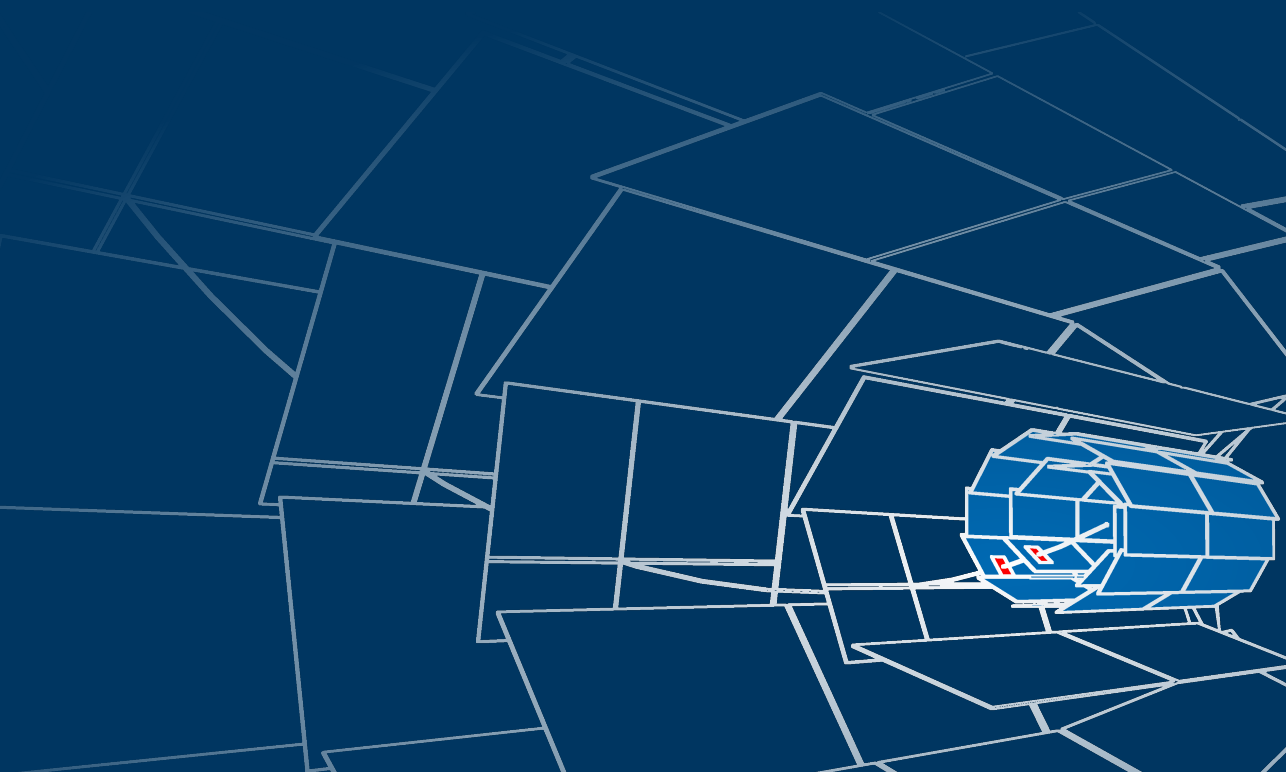
Thomas Geßler

JUSTUS-LIEBIG-
UNIVERSITÄT
GIESSEN

Development of FPGA-Based Algorithms for the Data Acquisition of the Belle II Pixel Detector

Dissertation



Development of FPGA-Based Algorithms for the Data Acquisition of the Belle II Pixel Detector


Belle II

Development of FPGA-Based Algorithms for the Data Acquisition of the Belle II Pixel Detector

Inauguraldissertation

zur Erlangung des Doktorgrades am

Fachbereich Mathematik und Informatik,
Physik, Geographie

der

Justus-Liebig-Universität Gießen

vorgelegt von

Thomas Geßler

aus Gießen

Gießen (2015)

Aus dem II. Physikalischen Institut

Dekan: Prof. Dr. Peter Jens Klar
Gutachter: Prof. Dr. Wolfgang Kühn
Gutachter: Prof. Dr. Alfred Müller

This document was typeset using LuaLaTeX and the memoir class. The text is set in Libertine, Biolinum, and Inconsolata.

Danksagung

Die Arbeit an dieser Dissertation hat mir die aufregende Chance gegeben, mich innerhalb einer internationalen Kollaboration an der Entwicklung eines großen künftigen Physikexperiments zu beteiligen. Für diese Möglichkeit möchte ich mich zuallererst bei meinem Betreuer, Prof. Dr. Wolfgang Kühn, bedanken. Weiterhin bedanke ich mich bei PD Dr. Jens Sören Lange, der sich unermüdlich um die Koordination der Belle- und Belle II-Projekte in Gießen kümmert. Herzlicher Dank gilt auch Christa Momberger, die mit viel Geduld alle organisatorischen und bürokratischen Hürden meistert, sowie Thomas Köster für seine Hilfestellung bei zahlreichen technischen Belangen.

Meine Dissertation war Teil eines Gemeinschaftsprojekts und wäre ohne das Teamwork mit meinen Gießener Kollegen kaum machbar gewesen. Ich bedanke mich bei Dr. Björn Spruck und Dr. David Münchow für die produktive Zusammenarbeit und die angenehme Arbeitsatmosphäre. Für die erfolgreiche Kooperation bei Integrationstests an anderen Instituten bedanke ich mich außerdem bei den Mitgliedern der Belle II- und DEPFET-Kollaborationen, allen voran Dipl.-Phys. Dmytro Levit von der Technischen Universität München.

Schließlich bedanke ich mich bei meinen Eltern, die mir durch ihre jahrelange moralische, finanzielle und kulinarische Unterstützung das Studium und die Promotion erst ermöglicht haben, und bei meiner Freundin Steffi. Danke, dass Du mir in dieser stressigen und fordernden Zeit immer zur Seite stehst.

Selbstständigkeitserklärung

Ich erkläre: Ich habe die vorgelegte Dissertation selbstständig und ohne unerlaubte fremde Hilfe und nur mit den Hilfen angefertigt, die ich in der Dissertation angegeben habe. Alle Textstellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen sind, und alle Angaben, die auf mündlichen Auskünften beruhen, sind als solche kenntlich gemacht. Bei den von mir durchgeführten und in der Dissertation erwähnten Untersuchungen habe ich die Grundsätze guter wissenschaftlicher Praxis, wie sie in der „Satzung der Justus-Liebig-Universität Gießen zur Sicherung guter wissenschaftlicher Praxis“ niedergelegt sind, eingehalten.

Datum *Unterschrift*

Zusammenfassung

In der vorliegenden Arbeit stelle ich Details zur Entwicklung und Ergebnisse von Tests des ONSEN-Systems vor. Dieses neuartige Echtzeit-Datenverarbeitungssystem wird eine Online-Reduktion der Ausgabedaten des Belle II-Pixeldetektors vornehmen. Das Belle II-Experiment wird sich am künftigen SuperKEKB Elektron-Positron-Collider befinden. Mit seinen 40 Sensormodulen, die in einer zweilagigen zylindrischen Geometrie um das Strahlrohr herum angebracht sind, wird der Pixeldetektor der innerste Detektor von Belle II sein. Er befindet sich im Einfluss von erheblichen Untergrundsignalen, verursacht durch seinen Abstand von nur 14 mm zum Wechselwirkungspunkt und der beispiellosen Luminosität von $8 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, die SuperKEKB erreichen wird. Die Auslese der 8 Millionen Pixel des Pixeldetektors wird etwa 20 µs dauern, was ungefähr 5000 Kollisionen von Elektronen- und Positronen-Bunches entspricht. Während dieser langen Integrationszeit werden Treffer in bis zu 3 % aller Pixel angesammelt, die größtenteils auf Untergrundprozesse zurückzuführen sind. Die resultierende Ausgabedatenrate wird nahezu 20 GB/s betragen. Das vereinheitlichte Datenaufnahmesystem, das für alle anderen Subdetektoren von Belle II benutzt wird, ist für viel kleinere Datenraten entwickelt worden und kann für den Pixeldetektor nicht angewendet werden. Ein Online-Datenreduktionsmechanismus—basierend auf „Bereichen von Interesse“, die bei einer Echtzeit-Ereignisrekonstruktion bestimmt werden—wird benutzt werden um Untergrundtreffer aus den Pixeldaten zu eliminieren und dadurch deren Größe um den Faktor 30 zu reduzieren, bevor sie permanent gespeichert werden.

Zu den Aufgaben des ONSEN-Systems gehört die Zwischenspeicherung aller Pixeldetektor-Daten während die Ereignisrekonstruktion stattfindet sowie die Filterung der Treffer anhand von Bereichen von Interesse, die von zwei externen Systemen bestimmt werden. Seine FPGA-basierte Hardware-Plattform ist eine Entwicklung vom IHEP in Peking, während die FPGA-Firmware für die

Datenprozessierung an der Justus-Liebig-Universität Gießen entworfen wurde. Ein großer Teil der Firmware für das ONSEN-System ist im Rahmen der Arbeit an dieser Dissertation entstanden. Neben der Mitarbeit an der Konzeption des Gesamtsystems gehören dazu Mechanismen für den Datenaustausch mit anderen Teilen der Datenaufnahmekette unter Verwendung von verschiedenen Protokollen, das Puffern von Rohdaten und prozessierten Daten, und die Analyse von Datenströmen zur Extraktion von Ereignisinformationen und der Verifizierung der Datenintegrität. Zu diesem Zweck wurde Quellcode in einer Hardwarebeschreibungssprache für die prozessorbasierte FPGA-Architektur entwickelt, die die Überwachung und Steuerung der implementierten Logik erlaubt. Weitere Arbeit wurde bei der Inbetriebnahme und Fehlerbehebung der Hardware-Plattform gemeinsam mit den Entwicklern am IHEP geleistet.

Der Pixeldetektor und das Datenaufnahmesystem von Belle II stellen verschiedene Anforderungen an die Leistung des ONSEN-Systems, darunter ein Datenfluss von fast 600 MB/s und eine Speicherbandbreite von etwa 1 GB/s bei jedem der 32 Module, die die Datenreduktion vornehmen. Um diese Werte zu erreichen verwendet das ONSEN-System serielle Hochgeschwindigkeitsverbindungen und hardwarenahe Speichercontroller-Schnittstellen. Tests der implementierten Logik haben gezeigt, dass diese sämtliche Anforderungen übertrifft und eine durchgängige Datenrate von 621.6 MB/s und eine Speicherbandbreite von bis zu 1683 MB/s erreicht. Während eines Tests des Pixeldetektors am DESY-Teststrahl, bei dem auch ein Prototyp des Belle II Datenaufnahmesystems zum Einsatz kam, wurden mehr als 20 Millionen Ereignisse mit dem ONSEN-System aufgenommen. Eine Analyse der Daten zeigte, dass die datenprozessierende Logik des ONSEN-Systems bei diesen Ereignissen stabil und fehlerfrei arbeitete. Weitere Änderungen an der Firmware sind nötig, wenn das System zum vorgesehenen Format skaliert wird. Machbarkeitsstudien haben gezeigt, dass alle Komponenten für das finale System einsatzbereit sind, sodass die nötigen Anpassungen kein Problem darstellen werden.

Abstract

In this thesis, I present development details and test results for the ONSEN system, a novel real-time data-processing system that will perform an online reduction of the output data from the Belle II pixel detector. The Belle II experiment will be located at the future SuperKEKB electron-positron collider. With its 40 sensor modules, arranged in a two-layer barrel geometry directly around the beam pipe, the pixel detector will be Belle II's innermost detector. It is subject to a harsh background environment, caused by its distance of only 14 mm from the interaction point and SuperKEKB's unprecedented instantaneous luminosity of $8 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. The read-out of its almost 8 million pixels takes approximately 20 µs, corresponding to about 5000 electron-positron bunch crossings. During this long integration time, hits in up to 3 % of all pixels will be accumulated, mostly stemming from background processes. The resulting output data rate will be close to 20 GB/s. The common data-acquisition system, used for all other Belle II subdetectors, was designed for much smaller rates and cannot be adapted to incorporate the pixel detector. An online data-reduction mechanism, based on regions of interest from a real-time event reconstruction, will be used to eliminate background hits from the pixel data and thereby reduce its size by a factor of 30 before it is put to permanent storage.

The ONSEN system is responsible for the buffering of the complete pixel-detector data while the event reconstruction takes place, and for performing the filtering of pixels according to the regions of interest determined by two external systems. Its FPGA-based hardware platform is a development from the IHEP in Beijing, while the FPGA firmware performing the data-processing functions was designed at the University of Gießen. A large part of the ONSEN system's firmware is a result from the work on this thesis. This includes: the co-design of the overall system architecture; I/O mechanisms for the data exchange with other subsystems of the data-acquisition chain, using different protocols;

the buffering of the raw and processed pixel data in memory; and the parsing of data streams for the extraction of event information and data-integrity tests. To this end, code in a hardware description language was developed for the processor-based FPGA architecture, allowing the online monitoring and control of the implemented logic. Additional work was invested in the the commissioning and co-debugging of the hardware platform together with the developers from the IHEP.

The pixel detector and Belle II data-acquisition systems impose various requirements on the performance of the ONSEN system, including a data throughput of almost 600 MB/s and a memory bandwidth of about 1 GB/s for every of the 32 modules performing the data reduction. The ONSEN system uses high-speed serial I/O links and low-level memory-controller interfaces to achieve these values. Small-scale tests show that the performance of the implemented logic surpasses the requirements, with a maintained input data rate of 621.6 MB/s and a memory bandwidth of up to 1683 MB/s. During tests of a pixel-detector module at the DESY test-beam facility, including the scaled-down Belle II data-acquisition system, more than 20 million events were recorded with the ONSEN system. An offline analysis of the data showed that the ONSEN system's data processing logic performed stably and without errors for these events. Further changes to the firmware are required to scale the system up to its design architecture. Feasibility tests have shown that all components for the final system are in a working state, and the required changes to the firmware will not pose a problem.

Contents

1 Introduction **1**

2 B-Factory Physics **5**
2.1 The Standard Model of Particle Physics 5
2.2 *CP* Violation and the CKM Matrix 8
2.3 Measurements at the B-Factories 13
2.4 Charmonium Spectroscopy and Exotic States 22
2.5 New Physics and the Need for a Super B-Factory 26

3 Belle II—A Super B Factory **33**
3.1 The SuperKEKB Accelerator 33
3.2 The Belle II Detector 36
3.3 The DEPFET Pixel Detector 44
3.4 Trigger and Data Acquisition 50

4 The ONSEN Data Reduction System **59**
4.1 Hardware Platform 59
4.2 System Architecture 71
4.3 Design Aspects 77
4.4 Contributions from the Work on this Thesis 86
4.5 Remaining Issues 90

5 Test Results **93**
5.1 Laboratory and Benchmark Tests 93
5.2 System Integration Tests 98
5.3 Carrier Board Tests 108

6 Conclusion and Outlook **113**

A Node Architecture Details **117**
A.1 Overview . . . 117
A.2 Merger Node Dataflow . . . 118
A.3 Selector Node Dataflow . . . 120

B IP Cores **123**
B.1 Common Features . . . 123
B.2 xFP Aurora Wrapper . . . 129
B.3 SiTCP Wrapper . . . 139
B.4 Belle II Format Handler . . . 145
B.5 NPI Writer . . . 149
B.6 NPI Reader . . . 155
B.7 Other ONSEN IP Cores . . . 161

C Data Formats **163**
C.1 Pixel Data from DHH . . . 163
C.2 ROI Data . . . 166
C.3 ONSEN Memory Management . . . 169
C.4 ONSEN Output Data . . . 170
C.5 Checksum Format . . . 172

D Hardware Details **173**
D.1 xFP v4.0 . . . 173
D.2 CNCB v3.3 . . . 179
D.3 UCF Files . . . 185

Bibliography **203**

Chapter 1

Introduction

Modern experiments in the field of particle physics are the result of the collaborative work of hundreds, sometimes thousands of researchers. With these experiments, physicists aim to investigate nature at its smallest scales and study the rarest processes. The motivation for a new experiment often arises from the desire of theoretical physicists to test the predictions of a promising new theory or explore the limits of an established model. Experimental physicists then take up the challenge and design new detectors and, in many cases, dedicated particle accelerators that are up to the job. The planning and construction of these machines is an extremely complex endeavor. It involves the development of new technologies and methods, and theses like this one emerge as a result of the many individual tasks. Only after the numerous components of the experiment—including not only detector parts but also cooling and high-voltage, data acquisition and trigger, control and monitoring, and many more systems—are assembled and working can the actual data taking and the "harvest" of the long-awaited results begin.

Most present-day accelerator experiments fall in one of two classes: *Energy frontier* experiments, like the ATLAS and CMS experiments at the LHC proton-proton collider, use large instruments to generate particle beams at extremely high energies. This allows them to investigate physics processes that are not accessible at lower energies and discover new, very massive particles. They also produce lighter particles in copious amounts, allowing them to study rare processes, albeit with large backgrounds. Their most famous recent result was the discovery of the Higgs boson in 2012, upon which theoretical physicists Peter Higgs and François Englert were awarded the 2013 Nobel Prize in Physics. *Intensity frontier* experiments, on the other hand, focus on the precise investigation of rare processes, using intense particle beams, high collision rates, strictly defined initial conditions, and very sensitive detectors. The Belle exper-

iment at the KEKB electron-positron collider in Tsukuba, Japan, and the BaBar experiment at the PEP-II electron-positron collider at Stanford University, USA, belong to this category. In 2001, they found evidence for the violation of *CP* symmetry in the neutral B meson system, which had been predicted by the Kobayashi-Maskawa mechanism. This discovery, too, led to the bestowal of a Physics Nobel Prize, this time on theoretical physicists Makoto Kobayashi and Toshihide Maskawa.

Both BaBar and Belle started data taking in 1999. The BaBar experiment was concluded in 2008, the Belle Experiment in 2010. Plans for an upgrade of Belle had been in the making for several years however, and the Japanese Ministry of Education, Culture, Sports, Science and Technology approved them in 2010, endorsing the continuation of the experiment under the name Belle II. BaBar did not receive an upgrade. The Italian Istituto Nazionale di Fisica Nucleare planned the construction of the SuperB experiment near Rome with many members of the BaBar collaboration, but this project was eventually cancelled. Belle II will therefore be the only B-factory of the second-generation, a so-called *Super B-factory*.

The SuperKEKB accelerator will surpass KEKB's luminosity by a factor of 40, and a new pixel detector for Belle II, based on the novel DEPFET technology, will vastly enhance the experiment's vertex resolution. These improvements will allow Belle II to study processes that were beyond the capabilities of the previous B-factories; they will however, also bring about new challenges. The higher luminosity will give rise to an increased background; this environment will be particularly demanding for the new pixel detector, which is expected to produce much more output data than can be handled by the permanent-storage system. Most of the hits detected by the pixel detector will be caused by background events.

This thesis describes the development and test of an integral part of the Belle II data-acquisition system that tackles this problem: The *Online Selection Nodes* (ONSEN) system is responsible for the online reduction of the Belle II pixel detector data. It is a development of the Belle II group at the University of Gießen, using a hardware platform designed at the IHEP in Beijing, China. In the following chapters, I specify why such a system is needed, explain it in detail, emphasizing the parts that were created during the work on this thesis, and show recent test results.

In chapter 2 I give an overview of B physics and explain the motivation for the construction of Belle II. After a brief introduction of the Standard Model of particle physics, I concentrate on the topics that are specific to the physics of Belle and Belle II, like *CP* violation and the quest for New Physics in the oscillation and decay of B mesons.

Belle II builds on the physics program of Belle, and many of its goals involve

the same techniques and methods that were used in Belle; it is therefore natural to begin by describing the physics and measurements of the B-factories and discuss their most important results. This includes the discovery of new, as-yet mysterious charmonium-like states that have sparked new interest in the field of charmonium spectroscopy. The latter part of the chapter then concentrates on the search for physics beyond the Standard Model, where the ultimate goal of the Belle II physics program lies. I briefly introduce some New Physics models, as well as processes that can be exploited to probe (and possibly exclude) them. I give a review of the current state and explain which improvements an upgrade to a Super B-factory will bring, and how they will allow Belle II to surpass the sensitivity of its predecessor and extend its reach into unmapped regions that are promising to contain physics beyond the Standard Model.

Chapter 3 describes the various parts of the Belle II detector and the SuperKEKB accelerator, and explains how they will achieve the proposed improvements. I discuss Belle II's subsystems, focusing on the new pixel detector with a short introduction of the DEPFET technology. Then I turn to the data-acquisition and trigger systems and point out the particular challenges for the pixel detector. This provides the motivation for the development of a data-reduction system.

With this groundwork, I begin the discussion of the work that was done for this thesis in chapter 4. I introduce the ONSEN system and its modular architecture, and describe how the various building blocks work together to achieve the required data reduction. The co-design and debugging of the hardware platform were a large part of the work on this thesis. I give a review of the hardware development, from the early stages to the current and final design. This includes a short introduction of field-programmable gate arrays (FPGAs) that are used for the data processing. I describe which parts of the ONSEN system have evolved from the work on this thesis. Details that are mainly of interest for future developers, including interface and data-format descriptions, have been moved to the appendix (see below).

Chapter 5 shows test results of the ONSEN system from various testing environments. I discuss feasibility and benchmark test that were done in Gießen to evaluate the performance of the hardware and individual parts of the firmware. Of particular interest are two experiments that were performed at the DESY test beam facility in 2013 and 2014. They provided the first opportunity for trial runs with other parts of the detector. During these tests, the interoperability of the various components of the Belle II data-acquisition system could be verified.

Chapter 6 concludes the main body of the thesis. Here I summarize the most important aspects and review the test results and their implications. I present an outlook, discussing future plans and possible changes and upgrades to the system that could augment its functionalities.

The extensive appendix lists the gritty details about the inner workings of the ONSEN system. It is meant as a reference for users and future developers. Appendix A lists the different node types and their inner structure, including the interconnection of the logic blocks (IP cores) used in each node. The IP cores themselves are explained in appendix B; this chapter is the documentation of the cores that were developed as part of the work on this thesis, including descriptions of the cores' functionalities and interfaces. In appendix C, I explain the data formats that are used for the inbound and outbound data streams of the ONSEN system and the internal data formats that are used in the communication between cores and for the memory management. Appendix D lists various details about the hardware that did not fit in the main text.

Chapter 2

B-Factory Physics

This chapter gives an overview of the physics phenomena that will be studied with the Belle II experiment. It begins with a short overview of the Standard Model of particle physics. A historical review follows, introducing the development and theoretical basics of B flavor physics and describing how the idea of the e^+e^--collider B-factories came to be. The main part of this chapter is dedicated to the discussion of the methods, discoveries, and results of the two B-factories constructed in the 1990s, Belle and BaBar. The last section discusses models of New Physics and ways to observe their effects at a future Super B-factory.

2.1 The Standard Model of Particle Physics

Particle physics is the study of the elementary constituents of matter and the forces acting between them. Since the mid-twentieth century, a number of quantum field theories have evolved that have been proven to describe the behavior of fundamental particles very successfully. The *Standard Model of particle physics* is a conglomerate of these theories, in particular of quantum chromodynamics (QCD) and the electroweak theory of Glashow, Weinberg, and Salam [1–3]. It can explain three of the four observed fundamental forces of nature: the electromagnetic force between electrically charged particles; the strong interaction, which is responsible for the binding of protons and neutrons in atomic nuclei; and the weak interaction, which manifests itself in the β-decay of radioactive elements. It also incorporates the Higgs mechanism, by which elementary particles can acquire mass [4–6]. The fourth fundamental force, gravitation, is not part of the Standard Model.

The constituents of matter in the Standard Model are fermions with spin 1/2. They are categorized into quarks and leptons. The quarks are bound by the

strong interaction to hadrons, like the protons and neutrons that form atomic nuclei. Leptons, on the other hand, are not subject to the strong interaction. The electrons in the atomic shell belong to this class. While protons, neutrons, and electrons are sufficient to describe "conventional" matter, the Standard Model contains many more particles and allows for countless bound states.

