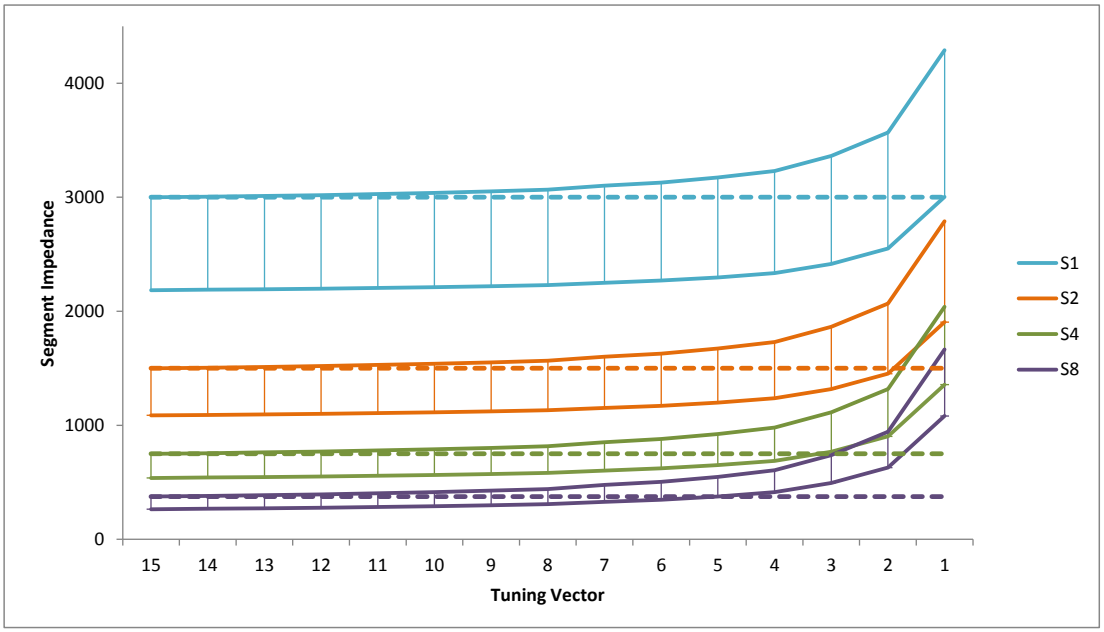


Figure 6.8: Variation of the impedance as a function of the tuning vector for four different segment types.

node for bandwidth extension. Within this context the application of T-coils has been reasonable in the past, since they offer a even all-pass filter. Unfortunately the group delay is altered and therefore leads to output signal distortion. Since these overshoots consist of high-frequency content, they are also attenuated depending on the transmission channel. The topic which trade-off between bandwidth and signal distortion is necessary for a transmission channel is still under research at the computer architecture group.

6.3 Design

The development of a SST transmitter leads to very complex wiring structures. Especially the rather digital circuits benefit from a digital verification framework to ensure functional correctness. To keep track on the large mixed-signal design a design and verification methodology has been used, which allows faster system simulation by using different abstraction level models for full custom circuits [68]. In this chapter the design methodology is described and the use of real number

models in the context of the transmitter full-custom part is elaborated to reduce simulation times and ease system verification.

6.3.1 Methodology

A SerDes implementation is an extensive development of high-speed mixed-signal circuitry, which also requires a well defined methodology for design and verification. For digital designers already many metric-driven verification frameworks exist, like the Universal Verification Methodology [6]. If it comes to analog design verification, usually only a SPICE simulation of a circuit against corners is done to meet a specification. This requires very well defined interfaces between these modules to ensure a correct behavior of the whole design later. In case of a SerDes development the complexity is even higher, because many rather small full-custom cells are combined to fulfill a superior function, like many current sources are interconnected to deliver a DAC or filter functionality. The verification of such a setup is very difficult, since the SPICE simulation time of analog circuits is magnitudes higher than a digital HDL simulation using EDA tools for digital design verification. Furthermore, while digital simulators solve logical expressions and show event-based behavior, analog simulations require a resolution of the continuous time, voltage and current values at every time step. To solve this problem, also different modeling languages are available, which allow a reasonable abstraction and thereby simulation in a shorter time frame. Nonetheless, a challenge is here to have an accurate representation of the analog circuit, and to keep models and implemented circuits in sync during the whole design process. Therefore, for the SerDes development a top-down methodology has been used, where the design is subdivided into several different cell types. They can just contain hierarchical structure information, which is described in Verilog, or synthesizeable logic, which is also described in Verilog. Further, cells modeling analog behavior can be integrated by a structural module, like depicted in fig. 6.9. The representation within such a leaf cell can contain different abstraction levels for several kinds of simulation. A functional simulation for example reveals erroneous connections and can be run very fast. Every leaf cell finally also has a transistor implementation, but the verification of the reference against the model is much easier to perform, since the same block-level verification test benches can be run on both for comparison. This methodology has been developed in

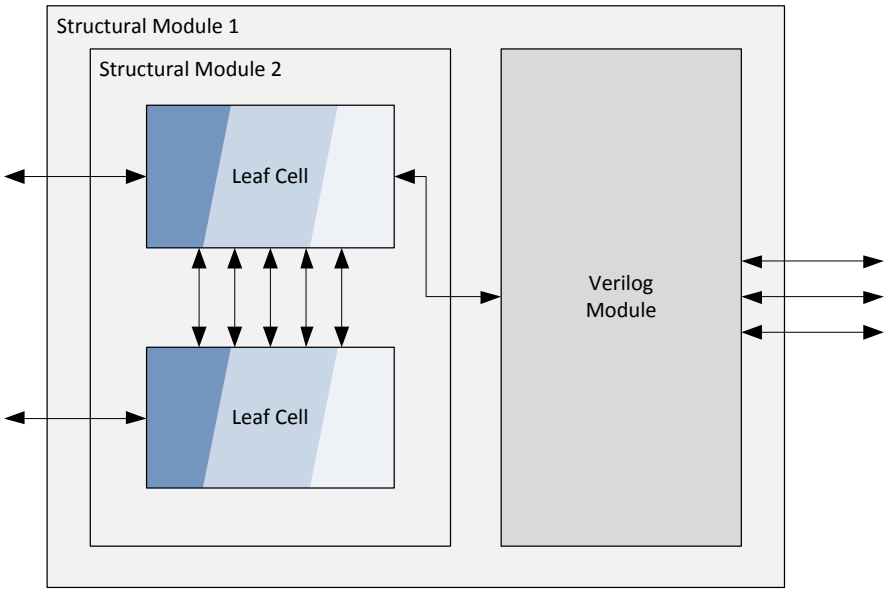


Figure 6.9: Design hierarchy example using the top-down methodology from [68]. The leaf cells can contain different abstraction level models, while structural cells are only used for connectivity and do not contain any behavioral description.

[68], and allows a very efficient mixed-signal design-flow, regarding specification, implementation and verification.

6.3.2 Behavior Modeling

The verification and simulation of mixed-signal circuits is a challenging task, because transistor-level SPICE simulations are too time consuming on system level. Especially if all possible analog and digital interactions need to be covered. Thus, the creation of simulation models, which describe the analog circuit behavior, is mandatory. Unless they provide a small simulation speed-up, languages like Verilog-A are not suitable for mixed-signal simulation, because of the continuous-time modeling semantics and the need for a SPICE simulator. With Verilog-AMS or VHDL-AMS also event-driven circuit characteristics can be described. Moreover, compared to Verilog, with the *wreal* type discrete floating-point real-numbers can be used to represent voltage levels but still allow a use of the digital simulation environment. Therefore real number models are the most effective way to abstract

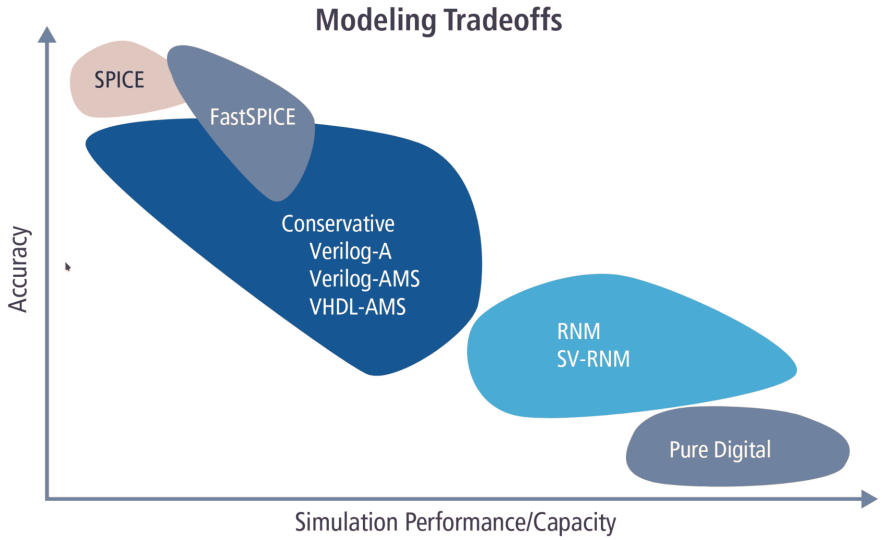


Figure 6.10: Model accuracy versus performance gain for mixed-signal simulation [15].

analog circuit behavior for a complex design simulation and verification, like depicted in fig. 6.10. To overcome the limitation that the wreal type can only carry one RNM value, SystemVerilog can be used. With the latest standard (IEEE 1800-2012 LRM) [1] also the use of User-Defined Types (UDTs) and User-Defined Resolutions (UDRs) is possible, which allows for multi-value nets and user-defined records. UDRs are functions which specify how the UDTs are combined and resolved, especially in case of multiple drivers.

To model the FIR behavior of the transmitter, where many driver segments are connected together depending on the emphasis settings, the real-number modeling capabilities of SystemVerilog have been used. The electrical package delivered within the Cadence Design Tools provide a UDT called EEstruct, which consists of three real values for voltage, current and resistance [19]. This allows to represent the weighted impact of a segment correctly, while current and impedance information are passed on. At the output node, where the segments are connected together, the resolution function calculates the final values conform to Kirchhoff's laws. Special adaption of the package was required regarding the resolution functions since by default it can not be distinguished between current directions, but this feature is necessary for the FFE modeling. To verify the functional and electrical behavior of the whole mixed-signal circuit, like emphasis and impedance,

the output voltage levels are compared to a reference transmitter which sends the same pattern, while assertions monitor the impedance.

As a simple example, the EEnet description of a driver segment in SystemVerilog is presented. The type of segment is determined by the subsequent added resistor in series.

```
module MGT_TX_OBUF_DRIVER #(
    parameter                                NUM_DRV_TUNE_BITS = 4
)(
    input wSup                               VDD,
    input wSup                               VSS,
    input wire                               IN,
    input wire  [NUM_DRV_TUNE_BITS-1:0] CFG_DRV_TUNE_P,
    input wire  [NUM_DRV_TUNE_BITS-1:0] CFG_DRV_TUNE_N,
    output TX_EEnet                          OUT
);

localparam  RS_PFET             = 1500.0;
localparam  RS_NFET             = 1500.0;
localparam  I_PMOS              = 2275e-7;
localparam  I_NMOS              = 2275e-7;

real        r_pfet, r_nfet;
real        vout, rout, iout;

wire        driver_idle;
wire [3:0]  tune_pfet;

assign tune_pfet = ~CFG_DRV_TUNE_P;

//detect electrical idle condition
assign driver_idle = ((CFG_DRV_TUNE_N == {NUM_DRV_TUNE_BITS{1'b0}}
                       &&
                       CFG_DRV_TUNE_P == {NUM_DRV_TUNE_BITS{1'b1}})
                        || (|CFG_DRV_TUNE_P===1'bx)
                        || (|CFG_DRV_TUNE_N===1'bx)) ? 1'b1 : 1'b0;

always @(*) begin

    //calculating output resistance depending on tune vector
    r_nfet = (RS_NFET / CFG_DRV_TUNE_N);
    r_pfet = (RS_PFET / tune_pfet);

```

```
38      //assigning output voltage, current and resistance values
39      casex ({{IN, driver_idle}})
40          2'b10: begin
41              vout <= VDD.V;
42              rout <= r_pfet;
43              iout <= I_PMOS;
44          end
45
46          2'b00: begin
47              vout <= VSS.V;
48              rout <= r_nfet;
49              iout <= -I_NMOS;
50          end
51          2'bx1: begin
52              vout <= 'Z;
53              rout <= 'Z;
54              iout <= 0;
55          end
56      endcase
57  end
58
59  assign OUT = '{vout, iout, rout, 0};
60
61  endmodule
```

In this case on driver segment consists of an input for the digital values, two different tuning inputs for PFETs and NFETs, and the UDT output TX_EEnet, which delivers voltage, current and resistance information. Depending on the tuning vector r_nfet, respectively r_pfet are calculated, which together with the subsequent termination resistor determine the output impedance of the segment. Finally the individual voltage contribution is calculated with a resolution function, depending on current and resistance. If all tuning bits are unset, the segments output switches to high-z, which is equitable to electrical idle.

6.4 Implementation

The implementation of the transmitter is done using the benefits from the top down methodology and the real number based modeling of the FIR filter. Speed limitations from the the design kit technology require the division into a full digital

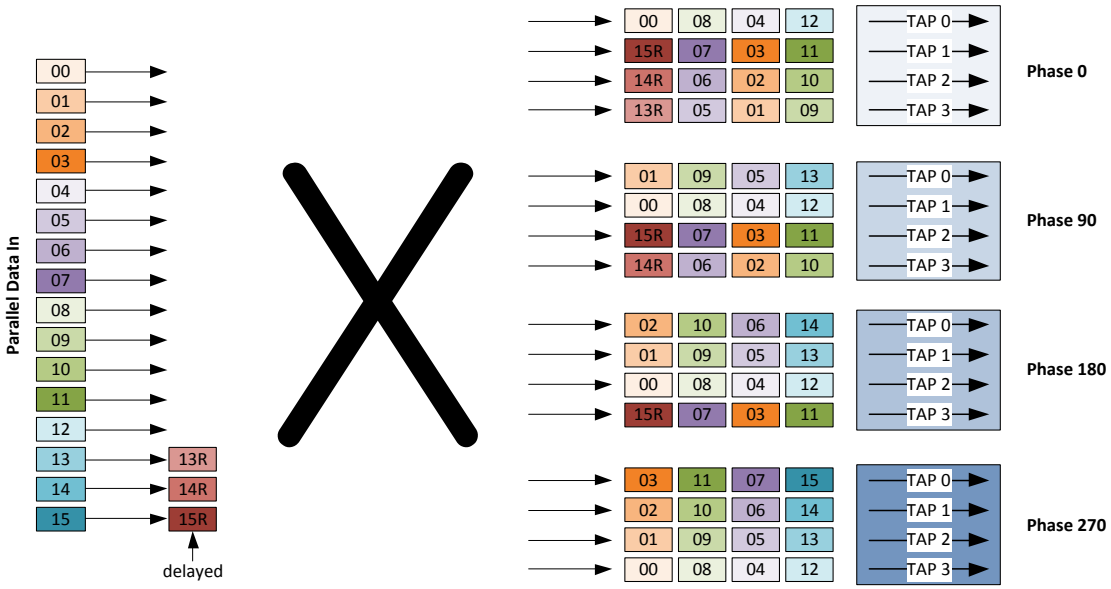


Figure 6.11: The switch matrix connects the 16 bit parallel input data to the four taps, while every tap contains four different phases.

semi-custom design part with synthesizable logic, consisting of a complex switch matrix and a multiplexer tree to support a segmented driver, and a full-custom multi-segment output buffer. In this section the SSTL design structure is described and techniques are elaborated how data dependent jitter is reduced and impedance tuning is realized. A big advantage of this mixed-signal design is, that all complex circuitry could shoved easily into the pure digital part since the structural verilog files interconnecting the full custom modules.