Both quarks and leptons can be subdivided into three families, each of which contains two elementary particles. Somewhat analogous to the rows of the periodic table of elements, the particles in different families share similar properties but have different masses.

Each of the three quark families consists of a down-type quark with electric charge −1/3 e and an up-type quark with electric charge +2/3 e. The members of the first quark family are the down quark (d) with a mass[1] of 4.8 MeV and the up quark (u) with a mass of 2.3 MeV. As the lightest quarks, they are the only types that can form a stable bound state: the proton. The second family consists of the strange quark (s) with a mass of 95 MeV and the considerably heavier charm quark (c) with a mass of 1.3 GeV. The third family contains the bottom quark (b) with a mass of 4.2 GeV and the heaviest elementary particle: the top quark (t) with a mass of 173 GeV.

The lepton families each contain a charged lepton with $Q = -1\,\mathrm{e}$ and a neutral, very light neutrino. The charged leptons show a mass hierarchy similar to that of the quark families: The electron (e^-) has a mass of 511 keV, the muon (μ^-) a mass of 106 MeV, and the tauon (τ^-) a mass of 1.8 GeV. The names of the three neutrinos are derived from their charged lepton-family partner: ν_e, ν_μ, and ν_τ. The Standard Model originally treated neutrinos as massless. This notion had to be revised after the discovery of neutrino oscillations [7], which can only be accounted for if the neutrinos have a finite mass. The current experimental limit constrains it to $< 2\,\mathrm{eV}$.

The fundamental interactions in the Standard Model can be attributed to the exchange of bosons with spin 1. The most well-known such *gauge boson* is the photon (γ), which mediates the electromagnetic force between electrically charged particles. The weak interaction is carried by the charged $W^\pm$ bosons and the neutral Z^0 boson. In contrast to the massless photon, they are very heavy, with masses of 80 GeV and 91 GeV respectively. As a consequence, the range of the weak interaction is very short, while the electromagnetic interaction has infinite range. At low energies, the weak interaction is about four orders of magnitudes weaker than the electromagnetic force, while both reduce to a single, *electroweak* force at very high energies. An important aspect of the charged

[1] I list all masses in units of energy, using the usual convention of "natural units" where c = 1. The u, d, and s masses given here are the current values for the *bare* or *current-quark* masses. They are different from the much larger *constituent* masses that can be assigned to quarks based on their effective contribution to hadron masses.

weak current mediated by the $W^{\pm}$ is its ability to transform up-type quarks into down-type quarks, even between families. This is explicitly forbidden for the neutral weak current mediated by the Z^0, which only couples to two particles of the same type: The Standard Model forbids flavor-changing neutral currents in leading-order processes in which a single boson in exchanged. In higher-order processes, including loops of virtual particles, they are allowed but heavily suppressed by the GIM mechanism [8].

The gauge boson of the strong interaction is the gluon (g). Like the photon, the gluon is massless, but it couples to a different type of charge: the *color*. Every quark carries this quantum number, and its value is one of red, green, and blue. Correspondingly, every antiquark carries an anticolor: antired, antigreen, or antiblue. As a unique feature of the strong interaction, the gluons themselves also carry color charge and couple to themselves. This leads to a principle called *confinement*, which states that particles with color can never be observed individually, but only in groups that add up to a colorless state: three different colors, three different anticolors, or a color and its anticolor. As a consequence of color confinement, the range of the strong interaction is not infinite, as gluons can be exchanged only between color-neutral particles.

Bound states of quarks are called *hadrons*. The simplest hadrons allowed by confinement are groups of three quarks, called *baryons*, groups of three antiquarks, called *antibaryons*, and groups of a quark and an antiquark, called *mesons*. In principle, the Standard Model allows other colorless combinations, such as pentaquarks (four quarks and an antiquark), tetraquarks (two quarks and two antiquarks), or molecules of two mesons, bound loosely together by the strong force. According to QCD, gluons can also contribute to the quantum numbers of a meson, resulting in hybrid meson states with valence gluons. Because gluons couple to each other, even glueballs, containing no quarks and only gluons, should be allowed. Until today, no unambiguous evidence for any such state has been found, although many candidates exist.

The Standard Model is a very successful description of most particle physics phenomena. Its latest triumph was the discovery of its last missing component in 2012, the Higgs boson, H^0. Despite its success, however, the Standard Model is known to be incomplete. A glaring defect is its inability to describe the gravitational force. There are more shortcomings, though: The Standard Model cannot predict the huge amounts of dark matter and dark energy in the universe. The intriguing, symmetric pattern of the quark and lepton families and the large number of input parameters—if neutrino oscillations are taken into account, they amount to 25 or 26 [9, p. 500]—beg the question whether a hidden substructure exists.

A problem that is of particular interest for the Belle II physics program is the observed asymmetry of matter and antimatter in the universe. This

phenomenon has been discussed for a long time. In 1967, Sakharov formulated three conditions that must be fulfilled by baryon-generating interactions in order to establish such an asymmetry [10]: They must violate baryon number conservation; they must violate C and CP conservation; and they must occur outside of thermal equilibrium.

While no baryon-number violating processes have ever been observed, CP violation is firmly established in the Standard Model. The current model is, however, not sufficient to account for the observed matter-antimatter asymmetry, since the predicted CP violation is too small by several orders of magnitude [11, p. 180]. Physics processes beyond the Standard Model could lead to additional contributions and enhance CP violating effects. Such *New Physics* phenomena are widely sought after, and their search will be a major part of the physics program of Belle II.

2.2 CP Violation and the CKM Matrix

Historical background

In the 1950s, the common consensus in the physics community was that all physical processes should obey a principle known as *parity invariance*: The "mirror image" of a process, denoted by the parity operator $\mathcal{P}$ that inverts all spacial coordinates in one point, should obey the same laws as the original process. This notion was falsified by an experiment performed by Wu in 1956 [12].

Wu was able to align the spins of radioactive cobalt nuclei using a strong magnetic field at very low temperature. She monitored the angular distribution of the electrons emitted in the β^- decay of ^{60}Co. Any anisotropy in this distribution was considered an indicator for parity violation: In the parity-inverted version of the experiment, the spin direction of the nuclei is unaffected[2] while the momentum vector of the β electrons is inverted. Parity conservation dictated that electrons be emitted equally in the direction of the nuclei's spin and in the opposite direction, so that both versions are equal. Wu found, however, that the emission was preferably in the direction of the spin vector, and thus showed that parity was *not* conserved in weak interactions.

An experiment on the decay $\pi^+ \longrightarrow \mu^+\nu_\mu$ in the same year showed that this process, too, violated parity conservation [13]: The *handedness* of the produced antimuons—the direction of their spin with respect to their momentum—was monitored. Parity inversion flips a particle's handedness, since the spin's orientation is conserved while the momentum vector is reversed. Again, parity

[2] Like any angular momentum vector, spin is a pseudovector. Mathematically, it is the cross product of two polar vectors. Since both polar vectors change their sign under parity inversion, their cross product is preserved.

invariance predicted an equal number of "right-handed" and "left-handed" antimuons. It turned out, however, that all of them were produced with left-handed orientation, their spin pointing in the direction opposite to their momentum.

Symmetry could be ostensibly restored to this situation by inverting *charge conjugation* as well as parity: The charge conjugation operator C transforms every particle into its antiparticle. The combined operation $\mathcal{CP}$ would turn the process $\pi^+ \longrightarrow \mu^+\nu_\mu$ with left-handed antimuons into $\pi^- \longrightarrow \mu^-\overline{\nu}_\mu$ with right-handed muons, seemingly averting the contradiction. This notion broke down as well, when in 1964 Cronin and Fitch showed that the combined quantum number CP was not conserved in the decay of neutral kaons [14].

At that time it was believed that neutral kaons propagate as a mixture of two CP eigenstates with different lifetimes[3]: the short-lived K_1^0 with $CP = +1$ and $c\tau \approx 2.7\,\text{cm}$; and the long-lived K_2^0 with $CP = -1$ and $c\tau \approx 15.3\,\text{m}$. They decayed into two or three pions, the two-pion system with $CP = +1$ and the three-pion system with $CP = -1$. According to CP conservation, the K_1^0 should *never* decay into three pions, and the K_2^0 *never* into two pions. The smaller phase space of the three-pion decay explains the much longer lifetime of the K_2^0.

In their famous experiment, Cronin and Fitch produced a beam of neutral kaons and allowed it to propagate for more than 300 K_1^0 decay lengths. After this distance, the K_1^0 component of the kaon beam was expected to have completely disappeared. They searched for decays of the remaining kaons into two charged pions, and found a significant number of such events. Cronin and Fitch concluded that the observed decays could only be a result of CP violation, and that the measured particles were not real eigenstates of $\mathcal{CP}$, but a mixture of such. Today they are known today as K_S^0 and K_L^0—"short" and "long", referring to their lifetimes.

In the theoretical framework of the time, this effect could not be explained. In 1973, the Japanese physicists Makoto Kobayashi and Toshihide Maskawa provided the explanation that has become the main source for CP violation in today's Standard Model. Their theory was an extension of the mechanism that Nicola Cabibbo had proposed in 1963 [15] to describe the weak interactions of the up, down, and strange quark[4]. Cabibbo had proposed that in charged weak interactions, the up quark couples to a *mixture* of the down and strange quarks that can be quantified by an angle known today as the *Cabibbo angle*, θ_C. His theory could model weak universality—the principle connecting the weak interactions of leptons to those of quarks—but it could not explain CP

[3] This concept is explained in more detail in the next section.

[4] The physical processes are described here in modern terms, although the quark model had not been established in the early 1960s.

violation.

Kobayashi and Maskawa noted—at a time when there was experimental evidence for only three quarks and predictions for a fourth—that it was possible to introduce a *CP*-violating parameter in a more general model with six quarks from three families [16]. They suggested that the charged weak current between up-type anti-quark fields ($\bar{u}_{\mathrm{L}}$, $\bar{c}_{\mathrm{L}}$, and $\bar{t}_{\mathrm{L}}$) and down-type quark fields (d_{L}, s_{L}, and b_{L}) was given by the expression[5]

$$-\frac{g}{\sqrt{2}}\left(\bar{u}_{\mathrm{L}}, \bar{c}_{\mathrm{L}}, \bar{t}_{\mathrm{L}}\right) \gamma^{\mu} W_{\mu}^{+} V_{\mathrm{CKM}} \begin{pmatrix} d_{\mathrm{L}} \\ s_{\mathrm{L}} \\ b_{\mathrm{L}} \end{pmatrix} + \mathrm{h.c.} \tag{2.1}$$

with the unitary matrix

$$V_{\mathrm{CKM}} = \begin{pmatrix} V_{\mathrm{ud}} & V_{\mathrm{us}} & V_{\mathrm{ub}} \\ V_{\mathrm{cd}} & V_{\mathrm{cs}} & V_{\mathrm{cb}} \\ V_{\mathrm{td}} & V_{\mathrm{ts}} & V_{\mathrm{tb}} \end{pmatrix},$$

the weak coupling constant g, the gamma matrices γ^{μ}, and the vector-boson field W_{μ}^{+}.

CKM-matrix representations and observables

Each of the nine matrix elements V_{ij} is complex, so the CKM matrix can be described by eighteen real parameters. This number is decreased significantly by the unitarity condition, $V_{\mathrm{CKM}}^{\dagger} V_{\mathrm{CKM}} = \mathbb{1}$, which is equivalent to the three equations

$$\sum_{i \in \{\mathrm{u,c,t}\}} |V_{ij}|^2 = 1, \quad j \in \{\mathrm{d,s,b}\},$$

plus the six equations

$$\sum_{i \in \{\mathrm{u,c,t}\}} V_{ij}^{*} V_{ik} = 0, \quad j,k \in \{\mathrm{d,s,b}\}, \quad j \neq k. \tag{2.2}$$

The remaining degrees of freedom can be written as three angles and six complex phases. The latter can be eliminated further using the phase invariance of the quark fields: Any of the six quark fields q_{L} in equation 2.1 can be multiplied with an arbitrary complex phase factor $\mathrm{e}^{\mathrm{i}\varphi}$ without changing the physical

[5] The nomenclature chosen here is the one used by the Particle Data Group [17].

interpretation. These factors can be extracted as diagonal matrices that are multiplied with the CKM matrix from both sides:

$$V'_{\mathrm{CKM}} = \begin{pmatrix} e^{i\alpha} & 0 & 0 \\ 0 & e^{i\beta} & 0 \\ 0 & 0 & e^{i\gamma} \end{pmatrix} V_{\mathrm{CKM}} \begin{pmatrix} e^{-i\delta} & 0 & 0 \\ 0 & e^{-i\epsilon} & 0 \\ 0 & 0 & e^{-i\zeta} \end{pmatrix} = \begin{pmatrix} V_{ud}e^{i(\alpha-\delta)} & V_{us}e^{i(\alpha-\epsilon)} & V_{ub}e^{i(\alpha-\zeta)} \\ V_{cd}e^{i(\beta-\delta)} & V_{cs}e^{i(\beta-\epsilon)} & V_{cb}e^{i(\beta-\zeta)} \\ V_{td}e^{i(\gamma-\delta)} & V_{ts}e^{i(\gamma-\epsilon)} & V_{tb}e^{i(\gamma-\zeta)} \end{pmatrix}. \tag{2.3}$$

The six phase factors can be chosen in such a way that five of the complex phases in the CKM matrix are eliminated. The remaining, *irreducible* complex phase is a source of CP violation if it is different from zero.

Further rephasing of the CKM matrix can produce many different representations, all of which have at least three real parameters and one complex phase. They all yield identical physical predictions; the square of each matrix element, in particular, is *rephasing invariant*, but the real and imaginary parts can differ between representations. The most common parameterization was proposed by Chau and Keung in 1984, when the size of some CKM observables was already known to a degree [18]. It uses the three mixing angles θ_{12}, θ_{13}, and θ_{23} and the phase factor δ, with $s_{ij} = \sin\theta_{ij}$ and $c_{ij} = \cos\theta_{ij}$:

$$V_{\mathrm{CKM}} = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix}$$

The coupling within a quark family is strong, the one between families very weak, so the matrix is more or less diagonal. The hierarchy of the mixing angles, $\theta_{13} \ll \theta_{23} \ll \theta_{12} \ll 1$ reflects the different coupling strengths between the families. The complex phase is small, and it appears only in products with s_{13}, so this representation underlines the smallness of the CP violating effect.

Another useful parameterization was introduced by Wolfenstein in 1983 [19]:

$$V_{\mathrm{CKM}} = \begin{pmatrix} 1-\lambda^2/2 & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1-\lambda^2/2 & A\lambda^2 \\ A\lambda^3(1 - \rho - i\eta) & -A\lambda^2 & 1 \end{pmatrix} + \mathcal{O}(\lambda^4)$$

It approximates the matrix elements with a power series of the parameter λ, which is approximately V_{us}, with $|V_{us}| \approx 0.2$. The other parameters, A, ρ, and η, are also of order one. This representation serves to show that the CKM matrix does not deviate from a diagonal matrix before the second power of λ; complex (CP violating) quantities appear with the third power of λ; all other deviations from the explicit approximation are at least of the order λ^4.

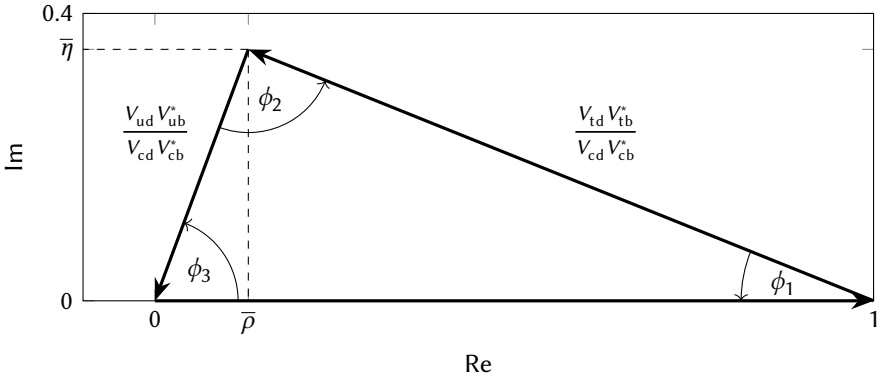


Figure 2.1: The most commonly used unitarity triangle

The unitarity triangle

In order to quantify the magnitude of *CP* violation in a way that is independent of the choice of parameterization, one of the unitarity conditions in equation 2.2 can be arranged into the form

$$1 + \frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*} + \frac{V_{td}V_{tb}^*}{V_{cd}V_{cb}^*} = 0.$$

Each of the two fractions in this equation is rephasing invariant—this can be seen by applying the phase factors from equation 2.3—so they correspond to physical observables. The equation describes a triangle in the complex plane, with its apex at the point

$$\overline{\rho} + i\overline{\eta} = -\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*},$$

$\overline{\rho}$ and $\overline{\eta}$ being variants of the corresponding Wolfenstein parameters. This *unitarity triangle*, shown in figure 2.1, is the most common of six possible triangles that can be constructed from the conditions in equation 2.2.

Besides the parameters $\overline{\rho}$ and $\overline{\eta}$, the three angles of the unitarity triangle

are also rephasing invariant parameters[6]:

$$\begin{aligned} \phi_1 = \beta = \arg\left(-\frac{V_{cd}V_{cb}^*}{V_{td}V_{tb}^*}\right), \quad \phi_2 = \alpha = \arg\left(-\frac{V_{td}V_{tb}^*}{V_{ud}V_{ub}^*}\right), \text{ and} \\ \phi_3 = \gamma = \arg\left(-\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*}\right). \end{aligned} \tag{2.4}$$

The magnitude of the various observables is a measure for the extent of *CP* violation: No *CP* violation would imply $\phi_1 = \phi_3 = \overline{\eta} = 0$ and $\phi_2 = \pi$, while significant *CP* violation would lead to significantly different values.

2.3 Measurements at the B-Factories

Flavor Oscillation

By 1980, both the quark model and the KM mechanism had become established after the discoveries of the predicted charm quark in 1974 [20, 21] and the bottom quark in 1977 [22]. Both quarks were found through the production of their *quarkonia*, the J/ψ meson being a $c\bar{c}$ bound state and the ϒ meson being a $b\bar{b}$ bound state. *CP* violation had, however, only been observed in the neutral kaon system. In 1981, Bigi and Sanda published an article that proposed a method for the search for *CP* violation in the decay of B mesons [23]. This paper eventually prompted the construction of the B-factories[7]. The suggested experiments relied on a mechanism often referred to as *flavor oscillation*. This phenomenon was already well known from the kaon system, and it will be introduced here shortly on the example of B meson oscillation. For a complete theoretical treatment, see, for example, the text book on *CP* violation by Bigi and Sanda [24] or the B-factory "Legacy Book" [11].

The CKM matrix can be understood as relating quark flavor eigenstates to their mass eigenstates: A B meson, for example, is *produced* in a state with definite quark flavors, but it *propagates* as a state with definite mass. The propagation eigenstate can be a superposition of different flavor eigenstates. The produced meson flavor defines the initial state of the propagation, but a

[6]There are two different naming conventions for the angles: The Belle collaboration used ϕ_l, ϕ_2, and ϕ_3 while the BaBar collaboration used γ, α, and β. The Belle convention is used in this thesis.

[7]In this thesis, I use the term "B-factory" exclusively for electron-positron colliders designed to produce an abundance of B mesons. It should be noted that other experiments also study B physics, the most notable being the LHCb experiment mentioned later. LHCb exploits the high cross section for B production in high-energy proton-proton collisions at the LHC, and can therefore be considered a hadronic B-factory.

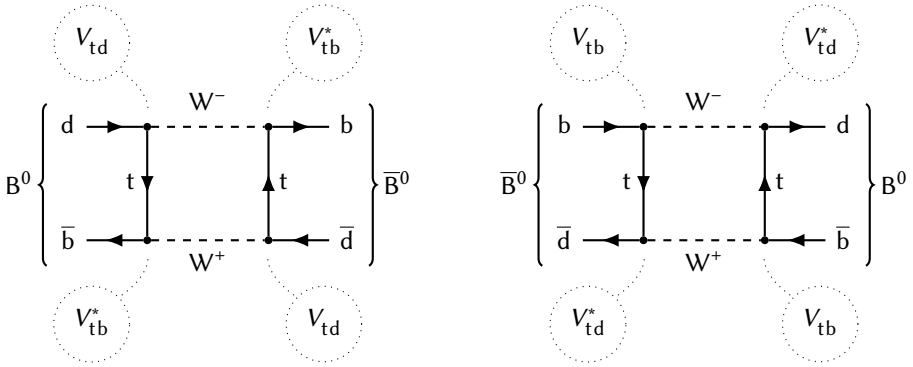


Figure 2.2: Feynman diagrams contributing to the mixing between B^0 and $\overline{B}^0$. Processes where a virtual u or c quark is exchanged instead of a t quark also exist, but are heavily suppressed by the small CKM matrix elements and the large t quark mass [11, p. 119]. The contribution of complex-conjugated CKM matrix elements to both diagrams leads to *CP* violation in the mixing if a complex CKM phase exists.

measurement at a later time can yield a different flavor. Consequently, certain neutral mesons can "oscillate" into their antiparticles and back through box diagrams like the ones depicted in figure 2.2. Oscillations of this type are only possible for the K^0, D^0, B^0, and B_s^0.

In the case of the B^0, the mass eigenstates are called B_L and B_H—"light" and "heavy", since they have slightly different masses. They correspond to the K_S^0 and K_L^0 states from the kaon system, but have almost identical lifetimes. The relation between the mass and flavor eigenstates is

$$|B_L\rangle = p|B^0\rangle + q|\overline{B}^0\rangle,$$
$$|B_H\rangle = p|B^0\rangle - q|\overline{B}^0\rangle.$$

B^0 and $\overline{B}^0$ are odd eigenstates of $\mathcal{P}$[8], and they are each other's antiparticles, so $\mathcal{P}|B^0\rangle = -|B^0\rangle$, $\mathcal{P}|\overline{B}^0\rangle = -|\overline{B}^0\rangle$, $\mathcal{C}|B^0\rangle = |\overline{B}^0\rangle$, and $\mathcal{C}|\overline{B}^0\rangle = |B^0\rangle$. If $p = q$ held, we could write

$$\mathcal{CP}\left(p|B^0\rangle + q|\overline{B}^0\rangle\right) = -\left(p|B^0\rangle + q|\overline{B}^0\rangle\right) \text{ and}$$
$$\mathcal{CP}\left(p|B^0\rangle - q|\overline{B}^0\rangle\right) = +\left(p|B^0\rangle - q|\overline{B}^0\rangle\right),$$

[8] As $d\overline{b}$ ground states, the B^0 has orbital angular momentum $L = 0$. For mesons, the parity can be calculated as $P = -(-1)^L$.

meaning the mass eigenstates would also be *CP* eigenstates. This implies that a superposition of B mesons in a definite state of *CP* would also propagate in that state and could not be measured with another value at a later time. Conversely, in the case that $|p| \neq |q|$, the measured *CP* value can change. This effect is known as *CP* violation *in the mixing*; it is considered an *indirect* form of *CP* violation[9].

A second, *direct* form of *CP* violation can be observed *in the decay* of a particle M and its *CP* conjugate $\overline{M}$ to the final state f and its *CP* conjugate $\overline{f}$. *CP* conservation implies that *CP* conjugated processes occur with the same rate, so that $A_f = \Gamma(M \longrightarrow f)$ should be equal to $\overline{A}_{\overline{f}} = \Gamma(\overline{M} \longrightarrow \overline{f})$. *CP* violation, on the other hand, implies $|A_f| \neq |\overline{A}_{\overline{f}}|$. Unlike *CP* violation in the mixing, *CP* violation in the decay is also possible for charged particles.