6.4.1 Switch Matrix

To feed the segments of the FIR with the correct bits, a switch matrix has been described to distribute the 16 bit parallel input data to the four phases of the quarter rate architecture (000, 090, 180, 270), while providing the four FFE taps (PRE, MAIN, POST1, POST2) at the same time. This means, that the signal bunch for one phase consists of signal groups for the four taps. It can be recognized, that in the phase 90 group subsequent bits of the phase 0 group are used and so on. The input bits 13, 14 and 15 are additionally registered, because the delayed

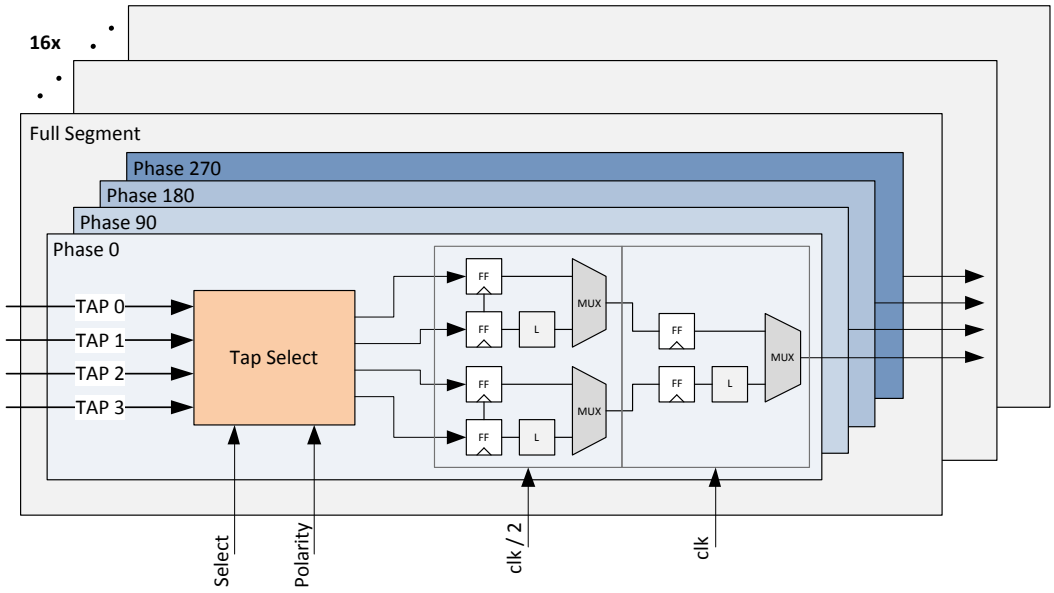


Figure 6.12: The digital segment structure of MUX trees.

values are also required. The data are selected in a manner, that by multiplexing them onto one bit stream, the correct sequence is obtained.

6.4.2 MUX Segments

The second unit in the digital part consists of 16 MUX segments, while every full segment is subdivided into four phases. This structure is depicted in fig. 6.12. The tap selection the individual assignment of a segment to one of the four taps, depending on the FIR settings. If the cursor should contribute with negative weight, also data inversion is possible by configuration. The multiplexer tree serializes the particular tap data into the quarter rate data streams, which are required by the analog core. The data are re-timed by flip-flops, while in one path the data are also saved with a latch to ensure a timing-correct selection by the subsequent multiplexer. The first MUX stage is also switched with a divided clock compared to the second MUX stage. Thereby the data rate is doubled. To ease the clock distribution within the digital part, the sequential logic runs with the same clock and therefore the output data of all segments are in phase. Since this part is fully described in digital logic, the fulfillment of setup and hold times is ensured by constraints and the semi-custom design flow timing analysis.

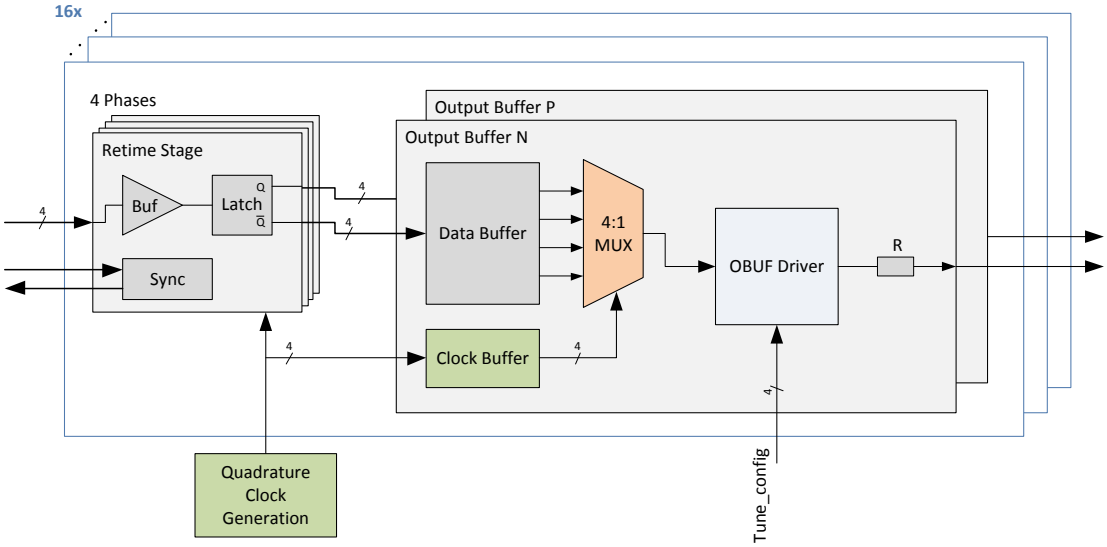


Figure 6.13: The full-custom core driver, consisting of a retime stage and the pseudo-differential output buffers. Segment impedance is determined by the source series resistor.

6.4.3 Core Driver

The core driver represents the full-custom top module, which is subdivided into retime stages and the pseudo-differential pairs of output buffers. A retime stage takes the data from the semi-custom digital design interface and synchronizes them into the full-custom domain. Thereby the four data phases of the quarter rate are retimed with a particular clock to achieve enough positive slack to be safely selected by the output multiplexer. Additional data and clock buffers are needed, since input loads of the MUX and driver are very high compared to the standard cell logic flip flops used in the digital domain. The subsequent resistor determines the type of segment and its contribution factor to the output signal. Since this part of the PHY is designed without any simulation models for timing analysis, the fulfillment of setup and hold times needs to be verified for every corner. Moreover, since the transmitter also has to be able to run with different speeds, a fixed time relation between the digital part and the full-custom part is not possible. For that reason, a synchronization stage is used to shift the data in the digital domain to the correct position for safely sampling the data in the full-custom domain.

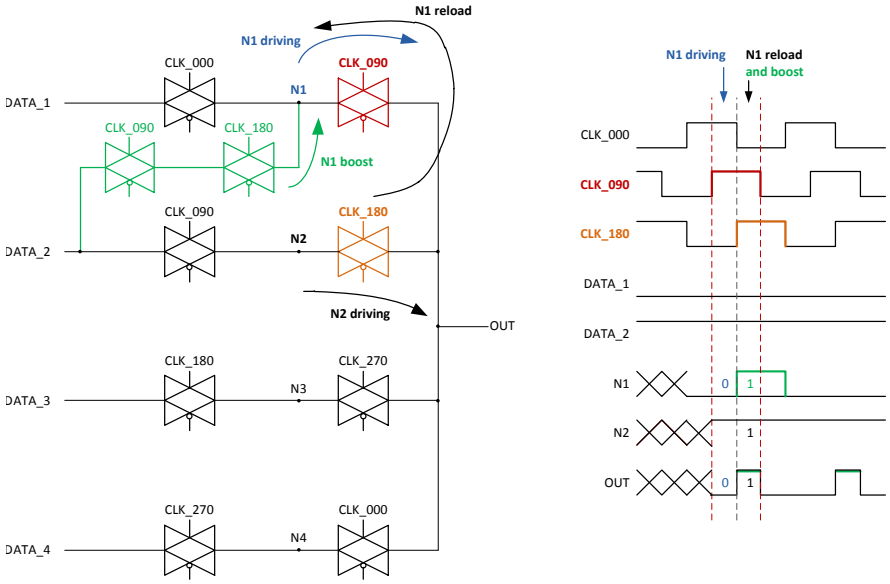


Figure 6.14: The 4 to 1 multiplexer implemented with transmission gates and using feed-forward charge injection to reduce the effective MUX output load.

The 4:1 multiplexer is built-up of two transmission gates in a row in the data path for every phase. Every transmission gate consists of two MOSFETs and needs complementary clocks. The two transmission gates in one path run with different clocks, with a phase difference of exactly one UI. Thereby the particular data value is sliced and driven to the output node of the MUX. A general problem of a quarter rate MUX is its very high output capacity. This leads to a maximum slew rate of 15 ns for the given process in the case that all other transmission gates have a different digital value and must be reloaded, even if no further output load is connected. Also a modification in size, resulting in higher currents, does not affect rise times, since capacities also increase in the same magnitude. Moreover, this estimation is only valid for ideal settings, like ideal clock pulses and power supply, but under meaningful simulation conditions, rise times are far more worse and have strong impact on the subsequent OBUF driver and the output signal. When using a transmission gate MUX also another problem comes into play, like depicted in fig. 6.14. Since the second pass gate in a row only drives a value for one UI, but is enabled for two UIs, the actual output load is even more big. If a clock pattern is sent, an output transmission gate also has to reload the internal node of the previous data path. Since the load can not be reduced, but the problem

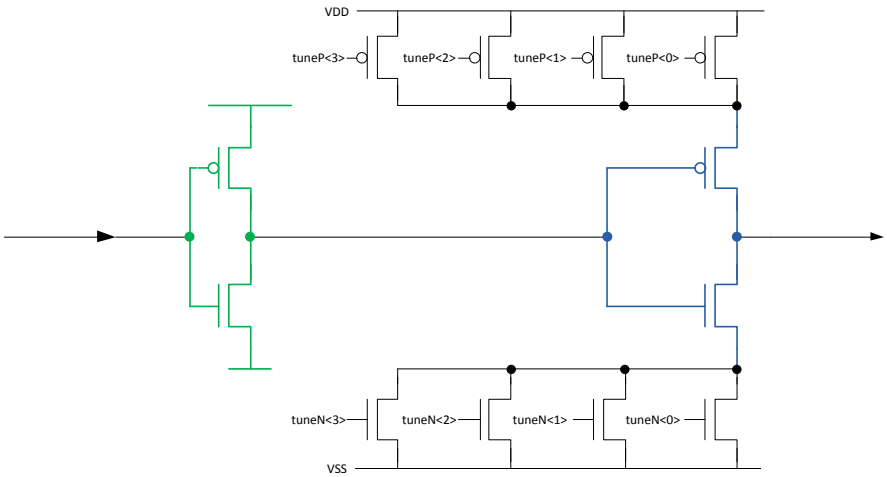
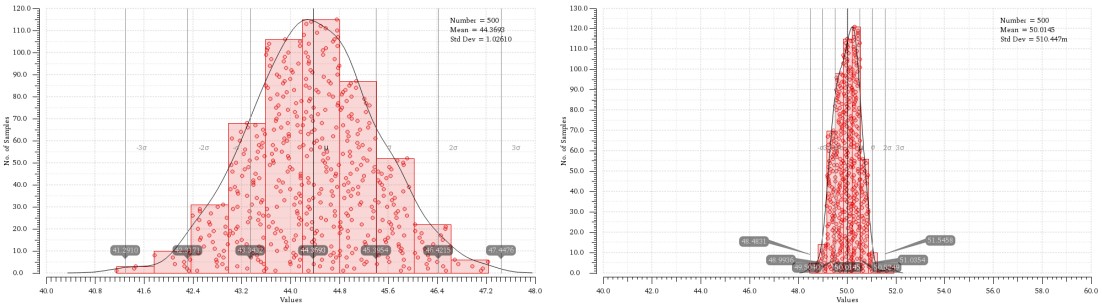


Figure 6.15: The output buffer driver consists of a pre driver (green), the main driver (blue) and the stacked impedance tuning transistors above and below.

only exists if a reload is necessary, a feed-forward charge injection technique has been used to simultaneously recharge the intermediate node. Thereby for every particular data path a second feed forward path is added using subsequent bit information to reload the intermediate node if necessary. In fig. 6.14 the principle of charge injection is explained for the path DATA_2. While CLK_090 activates the second pass gate in path DATA_1, only the first half of the clock period the intermediate node N1 drives the output node. During the second half, the pass gate is sill open, but N1 needs to be recharged from N2. To reduce the apparent load, a feed-forward charge injection path has been added to boost the same value to node N1 and improve output rise times. This technique has also been proposed in [37] to reduce output ISI by 77 %.

The subsequent output buffer driver consists of a predriver and main driver stage in series to improve rise times and deliver enough current to driver output loads. Although a larger buffer chain could further improve rise times, there is also a risk of increased duty cycle distortion and phase differences between the segments, since no further retiming is done. Impedance tuning is realized by two rows of four stacked transistors in parallel with binary weighted output resistances above and below, like depicted in fig. 6.15. By turning off tuning transistors it is possible to increase the output resistance of the output buffer driver. Since there are differences in the switching characteristics of NFETs and PFETs, they are tunable



(a) Untuned. **(b)** Tuned.

Figure 6.16: The result of a Monte Carlo simulation over process and temperature variations before and after the tuning mechanism.

independently.

6.5 Simulation and Verification

Besides the decision for the general structure of the transmitter, the dimensioning of all full custom circuit parts is required to meet a specification that supports multiple standards regarding line codings and transmission channels. This specification mainly results from three settings. The source impedance, the line rate and the equalization capabilities. While the first needs an accurate tuning mechanism, that has to be verified for all process and temperature corner cases, the line rate defines the minimum rise times necessary for the output driver. Equalization can be adapted in the range of the available FIR resolution. As already described in 6.2.4 the minimum granularity is $1/44$ and a free assignment of segments to all cursors is possible. To assure linear steps the current of one segment has to be in a fixed relation to the other segments. Since all segments are built up identically and the type is only determined by the subsequent series resistor, linearity is achieved by accurate individual segment impedance tuning.

So, the source impedance needs to be tuned accurately for every process/temperature setting, since a mismatch will cause reflections and lead to additional signal disturbance. To verify a correct tuning mechanism and that all corner variations can be compensated a special test bench with four tests has been set up. The first three tests enable only the segments of one segment type and sweep the tuning

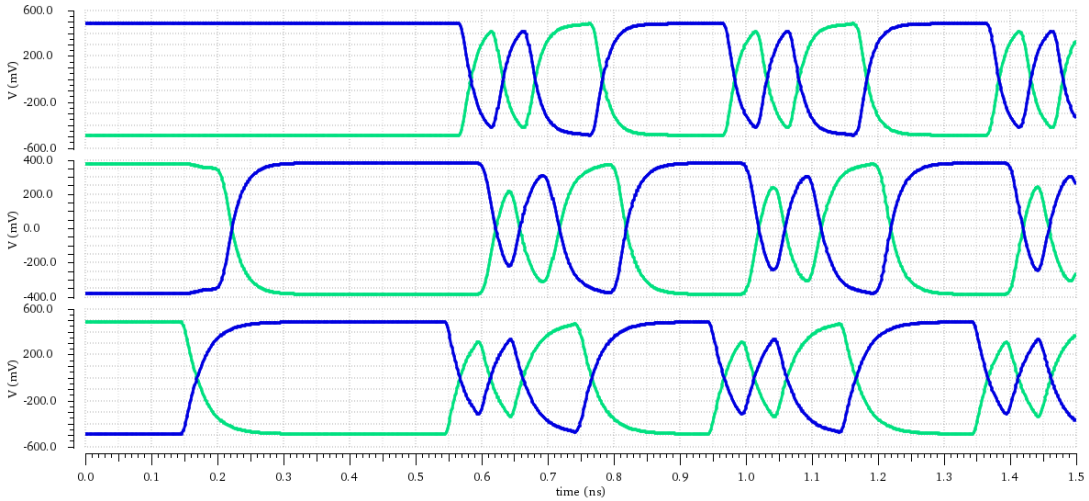
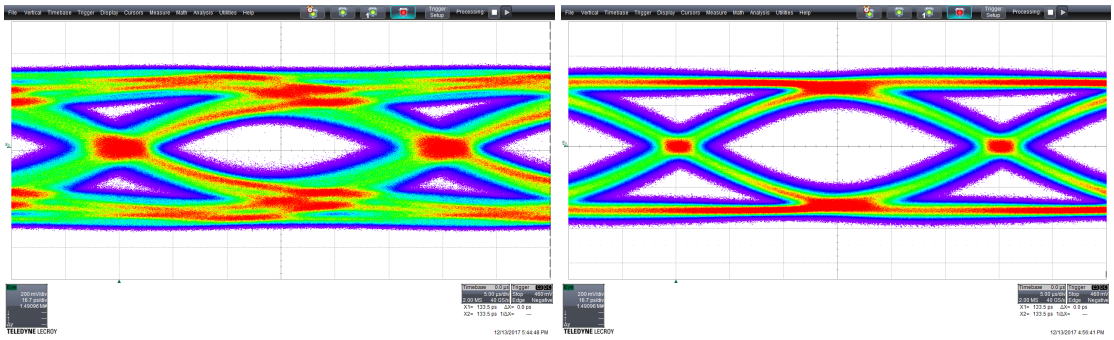