In their paper, Bigi and Sanda proposed a method to investigate the B meson system for a combined effect: *CP* violation *in the interference between mixing and decay*. This form of *CP* violation occurs when two *CP* conjugated states decay into the same final state. If the final state is a *CP* eigenstate f, the magnitude of the *CP* violation can be expressed by the parameter

$$\lambda = \frac{q}{p}\frac{A_f}{\overline{A}_f}.$$

For a *CP*-odd final state f, the decay rate f_+ of the process $\mathrm{B}^0 \longrightarrow f$ and the decay rate f_- of the process $\overline{\mathrm{B}}^0 \longrightarrow f$ are then approximately given by [11, p.122]

$$f_\pm(\Delta t) = \frac{\mathrm{e}^{-|\Delta t|/\tau_{\mathrm{B}^0}}}{4\tau_{\mathrm{B}^0}}\left[1 \pm \frac{2\,\mathrm{Im}(\lambda)}{1+|\lambda|^2}\sin(\Delta m_\mathrm{d}\Delta t) \mp \frac{1-|\lambda|^2}{1+|\lambda|^2}\cos(\Delta m_\mathrm{d}\Delta t)\right], \qquad (2.5)$$

where Δm_d is the mass difference between the two B mass eigenstates and τ_{B^0} is their lifetime, which is assumed to be equal in this case. For a *CP*-even final state, the signs of the sine- and cosine-terms in equation 2.5 must be inverted. The time difference Δt can be extracted from the decays of a flavor-tagged B meson pair; this concept will be explained in the next section.

The most important observable for the extraction of Δm_d and various other parameters is the *time-dependent asymmetry*

$$\mathcal{A}(\Delta t) = \frac{f_+(\Delta t) - f_-(\Delta t)}{f_+(\Delta t) + f_-(\Delta t)} = S\sin(\Delta m_\mathrm{d}\Delta t) - C\cos(\Delta m_\mathrm{d}\Delta t), \qquad (2.6)$$

where the amplitudes

$$S = \frac{2\,\mathrm{Im}(\lambda)}{1+|\lambda|^2} \quad \text{and} \quad C = \frac{1-|\lambda|^2}{1+|\lambda|^2} \qquad (2.7)$$

[9] I use here the nomenclature given by the PDG [17, pp. 225–226].

are parameters that depend on the investigated final state. Physical quantities can be extracted from the fitted values of S and C in the measured asymmetries. This technique, however, requires a sufficiently large value for the mass difference (and "oscillation frequency") Δm_d, so that the effect can be observed within the B meson lifetime.

The oscillation of neutral B mesons was first observed by the ARGUS experiment in 1987 [25]. The mixing was found to be considerable. This discovery prompted the planning and construction of two B-factories during the 1990s: The Belle experiment at the KEKB electron-positron collider, belonging to the KEK High Energy Accelerator Research Organization in Tsukuba, Japan; and the BaBar experiment at the PEP-II electron-positron collider, belonging to the SLAC National Accelerator Laboratory at Stanford University, USA.

Flavor Tagging and Vertexing

The measurement of time-dependent asymmetries is possible using B mesons from the decay of the $\Upsilon(4S)$ meson. The $\Upsilon(4S)$ is the first bottomonium state above the open-bottom threshold, meaning that its mass is greater than the combined mass of two B mesons. This makes a decay into a $\mathrm{B^0\overline{B}^0}$ or a $\mathrm{B^+B^-}$ pair possible. With a branching ratio of more than 96 %, this decay mode is very strong, which explains the short lifetime of the $\Upsilon(4S)$: It decays more than three orders of magnitude faster than the lower-lying Υ states.

A striking feature of this decay is the quantum entanglement of the two mesons in the final state: The $\mathrm{B\overline{B}}$ pair is produced in a *flavor singlet* state that can be written as $1/\sqrt{2}\left(|\mathrm{B^0}\rangle|\overline{\mathrm{B}}^0\rangle - |\overline{\mathrm{B}}^0\rangle|\mathrm{B^0}\rangle\right)$. As a consequence, the individual flavor of each meson is undetermined at each time t. Their combined bottomness is always 0, even though the entangled state is subject to flavor oscillation. The coherence of this state holds until one of the mesons decays. If the decay mode and its final state particles are flavor specific—for example, only possible for a $\mathrm{B^0}$ but not for a $\overline{\mathrm{B}}^0$—a determination of the decay products constitutes a measurement of the meson flavor at the time of the decay. The decaying meson is then called the *flavor-tagging* meson, $\mathrm{B_{tag}}$. Figure 2.3 shows such a decay: The processes $\mathrm{B^0 \longrightarrow D^-\ell^+\nu_\ell}$ and $\mathrm{\overline{B}^0 \longrightarrow D^+\ell^-\bar{\nu}_\ell}$ allow the determination of the meson flavor from the charge of the lepton in the final state.

The measurement of the $\mathrm{B_{tag}}$ flavor results in the decay of the singlet wave function: The flavor of the second meson at the time of the $\mathrm{B_{tag}}$ decay is determined to be the opposite flavor. From that time, it propagates as an individual particle, still experiencing flavor oscillation. It can, therefore, decay with the same flavor as $\mathrm{B_{tag}}$ or with the opposite flavor.

If the second B meson decays into a *CP* eigenstate, as explained in the previous section, measurements of *CP* violation in the interference between

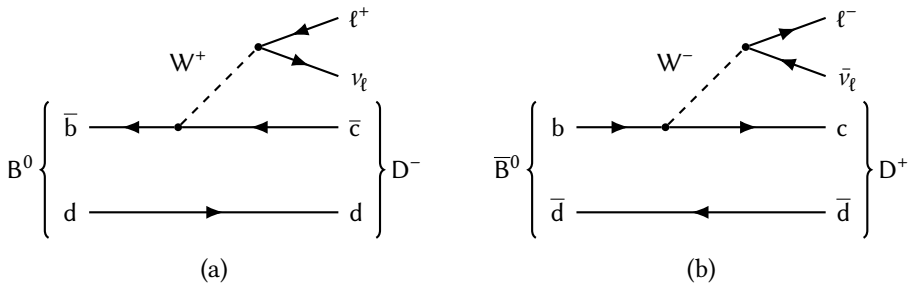


Figure 2.3: Feynman diagrams of **(a)** the flavor-tagging decay $B^0 \longrightarrow D^-\ell^+\nu_\ell$ and **(b)** its *CP*-conjugated process $\overline{B}^0 \longrightarrow D^+\ell^-\bar{\nu}_\ell$. The decays have flavor-specific final states.

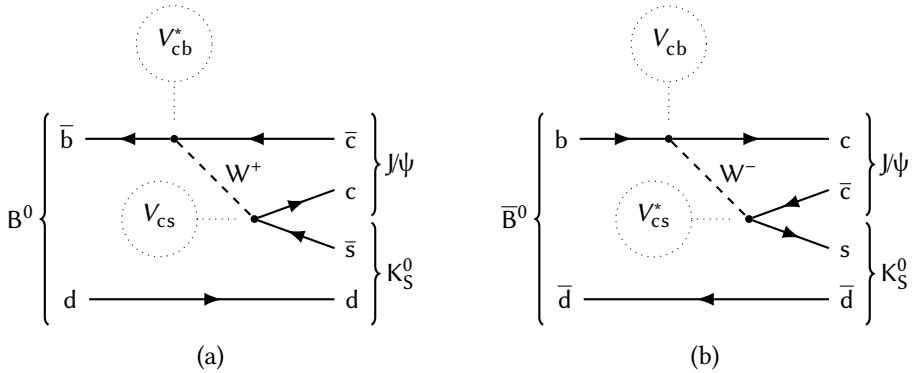


Figure 2.4: Feynman diagram of **(a)** the *CP*-eigenstate decay $B^0 \longrightarrow J/\psi K_S^0$ and **(b)** its *CP*-conjugated process $\overline{B}^0 \longrightarrow J/\psi K_S^0$. Both decays have the same final state. The contribution of complex conjugated CKM matrix elements to both diagrams leads to *CP* violation in the decay if a complex CKM phase exists.

mixing and decay become possible. Bigi and Sanda identified the process $B^0/\overline{B}^0 \longrightarrow J/\psi K_S^0$, shown in figure 2.4, as the most promising decay for the measurement of the angle $\sin(2\phi_1)$. Such a measurement requires the determination of the time difference Δt between the B_{tag} and B_{CP} decays. The closeness of the $\Upsilon(4S)$ to the $B\overline{B}$ threshold and the B meson life time of about $\tau_B = 1.5 \times 10^{-12}$ s make this difficult: If an $\Upsilon(4S)$ is produced at rest in the laboratory frame, for example by a symmetric electron-positron collider, and decays into a $B^0\overline{B}^0$ pair, the B mesons carry a momentum of only $p = 327\,\mathrm{MeV/c}$. The distance they travel before decaying is in the order of $\beta\gamma c\tau_B = (p/m_B)\tau_B \approx 28\,\mu\mathrm{m}$.

The B-factories therefore used a different approach. Both accelerators were constructed as *asymmetric* electron-positron colliders. KEKB used an electron

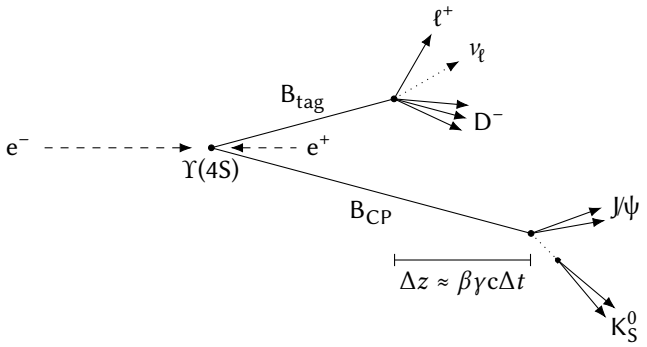


Figure 2.5: Schematic illustration of the flavor tagging process. An $\Upsilon(4S)$ is produced in the boosted e^+e^- rest frame and decays immediately into a B meson pair. One B meson decays in a flavor specific mode, so that the flavor of *both* B mesons is fixed at the time of the decay. The second B meson decays Δt later into a *CP* eigenstate. The distance of the decay vertices in z direction can be used for an approximate determination of Δt. If B_{CP} decays before B_{tag}, Δt is negative.

energy of 8 GeV and a positron energy of 3.5 GeV, resulting in a boost of the center-of-mass frame of $\beta\gamma = 0.43$. PEP-II used an electron energy of 9 GeV and a positron energy of 3.1 GeV, resulting in a boost of the center-of-mass frame of $\beta\gamma = 0.56$. In both cases, the energy in the center-of-mass frame is approximately 10.58 GeV: the mass of the $\Upsilon(4S)$ resonance.

In the laboratory frame, the relativistic boost from the asymmetric collision results in an additional contribution to the decay length of the B mesons from $\Upsilon(4S)$ decays of about $0.5c\tau_B = 225\,\mu m$. Neglecting the relative movement of the B mesons, this allows the determination of the decay-time difference from the vertex distance, $\Delta t = \Delta z/\beta\gamma c$, as illustrated in figure 2.5.

For determining the decay vertices, Belle and BaBar both used double-sided silicon-strip detectors in a barrel-arrangement around the beam pipes. They achieved a Δz-resolution in the order of 100 µm. Both experiments used drift chambers for particle-track reconstruction and momentum measurement and electromagnetic calorimeters for energy determination. For particle identification, Belle relied on Cherenkov counters and a time-of-flight detector while BaBar used a DIRC detector. In both cases, resistive plate chambers as the outermost detector layer provided detection for muons and neutral hadrons, especially K^0_L. Details about the detectors can be found in the respective design reports [26, 27]. They are not elaborated here, because an in-depth description of the Belle II detector follows in the next chapter.

Results

In 2001, the Belle and BaBar collaborations published the long-awaited measurements of the angle $\sin(2\phi_1)$ of the unitarity triangle, firmly establishing the existence of *CP* violation in the B meson system [28, 29]. They measured time-dependent *CP* asymmetries in the interference between mixing and decay of the "gold-plated channel" $B^0 \longrightarrow J/\psi K^0_S$.

The mixing-asymmetry parameter for B mesons that arises from the diagrams in figure 2.2 is [11, p. 304]

$$\frac{q}{p} = \frac{V_{td} V_{tb}^*}{V_{td}^* V_{tb}},$$

while the decay amplitudes from the diagrams in figure 2.4 give

$$\frac{A_f}{\overline{A}_f} = \eta_f \frac{V_{cb} V_{cs}^*}{V_{cb}^* V_{cs}} \frac{V_{cs} V_{cd}^*}{V_{cs}^* V_{cd}}.$$

Here η_f is the *CP* eigenvalue of the final state. It is -1 for the decay into $J/\psi K^0_S$, but the equation also holds for final states with $\eta_f = +1$ like $J/\psi K^0_L$. The factor $V_{cs} V_{cd}^* / V_{cs}^* V_{cd}$ arises from the K^0-$\overline{K}^0$ oscillations in the final state. Note that if the CKM elements were real, both fractions would be one and no *CP* violation could be observed. For the mixing parameter λ we then arrive at

$$\begin{aligned}
\lambda = \frac{q}{p}\frac{A_f}{\overline{A}_f} &= \eta_f \frac{V_{td} V_{tb}^*}{V_{td}^* V_{tb}} \frac{V_{cb} V_{cd}^*}{V_{cb}^* V_{cd}} = \eta_f \left(\frac{V_{cd} V_{cb}^*}{V_{td} V_{tb}^*} \right)^* \left(\frac{V_{cd} V_{cb}^*}{V_{td} V_{tb}^*} \right)^{-1} \\
&= \eta_f \exp\left[-2i \arg\left(\frac{V_{cd} V_{cb}^*}{V_{td} V_{tb}^*} \right) \right] \\
&= \eta_f \, e^{-2i\phi_1} = \eta_f \cos(2\phi_1) - i\eta_f \sin(2\phi_1)
\end{aligned}$$

using the definition of ϕ_1 from equation 2.4. This allows us to calculate the asymmetry amplitudes for this decay with equation 2.7,

$$S = \eta_f \sin(2\phi_1) \quad \text{and} \quad C = 0.$$

The time-dependent asymmetry is therefore

$$\mathcal{A}(\Delta t) = \eta_f \sin(2\phi_1) \sin(\Delta m_d \Delta t).$$

This function was measured by the B-factories with the goal of determining $\sin(2\phi_1)$. Figure 2.6 shows the latest results of such an analysis using the full Belle data set [30]. The sinusoidal form of the asymmetry can be clearly seen.

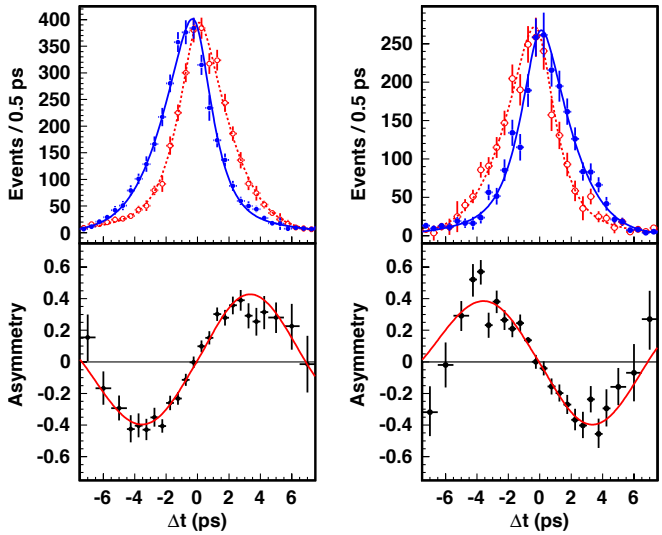


Figure 2.6: Measurements of time-dependent *CP* asymmetry with the full Belle data set. The left side shows data for *CP*-odd final states like $K^0_S J/\psi$. The right side shows data for *CP*-even final states like $K^0_L J/\psi$. The top plots correspond to the decay rates in equation 2.5. The red, dashed lines show events where B_{tag} is a B^0; the blue, solid lines show events where B_{tag} is a $\overline{B}^0$. The bottom plots correspond to the asymmetry in equation 2.6 and allows the extraction of $\sin(2\phi_1)$ from the amplitude S. (Reprinted figure with permission from [30] I. Adachi et al., Phys. Rev. Lett. 108 (2012), p. 171802. Copyright 2012 by the American Physical Society.)

This measurement was a huge success for the Standard Model and the KM mechanism. It was the first observation of *CP* violation outside of the kaon system. Since then, there have been observations of *direct CP* violation in the B meson system by Belle and BaBar [31, 32] and possibly of *CP* violation in the D meson system by LHCb [33].

The measurements of Belle and BaBar set tight constraints on the angles of the unitarity triangles and the Wolfenstein parameters. The CKMfitter group calculates global averages of the measured values from different experiments and publishes graphical and numerical constraints [34]. Figure 2.7 shows the most current fits. The fitted values correspond to the Wolfenstein parameters

$$A = 0.810 \pm 0.024, \qquad \lambda = 0.22548 \pm 0.00068,$$
$$\overline{\rho} = 0.145 \pm 0.013, \qquad \overline{\eta} = 0.343 \pm 0.012$$

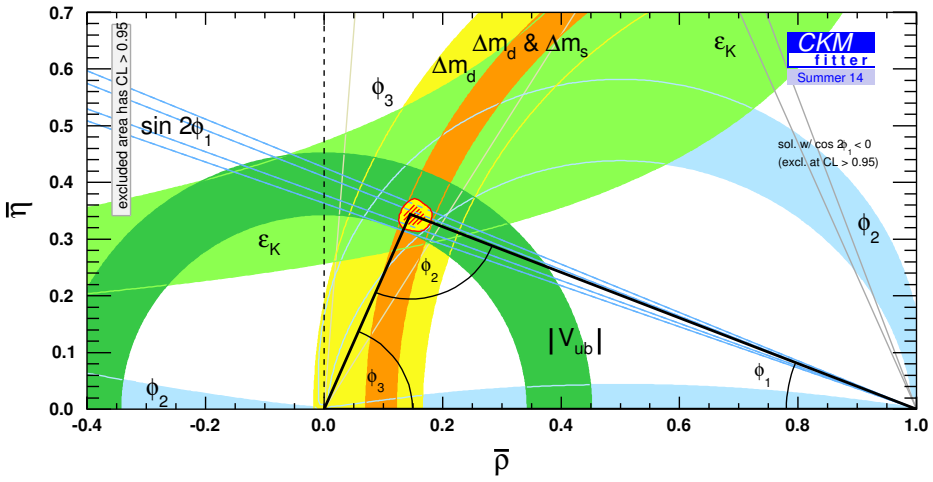


Figure 2.7: Graphical representation of CKM parameter fit results from the CKMfitter group [34]. The tip of the unitarity triangle determines the values of the parameters ϕ_1, ϕ_2, ϕ_3, $\overline{\rho}$, and $\overline{\eta}$. It is overconstrained by various measurements.

and the angles

$$\phi_1 = 21.89° \pm 0.77°, \quad \phi_2 = 91.0° \pm 2.3°, \quad \text{and} \quad \phi_3 = 67.1° \pm 2.2°.$$

The absolute values of the CKM matrix elements could be constrained to

$$\begin{pmatrix} |V_{ud}| & |V_{us}| & |V_{ub}| \\ |V_{cd}| & |V_{cs}| & |V_{cb}| \\ |V_{td}| & |V_{ts}| & |V_{tb}| \end{pmatrix} = \begin{pmatrix} 0.97424 & 0.22548 & 0.00355 \\ 0.22534 & 0.97341 & 0.0411 \\ 0.00855 & 0.0404 & 0.999146 \end{pmatrix} \pm \begin{pmatrix} 0.00016 & 0.00068 & 0.00017 \\ 0.00068 & 0.00018 & 0.0011 \\ 0.00027 & 0.0011 & 0.000038 \end{pmatrix} \begin{bmatrix} 0.016\,\% & 0.30\,\% & 4.8\,\% \\ 0.30\,\% & 0.018\,\% & 2.7\,\% \\ 3.2\,\% & 2.7\,\% & 0.0038\,\% \end{bmatrix}$$

Measurements of the off-diagonal elements involving top and bottom quarks, $|V_{ub}|$, $|V_{cb}|$, $|V_{td}|$, and $|V_{ts}|$, are particularly challenging. The B-factories contributed heavily to the current accuracy of their values. Still, these elements have the largest relative errors, leaving much room for improvement with future experiments.

2.4 Charmonium Spectroscopy and Exotic States

CP-violation studies are not the only field in which the B-factories made important discoveries. Another area that held particularly surprising results was charmonium spectroscopy. The detailed investigation of $c\bar{c}$ bound states was possible thanks to the copious production of charmed mesons in B decays.

Since the center-of-mass energy at B-factories is mostly fixed to the $\Upsilon(4S)$ mass, resonant production of charmonium, as in dedicated charm factories like the BES III experiment at the BEPC II electron-positron collider in Beijing, is not feasible. States with lower energies and $J^{PC} = 1^{--}$ can be produced if either the electron or the positron emits a photon before the collision—a process known as *initial state radiation*. Another possible production channel is the two-photon process $e^+e^- \longrightarrow e^+e^-(\gamma^*\gamma^*) \longrightarrow e^+e^-c\bar{c}$, which allows the quantum numbers $J^{PC} = 0^{\pm+}, 2^{\pm+}, 4^{\pm+}, \ldots$ and $3^{++}, 5^{++}, \ldots$ [35].

The most important $c\bar{c}$-production channel for B-factories, however, is through decays of B mesons: Their b quarks must eventually decay weakly into an up-type quark. Their coupling to the charm quark is much stronger than to the up quark, so the process $b \longrightarrow cW^- \longrightarrow c\bar{c}s$ is abundant. This mechanism can, in principle, produce any quantum number. It led to the discovery of the exotic charmonium states described below. Lastly, charmonium states can be produced via double-$c\bar{c}$ production, for example $e^+e^- \longrightarrow J/\psi + c\bar{c}$. This channel is particularly interesting because its cross section is much larger than predicted by theory [36].

In contrast to the light (u, d, and s) quarks, the mass of the charm quark is in the same order as that of its bound states, with $2m_c \approx 2550\,\text{MeV}$ and $2900\,\text{MeV} < m_{c\bar{c}} < 4700\,\text{MeV}$. The constituent quarks can therefore be associated with a small velocity, and the system can be approximately treated as non-relativistic. Similarly to the hydrogen and positronium systems in electrodynamics, a simple potential can then be used to model the force between the two quarks, and the energy levels of the system—the masses of the charmonium states—can be obtained by solving the Schrödinger equation.

An example for such a potential is [37]

$$V_0^{(c\bar{c})}(r) = -\frac{4}{3}\frac{\alpha_s}{r} + br + \frac{32\pi\alpha_s}{9m_c^2}\left(\frac{\sigma}{\sqrt{\pi}}\right)^3 e^{-\sigma^2 r^2}\,\vec{S}_c \cdot \vec{S}_{\bar{c}},$$

where α_s, b, m_c, and σ are parameters that are determined from fits using known charmonium masses as input. The first term is a Coulomb-like potential that models the binding force at short distances. The difference to the Coulomb potential from electrodynamics lies mainly in the much larger coupling constant α_s. The second term, which becomes dominant at larger distances, introduces a

linearly rising potential, resulting in a constant attractive force between the two quarks. I can be seen as a model for color confinement since an ever-increasing amount of energy must be expended to pull the quarks farther apart. The third term models the spin-spin hyperfine interactions between the two quarks. The referenced model treats additional spin-dependent terms, like spin-orbit coupling, as perturbations that lead to mass shifts of the determined states.

Figure 2.8 shows the masses of the charmonium states predicted by this model in comparison with experimental values. The correspondence between theoretical predictions and measurements are almost perfect in the mass region below the open-charm threshold—the energy above which decays into two charmed mesons are possible. Some of these states, like the h_c, have long evaded experimental discovery, but their masses have been correctly predicted for decades.

Above the open-charm threshold, the predictive power of potential models diminishes. While some of the predicted states have not yet been discovered, others miss the experimental values by tens of MeV. The decays into charmed mesons that become possible at these energies complicate the situation. What's more, other theoretical models predict the existence of exotic states at higher energies in the charmonium system. Potential models cannot predict such states, since they only describe two-quark systems, so more fundamental methods must be used.

Lattice QCD is such a method. It is a non-perturbative approach that calculates QCD on a discrete, four-dimensional spacetime grid using computer simulations. Since lattice QCD calculations are based on first principles of QCD, they allow the determination of all bound states that are possible in QCD, including so-called exotic states. Predictions based on lattice QCD exist for the masses of hybrid mesons that have gluonic degrees of freedom (valence gluons) [38] and even glueballs [39]. These calculations require large computational efforts and still suffer from uncertainties, including systematic errors from the discretization process and statistical errors from Monte Carlo calculations. They also require various input parameters like the strong coupling constant and quark masses.