Figure 6.17: The post layout RC extracted core driver differential 20 Gbit/s output signal against a 50 Ohm receiver over three corner cases (from the top: typical, RC worst, RC best).

code while the output impedance is compared against the nominal resistance. The crossing point provides the optimum setting for every segment type group. In the last test these codes are assigned to the segment types and the final output impedance is measured. The results of a Monte Carlo simulation in fig. 6.16 before and after the tuning mechanism show how process and temperature variations are balanced. Before the tuning mechanism impedance varies from 42.3 to 46.4 Ω, while afterwards there is only a variance of ± 1 Ω for a ± 2 σ normal distribution and an expected value of 50.02 Ω.

An overall verification of a transmitter is only meaningful for in the context of a whole SerDes interconnect, consisting of a transmitter, receiver and a transmission channel, including all capacitive and inductive elements. The dielectric and conductor losses depend on the wires, electrical or optical transmission mediums, backplanes, connectors, package, bonding type, PCB and so on. For this work the assumption has been made, that it has to be distinguished between the output load seen by the driver and the subsequent transmission channel. Regarding the dimensioning of the line drivers they should at least have a reasonable slew rate to reload the output capacitance with the target frequency. The target rate of 20 Gbit/s defines a UI width of 50 ps. To achieve full signal swing, usually the rise times should be better than 50 % of one UI, respectively 25 ps in all corners. This



(a) FIR off **(b)** FIR on

Figure 6.18: Eye measurement of the transmitter output running a PRBS-31 pattern at 10 Gb/s with disabled and enabled equalization.

sets hard requirements on the output driver and all previous circuitry while trying to minimize the power consumption. Furthermore, the low pass characteristics of the output capacitance can also be compensated by appropriate equalization settings. In a first approximation the final dimensioning has been determined by post layout simulations. For this the whole full-custom transmitter circuit has been RC extracted, while most capacitance is added by the output driver node itself, the series termination resistors, the connecting metal traces, the ESD diodes and the pad. The output signals of the simulation are depicted in fig. 6.17 for three different corners. Differences regarding the rise times are traced back to the fact, that different supply voltages are used and different driver tuning settings. As already mentioned, the overall verification is very difficult and only meaningful in the context of a whole SerDes interconnect. The elaborated performance analysis and verification, especially of the FIR filter, has been done using a budgeting procedure, in which the impact of is estimated by special algorithms for the developed real-number models. This budgeting procedure and the related algorithms are presented in [94]. For the results also be kindly referred to this thesis. In addition to the data path related challenges of a transmitter, also the clock sources and the clock distribution are very complex circuits which take much design effort. The research and development of these topics in the context of the OpenMGT project have been examined in [68].

Since the manufactured ASIC just arrived during the completion of this thesis, some first measurements could be made. An eye measurement of the transmitter output signal, running at 10 Gb/s a PRBS-31 pattern is depicted in fig. 6.18.

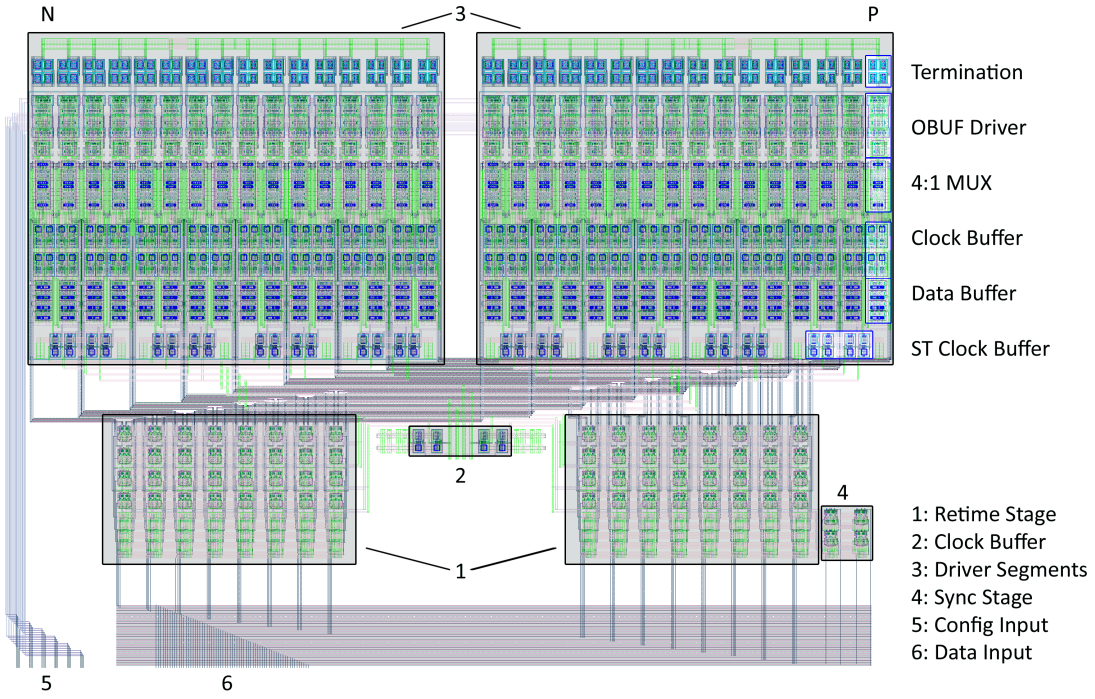


Figure 6.19: The core driver full-custom layout built-up of the retime stage and the two pseudo-differential output buffers.

While in the first picture (a) the FIR is disabled, and all segments drive the same value, in the second picture (b) the FIR is enabled, resulting in significantly more eye-margin. The channel attenuation is higher at the estimated data rate of 20 Gb/s, thus a prospective comparison at that speed will show even more contrast.

6.6 Layout

For the design part of the full-custom circuits also special care has to be taken regarding the layout. There are three main aspects which needs to be considered carefully in a segmented design of SSTL drivers. First, all segments contribute to the output signal and to minimize skew between single units, a symmetric structure is favorable, especially after the last retiming stage. Second, special care must be taken regarding the clock distribution, which needs to be arranged, so that all endpoints receive a synchronous signal. Therefore not only the trace

length should be equal, but also capacity and resistance, since variations modulate on the output signal. Last, same requirements apply for the design of the power grid. As already described in 6.4.3 the core driver is divided in a retime stage and the pseudo differential output buffers, built-up of segments, like depicted in fig. 6.19. Every segment is placed in a column consisting of the termination resistor, the OBUF driver, the 4:1 MUX, and individual data and clock buffers. Every group of the same segment type also obtains a common clock buffer. The wire structure for clock and data has been chosen to gain connections of equal length and capacity. For further capacity reduction and to reduce cross coupling, all traces of the same metal layer are spaced by at least 100 % of their own width and no metal trace on the directly upper or lower layer is placed in the same location. Test simulations with the design kit confirmed the application of the mentioned layout techniques.

6.7 Conclusion

In this chapter the design and verification of a multi gigabit transmitter physical layer has been described, using digital-like design techniques for an complex mixed-signal development flow. A great advantage regarding simulation times and system verification could be gained from the use of real-number models of the full-custom part. The requirements and technical background have been explained and the architectural design decisions have been elaborated. Main focus laid on the fact, that the implementation supports many standards and also fulfills the demands for the use in state-of-the-art and future data acquisition systems, like deterministic latency. Within the OpenMGT project, in conjunction with the receiver, also deterministic word alignment of serial data is possible, which assures deterministic latency for an network interconnect and this is mandatory for the accurate synchronization of front-end electronics over serial links. A tuning mechanism allows impedance balancing without affecting equalization settings. Finally, a first prototype ASIC to prove the concept has been submitted in the 28 nm TSMC HPC+ process and first results of the impedance tuning mechanism and segmented FIR implementation are available for 10 Gbit/s data rate.

Chapter 7

Conclusion and Outlook

Digital data acquisition systems based on state-of-the-art serial multi-gigabit interconnection networks are extremely important to realize complex scientific data-driven experiments, especially nowadays, since serial links allow the required high-speed data transmission across various media types with different speeds and over long distances. Unfortunately, neither pure SerDes solutions, nor most network protocols deliver all capabilities for the further demanding challenges of modern particle physics experiments in terms of reliability and synchronization. This point is particularly emphasized when it comes to the great effort and designing time which is put in the development of custom DAQ solutions in projects, which can be compared to CBM.

This work presented a complete and very reliable unified custom network protocol, for the physical, data link, and network layer of the CBM data acquisition system, which is partially exposed to heavy ionizing radiation.

As already mentioned in the introduction, due to the CBM detector upgrade and the planned implementation of trigger and synchronization logic in the front-end area, which is controlled by the network protocol; the earlier development came up against its limits regarding reliability, flexibility, and resource consumption.

This in turn led to the suggestion of a new network protocol solution for the CBM DAQ, based on a careful analysis and evaluation of the former implementation with increased demands on the CBM detector setup, since otherwise a systematic continuation of the experiment would not have been possible.

In this context completely new developed CBMnet cores and modules have been proposed, while the concept of three traffic classes and the use of deterministic

latency links for synchronization has been maintained and extended. Further, the successful development and integration of several types of transceivers, from high-speed LVDS to multi-gigabit SSTL circuits, with the network protocol has been shown in different front-end, intermediate HUB and back-end designs. The generic IP cores are highly configurable in terms of speed, bandwidth, number of interconnects and hierarchy, which makes them perfectly suitable for the flexible use in many FPGA and ASIC devices.

Since usual radiation mitigation techniques like TMR and scrubbing could not be used due to resource and space limitations, an adapted radiation mitigation strategy was invented, providing sophisticated algorithms for functional units against single and multi bit upsets, caused by ionizing radiation. The algorithms have been implemented into hardware modules and guarantee a continuous operation by avoiding data loss and dead times through reset procedures. Simulation and live application of the new network protocol implementations in laboratory tests and beam times showed a significant increase in reliability regarding induced soft errors, while the final resource consumption could even be slightly reduced, compared to the earlier development.

Further a set of useful tools has been invented to ease the built-up and testing of larger read-out setups. A diagnostics core provides assertion-based system and error information, which in larger setups can be obtained for statistic or debug purposes to identify immediately malfunctions or hardware defects. With the SEU fault injection simulator tool, which complements state-of-the-art simulation and verification tools, the testing of hardware design setups can be easily carried out in advance to be more efficient, save time and debug resources. It should be mentioned that meanwhile a similar tool for SEU injection is also available within the Cadence Incisive tools [2].

The very challenging tasks of developing, configuring and integrating SerDes technology in data acquisition networks has been elaborated in detail regarding multi-gigabit speed at least for the transmitter part in the context of the openMGT project. A full-digital architectural concept brings many advantages for future customization, verification, and porting to other process technologies. Furthermore, special attention is paid to the development of deterministic latency concepts for these circuits, facilitating a global clock distribution and ensuring the precise time synchronization of a whole read-out tree to less than 2 ns difference with a

deterministic link latency better than 20 ps, which has been verified successfully for the relevant CBMnet device prototypes in simulations and live applications.

It can be therefore concluded that the presented concepts and designs fulfill all requirements of the new experiment detector setup, which is why they could also be used in the final application.

In the future, improvements regarding power efficiency should be examined, since this will also become increasingly important in scientific experiments and has not yet been covered within this research. Especially the power consumption of serial interconnects gets more relevant as they demand lots of energy from the overall power budget of an integrated circuit. This applies in particular to equalization settings for channels with high insertion loss, since strong equalization leads to high cross currents within one output buffer. First architectural suggestions addressing this problem can be found in [38]. Furthermore, the development of efficient T-coil circuits to gain a bandwidth extension is still under research within the computer architecture group.

The new CBMnet IP cores and designs are available in the global CBM repository, allowing also other detector groups to evaluate its capabilities for their read-out setups. First implementation tests from the HADES group have been done on Lattice FPGAs, but it is likely that also other teams integrate features as the cores are highly configurable and widely usable within a data acquisition system.

List of Figures

List of Tables

Bibliography

[1] IEEE Standard for SystemVerilog–Unified Hardware Design, Specification, and Verification Language. *IEEE Std 1800-2012 (Revision of IEEE Std 1800-2009)*, pages 1–1315, Feb 2013.

[2] Incisive Functional Safety Simulator. Cadence, Datasheet, 2015.

[3] Continuing Experiments of Atmospheric Neutron Effects on Deep Submicron Integrated Circuits. *White Paper: Xilinx FPGA Families*, 2016.

[4] Device Reliability Report. *Xilinx*, 2016.

[5] European Organisation for Nuclear Research (CERN), Geneva, Switzerland. `http://www.cern.ch`, 2017. [Online; accessed 19-July-2017].

[6] IEEE Standard for Universal Verification Methodology Language Reference Manual. *IEEE Std 1800.2-2017*, pages 1–472, May 2017.

[7] Soft Error Mitigation Controller v4.1. *Xilinx LogiCORE IP Product Guide*, PG036, 2017.

[8] Website Facility for Antiproton and Ion Research in Europe GmbH. `http://fair-center.eu`, 2017. [Online; accessed 19-July-2017].

[9] Website GSI Helmholtzzentrum für Schwerionenforschung GmbH. `http://gsi.de`, 2017. [Online; accessed 19-July-2017].

[10] F. Abate, L. Sterpone, C. A. Lisboa, L. Carro, and M. Violante. New Techniques for Improving the Performance of the Lockstep Architecture for SEEs Mitigation in FPGA Embedded Processors. *IEEE Transactions on Nuclear Science*, 56(4):1992–2000, Aug 2009.

[11] Norbert Abel, Frank Lemke, and Wenxue Gao. Design and implementation of a hierarchical DAQ network. March 2008.

[12] G. Anelli, M. Campbell, M. Delmastro, F. Faccio, S. Floria, A. Giraldo, E. Heijne, P. Jarron, K. Kloukinas, A. Marchioro, P. Moreira, and W. Snoeys. Radiation tolerant VLSI circuits in standard deep submicron CMOS technologies for the LHC experiments: practical design aspects.

[13] Tim Armbruster. *SPADIC-A self-triggered detector readout asic with multichannel amplification and digitization.* PhD thesis, 2013.

[14] Tim Armbruster, Peter Fischer, and Ivan Perić. SPADIC—A self-triggered pulse amplification and digitization ASIC. In *Nuclear Science Symposium Conference Record (NSS/MIC), 2010 IEEE*, pages 1358–1362. IEEE, 2010.

[15] Sathishkumar Balasubramanian and Pete Hardee. Solutions for mixed-signal soc verification using real number models. *Cadence Design Systems*, 2013.

[16] M. Bassi, F. Radice, M. Bruccoleri, S. Erba, and A. Mazzanti. 3.6 a 45gb/s pam-4 transmitter delivering 1.3vppd output swing with 1v supply in 28nm cmos fdsoi. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 66–67, Jan 2016.

[17] R. C. Baumann. Radiation-induced soft errors in advanced semiconductor technologies. *IEEE Transactions on Device and Materials Reliability*, 5(3):305–316, Sept 2005.

[18] T. K. Bhattacharyya, A. Halder, I. Som, et al. Radiation-tolerant 2.5 GHz Clock Multiplier Unit and 5 Gbps SERDES. *CBM Progress Report*, page 85, 2012.

[19] John Brennan, Thomas Ziller, Kawe Fotouhi, and Ahmed Osman. The how to's of advanced mixed-signal verification. *Design and Verification Conference and Exhibition*, 2015.