The interest in charmonium spectroscopy was fueled by a discovery made in 2003: The Belle collaboration found a new state in the $\pi^+\pi^- J/\psi$ invariant-mass spectrum of the decay $B^\pm \longrightarrow K^\pm\pi^+\pi^- J/\psi$ [40]. This state, known as X(3872), appeared very close to the $\overline{D}^{*0}D^0$ threshold and could not be accounted for with naive potential models. It was classified as *charmonium-like*, since it decayed into final states with charmonium, indicating that the it must contain "hidden charm" (a $c\bar{c}$ pair). Its mass, however, did not fit any of the missing charmonium states, and its width was decidedly too narrow for a charmonium state above the open-charm threshold, which should be able to decay into a $D\overline{D}$ pair quickly. At

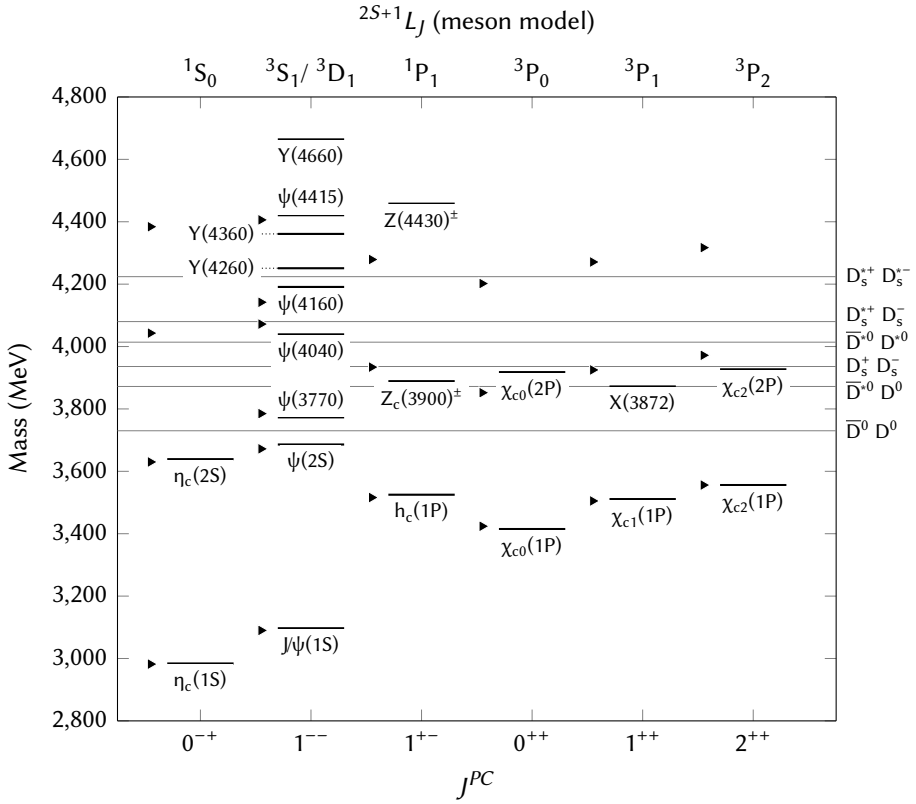


Figure 2.8: Charmonium and charmonium-like states that are listed as *confirmed* in the current PDG Review of Particle Physics [17]. The triangle marks are mass predictions from a non-relativistic potential model [37]. The vertical lines are thresholds for the production of charmed meson pairs. States are ordered in columns according to their quantum numbers J^{PC}. Quark spin (S) and orbital angular momentum (L) are assumptions based on the potential model and do not apply to the exotic states X, Y, and Z.

the time of its discovery, the quantum numbers of the X(3872) were not known. They were not completely established until 2013, when the LHCb collaboration reported the value $J^{PC} = 1^{++}$ [41]. The state is shown in figure 2.8 along with the conventional charmonium states. Its quantum numbers would fit the missing $\chi_{c1}(2P)$ state, but its mass is too far off. As of today, the situation is still unclear, but the closeness of the X(3872) mass to the combined masses of the $\overline{D}^{*0}$ and D^0 mesons indicates that it could be a loosely bound molecule of the two mesons.

Since the discovery of the X(3872), a number of new charmonium-like states have been identified. In 2004, the BaBar collaboration found the Y(4260) with quantum numbers $J^{PC} = 1^{--}$ in initial-state radiation processes [42][10]. Once again, potential models could not provide a fitting candidate (see figure 2.8), especially since the predicted 1^{--}-states in the mass regions of the Y(4260) had already been discovered. More states with the same quantum numbers showed up, including the Y(4360) and Y(4660). As in the case of the X(3872), their nature is still unknown. Possible explanations include tetraquarks, meson molecules, and hybrid mesons.

The clearest evidence to date of an exotic charmonium state was found in 2008 by the Belle collaboration [43]: In the decay $B \longrightarrow K\pi^{\pm}\psi(2S)$, Belle found a distinct peak in the $\pi^{\pm}\psi(2S)$ invariant-mass spectrum. Once again, the decay into $\psi(2S)$ indicated that the discovered state must contain a $c\bar{c}$ pair; in contrast to the X and Y states, however, the additional $\pi^{\pm}$, meant that the state carries electric charge. Consequently, it must be composed of at least two additional quarks, making it a very strong candidate for a tetraquark or meson molecule. The resonance was labeled $Z(4430)^{\pm}$. It was at first not seen by the BaBar collaboration [44], but it was later confirmed with high significance by LHCb [45], and its quantum numbers were determined to be $J^{PC} = 1^{+-}$. In the meantime, the $Z_c(3900)^{\pm}$ had been discovered by the BES III and Belle collaborations in 2013 [46, 47], making it the first charged charmonium-like state observed by to independent experiments.

More charged charmonium-like states have been observed since then, but there is still no unambiguous explanation for any of the exotic candidates. A similar situation has evolved in the $b\bar{b}$ system, where several "bottomonium-like" states, both neutral and charged, were discovered. Finding a theoretical model that is able to predict all of these mysterious states would greatly enhance our understanding of QCD. Precision measurements in the charmonium and bottomonium sector have therefore become a hot topic for current and future experiments.

[10] I use the nomenclature of X, Y, and Z that is currently prevalent among the physics community when referring to these exotic states. It should be noted that the PDG labels all mesons with unknown quark content with X and the state's mass [17, p. 120].

2.5 New Physics and the Need for a Super B-Factory

The results from the *CP*-violation measurements at the B-factories were a great success for the Standard Model and KM mechanism. They showed that the underlying processes are well understood, and we can predict many observables precisely based on the known physics. If New Physics has a large impact on the B-meson and the *CP*-violating sector, it should lead to discrepancies between Standard-Model predictions and experimental results. In addition, effects of New Physics may be seen by overconstraining measurements of correlated values, like the angles of the unitarity triangle, and checking that they remain consistent with each other.

The Standard Model is assumed to have no underlying structure that can explain the number of quark and lepton families. There is no first principle that limits the number of generations to three, so a *fourth generation* could theoretically exist, harboring very heavy up- and down-type quarks as well as a very heavy lepton and an additional neutrino. In a such a configuration, the CKM matrix would be a 4×4 unitary matrix with six real parameters and three complex phases instead of only one. This would also have indirect implications for *CP* violation measurements in the sector up to the third generation. For instance, possible transitions between known quarks and fourth-generation quarks would mean that the 3×3 unitarity conditions do not hold exactly. In that case, the angles of the unitarity triangle would not add up to 180°. A simple extension of the current Standard Model with a fourth generation (SM4) has, however, been ruled out by the discovery of the Higgs boson by the ATLAS and CMS experiments; a fourth quark family would impact the Higgs mass and suppress its decay into $\gamma\gamma$ [48]. A possible four-generation model must therefore be more complicated.

Flavor experiments at the intensity frontier are particularly sensitive to New Physics models that incorporate new particles contributing to processes via loop diagrams. This was demonstrated in the 1970s by the discovery of the charm quark: The smallness of the branching ratio for the flavor-changing neutral current process $K_L^0 \longrightarrow \mu^+\mu^-$ could not be explained with the particles known at that time. This process is only possible via a loop diagram with two virtual quarks. The GIM mechanism could explain the suppression with a new quark contributing to this loop. Even the mass of the charm quark could be predicted long before it was directly observed for the first time.

A current class of New-Physics candidates are type-II Two-Higgs-Doublet models (2HDM) [49]. They extend the particle zoo with a second Higgs doublet, which leads to a new type of gauge boson: a charged Higgs boson $H^\pm$ that can appear in loop diagrams in places where the Standard Model only allows a $W^\pm$. This can have dramatic influences on observables for which the enhanced

process plays a role. Processes that are hopelessly suppressed in the Standard Model, for example, could become observable through this mechanism.

The term *Supersymmetry* (SUSY) describes another class of physics models that goes even farther. These models assign a heavier *superpartner* to each particle: a boson to each fermion, and vice versa. A frequently mentioned motivation for this is the apparent smallness of the Higgs mass despite possible loop corrections that should make it heavier by many orders of magnitude. This is known as the *hierarchy problem*, and supersymmetric theories try to avert it by introducing superpartners whose contributions to these corrections cancel each other out [50]. The Minimal Supersymmetric Standard Model (MSSM) introduces superpartners only for the currently known particles [51]. It includes the second Higgs doublet of type-II 2HDM, and is therefore sensitive to the same loop-diagram contributions.

In B-meson decays, loop diagrams often appear in the form of so-called *penguin diagrams*[11]. Figure 2.9 shows a few such processes. In addition to charged Higgs contributions to the loops, SUSY models could enhance these processes through contributions of supersymmetric-quark (squark) loops. Any deviation of measured observables from processes involving these diagrams can be taken as an indication for New Physics.

Radiative penguin decays, like the one shown in figure 2.9b, have been studied extensively in the past. At B-factories they can be observed, for example, in the rare process $\overline{B}^0 \longrightarrow \overline{K}^*(892)^0\gamma$. This process is not allowed at leading order, and so the main contribution comes from penguin diagrams. There are theoretical predictions for the influence of supersymmetric particles to decays of this kind [54]; measurements by Belle [55] and other experiments have, however, yielded no significant deviation from Standard-Model expectations.

Other processes are allowed in leading-order diagrams as well as penguin diagrams. Figure 2.10 shows the relevant diagrams for the decay $B^0 \longrightarrow D^+D^-$. This decay is particularly interesting because its final state is a *CP* eigenstate that allows the measurement of $\sin(2\phi_1)$ in time-dependent *CP* asymmetries. The inclusive b decay $b \longrightarrow c\bar{c}d$ can proceed via leading-order or gluonic penguin diagrams, while for the "gold-plated channel" $B^0 \longrightarrow J/\psi K^0_S$, with $b \longrightarrow c\bar{c}s$, only leading-order diagrams are possible (see figure 2.4). Calculating the asymmetry amplitudes *S* and *C* for the D^+D^- final state, it turns out that they are identical to the $J/\psi K^0_S$ final state if only the leading order diagram is taken into account [56]. The correction from the penguin diagram amounts only to a few percent, but could be increased by New Physics contributions. These could be detected by unexpected amplitude values or a result for $\sin(2\phi_1)$

[11] According to lore [53], this name was conceived by CERN physicist John Ellis after a lost bet that required him to use the word "penguin" in his next publication.

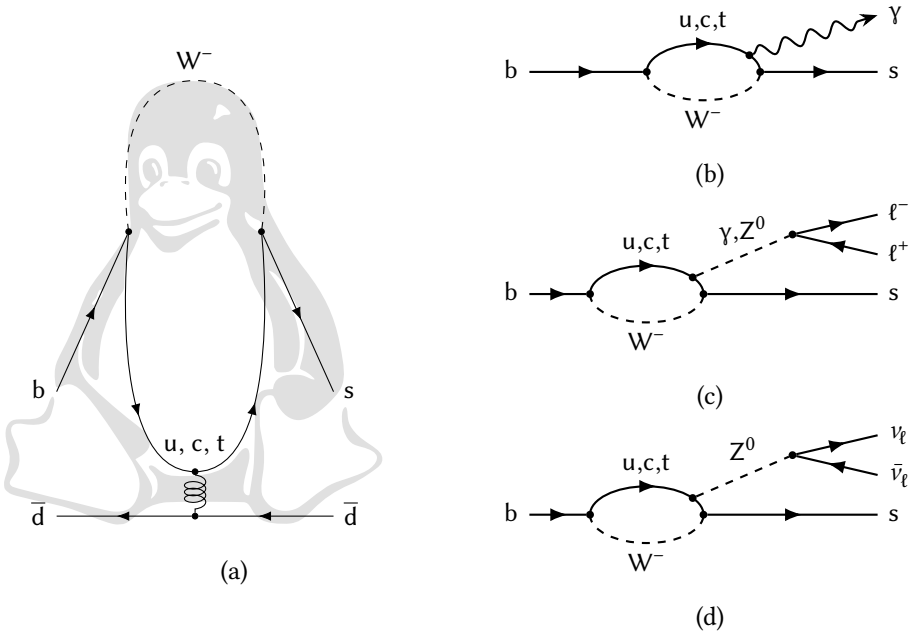


Figure 2.9: Penguin diagrams of involving $b \longrightarrow s$ transitions via up-type quark loops. **(a)** Illustrative gluonic penguin diagram. Note that the process $\overline{B}^0 \longrightarrow \overline{K}^0$ is energetically disallowed; the outer fermion lines must be rearranged to create an allowed process. (Background image adapted from the Tux logo [52].) **(b)** Inclusive radiative penguin decay $b \longrightarrow s\gamma$ with a real photon, observable in $\overline{B}^0 \longrightarrow \overline{K}^*(892)^0\gamma$. **(c)** Inclusive electroweak penguin decay $b \longrightarrow s\ell^+\ell^-$ with a virtual γ or Z^0, observable in $\overline{B}^0 \longrightarrow \overline{K}^*(892)^0\ell^+\ell^-$. **(d)** Inclusive electroweak penguin decay $b \longrightarrow s\nu_\ell\bar{\nu}_\ell$ with a virtual Z^0, observable in $\overline{B}^0 \longrightarrow \overline{K}^*(892)^0\bar{\nu}_\ell\nu_\ell$.

that differs from the $B^0 \longrightarrow J/\psi K_S^0$ value. A measurement by Belle could not find any such discrepancies [56].

Yet another channel for the determination of $\sin(2\phi_1)$ is $B^0 \longrightarrow J/\psi\phi$. With $b \longrightarrow s\bar{s}s$, this decay is forbidden at leading order and goes mainly via the gluonic penguin diagram, making it sensitive to New Physics. Here, Belle initially found a result for ϕ_1 that differed from the established value. This effect could, however not be confirmed by other analyses. With the sensitivity of the B-factories, no conclusive result could be obtained.

A final example for New Physics opportunities in B decays are searches for very rare processes that are forbidden in first-order and suppressed in loop diagrams. The decay $B_s^0 \longrightarrow \mu^-\mu^+$ is shown in figure 2.11. It can go via a box

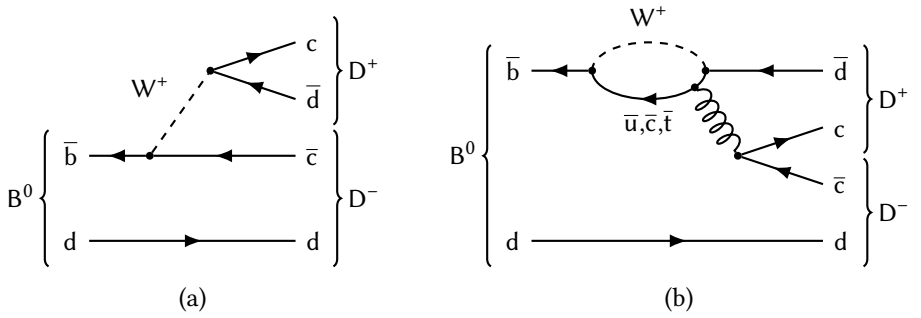


Figure 2.10: Two Feynman diagrams contributing to the process $B^0 \longrightarrow D^+D^-$: **(a)** a tree level diagram and **(b)** a gluonic penguin diagram.

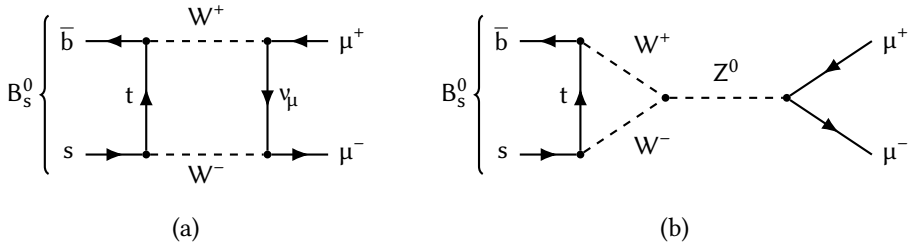


Figure 2.11: Two Feynman diagrams contributing to the rare process $B_s^0 \longrightarrow \mu^+\mu^-$: **(a)** a box diagram and **(b)** an electroweak penguin diagram.

diagram or an electroweak penguin diagram[12]. New Physics contributions could influence both the penguin and the loop diagrams and enhance the process above the small branching ratio predicted by the Standard Model. The branching ratio was measured in a combined analysis of the CMS and LHCb experiments in 2014 [57]. As in the previous cases, however, no significant hints for New Physics contributions could be found.

Besides B physics, any e^+e^- collider has access to continuum processes like $e^+e^- \longrightarrow \tau^+\tau^-$, and can therefore be considered a τ factory, providing a very clean environment for tauon-decay studies. Similar to B meson pairs from $\Upsilon(4S)$ decays, τ lepton pairs can be separated into a tagging and a signal side. On the tagging side, one of the tauons is identified with kinematic constraints in a known decay; the signal side can then be used to study signal decays, including possible New Physics processes.

Tauons are particularly well-suited to study lepton-flavor violating decays. Since the discovery of neutrino oscillations, it is clear that the three known neu-

[12]The penguin diagram is not immediately identifiable as such, but it can be rearranged to a form similar to figure 2.9c.

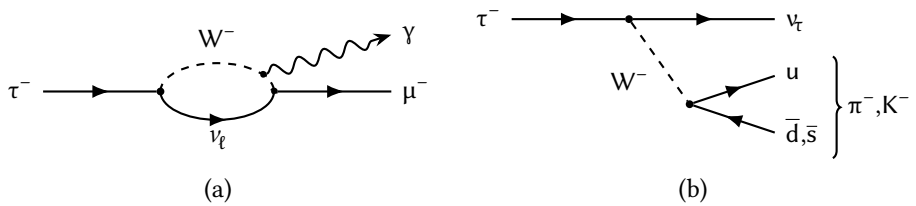


Figure 2.12: Feynman diagrams of possible τ^- decays: **(a)** the theoretical, strongly suppressed radiative penguin decay $\tau^- \longrightarrow \mu^-\gamma$ and **(b)** the semi-leptonic decay $\tau^- \longrightarrow \nu_\tau h^-$, where h^- is a π^- or K^-.

trino types must have finite, differing masses. The smallest change to the SM that incorporates the observed phenomenons explains the oscillations by a mixing mechanism very similar to that of quark mixing with the CKM matrix: the Pontecorvo-Maki-Nakagawa-Sakata matrix (PMNS matrix) [58]. This mixing mechanism makes even the charged lepton-flavor violating process $\tau^- \longrightarrow \mu^-\gamma$ possible via penguin diagrams with virtual neutrinos (see figure 2.12a). The branching ratio obtained from these diagrams, however, is extremely small [59]: It is suppressed by the factor $\Delta m^4/M_W^4$, where Δm is the neutrino mass splitting and M_W is the mass of the $W^\pm$ boson. With the current values, this amounts to a factor of at least 10^{-49}, making the decay unobservable for all practical purposes. Once again, supersymmetric particles in the loops can come to the rescue and enhance the value to observable quantities [60]. Current measurements [61] could not find large differences to Standard Model predictions. The same is true for the lepton-flavor violating decay $\tau^- \longrightarrow \mu^+\mu^-\mu^-$.

Since tauons are heavy enough to decay into hadrons, they can also provide insights to hadron-related observables through very clean processes. As an example, the decays $\tau^- \longrightarrow \nu_\tau\pi^-$ and $\tau^- \longrightarrow \nu_\tau K^-$, shown in figure 2.12b, can be used to determine the CKM matrix elements $|V_{ud}|$ and $|V_{us}|$ [11, p.651]. Discrepancies to measurements from other channels are a sign for New Physics.

These are just a few examples of how precision measurements of observables that are accessible at B-factories can provide hints to possible physics beyond the Standard Model. Past measurements have found at most indications, but no hard evidence, for deviations from Standard-Model expectations. Results from Belle, BaBar, and more recently LHCb could constrain the parameter space for New Physics models. It is clear that machines and techniques must be improved to become sensitive to effects that have so far evaded observation. In this endeavor, electron-positron colliders and hadron colliders will work complementarily: The former offer a cleaner environment that allows, for example, the investigation of final states with missing energy, whereas the latter reach higher cross sections

and energies, so that decays of $B_c^{\pm}$ or B_s^0 mesons can be studied.

In the case of B-factories, a luminosity increase is clearly the most important upgrade. It will allow the observation of rare events and improve the statistics of all measurements to more significant levels. In addition, a better vertex resolution is vital for precision measurements of B decays and time-dependent asymmetries. An improved vertex detector with a larger volume can also help to capture low-momentum pions from the decay of neutral kaons that appear in the final state of many important processes. These pions often escaped detection because they "curl" in helix-like tracks in the magnetic field around the beam pipe and never reached the outer detector layers. These changes and others—an improved particle identification system and the trigger, data-acquisition, and auxiliary systems that go along with the other upgrades—constitute a *Super B-factory*. The sensitivity of such an experiment would provide more constraints on New Physics scenarios that may allow us to exclude certain theories and pursue others. The clearest and most desirable result in this sense would be significant deviations from Standard Model predictions that give strong indications for one of the possible models.

Chapter 3

Belle II—A Super B Factory

This chapter gives an overview of the Belle II experiment as a whole, describing both the accelerator facilities and the components of the detector. The pixel detector, based on the novel DEPFET technology, and its data-acquisition system are explained in detail. The last section details why a data reduction system for the pixel detector is deemed necessary and which mechanism will be used for its implementation.

3.1 The SuperKEKB Accelerator

SuperKEKB is a direct upgrade of the KEKB accelerator that was used for the Belle experiment. It is placed in the same tunnel as KEKB and uses many of the same components, including quadrupole magnets and cavities. This section briefly explains the accelerator's main elements and explains how the improved luminosity–40 times that of KEKB—will be achieved. The information given here is mostly summarized from the Belle II Technical Design Report (TDR) [62].

Figure 3.1 shows the SuperKEKB main rings and its supporting facilities. Both rings have a circumference of 3016 m and can store 2506 particle bunches. In the high-energy ring (HER), electrons are stored with an energy of 7 GeV and a current of 2.62 A; in the low-energy ring (LER), positrons are stored with an energy of 4 GeV and a current of 3.60 A. Compared with Belle, these beam energies result in a reduced center-of-mass boost of $\beta\gamma = 0.28$. They were chosen to increase the lifetimes of the beams and reduce their emittance (the average deviation from nominal phase-space coordinates), while keeping the boost large enough for vertex determination with the upgraded detectors described in the next section.