[20] Niels Burkhardt. *A Hardware Verification Methodology for an Interconnection Network with fast Process Synchronization.* PhD thesis, Universität Mannheim, 2012.

[21] A Caratelli, S Bonacini, K Kloukinas, A Marchioro, P Moreira, R De Oliveira, and C Paillard. The gbt-sca, a radiation tolerant asic for detector control and monitoring applications in hep experiments. *Journal of Instrumentation*, 10(03):C03034, 2015.

[22] HyperTransport Consortium. The HyperTransport 3.1 specification. 2008.

[23] J De Cuveland, V Lindenstruth, CBM collaboration, et al. A First-level Event Selector for the CBM Experiment at FAIR. In *Journal of physics: Conference series*, volume 331, page 022006. IOP Publishing, 2011.

[24] C. de Jesús García Chávez and U. Kebschull. Design and evaluation of an fpga online feature extraction data pre-processing stage for the cbm-trd experiment. In *2016 IEEE-NPSS Real Time Conference (RT)*, pages 1–3, June 2016.

[25] V. Dumitriu, L. Kirischian, and V. Kirischian. Decentralized run-time recovery mechanism for transient and permanent hardware faults for space-borne FPGA-based computing systems. In *2014 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pages 47–54, July 2014.

[26] S. Erdogan, J. L. Gersting, T. Shaneyfelt, and E. L. Duke. Using FPGA technology towards the design of an adaptive fault tolerant framework. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3823–3827 Vol. 4, Oct 2005.

[27] Volker Friese. Computational challenges for the CBM experiment. In *Mathematical Modeling and Computational Science*, pages 17–27. Springer, 2012.

[28] J Gebelein, D Gottschalk, G May, and U Kebschull. SysCore3 A universal Read-Out Controller and Data Processing Board. *CBM Progress Report*, 2012.

[29] Jáno Gebelein. *FPGA fault tolerance in radiation environments*. PhD thesis, 2016.

[30] H. H. Gutbrod, I. Augustin, H. Eickhoff, K.-D. Groß, W. F. Henning, D. Krämer, and G. Walter. FAIR Baseline Technical Report. Technical report, 2006.

[31] D. L. Hansen, E. J. Miller, A. Kleinosowski, K. Kohnen, A. Le, D. Wong, K. Amador, M. Baze, D. DeSalvo, M. Dooley, K. Gerst, B. Hughlock, B. Jeppson, R. D. Jobe, D. Nardi, I. Ojalvo, B. Rasmussen, D. Sunderland, J. Truong, M. Yoo, and E. Zayas. Clock, flip-flop, and combinatorial logic contributions to the seu cross section in 90 nm asic technology. *IEEE Transactions on Nuclear Science*, 56(6):3542–3550, Dec 2009.

[32] H. Hatamkhani, Koon-Lun Jackie Wong, R. Drost, and Chih-Kong Ken Yang. A 10-mw 3.6-gbps i/o transmitter. In *2003 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.03CH37408)*, pages 97–98, June 2003.

[33] Johann M Heuser. Status of the CBM experiment. In *EPJ Web of Conferences*, volume 95, page 01006. EDP Sciences, 2015.

[34] Johann M Heuser, M Deveaux, C Müntz, and J Stroth. Requirements for the silicon tracking system of cbm at fair. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 568(1):258–262, 2006.

[35] Claudia Höhne, S Das, M Dürr, T Galatyuk, P Koczon, S Lebedev, A Maevskaya, G Ososkov, Cbm Collaboration, et al. Development of a rich detector for electron identification in cbm. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 595(1):187–189, 2008.

[36] K. Kasinski, R. Kleczek, P. Otfinowski, R. Szczygiel, and P. Grybos. STS-XYTER, a high count-rate self-triggering silicon strip detector readout IC for high resolution time and energy measurements. In *2014 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pages 1–6, Nov 2014.

[37] Jihwan Kim, Ajay Balankutty, Amr Elshazly, Yan-Yu Huang, Hang Song, Kai Yu, and Frank O'Mahony. 3.5 a 16-to-40gb/s quarter-rate nrz/pam4 dual-mode transmitter in 14nm cmos. In *Solid-State Circuits Conference-(ISSCC), 2015 IEEE International*, pages 1–3. IEEE, 2015.

[38] N. Kocaman, T. Ali, L. P. Rao, U. Singh, M. Abdul-Latif, Y. Liu, A. A. Hafez, H. Park, A. Vasani, Z. Huang, A. Iyer, B. Zhang, and A. Momtaz. A 3.8 mw/gbps quad-channel 8.5 x2013;13 gbps serial link with a 5 tap dfe and a 4 tap transmit ffe in 28 nm cmos. *IEEE Journal of Solid-State Circuits*, 51(4):881–892, April 2016.

[39] M Krieger. Design of new spadic front-end boards for trd readout.

[40] Christian Leber. *Efficient hardware for low latency applications*. PhD thesis, Universität Mannheim, 2012.

[41] J. Lehnert, A.P. Byszuk, D. Emschermann, K. Kasinski, W.F.J. Müller, C.J. Schmidt, R. Szczygiel, and W.M. Zabolotny. GBT based readout in the CBM experiment. *Journal of Instrumentation*, 12(02):C02061, 2017.

[42] F. Lemke, D. Slogsnat, N. Burkhardt, and U. Bruening. A unified interconnection network with precise time synchronization for the CBM DAQ-system. In *2009 16th IEEE-NPSS Real Time Conference*, pages 506–511, May 2009.

[43] Frank Lemke. CBM Protocol Second Generation. 2010.

[44] Frank Lemke. *Unified Synchronized Data Acquisition Networks*. PhD thesis, 2012.

[45] Frank Lemke and Ulrich Brüning. A generic link protocol for the CBM DAQ system. *CBM Progress Report 2008*, page 57, 2009.

[46] Frank Lemke and Sven Schatral. Design concepts and measurements of the cbm daq network. *Verhandlungen der Deutschen Physikalischen Gesellschaft*, 2013.

[47] Frank Lemke, Sven Schenk, and Ulrich Brüning. The adapted CBM network structure design and CBMnet V2.0 implementation. *CBM Progress Report 2011*, page 61, 2012.

[48] William R Leo. *Techniques for nuclear and particle physics experiments: a how-to approach*. Springer Science & Business Media, 2012.

[49] Austin Lesea and Peter Alfke. Xilinx fpgas overcome the side effects of sub-90 nm technology. *WP256 (v1. 0.1), Xilinx Inc*, 2, 2007.

[50] S. Löchner and H. Deppe. Radiation studies on the umc 180nm cmos process at gsi. In *2009 European Conference on Radiation and Its Effects on Components and Systems*, pages 614–616, Sept 2009.

[51] Pierre Maillard. *Radiation-hardened-by-design (RHBD) delay locked loops (DLLs): single event transient analysis, simulation, and hardening*. PhD thesis, Vanderbilt University, 2010.

[52] LVDS Owners Manual. -low voltage differential signaling. *National Semiconductor Corp*, 2004.

[53] Sebastian Andreas Manz. *Radiation mitigation for SRAM-Based FPGAs in the CBM experiment*. PhD thesis, 2015.

[54] M Barros Marin, S Baron, SS Feger, P Leitao, ES Lupu, C Soos, P Vichoudis, and K Wyllie. The gbt-fpga core: features and challenges. *Journal of Instrumentation*, 10(03):C03021, 2015.

[55] David G Mavis and Paul H Eaton. Seu and set mitigation techniques for fpga circuit and configuration bit storage design. In *2001 MAPLD International Conference*, 2001.

[56] M. Meghelli. A 43-gb/s full-rate clock transmitter in 0.18- mu;m sige bicmos technology. *IEEE Journal of Solid-State Circuits*, 40(10):2046–2050, Oct 2005.

[57] C. Menolfi, T. Toifl, P. Buchmann, M. Kossel, T. Morf, J. Weiss, and M. Schmatz. A 16gb/s source-series terminated transmitter in 65nm cmos soi. In *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, pages 446–614, Feb 2007.

[58] J Michel, M Böhmer, M Kajetanowicz, G Korcyl, L Maier, M Palka, J Stroth, A Tarantola, M Traxler, C Ugur, and S Yurevich. The upgraded HADES trigger and data acquisition system. *Journal of Instrumentation*, 6(12):C12056, 2011.