The electrons for the HER are produced by a low-emittance photocathode RF electron gun and brought to the final energy of 7 GeV by a linear accelerator

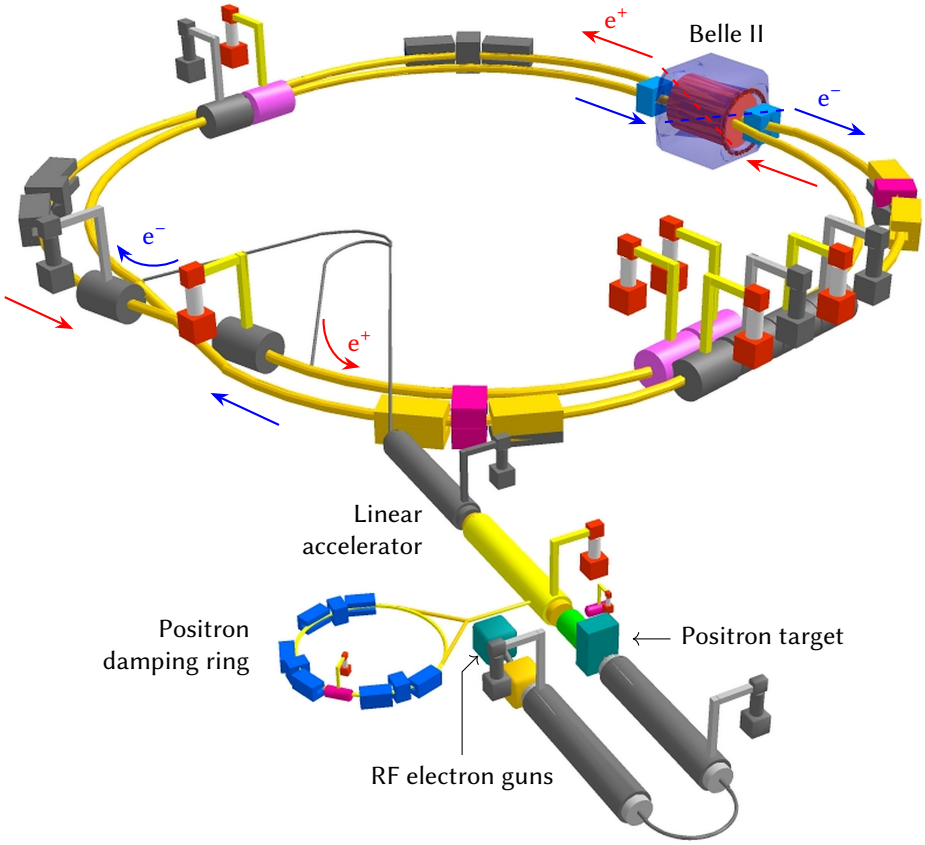


Figure 3.1: Rendering of the SuperKEKB accelerator, including the storage rings, the electron/positron source, and the linear accelerator used for injection into the main rings (Source: The SuperKEKB collaboration)

(linac). Positrons are obtained from pair production of bremsstrahlung photons, produced by firing 3.3 GeV electrons from a thermionic RF electron gun (with higher intensity but also higher emittance than the photocathode gun) on a tungsten target. The positrons are accelerated to 1 GeV, directed through a damping ring, which reduces their emittance, and further accelerated to 4 GeV before injection. Electrons and positrons are accelerated by the same linac, alternating on a pulse-by-pulse basis. The injection into the main rings uses a *continuous-injection* mode, where bunches are topped off with a frequency of 50 Hz. This scheme allows for an almost constant luminosity, but periodically introduces "noisy bunches" into the rings, causing a dramatically increased background. The noisy bunches take several milliseconds to reach design emittance through radiation damping. This leads to a detector dead time, during

which all triggers are blocked, of up to 1 ms (5 % of the time); during an additional period of about 2.5 ms (12.5 % of the time), the background in other areas has died down sufficiently, and triggers are only vetoed while the noisy bunches pass the interaction point.

The most significant change form KEKB to SuperKEKB, making the required luminosity increase possible, is the adoption of the *nanobeam* scheme. This technique was initially developed for the canceled SuperB experiment [63]. It increases the luminosity by extremely compressing the colliding beams in the vertical dimension and crossing them at a relatively large angle instead of head-on.

The luminosity for a collision of two very flat beams[1] with equal beam sizes at the interaction point is

$$L = \frac{\gamma_\pm}{2 e r_e} \left(\frac{I_\pm \xi_{y\pm}}{\beta^*_{y\pm}} \right) \left(\frac{R_L}{R_{\xi_y}} \right),$$

with quantities of either the electron beam (subscript −) or the positron beam (subscript +). γ is the Lorentz factor, e the elementary charge, and r_e the classical electron radius. The fraction of the reduction factors R_L and R_{ξ_y} is close to one. The remaining factors can be tuned to achieve a higher luminosity: the beam current I; the beam-beam parameter ξ_y, characterizing the force on a particle due to the potential created by an opposing bunch; and the vertical beta function at the interaction point β^*_y, connected to the vertical beam size at the interaction point σ^*_y and the vertical emittance ϵ_y by the relation $\sigma^*_y = \sqrt{\epsilon_y \beta^*_y}$.

While the emittance (outside of the injection-noise period) is more or less constant along the beam trajectory, magnetic lenses like quadrupole magnets can be used to manipulate the beta function at various positions of the storage ring. Naively, this would make it possible to enhance the luminosity by making the beta function at the interaction point very small. In reality, this method is constrained by the *hourglass effect*: Since the beta function can be maximally compressed only at a single point, growing quadratically with distance, only a small part of the bunch length effectively contributes to the luminosity [17, p. 387]. For bunches colliding head-on, the beta function at the interaction point can therefore *not* be made arbitrarily small, but is constrained by the bunch length: $\beta^*_y > \sigma_z$, where the bunch length σ_{z-} in SuperKEKB is about 5 mm.

In the nanobeam scheme, this effect is mitigated by choosing a finite crossing angle φ. The effective overlap length in this case is $d \approx \sigma^*_x / \varphi$, where σ^*_x is

[1] "Flat" means that the ratio of the vertical and horizontal beam sizes at the interaction point, σ^*_y / σ^*_x, is close to zero.

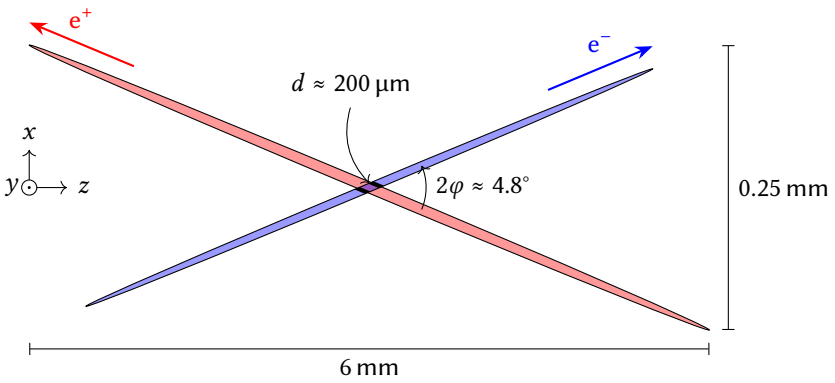


Figure 3.2: Top-view illustration of the nanobeam collision scheme. The magnitude of the beta function at the interaction point is limited by d instead of the much larger bunch length. Note that the vertical direction in this drawing is stretched by a factor of 10.

the horizontal beam size at the interaction point, and the constraint for the beta function becomes $\beta_y^* > d$. This principle is illustrated in figure 3.2. The design values for SuperKEKB are $\varphi = 41.5\,\text{mrad} \approx 2.4°$ and $\sigma_{x-}^* = 7.75\,\mu\text{m}$, so that $d \approx 200\,\mu\text{m}$, about a factor of 25 smaller than the bunch length.

With a projected beam-beam parameter of $\xi_{y-} = 0.088$, based on Belle experience, and a vertical beta function of $\beta_{y-}^* = 0.41$ mm, the design luminosity of Belle II is $L = 8 \times 10^{35}\,\text{cm}^{-2}\,\text{s}^{-1}$, 40 times that of Belle. With this value, Belle's record integrated luminosity of about $1\,\text{ab}^{-1}$ could be accumulated in a net data-taking time of only 15 days. The goal for Belle II is the collection of an integrated luminosity of $50\,\text{ab}^{-1}$.

3.2 The Belle II Detector

The Belle II detector is situated at the SuperKEKB interaction point. It is a multi-layer detector with advanced tracking, particle-identification, and energy-measurement capabilities. Subdetectors are placed in a barrel around the interaction point and an end-cap section that expands the acceptance in the forward (boost) direction. The total acceptance in the laboratory frame is 2π in the polar plane and $17° < \theta < 150°$ in the azimuthal plane, where $\theta = 0$ corresponds to the forward direction. Figure 3.3 shows the profile of the detector with its various subsystems. The individual subdetectors are briefly introduced in the following paragraphs. The information given here is summarized from the Belle II TDR. Other sources are explicitly referenced where used.

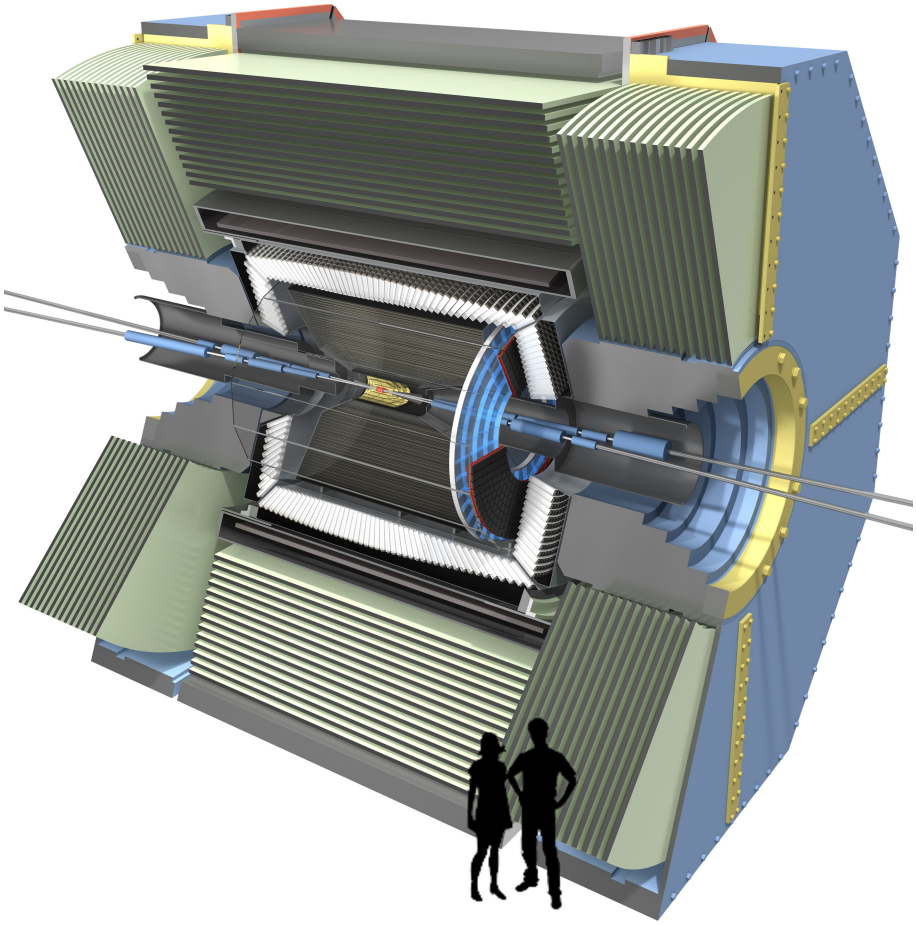


Figure 3.3: Cutaway rendering of the Belle II detector. The components, from the interaction point outwards, are: The PXD (shown in red), the SVD (shown in yellow), the CDC (shown as gray wires), the iTOP (shown as transparent slabs in the barrel region), the ARICH (shown as bluish tiles in the forward region), the ECL (shown as white crystals), the solenoid coil, and the KLM (shown as green sandwich structure). The various subdetectors are explained in the text. (Source: The Belle II collaboration)

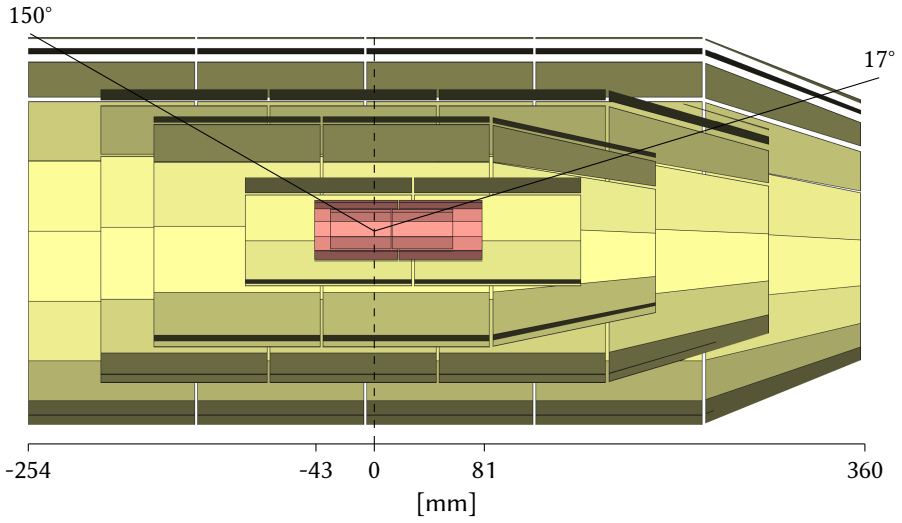
The vertex detector

The *vertex detector* (VXD) is responsible for the precise reconstruction of B meson decay vertices close to the interaction point. It consists of six silicon-detector layers in a barrel configuration around the beam pipe. This arrangement is shown in figure 3.4. The inner two layers belong to the DEPFET *pixel detector* (PXD). With a total of almost 8 million pixels and a distance to the interaction point of only 14 mm, this detector delivers very precise position information, but is also susceptible to background hits. A detailed explanation of the PXD follows in the next section.

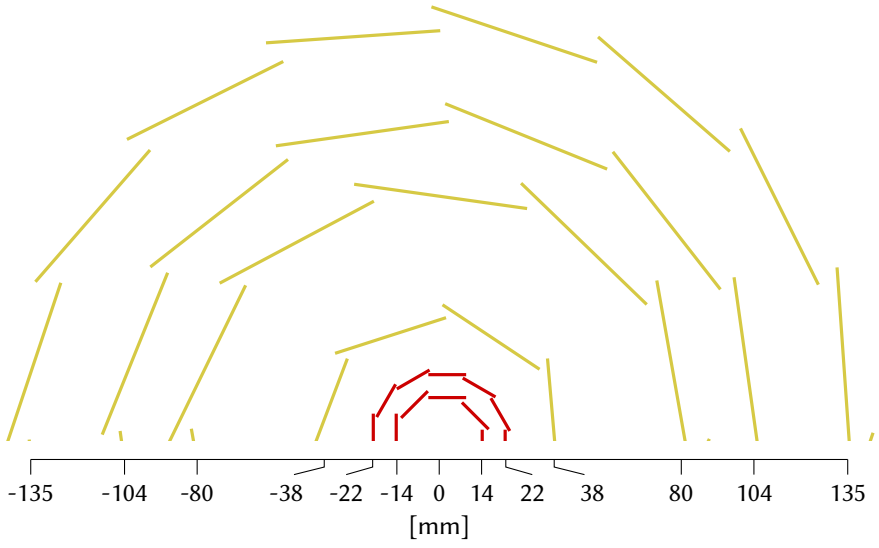
The outer four VXD layers make up the *silicon vertex detector* (SVD) [64], the direct successor of the Belle vertex detector. The SVD is a double-sided silicon strip detector made from an n-doped bulk region, implanted on one side with long, highly p-doped strips parallel to the beam and on the other side with short, highly n-doped strips perpendicular to the beam. Charged particles passing through an SVD module's bulk region produce electron-hole pairs via ionization. The p-n strips are reverse biased, so that the electrons drift to the nearest n-strip and the holes drift to the nearest p-strip. The signals generated by each strip are amplified, shaped, buffered in an analog pipeline, and digitized by a Flash ADC upon a trigger.

The SVD consists of 187 sensors with a thickness of 300 µm. There are three basic module shapes: rectangular modules for the inner SVD layer with a size of 122.8 mm × 38.4 mm; rectangular modules for the three outer SVD layers with a size of 122.8 mm × 57.6 mm; and trapezoidal modules with a size of 122.8 mm × 57.6–38.4 mm. The latter are used as slanted modules for the most forward sensors of the three outer SVD layers in order to cover the complete acceptance region of Belle II (see figure 3.4a). All SVD sensors have 768 long p-strips. The sensors of the innermost layer have 768 short n-strips, while all other modules have 512 short n-strips. The modules of both the PXD and SVD are arranged in an overlapping "windmill" structure (see figure 3.4b). This geometry minimizes inactive gaps at the sensor junctions and ensures that most particle tracks originating from the interaction point must pass through an active detector surface in every VXD layer.

Compared with the PXD, the SVD has a shorter read cycle, but, like every strip detector, it is subject to *ghost hits*: If n particles pass the detector at the same time at different vertical and horizontal positions, n p-strips and n n-strips will register a signal. In the reconstruction, all n^2 combinations of firing p- and n-strips must be considered as the possible cause for the signal. In a very high background environment, many strips will fire at the same time, making strip detectors virtually useless. Pixel detectors are more suitable in this case, but they are more expensive, more complicated to read out, and large sensors are



(a)



(b)

Figure 3.4: Geometry of the VXD. Only active detector surfaces, excluding slanted SVD modules, are shown. **(a)** Cutaway side view, showing the detector size and acceptance. The origin corresponds to the interaction point. The boost direction is to the right. The two central PXD layers are shown in red; the four outer SVD layers are shown in yellow. **(b)** View along the beam direction from the low-energy (e^+) side, showing the windmill structure of the modules.

harder to manufacture. The combination of both technologies for the Belle II VXD is a compromise between these concerns. The expected resolution for the z-coordinate of B decay vertices—the most important observable for the measurement of time-dependent CP asymmetries—is in the order of 20 µm.

The central drift chamber

Surrounding the vertex detector in the barrel region of Belle II is the *central drift chamber* (CDC) [65]. The CDC extends over a length of about 2.4 m and covers the radial region between 160 cm and 1130 cm. It consists of eight superlayers that are assembled alternatingly as axial layers (with wires parallel to the beam) and stereo layers (with wires skewed by an angle between −74 mrad and 70 mrad). Each superlayer is composed of six individual layers of sense wires (eight for the innermost superlayer), made from gold-plated tungsten. The total number of sense wires is 14 336.

The CDC volume is filled with a gas mixture of 50 % helium and 50 % ethane (He-C_2H_6). Since the CDC is inside the magnetic field of the Belle II solenoid, a charged particle passing through this volume moves on a helical trajectory and ionizes the gas atoms along its path. In the electric field between the CDC's field wires and the sense wires, the liberated electrons drift toward the sense wires, where they produce a current signal. The signal timing, in combination with the known drift velocity, allows the determination of the particle's distance from the firing wire. The trajectory of the particle in the x-y plane (perpendicular to the beam) can then be reconstructed by fitting the individual hits to a circle. The stereo layers additionally allow the determination of the z coordinate.

The CDC can match vertex-detector hits with signals of the outer detectors. Its most important application is the determination of a particle's momentum from the bending radius of its trajectory in the magnetic field. Since a particle deposits only a small amount of energy in the spacial region of each sense wire, the mean energy loss in all drift cells along the particle's trajectory allows the determination of the energy loss per distance, dE/dx. This quantity can be used for particle identification, since the correlation of dE/dx and momentum depends on the particle type.

The resolution for the transverse momentum p_t with the CDC is expected to be similar to or better than the one achieved by Belle, $\sigma_{p_t}/p_t = 0.0019 p_t[\mathrm{GeV/c}] \oplus 0.0030/\beta$ [66]. The CDC is furthermore an important trigger source for Belle II (see section 3.4).

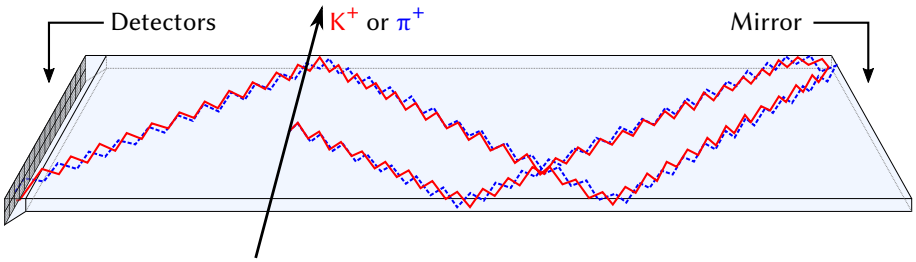


Figure 3.5: Working principle of the Belle II iTOP: A charged kaon and pion, traversing the detector at the same position and with the same momentum, produce Cherenkov radiation under different angles. The photons emitted by the heavier, slower kaon have a smaller angle (red, solid line). They arrive at the detector plane after a longer time and at a different position than the photons emitted by the pion (blue, dashed line). (Picture adapted from reference [68])

The particle identification system

Belle II will use two separate types of *ring-imaging Cherenkov* detectors (RICH) for particle identification in the barrel and end-cap regions. These detectors exploit the Cherenkov light radiated by a particle moving through a medium with a velocity β greater than the speed of light in that medium. Such a particle emits photons under the angle $\theta = \arccos(1/n\beta)$ relative to its direction of motion, where n is the medium's refractive index. Since the emission is isotropic in the polar direction, the Cherenkov radiation originating from any one point propagates forward as a light cone. If this cone hits a screen after a known distance d, its projection is a ring with radius r. Knowing n and d, the particle's original velocity can be determined by measuring r.

The *imaging time-of-propagation* detector (iTOP) [67] in the Belle II barrel region is a special type of *detection of internally reflected Cherenkov light* detector (DIRC). In a DIRC, the Cherenkov light cone is not allowed to propagate freely from its point of origin. The radiator material and geometry are chosen in such a way that most Cherenkov photons undergo total internal reflection when they reach the medium boundary. This effect is used to guide the produced light out of the radiator at a distant position, where detectors can be placed conveniently. The radiators of the iTOP are fused-silica quartz bars with a size of 2.7 m × 450 mm × 20 mm. Sixteen such bars are placed around the CDC (see figure 3.3).

Figure 3.5 shows the geometry of a single module. On one end of the bar, a 100 mm long prism expands the vertical dimension of the radiator to 51 mm. Two rows of 16 micro-channel plate photomultiplier tubes (MCP-PMTs) are attached directly to the end of the prism to detect the arriving photons. On the

opposite end of the bar, a spherical mirror reflects the Cherenkov light to the detector plane, focusing photons that arrive under the same angle.

The MCP-PMTs are divided into 4 × 4 individual segments, so that each bar is read out by 64 × 8 channels. The sensors have a very good timing resolution of about 50 ps, allowing the precise measurement of the time between the bunch crossing in the collider and the Cherenkov-light detection. This time difference corresponds to the sum of the charged particle's time-of-flight and the time-of-propagation of the Cherenkov photons. The combination of timing and two-dimensional position information allows a reconstruction of the Cherenkov ring. In practice, iTOP hits are associated with CDC tracks to determine the momentum, position, and angle of impact on the quartz bar. The measured coordinates are compared with predictions based on the assumption that the particle was a kaon or a pion. The result is a likelihood with an expected fake rate that is 2 to 5 times smaller than the one achieved by Belle.

The *Aerogel ring-imaging Cherenkov* detector (ARICH) [69] used in the end-cap region is a simpler RICH detector with an expansion volume: Particles pass through a 4 cm thick Aerogel radiator. Hexagonal Aerogel tiles are arranged to cover the area at a distance of of 167 cm from the interaction point between radii of 410 mm and 1140 mm. Following the radiator is an approximately 20 cm wide gap where the Cherenkov light cone is allowed to spread. The photons are then detected by nine radial layers with a total of 540 *hybrid avalanche photo detector* (HAPDs), each of which is is segmented into 12 × 12 pads.

The thickness of the Aerogel tiles is a trade-off between the number of produced photons and the resolution of the Cherenkov angle measurement: Making the radiator thicker increases the number of Cherenkov photons, improving the chance to obtain a reconstructable ring image; at the same time, it smears the radius, because the charged particle can emit photons at any point of its path through the radiator, resulting in a ring with non-zero thickness. The ARICH radiators are therefore divided into two 2 cm thick pads with refractive indices 1.055 and 1.065. The parameters are chosen in such a way that the second Aerogel pad produces photons with a larger Cherenkov angle; since it is also closer to the detectors, the emitted light cone overlaps with the one from the first radiator at the detector plane, enhancing the signal yield without adding to the uncertainty of the measurement.

The electromagnetic calorimeter

The Belle II electromagnetic calorimeter (ECL) is a homogeneous calorimeter that reuses most components of Belle's ECL. It is separated into a 3 m long barrel part directly behind the iTOP, a forward end-cap at $z = 196$ cm, and a backward end-cap at $z = -102$ cm. The ECL consists of 8736 crystals made from

thallium-doped caesium iodine (CsI(Tl)) and cut into truncated-pyramid shapes. Photodiodes capture the scintillation light produced by traversing particles. Their output signals are amplified, shaped, and digitized by Flash ADCs.

The main application of the ECL is the measurement of photon and electron energies with an expected resolution of

$$\frac{\sigma_E}{E} = \sqrt{\left(\frac{0.066\,\%}{E/\mathrm{GeV}}\right)^2 + \left(\frac{0.81\,\%}{\sqrt[4]{E/\mathrm{GeV}}}\right)^2 + (1.34\,\%)^2}.$$

It is also one of the main trigger sources for the experiment (see section 3.4).

The K_L^0 and muon detector

The previously described Belle II subdetectors are located inside the field of a 1.5 T superconducting solenoid magnet. The cylindrical magnet coil surrounds the ECL in the barrel region. It has a radius of 1.7 m and a length of 4.4 m in z-direction. A yoke constructed from iron plates serves as a flux return path for the magnetic field. The plates are interspersed with detector layers, constituting a sampling detector with a sandwich structure. This *K_L^0 and muon detector* (KLM) identifies passing muons and long-living neutral kaons with high efficiency.

In the barrel region, forward end-cap, and backward end-cap, 14 iron plates with a thickness of 47 mm serve as absorbers that slow down traversing muons and cause kaons to initiate hadronic showers. The gaps between the plates are 44 mm wide. In the barrel region, 15 *resistive plate chambers* (RPCs) are placed in the gaps. They consist of 2.4 mm thick float glass plates with a very high electrical resistivity. A high voltage is applied between two plates separated by 1.9 mm, and the gap is filled with a gas mixture. Traversing charged particles ionize the gas, leading to an avalanche breakdown and formation of a streamer (an ionized, conductive channel) between the electrodes. Electric charge flows from one plate to the other along the streamer, causing a local discharge, limited in size by the glass's resistivity. The affected region is left blind until the charge is restored over a time scale of seconds. External pickup strips with a width of 5 cm register the discharge as a signal. Each module is a superlayer of two back-to-back RPCs with pickup strips in perpendicular directions. Discriminators process the signals, detect hits, and pass them to the data-acquisition system.

The higher particle flux in the end-caps, especially due to background events, make the inherent RPC dead time unacceptable. In the gaps between the absorber plates in the end-caps, 14 superlayers of organic scintillator strips with a width of 40 mm are therefore used. Each superlayer consists of two layers with

orthogonal strips. Wavelength-shifting fibers transport the scintillation light to silicon photomultipliers (SiPMs) for detection. The SiPM signal is amplified, digitized, and processed by frontend electronics.

Muons are identified by tracks in the CDC that have corresponding KLM hits. In order to distinguish muons from charged hadrons, the range of a track with the given momentum in the KLM is predicted for a muon. The comparison of the observed range with the prediction gives a muon likelihood. The remaining false positives are mostly misidentified charged pions. K_L^0 are identified by hits in the KLM without corresponding CDC tracks. The kaon can initiate a hadronic shower in either the ECL or KLM. To qualify as a K_L^0 candidate, at least two clusters, projecting roughly to the interaction point, are required: two in the KLM or one in the KLM and one in the ECL. A crude momentum reconstruction for the K_L^0 is possible using time-of-flight information from the KLM.

3.3 The DEPFET Pixel Detector

Background sources

The geometrical arrangement of the 40 PXD modules was explained in the previous section. This section gives a more comprehensive overview of the sensor layout, the employed technology, and the read-out system. An exhaustive report with detailed information about the PXD can be found in the *PXD Whitebook* [70].

The PXD consists of two radial layers at very small distances (14 mm and 22 mm) from the interaction point. In order to make meaningful predictions about the feasibility, performance, and expected data rate of the detector, it is necessary to have a good understanding of the processes that contribute to the background in this region and pollute the PXD data. These processes fall in one of two categories: *luminosity-dependent* processes and *beam-induced* processes.

Luminosity-dependent processes are QED reactions of electrons and positrons from the opposing beams. Their rate is directly proportional to the instantaneous luminosity and will therefore increase by a factor of 40 compared with Belle. Nevertheless, the impact on the PXD performance cannot be exactly predicted based on past experience, since Belle did not have a comparable detector.

The final state of the *two-photon process* $e^+e^- \longrightarrow e^+e^-(\gamma^*\gamma^*) \longrightarrow e^+e^-e^+e^-$ usually contains two low-energetic particles that curl in a helical track close to the beam pipe; they produce signals in the PXD but do not reach the outer detectors. This process is expected to be the dominant source of background for the PXD by far. The second important QED background process is *radiative*

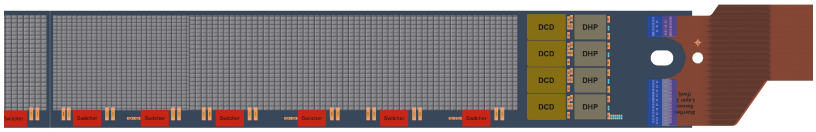


Figure 3.6: Rendering of a PXD half-ladder for the forward direction in the outer layer. The print is life-sized. The cross-hatched area corresponds to the thinned, active surface. ASICs are bonded to the surrounding frame. The module for the backward direction is attached on the left side. The Kapton cable on the right provides outward connections. (Picture courtesy of K. Ackermann, MPI Munich)

Bhabha scattering: electron-positron scattering, with the emission of one or more photons, $e^+e^- \longrightarrow e^+e^- + n\gamma$. While the particles in the final state of this process are usually emitted under very small angles, they can hit the beam pipe and generate secondary particles which are scattered back to the PXD.

Beam-induced processes, on the other hand, take place within the two separate beams and do not stem from their collisions. They are largely proportional on the beam current, which is only increased by a factor of approximately 2.2 compared with Belle. While this means that these backgrounds will not scale as much as the luminosity-dependent processes, their rate can have additional dependencies. This is especially true for the *Touschek effect*, the intra-bunch scattering of two electrons or two positrons. The scattered particles can leave the accelerator's acceptance and collide with the beam pipe, producing showers that can be seen as clusters in the PXD. This process is inversely proportional to the beam size, which is extremely small at the interaction point to facilitate Belle II's nanobeam scheme.

Other beam-induced processes are *beam-gas scattering*—Coulomb scattering and bremsstrahlung following the collision of a beam particle with a residual gas molecule inside the beam pipe—and *synchrotron radiation*, mainly produced by the electrons and positrons in the focusing magnets around the interaction point. With the exception of beam-gas bremsstrahlung, all of these processes are expected to contribute measurably to the PXD background.

The exact contribution of the background to the fraction of firing pixels in the PXD (the occupancy) depends strongly on the integration time: the length of the time window during which the detector collects charge that is associated with a single trigger or event. The impact will therefore be discussed after the working principle and read-out of the PXD have been explained.

DEPFET principle

Figure 3.6 shows the layout of a single half-ladder. The sensors are produced from an n-type silicon wafer through a large number of processing steps. They are 15.4 mm wide, 420 µm thick, and have a length of 68.0 mm for the inner layer and 85.0 mm for the outer layer. The active area is thinned down to only 75 µm. It is 12.5 mm wide and has a length of 44.8 mm for the inner layer and 61.44 mm for the outer layer. The *application-specific integrated circuits* (ASICs) required for control and read-out are bump-bonded to the surrounding rim, which also provides mechanical stability to the half-ladder. Supply voltages, control signals, and data enter and leave the module through a flexible printed circuit (Kapton) cable.

Each sensor has an array of 768 × 250 pixels, making a total of 7 680 000. In the horizontal direction (measuring θ), the 256 pixels closest to the interaction point are smaller than the remaining 512 pixels to accommodate for the higher track density in this area. The respective pixel widths are: 55 µm and 60 µm in the inner layer; and 70 µm and 80 µm in the outer layer. The height of all pixels (measuring φ) is 50 µm.

Each pixel is a single *depleted field-effect transistor* (DEPFET). This technology was proposed for use as a particle detector in 1986 [71]. In contrast to pixel detectors used by particle-physics experiments in the past, charge generation and amplification in a DEPFET pixel take place in a single, monolithic structure, avoiding charge loss and making it less susceptible to noise. Over the last decade, large DEPFET sensors have been developed and implemented for the first time for the use in Belle II, the International Linear Collider (ILC), and other experiments [72, 73].

The structure of a DEPFET pixel is shown in figure 3.7. It is based on the principle of an enhancement-mode p-channel *metal-oxide-semiconductor field-effect transistor* (pMOSFET): Two strongly p-doped regions implanted in the n-type substrate serve as the source and drain contacts. A negative drain-source voltage does not lead to a current flow because the charge carriers of the source (holes) cannot drift to the drain through the depletion regions at the two pn-junctions[2]. The situation changes when an additional contact (gate) is added at the bulk surface between source and drain. The gate is electrically isolated from the substrate by an oxide layer, but a sufficiently strong negative gate-source voltage attracts holes to the bulk surface, establishing a conductive channel. If the drain-source voltage is kept constant, the current through this channel can be steered with the gate-source voltage.

[2]For this simplified explanation, I assume that there is no potential difference between the source and the bulk. This is usually not the case for the DEPFET operation.

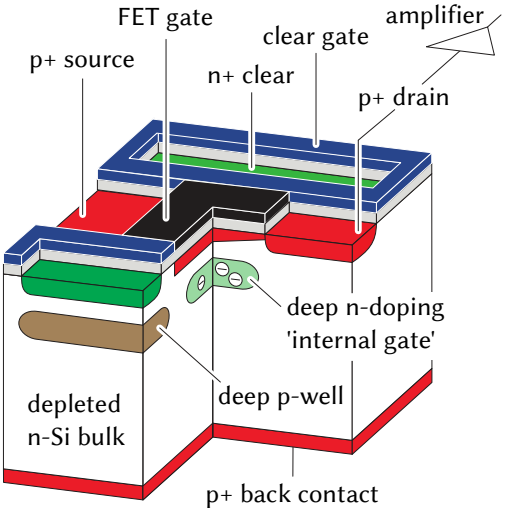


Figure 3.7: Schematic view of a single DEPFET pixel. The purpose of the various regions and contacts is explained in the text. (Source: The DEPFET collaboration)

A MOSFET becomes a DEPFET pixel by the addition of a strongly p-doped back contact. This contact is biased with a very negative voltage, causing the entire bulk volume to be depleted of charge carriers. The device is now sensitive to ionizing radiation: When electron-hole pairs are created in the bulk by a charged particle, the holes immediately drift to the back contact. The electrons, on the other hand, move toward a potential minimum—formed by the applied voltages and an additional strong n-doping—directly underneath the gate contact. They are trapped at this location, modulating the gate potential with their charge. When the gate is active, the magnitude of the drain current is a measure for the number of electrons generated by the traversing particle. The potential minimum is therefore referred to as the *internal gate*.

After the current has been read out, the electrons must be cleared out of the internal gate for the next charge-collection period. To this end, a strongly n-doped contact (clear) can be biased with a very positive voltage, creating a conductive path for the captured electrons to the clear electrode. The clear contact is embedded in a p-doped region (deep p-well) that prevents electrons from drifting to clear instead of the internal gate during charge collection. An additional clear-gate contact can be used to fine-tune the potential barrier between the internal gate and the clear electrode and speed up the clear process.

Read-out

The pixels on a DEPFET sensor are arranged in a matrix with 250 long columns and 768 short rows. Logically and electrically, this layout is rearranged by grouping four rows into a row group, resulting in an array of 1000 logical columns and 192 logical rows. This reduces the read-out time for the whole sensor, as will become apparent in the following paragraphs.

All drain outputs of the pixels within a logical column are connected to the same drain line that is read out by a single ADC channel. Only one pixel per column can therefore have an active gate voltage and drive the drain line at a time. This is achieved by a rolling-shutter read-out scheme: The gates of all pixels within a logical row are driven by the same voltage. The same goes for the clear inputs. After charge collection (inactive gate and clear), a row is read out (active gate and inactive clear) and the collected charge is removed (active gate and clear). Then the processing steps to the next row.

The read-out process for a half-ladder is performed by three types of ASICs [74]: Six SWITCHERs, four *data handling processors* (DHPs) [75] and four *drain current digitizers* (DCDs) [76][3]. The SWITCHERs are mounted on the thin rim on the long side of the half-ladder. Each SWITCHER drives the gate and clear voltages for 32 pixel row-groups, and thereby controls the timing of the read-out process. The DCDs are mounted on the inactive silicon surface on the end of the pixel columns. This area is outside of the detector acceptance, where active cooling can be supplied. Each DCD amplifies and digitizes the current from 250 column drain lines using 8-bit ADCs. In order to compensate dark currents offsets (pedestals) for the individual pixels, it can subtract a coarsely configurable current before digitization. This increases the dynamic range of the ADCs.

Each DCD connects to a DHP mounted next to it. The DHP receives the digitized pixel data and stores them in a ring buffer. This buffer always contains the raw data from one complete sensor frame, up to the current row. It acts as a programmable delay element: When a trigger arrives, data is read from a variable relative position in the buffer. This mechanism makes it possible to adapt the read offset to the trigger delay in the experiment, so that all hits belonging to the triggered event are read.

The DHP reduces triggered data in a two-step process: First, it subtracts digital pedestal offsets from each ADC value, accounting for shifts that could not be taken care of with the CDC's rough analog compensation mechanism. Next, it uses a two-pass process to calculate the average remaining zero offsets for all pixels of a row segment sampled at the same time. This so-called *common-mode*

[3]The ASICs are sometimes referred to with version numbers (e.g, DCDv2) or Belle-specific names with an appended "B" (e.g., SWITCHERB). These suffixes are omitted in this thesis.

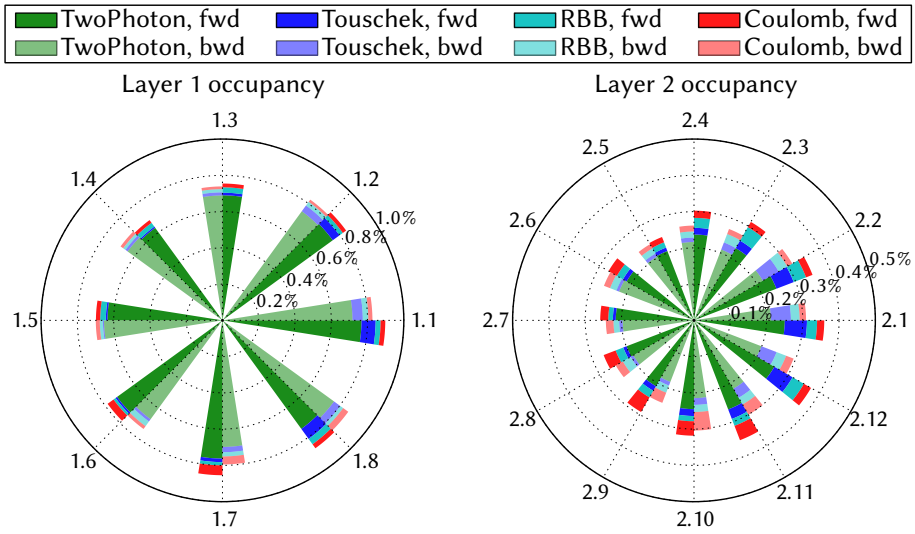


Figure 3.8: Contributions to the PXD background occupancy from various processes. Synchrotron background is not included. Note the different plot scales for the two layers (up to 1 % for the inner and 0.5 % for the outer layer). (Picture courtesy of M. Ritter, LMU Munich, from the 9th background campaign)

noise is also subtracted from all pixels. Only pixels with a non-zero ADC value after these steps are put in an output queue and sent to the data-acquisition system. This process is called *zero suppression*.

Pedestal values for every pixel are stored in the DHP memory. They are continuously calculated by a part of the data-acquisition system and uploaded to the DHP. In order to produce data for these calculations, the DHP provides an alternative read-out mode where it outputs a complete, unprocessed frame containing the raw ADC values of all pixels. The values from a number of these frames are averaged and used as pedestals. Random triggers should be used to initiate a full-frame read-out because pixels with real hits complicate the pedestal calculation. A possible opportunity for this is a special trigger that is sent to all detectors shortly before a new SuperKEKB injection. Triggers are blocked for some time after the injection, so this period can be used to perform the time-consuming process.

The DCD's sampling frequency constrains the read-out time for a single row group to about 100 ns. The number of row groups is 192, so each pixel is read out approximately every 20 µs. All hits accumulated during this time contribute to the detected charge. This includes signal as well as background hits. The signal contribution depends on the Belle II trigger rate, which will be discussed in the next section. The contribution from background processes

can be estimated from simulations. Recent results show that the maximum occupancy for any half-ladder is well below 1 % for the inner PXD layer and below 0.5 % for the outer PXD layer, dominated by two-photon processes (see figure 3.8). As a conservative estimate, we assume a value of 3 %.

The long integration time of the PXD brings an additional problem: As discussed earlier, the injection of new particles into the SuperKEKB main rings causes a period of increased background from noisy bunches. During the latter part of this time interval, triggers are blocked only for collisions of these bunches. The revolution time for each bunch is approximately 10 µs, so that subdetectors with a significantly shorter integration time can take data during this phase. For the PXD, however, every read-out frame will be polluted by background from noisy-bunch crossings. This makes all data taken during a time of about 3.5 ms after every injection (every 20 ms) unusable, leading to an effective dead time of 17.5 %.

The so-called *gated mode* is a new feature of the Belle II DEPFET sensors that was developed to counter this problem. It uses the established pixel design, but requires a new version of the SWITCHER that is able to drive the correct voltages for the new mode. In gated mode, the clear voltages for all pixels of the sensor are driven to a very positive voltage, while the gates stay inactive. In this configuration, electrons from newly created electron-hole pairs are strongly attracted to the clear electrode instead of being collected in the internal gate. Electrons collected before are unaffected and are kept in the internal gate by the positive voltage on the external gate. If the gated (or blind) mode is activated during the noisy bunch crossings, signal hits from events outside this time can still be collected, leading to a significantly reduced dead time.

3.4 Trigger and Data Acquisition

The trigger system

Events that are of interest for the Belle II physics program can be associated with certain characteristic responses of the various subdetectors. Several subdetectors are therefore equipped with configurable logic circuits (FPGAs—see section 4.1) that allow a coarse but very fast online analysis of the detector output. They compare the observed values and event topologies to the signatures of signal events. A match indicates that an interesting event may have occurred, and the subdetector logic issues a trigger signal.

The CDC and ECL are the two most important trigger sources. Many signal events involve at least two charged particle tracks reaching the drift chamber, so track multiplicity is a first good discriminator. At the same time, bogus triggers can be rejected by requiring that the tracks originate from the region around

the interaction point. A constraint on the z coordinate can therefore reduce the number of background triggers. The ECL checks the number of clusters, their energy, and other variables to generate triggers. This also allows online luminosity measurement: Well-known electromagnetic processes like $e^+e^- \longrightarrow e^+e^-$ (Bhabha scattering) and $e^+e^- \longrightarrow \gamma\gamma$ (electron-positron annihilation) leave characteristic signatures in the ECL. By measuring the rate of these processes, it is possible to calculate the instantaneous luminosity based on the known cross-sections. Additional triggers come from the ARICH and iTOP, both of which can provide good timing information, and the KLM, which identifies events with muon tracks.

The *global decision logic* (GDL) is the final arbiter for the trigger decision. It collects the trigger signals and summarized topology information from all subsystems and makes a global trigger decision based on this input. Upon a positive decision, the so-called *level-1 trigger* signal is is distributed by a network of *frontend timing switches* (FTSWs) [77] to to about 1000 destinations at the detector frontends. The level-1 trigger has a unique combination of a 32-bit trigger number (sometimes called event number; incremented for every issued level-1 trigger), an 8-bit subrun number (incremented in case of a locally recovered subsystem error that does not require a stop of the complete data-acquisition system; in this case, the trigger number is not reset), a 14-bit run number (incremented for a cold start of all subsystems; in this case, both the subrun number and trigger number are reset), and a 10-bit experiment number (incremented for major changes of the detector or accelerator, if background or cosmics data is recorded, if the beam energy is altered, etc.).

The allowed trigger delay in this architecture, from bunch collision to the arrival of the trigger at the subdetectors, is 5 µs. The GDL enforces a minimum temporal separation of 200 ns for two triggers. The sensitivity of the trigger logic is a compromise between a high efficiency for the detection of signal events and a good background rejection to limit the experiment's output data rate. For events with B decays, an efficiency of more than 99 % is expected.

The level-1 trigger rate for signal processes in Belle II can be predicted using the known cross section of the reactions and the SuperKEKB luminosity. The rate is dominated by two-photon processes with final state particles in the detector acceptance and transverse momenta over 100 MeV/c (~15 kHz). Another large contribution comes from continuum production of hadrons (2.2 kHz) and muon or tauon pairs (640 Hz each). The rate from B meson pair production from $\Upsilon(4S)$ decays is 960 Hz. Together with the (downscaled) calibration triggers, a total rate of 20 kHz is expected. The data-acquisition system is therefore designed to cope with a rate of up to 30 kHz.

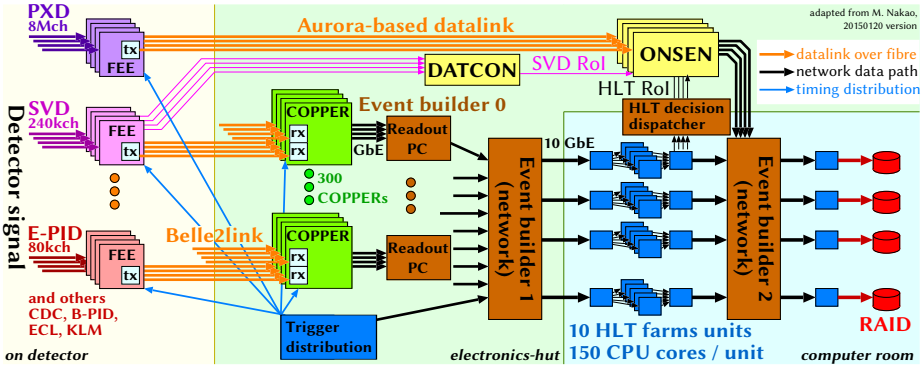


Figure 3.9: Simplified diagram of the Belle II data-acquisition system. Subsystems exclusive to the PXD data-acquisition, like the DATCON and ONSEN, are introduced later. (Original figure courtesy of M. Nakao, KEK)

Data acquisition for the outer detectors

The Belle II data-acquisition system collects the data for triggered events from all subdetectors, merges the data packets belonging to each trigger number (event building), and writes them to permanent storage. Its architecture is shown in figure 3.9.

The data acquisition for all subdetectors *except* the PXD (referred to as *outer detectors* here) has similar requirements. In particular, the data rates are moderate (from 2.6 kB per event for the KLM to 14.9 kB per event for the SVD [78]), so that data streams can be processed by CPU-based systems. In order to reduce cost and development effort, a unified architecture is therefore used for all systems. The individual frontend-electronics (FEE) boards of each subsystem perform the read-out and preprocessing of the triggered detector data. They are equipped with FPGAs that use a common transmission logic to encode the data with the unified Belle2Link serial protocol [79] and send it to a *common pipeline platform for electronics readout* (COPPER) board [80]. The COPPER boards obtain timing and trigger signals from the FTSWs. They can receive data from multiple frontend boards and perform local event building on a CPU. A readout PC collects the data from several COPPERs over point-to-point Ethernet connections, checks them, and packs them into an event fragment. This stage is also known as *event builder 0* (EB0). In a final step, the *event builder 1* (EB1) PC farm receives the output from all readout PCs over an Ethernet network and combines all data belonging to an event into a single packet.