[59] J. Michel, I. Fröhlich, M. Böhmer, G. Korcyl, L. Maier, M. Palka, J. Stroth, M. Traxler, and S. Yurevich. The HADES trigger and readout board network (TrbNet). In *2010 17th IEEE-NPSS Real Time Conference*, pages 1–5, May 2010.

[60] J Michel, G Korcyl, L Maier, and M Traxler. In-beam experience with a highly granular DAQ and control network: TrbNet. *Journal of Instrumentation*, 8(02):C02034, 2013.

[61] Martin Miller and Michael Schnecker. Quantifying crosstalk-induced jitter in multi-lane serial data system. *DesignCon, Santa Clara, CA, USA*, 2009.

[62] P Moreira, S Baron, S Bonacini, O Cobanoglu, F Faccio, S Feger, R Francisco, P Gui, J Li, A Marchioro, et al. The gbt-serdes asic prototype. *Journal of Instrumentation*, 5(11):C11022, 2010.

[63] Paulo Moreira. The radiation hard gbtx link interface chip. In *PH-ESE Electronics Seminars, November*, volume 26, 2013.

[64] Paulo Moreira, A Marchioro, et al. The gbt: a proposed architecure for multi-gb/s data transmission in high energy physics. 2007.

[65] Paulo Moreira, K Wyllie, B Yu, A Marchioro, C Paillard, K Kloukinas, T Fedorov, N Pinilla, R Ballabriga, S Bonacini, et al. The gbt project. 2009.

[66] Walter F. J. Müller. DAQ Summary. CBM Collaboration Meeting. GSI Helmholtzzentrum für Schwerionenforschung GmbH, 2012.

[67] Walter F. J. Müller. Towards CBMnet V3.0. FLESDAQ Workgroup Meeting. GSI Helmholtzzentrum für Schwerionenforschung GmbH, 2013.

[68] Markus Müller. *Digital Centric Multi-Gigabit SerDes Design and Verification.* PhD thesis, 2017.

[69] K. Nakahara, S. Kouyama, T. Izumi, H. Ochi, and Y. Nakamura. Fault Tolerant Reconfigurable Device Based on Autonomous-Repair Cells. In *2006 International Conference on Field Programmable Logic and Applications*, pages 1–6, Aug 2006.

[70] NVIDIA. Nvlink 1.0 dl/pl specification. 2014.

[71] HK Pandey and TK Bhattacharya. Total ionizing dose (tid) effect test of developed 2.5 ghz radiation hardened clock multiplier unit (cmu). In *Proceedings of the DAE-BRNS Symp. on Nucl. Phys*, volume 60, page 1076, 2015.

[72] Rick A Philpott, James S Humble, Robert A Kertis, Karl E Fritz, Barry K Gilbert, and Erik S Daniel. A 20gb/s serdes transmitter with adjustable source impedance and 4-tap feed-forward equalization in 65nm bulk cmos. In *Custom Integrated Circuits Conference, 2008. CICC 2008. IEEE*, pages 623–626. IEEE, 2008.

[73] M. Portela-Garcia, C. Lopez-Ongil, M. Garcia-Valderas, L. Entrena, G. Thys, and S. Redant. Assessing set sensitivity of a pll. In *Design of Circuits and Integrated Systems*, pages 1–6, Nov 2014.

[74] M. Psarakis and A. Apostolakis. Fault tolerant FPGA processor based on runtime reconfigurable modules. In *2012 17th IEEE European Test Symposium (ETS)*, pages 1–6, May 2012.

[75] H. Quinn, P. Graham, K. Morgan, Z. Baker, M. Caffrey, D. Smith, M. Wirthlin, and R. Bell. Flight experience of the xilinx virtex-4. *IEEE Transactions on Nuclear Science*, 60(4):2682–2690, Aug 2013.

[76] P. Roche, G. Gasiot, S. Uznanski, J. M. Daveau, J. Torras-Flaquer, S. Clerc, and R. Harboe-Sørensen. A commercial 65nm cmos technology for space applications: Heavy ion, proton and gamma test results and modeling. In *2009 European Conference on Radiation and Its Effects on Components and Systems*, pages 456–464, Sept 2009.

[77] Sven Schatral and Ulrich Bruening. SPADIC ASIC 1.0 CBMnet link implementation and read-out. CBM DAQ Meeting, 2012.

[78] Sven Schatral, Frank Lemke, and Ulrich Bruening. Design of a deterministic link initialization mechanism for serial LVDS interconnects. *Journal of Instrumentation*, 9(03):C03022, 2014.

[79] Sven Schatral, Frank Lemke, and Ulrich Brüning. HUB ASIC Development. CBM Collaboration Meeting, 2013.

[80] Sven Schatral, Frank Lemke, and Ulrich Brüning. Status of CBMnet Implementation in current Read-out Chains. CBM Collaboration Meeting, 2013.

[81] Sven Schatral, Frank Lemke, and Ulrich Brüning. Status of the CBMnet based Front-end read-out. CBM FEE/DAQ Workshop, 2014.

[82] Sven Schatral, Frank Lemke, Indranil Som, Tarun Bhattacharyya, and Ulrich Bruening. Status of cbmnet read-out and the prototype asic. *GSI Scientific Report*, 2014.

[83] Sven Schatral, Frank Lemke, Indranil Som, Tarun Bhattacharyya, and Ulrich Bruening. Design and prototyping of a read-out aggregation asic. *Fruehjahrstagung der Deutschen Physikalischen Gesellschaft e.v. (DPG15)*, 2015.

[84] Sven Schatral, Philipp Schaefer, Frank Lemke, and Ulrich Bruening. Read-out over Optics with CBMnet 2.0. CBM DAQ Meeting, 2012.

[85] Philipp Schäfer. Synchronization of Front-End Electronics in a Data Acquisition Interconnection Network, June 2012.

[86] Philipp Schäfer. Design and Implementation of a Prototype ASIC for an Unified DAQ Interconnection Network. Master's thesis, March 2015.

[87] Norbert Abel Sebastian Manz. The Read Out Controller for the FEET Boards Using the Optical Communication Module. 2010.

[88] Sélim Seddiki and the Cbm Collaboration. The Compressed Baryonic Matter experiment. *Journal of Physics: Conference Series*, 503(1):012027, 2014.

[89] A. Senger. FLUKA Calculations for CBM. GSI Helmholtzzentrum für Schwerionenforschung GmbH, 2011.

[90] Peter Senger. Goals and status of the cbm experiment. In *EPJ Web of Conferences*, volume 7, page 02003. EDP Sciences, 2010.

[91] PCI Sig. Pci express base specification revision 4.0 version 0.7. *PCI SIG*, 2016.

[92] Charles Slayman. Soft error trends and mitigation techniques in memory devices. In *Reliability and Maintainability Symposium (RAMS), 2011 Proceedings-Annual*, pages 1–5. IEEE, 2011.

[93] Indranil Som, Sven Schatral, Frank Lemke, and Ulrich Brüning. A High-Speed Serializer Read-out ASIC prototype. CBM Collaboration Meeting, 2014.

[94] Maximilian Thürmer. *Modelling and performance analysis of multigigabit serial interconnects using real number based analog verification methods*. PhD thesis, 2017.

[95] Thomas Toifl. Low-power high-speed cmos i/os: Design challenges and solutions. *Topical Workshop on Electronics for Particle Physics*, 2012.

[96] Dagan White. Considerations Surrounding Single Event Effects in FPGAs, ASICs, and Processors. *White Paper: Xilinx FPGAs*, 2012.

[97] Denis Wohlfeld, Frank Lemke, Holger Froening, Sven Schenk, and Ulrich Bruening. High-density active optical cable: from a new concept to a prototype. In *SPIE OPTO*, pages 79440L–79440L. International Society for Optics and Photonics, 2011.

[98] Xilinx. Spartan-6 Family Overview. *DS160*, 2011.

[99] Xilinx. Spartan-6 FPGA Configuration User Guide. *UG380*, 2017.

[100] Hubert Zimmermann. OSI reference model–The ISO model of architecture for open systems interconnection. *IEEE Transactions on communications*, 28(4):425–432, 1980.