Including a safety factor of 2, a rough estimation puts the total level-1 data rate of all outer detectors at 2.5 GB/s. Before the permanent-storage stage, this amount is further reduced by the *high-level trigger* (HLT) [81] PC farm. The

HLT performs a full online event reconstruction, including tracking in the CDC and SVD and association of particle tracks with ECL clusters. With this information, background can be further reduced by applying cuts on the vertex position, energy deposition, and other variables. The remaining events are filtered based on physics content relevant for analyses (skim). The HLT trigger rate is assumed to be 10 kHz, corresponding to a reduction factor of 3.

The HLT processes many events in parallel. Depending on the event topology, the processing time per event can vary between several milliseconds and (applying a safety margin) five seconds. The average processing time is expected to be below one second. Events that pass the HLT selection are forwarded to the *event builder 2* (EB2), where they are combined with PXD data and put to permanent storage. Up to this point, the dataflow is completely decoupled from the one of the PXD. Due to the parallel processing, events arriving at the EB2 are in general not in trigger order.

Data acquisition for the PXD

Knowing the level-1 trigger rate, $f_\mathrm{t} = 30\,\mathrm{kHz}$, the PXD's output data rate can be estimated: As discussed before, we assume a background occupancy of 3 %. To compare this to the occupancy from signal events, we must first calculate how many triggers can occur during one PXD integration period. The minimum separation between two triggers is 200 ns, but otherwise, every $\mathrm{e}^+\mathrm{e}^-$ bunch crossing in the accelerator (every 4 ns) has a chance to produce a signal event. The probability p_n to have n triggers during one read-out interval $t_\mathrm{r} = 20\,\mu\mathrm{s}$ therefore approximately follows a Poisson distribution:

$$p_n = \frac{(f_\mathrm{t} t_\mathrm{r})^n \, \mathrm{e}^{-f_\mathrm{t} t_\mathrm{r}}}{n!}$$

The resulting chances are: 55 % for no trigger, 33 % for one trigger, and 12 % for two or more triggers.

We make an extremely generous estimate for the signal data rate by assuming one event per PXD frame, 50 tracks with hits in both layers, and a cluster with 50 firing pixels per hit. Even with these numbers, the signal contribution to the occupancy is well below 0.1 %. We see that the background dominates the PXD's occupancy completely, and we can neglect signal contributions in discussions of the data rate.

Assuming the worst-case hit distribution, approximately 2.5 bytes are needed to encode the position and ADC value of a fired pixel in the zero-suppressed format (see appendix section C.1.7 for a more detailed analysis). We arrive at a raw data rate of $3\,\% \times 7\,680\,000 \times 2.5\,\mathrm{B} \times 30\,\mathrm{kHz} \approx 17.3\,\mathrm{GB/s}$. During data acquisition, metainformation is added to the data packets, including DHP

and sensor information, trigger numbers, and checksums. Another contribution comes from the full-frame read-outs required for pedestal calculations. The actual data rate is therefore slightly larger, and we assume 18 GB/s. Compared with the summed-up data rates for the outer detectors, the PXD data rate is about an order of magnitude higher. The unified data-acquisition and permanent-storage systems cannot handle this amount of data without being scaled up by an unfeasible degree. As the PXD produces mostly background, a specialized data-acquisition system an an additional online reduction mechanism are needed.

The data-acquisition chain of the PXD begins with the read-out of the triggered data by the *data handling hybrid* (DHH)[4] system [82]. The DHH consists of 48 modules equipped with FPGAs and DDR3 memory. All modules use the same hardware design, but perform two separate functions: The 40 *data handling engines* (DHEs) are connected to the PXD half-ladders via InfiniBand cables, attached to the sensor Kapton cables through a patch panel. A DHE configures the ASICs on each half-ladder, initiates the read-out following a level-1 trigger, and receives the pixel data. As a consequence of the PXD's rolling-shutter read-out mode, the data can start at any pixel row and wrap around to the top during the frame. Multiple triggers during one PXD frame pose another complication, since the hits in the overlapping region can belong to two distinct events. The DHH resolves these problems by reordering the data, so that frames start with the first firing row, and duplicating frame segments between events in case of overlapping triggers. In addition, it provides the option to perform clustering on the hits found by each DHP. In this mode, neighboring fired pixels are combined into a single logical entity (cluster) in the output data. This feature can be used during later data-processing steps.

The DHEs combine the processed data from the four DHPs into a single data packet and send it to one of the eight *data handling concentrators* (DHCs). Each DHC provides clock and trigger interfaces to five connected DHEs, receives their output data, and performs a 5-to-4 multiplexing. The process is illustrated in figure 3.10: Each DHC carries out a local subevent building by combining the event data of the five connected DHEs into a single packet. The packets are then sent out on one of four links, alternating on an event-by-event basis. With this mechanism, it is possible to average out the varying data rates between PXD modules in different detector regions. As an example, each DHC could process the data from three modules of the outer layer and two modules of the

[4] The naming of this system has changed recently. Previously, the 40 read-out boards were called DHH (now DHE) and the 8 concentrator boards were called DHHC (now DHC). This led to some ambiguity, because the term "DHH" could refer to the system as a whole as well as the individual boards.

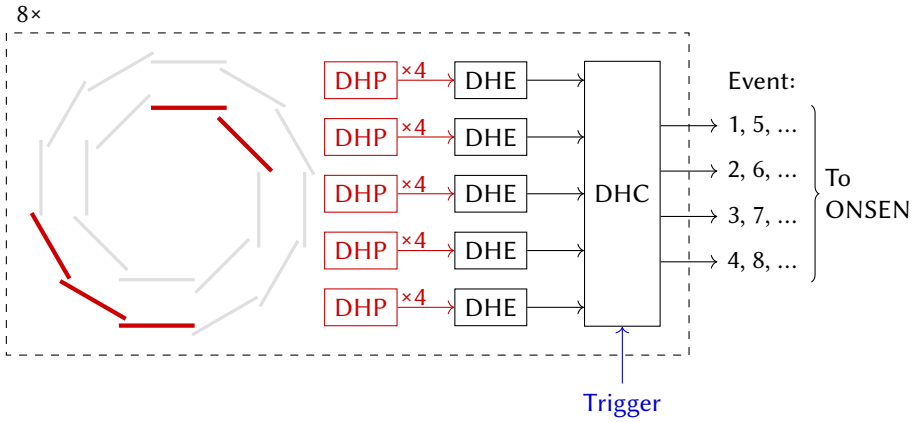


Figure 3.10: Dataflow for the PXD modules: A DHE configures and reads out each half-ladder. A DHC collects the data from five DHE and sends them out on one of four links, depending on the event number. To cover all 40 PXD modules, 40 DHEs and 8 DHCs are needed.

inner layer, where the expected occupancy is much higher. The connections can be rearranged to adapt to the observed data rates in the experiment.

Assuming equal flows on all DHC output links, each link sees a maximum data rate of 18 GB/s/32 = 562.5 MB/s. The outbound data transport is accomplished with multi-gigabit optical links running at 6.25 Gbps. With the 8b/10b-encoded Aurora protocol (see section 4.3), a payload data rate of almost 625 MB/s can be achieved. On the receiving end of this connection is the ONSEN system, which will be introduced in the next chapter.

Data reduction

The first step in the reduction of the PXD data is identical to that for the outer detectors: the downscaling of the event rate by the HLT decision, corresponding to a reduction factor of 3. Since the PXD data are not processed by the unified data-acquisition scheme and therefore not included in the EB1 output, a different data path must be established. It is clear that the read-out of the PXD sensors cannot be delayed until the completion of the HLT decision, for which a time of up to five seconds is allowed. The read-out must therefore be based on the full 30 kHz level-1 trigger, and the data must be stored by a specialized subsystem, which waits for the HLT output and sends only triggered data to the EB2 for integration with the data from the outer detectors. The first requirements for a system that handles the PXD data reduction are therefore a large input bandwidth, compatible with the 6.25 Gbps outputs from the DHH; an output

interface to the EB2, preferably based on a TCP/IP Ethernet connection; and sufficient memory capacity and bandwidth. With the previously calculated raw data rate, a system that retains all PXD data for twice the average HLT processing time (2 s) must have a total memory capacity of 36 GB. This translates into 1.125 GB per DHH output link.

From the background-dominated PXD data alone, not much meaningful information can be extracted. In particular, it is difficult to determine whether any one hit stems from a signal track or a background process[5]. The PXD clusters become useful when they can be associated with already reconstructed particle tracks from the SVD and CDC. These detectors have a much lower background level, so that particle trajectories can be determined without an overwhelming amount of fake tracks. If these trajectories are extended with PXD clusters very close to the interaction point, the vertex resolution is greatly enhanced.

With this application of the PXD hits in mind, the natural extension of the data reduction is to discard those hits that *cannot* be associated with an SVD or CDC track. It is implemented as an extension of the track-finder algorithm running as part of the online event reconstruction on the HLT. The concept is illustrated in figure 3.11: Particle tracks, found by the HLT based on hits in the SVD and CDC, are extrapolated back to the interaction region, giving an estimate of the vertex position. The PXD hits produced by the particle are expected near the position where the extrapolated trajectory intercepts a PXD sensor. The area around the intercept is therefore marked as a *region of interest* (ROI). The ROI information, together with the HLT decision, is the basis for the operation of the data-reduction system.

Ideally, the rejection of all PXD hits that are not inside an ROI would dispose of the majority of the data stemming from background and keep only the very small amount of signal-related hits. In reality, the reduction depends on the number of reconstructed tracks, including fakes, and the quality of the track reconstruction. The resolution of the track position at the point of the PXD intercept determines the size of the created ROI, so that the ROI can be guaranteed to include the relevant pixels. The requirement form the event-building system is a reduction factor of 10 at this stage, meaning that about a tenth of the sensor areas can be covered by ROIs. With the combination of both concepts, a total data-reduction factor of 30 is achieved. This puts the reduced PXD data rate at 600 MB/s, or 18.75 MB/s for a each DHH output link. Taking into account the unreduced raw-data read-outs, the result is closer to

[5] There is an ongoing development of a “cluster-rescue” mechanism that tries to accomplish just this by an FPGA-based online analysis of cluster properties based on a neural network. It is meant as a possible extension to the ROI mechanism described here.

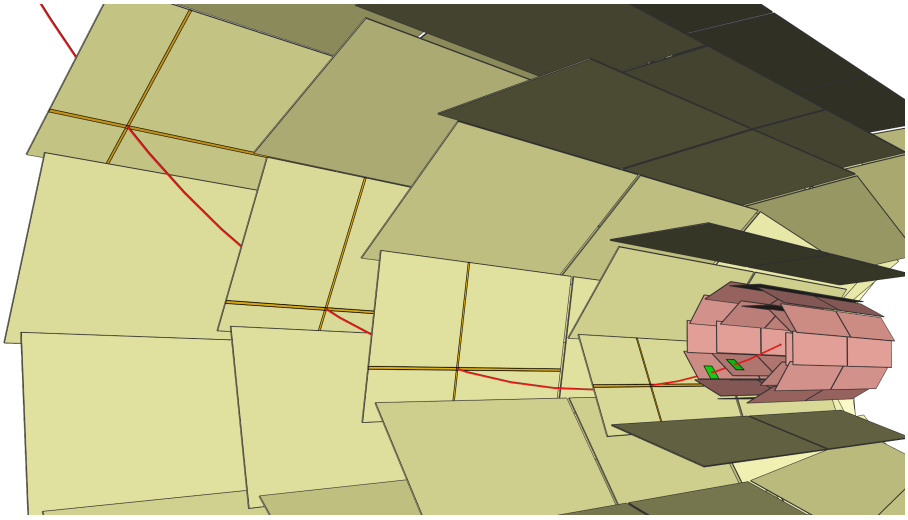


Figure 3.11: Illustration of the ROI generation mechanism, showing the active surfaces of the PXD and half of the SVD: A charged particle moves outward from the interaction point on a helical trajectory, leaving hits in the strips of every SVD layer. This allows the reconstruction of the particle's track and an extrapolation to the two PXD layers. The most probable intercept positions with the PXD are inferred, and regions of interest are defined around them.

30 MB/s (see appendix section C.4.4).

A second set of ROIs for every event with a level-1 trigger is produced by the *Data Acquisition Tracking and Concentrator Online Node* (DATCON) [83]. This FPGA-based system uses the same hardware platform as the PXD data-reduction system described in the next chapter. The DATCON works exclusively on SVD data. It uses a sector-neighbor finder and a fast Hough transformation to reconstruct SVD tracks, extrapolates the tracks to the PXD layers, and defines ROIs around the intercepts, similar to the HLT concept. The output from the DATCON for each event will be ready in a much shorter time than that from the HLT. The two ROI sources are redundant, and the selected areas must be logically combined, but only the HLT can select an event for storage. DATCON ROIs for events that do not pass the HLT's event selection process are therefore ignored. The complete dataflow, including both ROI sources, is included in figure 3.9.

The ROI mechanism adds more prerequisites to the data reduction system: It must provide interfaces to the HLT (Ethernet) and DATCON (multi-gigabit optical links), and it must have sufficient processing power and throughput to perform the matching of ROIs from both sources with the correct data and

filter the hits accordingly. The ONSEN system, which we developed for this purpose, fulfills or exceeds all of the mentioned requirements. The next chapter discusses this system in detail.

Chapter 4

The ONSEN Data-Reduction System

The work on this thesis revolved around the design and implementation of the *ONSEN* data-reduction system for the Belle II pixel detector. This chapter gives a comprehensive description of the hardware used by the ONSEN system, the overall architecture, and the design of the component modules. It begins with an introduction to the concept of FPGA-based data processing and a comparison with other approaches, followed by an overview of the *Compute Node* hardware platform. Then the developed firmware is explained, focusing on the data processing mechanisms, input and output, memory management, and various central design aspects. The last section gives an outlook to remaining tasks and possible future changes. Where details were deemed too involved or disruptive, they were moved to the appendix and referenced at the relevant locations.

4.1 Hardware Platform

A list of requirements for the Belle II subsystem responsible for the buffering and reduction of PXD data was presented at the end of the previous chapter. Based on these requirements, we developed the *online selection nodes* (ONSEN) system. The ONSEN system has been presented in several peer-reviewed publications [84–86], one of which has evolved from the work on this thesis. It is capable of buffering the unprocessed PXD data for an average time of more than 2.5 seconds, combine ROIs from the two inputs, perform the filtering of the selected pixels, and pass the processed data on to the Belle II event-building system. These features are achieved by using a hardware platform based on *field-programmable gate arrays* (FPGAs).

FPGAs as data processors

An FPGA is an integrated circuit (IC) built from reconfigurable logic components that can be adapted to many different purposes. The basic constituents of most FPGAs are flip-flops (1-bit storage elements) and lookup tables (LUTs). LUTs are function generators with a fixed number of inputs; they can can be understood as versatile *n*-bit logic gates that can be configured to produce any Boolean function with *n* or fewer inputs. The combination of flip-flops and LUTs allows the construction of complex sequential logic circuits that can be modeled at the register-transfer level (RTL): a number of synchronously clocked register stages (flip-flops), where the output from one stage is processed by combinational logic (LUTs) and used as input for the next stage. Flip-flops and LUTs are arranged in *slices*. FPGA vendors often offer differently sized devices of the same "family" where a main distinction is the available number of these slices[1]. Other design elements found in FPGAs include memory cells (block RAM), clocking components like frequency synthesizers, and even complete ICs like CPUs.

Modern FPGAs have a large number of general-purpose input/output (I/O) ports that support different signal standards, such as 3.3 V *low voltage transistor-transistor logic* (LVTTL) or 1.8 V *stub-series terminated logic* (SSTL). This allows connections to a large range of external devices, including memory chips, clock sources, and Ethernet physical transceivers (PHYs). In addition, some FPGAs offer *serializer/deserializer* (SerDes) blocks and differential transceivers for data transfer over high-speed serial links. These links range from standard *low-voltage differential signaling* (LVDS) ports, supporting data rates up to about 800 Mbps, to *multi-gigabit transceivers* (MGTs) that can reach 10 Gbps and more. MGTs can be used for connections to various standard I/O interfaces like PCI Express and 1000BASE-X Gigabit Ethernet (GbE).

To program an FPGA for a certain function, configuration bits must be written into SRAM cells that control the state of the various components and the connections between them. These RAM cells are volatile, which implies that the FPGA must be reconfigured after every power cycle. The configuration data is stored in a file called *bitstream*, which can be downloaded into the FPGA by one of several different methods. Some of these will be explained below.

The bitstream is the final product of a series of design processes that usually involve the use of vendor- and device-specific software on a PC. Different

[1]Since the ONSEN system uses FPGAs from the largest vendor, Xilinx, I use Xilinx jargon throughout this thesis and make generalizations that are sometimes not completely valid for other vendors. For example, the smallest groupings of logic components, here referred to as slices, are called *logic elements* for Altera FPGAs and *logic cells* for Lattice FPGAs. Because of different design choices by the vendors, the terms are usually not interchangeable.

design methods exist, but the most common one starts with the writing of source code in a hardware description language (HDL) like Verilog or VHDL[2]. Code written in an HDL describes the behavior or structure of a logic circuit. It includes statements like: "In case of a rising edge of input CLK, assign to output Q the inverted value of input D." A synthesis tool processes this code and infers from it a *netlist*: a circuit diagram of components available in the target FPGA family. The above example describes a D flip-flop with inverted input; it could be synthesized into a circuit of a flip-flop, a LUT configured to act as an inverter, two input buffers, and one output buffer. In a subsequent step, the elements of the netlist are mapped to the primitives available in the selected FPGA. If the design is too complex, so that enough resources are not available, the process terminates at this point. If the mapping succeeds, each component is assigned its final location inside the FPGA (placement) and the signal paths for all interconnects are determined (routing). Place-and-route is a complex process—especially in cases where a design uses most of the available FPGA resources—made even more complex when timing constraints are taken into account.

Timing constraints ascertain the correct functionality of the produced circuit by limiting the allowed signal propagation times through logic gates, routing paths, and I/O buffers. A timing constraint could be phrased like this: "The delay induced by the logic path between the output of flip-flop A and the input of flip-flop B must be 2 ns shorter than one clock period." This condition makes sure that the output from one register stage, after being processed by a chain of logic elements, is stable when it is registered at the input of the next stage, all while taking into account inherent hardware properties like setup times and clock-to-output delays of flip-flops. In order to obtain a realistic timing analysis, the designer must supply the correct constraints. This can be complicated for external connections, where the timing of PCB traces and other ICs plays a role; for internal paths, it is usually sufficient to specify all clock frequencies and adhere to proper design practices. A failed constraint often means that the desired clock frequency is too high for the produced circuit. A successful design has a timing score of 0, meaning that the cumulative time by which the timing constraints for all signals are violated is 0 ps.

Simulations are an additional method for verifying design integrity and discovering possible problems. They are performed by programs that are supplied with information about the simulated circuit, stimuli, which define the values of the circuit's inputs all all times of the simulation, and possibly device-specific information like logic delays and flip-flop switching characteristics. The simulator produces waveforms that represent the outputs produced by the circuit

[2]The "V" stands for VHSIC (very high speed integrated circuit).

in reaction to the stimuli. It can also provide insight into the state of internal signals. Simulations can be made at various points of the design process: In a *behavioral simulation*, the pure HDL code is interpreted. This can help to identify logic flaws at the RTL level. If device-specific information is supplied, the synthesized netlist and even the circuit after place-and-route can be analyzed. Such a *timing simulation* is much slower, more complicated to set up, and only allows limited analysis of internal signals and logic integrity. It gives, however, an extremely realistic representation of the circuit's behavior and can help to find problems that are not caught in the constraint-based timing analysis.

Comparison to other technologies

As processors in a real-time data acquisition system, FPGAs must be compared to *application-specific integrated circuit* (ASICs), *central processing units* (CPUs), and *general-purpose computing on graphics processing units* (GPGPU). ASICs are silicon chips designed for an explicit purpose, like the front-end electronics chips that perform the digitization and zero-suppression for the PXD. The ASIC development process is similar to that of FPGAs, but ASIC designs are not limited to the available resources and technologies of any given FPGA. They can include analog components and can be run with higher clock frequencies than FPGAs, as their trace lengths are not determined by the versatile FPGA routing infrastructure. On the down side, the design effort for ASICs is much higher and requires greater experience. The reprogrammability of FPGAs makes them more forgiving of errors: A new hardware version of an ASIC can easily come with a five- or six-figure price tag, whereas an FPGA can be reprogrammed at no cost. ASICS become economic especially for large production volumes, as their non-recurring engineering costs start out higher than that of FPGAs, but scale less steeply for higher unit quantities. For a data processing system, the reprogrammability of FPGAs is an especially desirable feature, as the system's functionality can be expanded, algorithms can be changed, and features can be added at a later point.

Processing farms based on CPUs or GPUs offer the greatest flexibility, and their programming process is arguably the most familiar. The GPGPU scheme has gained popularity in recent years for tasks that require a large number of simple computations performed in parallel. It can be argued that the pixel filtering is such a task, since it can be split into many individual pixel-to-ROI comparisons taking place at the same time. Both CPU and GPU systems, however, rely on PC systems that provide the I/O and memory framework for the processed data. In order to grant sufficient bandwidth for the complete PXD output data, a large-scale PC system would be needed. In addition, the type of the inbound data links is defined by the DHH to be multi-gigabit serial

links using the Aurora link-layer protocol. For input into a PC system, a special extension card, most likely based on an FPGA, would be needed in any case. Using FPGAs as data processors is therefore the most direct solution and provides the best balance of processing power, flexibility, and ease of programming.

The xTCA architectures

A second fundamental design choice for a custom hardware platform is the selection of a hardware architecture. The use of an industry standard reduces cost and design effort, because established solutions for power supply, cooling, hardware monitoring, and module interconnection can be used. Architectures used by physics experiments in the past include the VMEbus, FASTBUS, and CAMAC standards. In the light of the requirements of current experiments, these architectures are no longer up to par. Their data transfer bandwidth, in particular, is not sufficient for the high output rates of modern detectors. Many institutes have therefore switched their hardware-development efforts to standards that were developed for the telecommunication industry by the PCI Industrial Computer Manufacturers Group (PICMG) consortium. These include the *Advanced Telecommunications Computing Architecture* (ATCA or AdvancedTCA) [87], the *Advanced Mezzanine Card* (AMC) standard [88], and the *MicroTCA* architecture [89]. These specifications are sometimes summarized under the umbrella term xTCA.

The PCB of an ATCA board is 322.25 mm high and 280 mm deep. An ATCA shelf offers slots with a width of 30.48 mm for up to 16 boards and supplies them with power and cooling. Figure 4.1 shows a typical 14-slot shelf with a full-mesh backplane. In such a shelf, each board is connected to every other board with four bi-directional, differential links. Each link is referred to as a *port*, and the four ports between two boards constitute a *fabric channel*. In addition, each slot has a 5-port *update channel* to one of its neighbors in the shelf. Two *base channels* support Ethernet connections from every slot to the two central hub slots, where network switches can be placed. From the rear side of the shelf, smaller add-on boards with a depth of 70 mm, so-called *rear transition modules* (RTMs), can be inserted and connected to an ATCA front board through a custom connector.

ATCA shelves offer high reliability by making most critical systems hot-swappable, redundant, or both. Each shelf is controlled by a Shelf Manager module, which keeps track of all present *field-replaceable units* (FRUs) like ATCA boards, cooling fans, and power input modules. The Shelf Manager communicates with intelligent FRUs via the *Intelligent Platform Management Interface* (IPMI) [91]. Each ATCA board needs an IPM Controller (IPMC) that handles

Figure 4.1: Photograph of a 14-slot, full-mesh ATCA shelf (Schroff 11592-402 [90]). ATCA carrier boards with four single, full-size AMC cards are present in slots 5 and 6. A base-channel GbE switch with two single, mid-size AMC cards is present in the first hub slot (slot 7). The right half of the shelf is left open to show the backplane with the blue power connector (Zone 1) and the fabric- and base-channel connector (Zone 2). The open space above the backplane can be used for the custom RTM connector (Zone 3). A 1 kW, 48 V power supply sits on top of the shelf. The Shelf Manager is plugged into a dedicated bay right of ATCA slot 14.

Figure 4.2: Photograph of a 4-slot MicroTCA shelf with fixed backplane connections between the AMC ports of the slots (Schroff 11850-013 [92]). A single, full-size AMC card is plugged into each slot. The slots in this shelf are extra-wide to allow access to the cards for testing purposes.

this process. The IPMC reads out local sensors and reports parameters like power consumption and temperature to the Shelf Manager. Based on the global information from all subsystems, the Shelf Manager decides which boards are allowed to be powered, how fast each cooling fan runs, and whether an alarm should be issued to an external control instance.

The AMC standard adds another hierarchy layer to the system. AMCs are smaller cards that are plugged into special ATCA carrier boards. They come in different sizes: The height is either a quarter of an ATCA board height (single module) or a half of an ATCA board height (double module). The width is a half of an ATCA slot width (compact), two thirds of an ATCA slot width (mid-size), or a full ATCA slot width (full-size)[3]. The module depth is 180.6 mm. Power and data connections are provided by a defined AMC connector on the carrier board. This connector has 170 pins and offers 20 bi-directional, differential links called *AMC ports*. The ports are numbered 0 through 20 (excluding port 16, which has been repurposed in a revision of the specification). The remaining pins are used for management signals, five differential clocks (a *fabric clock* and four *telecom clocks*), and *Joint Test Action Group* (JTAG) signals for in-system testing and configuration of microchips.

For hardware management, a *Module Management Controller* (MMC) is required on every AMC. Its function is similar to that of an ATCA board's IPMC. The MMCs communicate with the carrier IPMC, which manages all inserted cards and relays relevant information to the Shelf Manager. The MicroTCA

[3]The ratios given here are only approximate, as gaps and tolerances must be taken into account.

standard adds the possibility to operate individual AMCs in a small shelf instead of an ATCA carrier board. Figure 4.2 shows a simple 4-slot MicroTCA shelf. The AMC ports in a MicroTCA shelf are either routed between slots in a fixed, shelf-dependent pattern, or connected in a star-topology to a *MicroTCA Carrier Hub* (MCH) in a special slot. The MCH routes data between the other cards, based on standard protocols like Ethernet and PCI Express, and is the central hardware-management entity, similar to the Shelf Manager in an ATCA shelf.

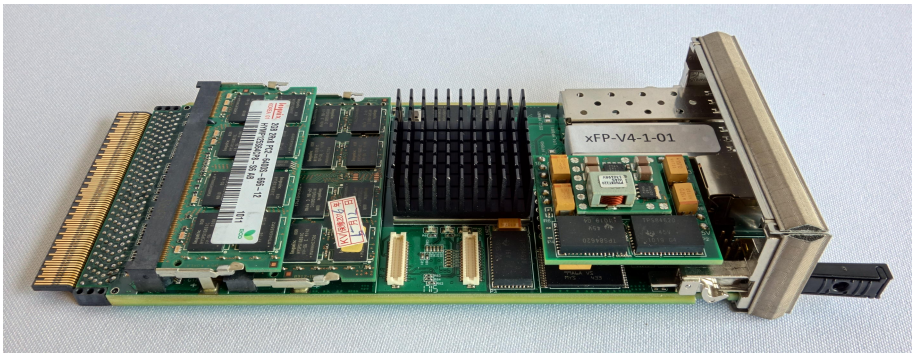
The Compute Node

The *Compute Node* (CN) is an FPGA-based data processing platform. It was developed by the Trigger Lab of the Institute of High Energy Physics (IHEP) in Beijing, China, in collaboration with our institute. The CN was originally conceived as a single ATCA board with five Xilinx Virtex-4 FPGAs for the data acquisition system of the $\overline{\text{P}}$ANDA experiment [93]. Since it was chosen as the hardware platform for the ONSEN system, the CN has been developed and used in parallel for both the $\overline{\text{P}}$ANDA and Belle II projects in Gießen.

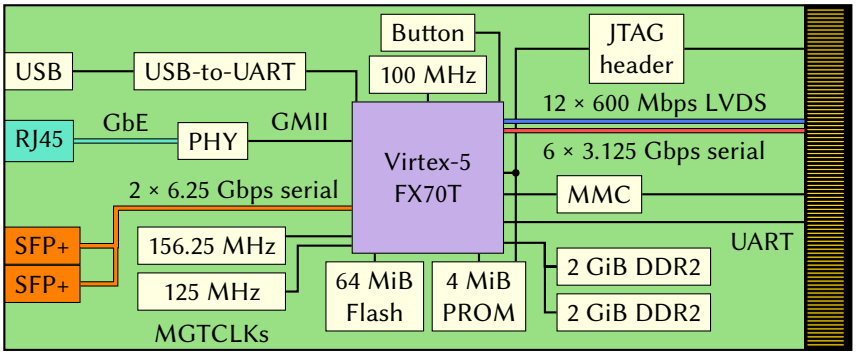
In its current incarnation [94], the CN consists of an ATCA carrier board with a single Xilinx Virtex-4 FPGA and up to four single-height, full-size AMC cards with Xilinx Virtex-5 FPGAs. In the following, the carrier board will be referred to as *Compute Node carrier board* (CNCB). The AMC card is called *xTCA-based FPGA Processor* (xFP).

Figure 4.3a shows an xFP of hardware revision 4.0. The FPGA on the xFP is a commercial-grade Xilinx Virtex-5 FX70T with speed grade −2 in an FFG1136 package. This FPGA contains 11 200 slices with a total of 44 800 6-input LUTs and 44 800 flip-flops. It has 148 block RAM units with a total capacity of 5328 kbit. I/O is provided by 640 general purpose pins and 16 GTX transceivers that support line rates of up to 6.5 Gbps. A PowerPC 440 CPU and four Ethernet MACs are included as embedded cores in the silicon chip. More details on the FPGA can be found in the device data sheet [95] and user guide [96].

A schematic view of the xFP's components is shown in figure 4.3b. The xFP has two SO-DIMM sockets designed for DDR2 SDRAM modules. 64 MiB of Flash memory can be used for non-volatile storage; a bitstream for automatic downloading upon power-on can be stored in a 4 MiB Xilinx Platform Flash chip. Two SFP+ cages connect to MGTs of the FPGA. They can be used either with an optical transceiver for high-speed serial links or with a 1000BASE-T Ethernet transceiver that allows GbE connections using standard copper network cables. Six more MGTs, as well as twelve bi-directional LVDS links, go to AMC connector ports. A GbE PHY chip and RJ45 connector provide an additional Ethernet connection. For debugging and monitoring purposes, a low-speed serial connection (universal asynchronous receiver/transmitter, UART) between



(a)



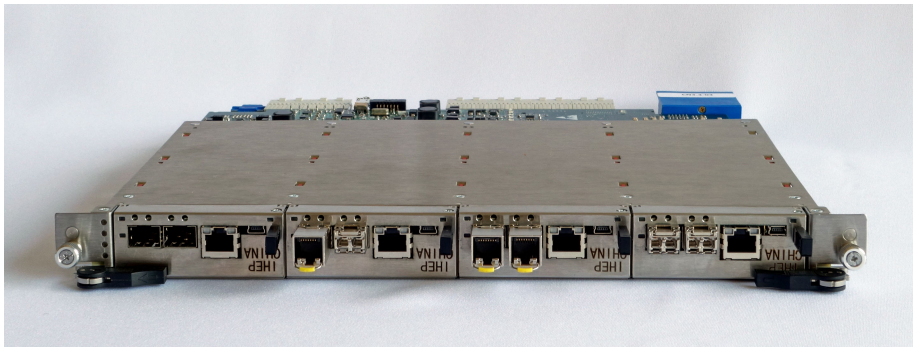
(b)

Figure 4.3: The xFP v4.0 AMC card. **(a)** Photograph of a card equipped with two DDR2 SO-DIMM modules. **(b)** Schematic view of the card's components. Not shown: power supplies, LEDs, sensors, and AMC clocks.

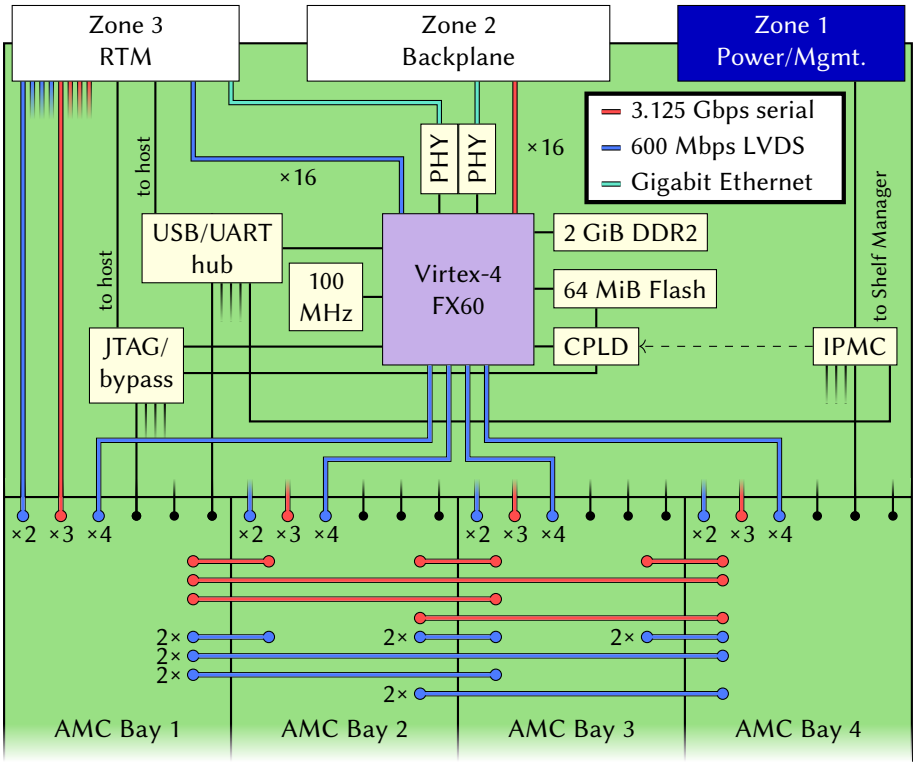
a PC and the FPGA can be established through a USB socket and UART-bridge chip. A second UART connection is accessible over the AMC connector. More details about the xFP are given in appendix section D.1.

Figure 4.4a shows a CNCB of hardware revision 3.3. The FPGA on the CNCB is a commercial-grade Xilinx Virtex-4 FX60 with speed grade −11 in an FFG1152 package. This FPGA contains 25 280 slices with a total of 50 560 4-input LUTs and 50 560 flip-flops. It has 232 block RAM units with a total capacity of 4176 kbit. I/O is provided by 576 general purpose pins and 16 GT11 transceivers that support line rates of up to 6.5 Gbps. Two PowerPC 405 CPUs and four Ethernet MACs are included as embedded cores in the silicon chip. More details can be found in the device data sheet [97] and user guide [98].

A schematic view of the CNCB's components is shown in figure 4.4b. Sim-



(a)



(b)

Figure 4.4: The CNCB v3.3 carrier board. **(a)** Photograph of a board equipped with four xFP cards and different SFP transceivers. **(b)** Schematic view of the board's components. Not shown: power supplies, LEDs, buttons, MGT clocks, sensors, serial programming chain, and clock fan-out.

ilarly to the xFP, the CNCB has 64 MiB of Flash, but only a single SO-DIMM socket for RAM. Its FPGA connects to the AMC connector of each of the four AMC bays with 4 bi-directional LVDS links. A GbE PHY connects the FPGA to the first ATCA backplane base channel. All 16 MGTs of the FPGA are routed to fabric channels. In a full-mesh ATCA shelf equipped with CNs, this layout makes it possible to route data packets from the FPGA on any xFP to any other one through the connections provided by their carrier boards. The FPGA on the CNCB is therefore referred to as the *Switch FPGA*.

Three AMC ports of each AMC bay are used to create a threefold full-mesh interconnection between the inserted cards. On the xFP, two of these ports are LVDS links and one is an MGT. Five more ports of each AMC (two LVDS and three MGT) are routed to the RTM connector. The Switch FPGA also has 16 LVDS links and an additional Ethernet interface via a second PHY connected to the RTM.

While the CNCB does not come with a Platform Flash chip for bitstream storage, it provides a different mechanism for automatic programming of both the Switch FPGA and the FPGAs on plugged-in xFPs. This method is based on a second programmable chip on the CNCB, a *complex programmable logic device* (CPLD). The structure of a CPLD is much simpler than that of an FPGA. It consists of an array of AND- and OR-gates arranged in *macrocells*. The CPLD on the CNCB is a commercial-grade Xilinx XC95144XL with speed grade −10 in a TQG100 package. It contains 144 macrocells with 3200 usable gates and 144 registers. More details can be found in the device data sheet [99].

The CPLD's configuration memory is non-volatile. Once programmed, the logic automatically starts whenever power is supplied to the device. On the CNCB, this is used to load a file, containing a combined bitstream in a special format, from a configurable offset in the Flash memory and program it to the Switch FPGA and up to four xFP FPGAs. For this process to work, the FPGAs must be placed in a slave-serial configuration daisy chain. The mechanism is explained in more detail in appendix section D.2.5. A set of backup bitstreams can be placed at a different address, and the IPMC can be used to initialize the configuration of either version. During the work on this thesis, a programming file for the CPLD was developed from scratch.

The CNCB places the Switch FPGA, CPLD, and devices on all inserted AMCs in a single JTAG chain. Empty AMC bays are decoupled by a bypass chip. UART connections to the Switch FPGA, the IPMC, and the UART pins of inserted xFP cards are all aggregated by a USB hub. The host-side signals of the JTAG chain and USB hub are routed to the RTM connector. A simple, passive RTM, called *xIO*, was designed and built by the IHEP group as an add-on to the CNCB. It provides sockets for external connections to the JTAG chain, the USB hub, and the Switch FPGA's secondary Ethernet interface.

The power supply for the CNCB is located on a detachable add-on board (CN_PWR). In this thesis, the term CNCB refers to the complete carrier board, including the power supply.

Hardware development

The following timeline shows the development from the first prototype of the CN to the latest hardware revisions, which will be used for the ONSEN system. The dates indicate the finalization of each design version by the developers from the IHEP:

- **Jan 2008:** CN v1.0
- **Dec 2008:** CN v2.0
- **Oct 2010:** CN v2.1
- **Dec 2010:** xFP v1.0
- **Oct 2011:** CNCB v3.0
  CN_PWR v1.0
- **Dec 2011:** xFP v2.0
- **Feb 2012:** CN_PWR v1.1
- **Dec 2012:** xFP v3.0
- **May 2013:** CN_PWR v1.2
- **May 2014:** CNCB v3.1
- **Nov 2014:** CNCB v3.2
  xFP v4.0
- **May 2015:** CNCB v3.3
  xIO v1.0

During the work on this thesis, a lot of effort was invested into the commissioning, testing, and debugging of these boards. Based on first-hand experience with the board prototypes (often as one of the first users) and careful reviews of the hardware designs, I was able to identify a number of critical design issues. These included routing errors like swapped signals, incorrect supply voltages, incompatible I/O standards, wrong capacitor and resistor values, falsely placed or omitted components, and more. Solutions to all problems were discussed and developed in close cooperation with the developers from the IHEP. As a result, the boards use for the ONSEN system are now in a mature state and provide the required functionality. An exception is the CNCB's power supply board; an additional hardware iteration is needed that fixes a possible incompatibility of the board's sensor interfaces with the ATCA hardware monitoring standard. In addition, the RTM requires a repositioning of one of its connectors to fit the required form factor.

For IPMI-based hardware management, an IPMC for the CNCB and an MMC for the xFP are required. An IPMC and the accompanying microcontroller firmware were previously developed for version 2.0 of the CN [100, 101]. They are, however, not compatible with the current CNCB, and the firmware did not include all features required by the ATCA specification. An MMC for the xFP was developed and built by the IHEP group, but lacks any IPMI functionality. The Belle II group at the University of Mainz is currently producing a new hardware revision of the IPMC and developing a new MMC from scratch. A reference MMC firmware implementation available from DESY [102] will be used as a basis for the firmware of both controllers.

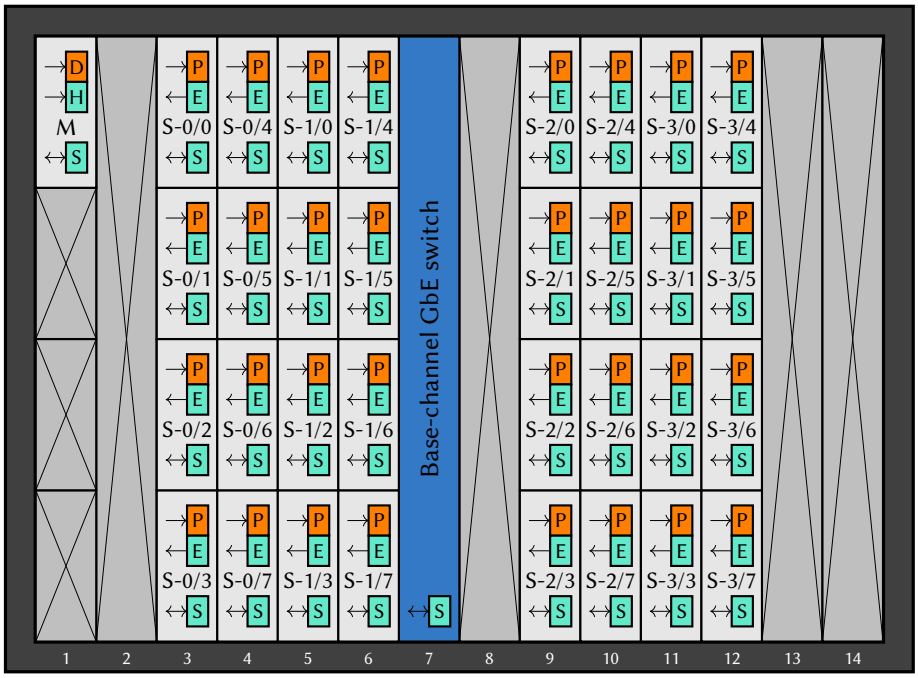
4.2 System Architecture

Overview

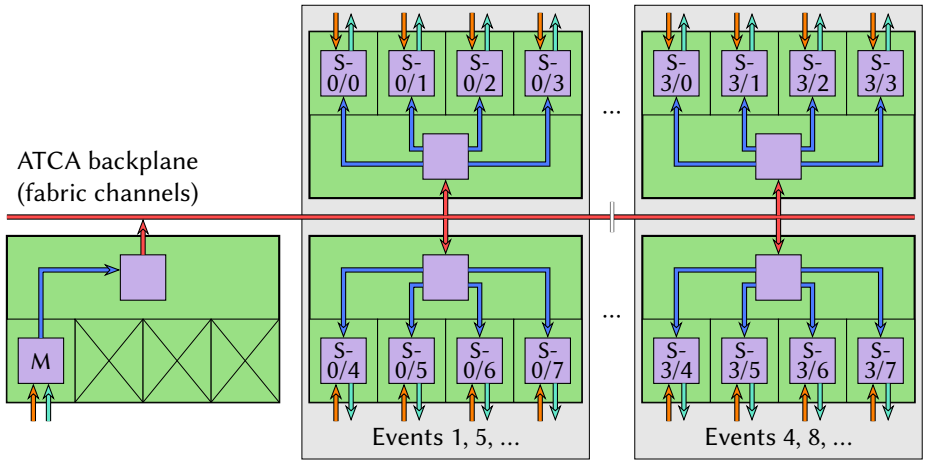
The ONSEN system uses 33 xFP cards and 9 CNCBs as carrier boards in a 14-slot, full-mesh ATCA shelf. The arrangement is shown in figure 4.5a. One of the xFPs acts as the *Merger* node. It has two inputs: an optical fiber link that receives all ROIs sent by the DATCON system, and a GbE link that receives all ROIs and trigger decisions sent by the HLT. Both sources are guaranteed to provide (possibly empty) ROI packets for every level-1 trigger, but the DATCON output is ready in a matter of microseconds, whereas the HLT output can be delayed by several seconds. The Merger therefore buffers the DATCON ROIs for every event until the HLT packets with the same event number arrive. Then it combines both into a single packet and sends it to the Switch FPGA, which distributes the packet to the other boards in the system A closer look at the operation of the Merger node will be given below.

The remaining 32 xFPs, hosted by the other 8 CNCBs, are the *Selector* nodes. Their purpose is the reduction of the PXD output data using the combined ROIs. Each Selector receives the pixel data from one of the four optical-fiber outputs of a DHC. This corresponds to the load-balanced data from five PXD half-ladders for every fourth level-1 trigger (see figure 3.10). The Selector buffers the pixel data for the assigned events—the mechanism is practically identical to how the Merger buffers the DATCON ROIs—and performs the data reduction as soon as it receives the merged ROIs with the same trigger number. After a final reformatting step, the Selector sends the processed data to the EB2 over a GbE interface.

The numbering scheme for the Selectors used in figure 4.5 (S-x/y) indicates which node processes which pixel data packets. Selectors with the same x process data with the same event number; Selectors with the same y receive data from the same DHC (i.e., the same five PXD half-ladders). In the current

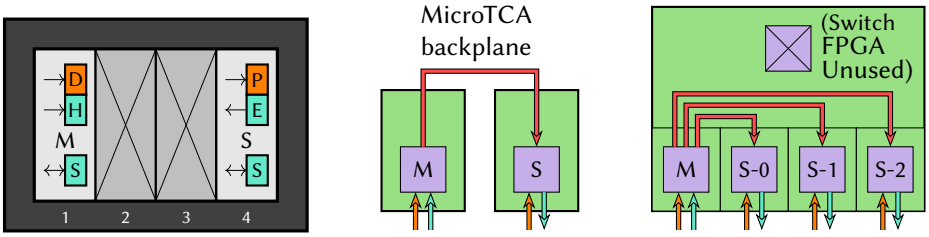


(a)



(b)

Figure 4.5: Architecture of the ONSEN system. **(a)** Arrangement of Merger (M) and Selector (S-x/y) nodes in an ATCA shelf. The I/O ports are: input from DATCON (D), input from HLT (H), pixel data input from DHC (P), output to EB2 (E), and slow control I/O (S). **(b)** Dataflow through the ATCA shelf. Internal arrows show distribution of merged ROIs. Slow control not shown.



(a) (b) (c)

Figure 4.6: Alternative setups for tests of the ONSEN Merger and Selector cards. **(a)** *Pocket ONSEN* setup: a Merger and Selector card in a MicroTCA shelf with fixed backplane connections. **(b)** Dataflow for merged ROIs in the Pocket ONSEN system. **(c)** Test setup with a Merger and three Selectors in a single carrier board, using the AMC interconnects provided by the CNCB.

scheme, the mapping is chosen in such a way that all Selectors in a single CNCB process data from the same event, and two neighboring CNCBs process one complete event. The Switch FPGA on the Merger-CNCB sends to each Selector-CNCB only the ROI packets for the relevant event numbers. It should be noted that this scheme is not final and depends heavily on the load-balancing and subevent-building mechanisms implemented by the DHH system.

In addition the I/O interfaces for the pixel and ROI data streams, figure 4.5 shows a GbE interface labeled “S” on each xFP. These interfaces are used for the slow-control (online control, monitoring, and debugging) of the FPGA logic. For the same purpose, all Switch FPGAs connect to a central Ethernet switch with their ATCA base-channel GbE interface. The slow-control mechanisms for the ONSEN system are explained in more detail in a later section.

At the time of writing of this thesis, the firmware for all FPGAs necessary for the setup shown in figure 4.5 has not yet been completed. The Switch FPGAs, in particular, lack the required routing functionality. The main reason for this is that the LVDS links between Switch FPGA and AMCs were not usable in previous hardware revisions of the CNCB. This problem was fixed in version 3.2 from late 2014. Development efforts before this time were therefore concentrated on the bitstreams for the Merger and Selector modules, using alternative setups that do not require dataflow via the ATCA backplane. One such setup, also used for system tests at external facilities, uses a small-scale MicroTCA-based system instead of the full ATCA-based one. It is shown in figures 4.6a and 4.6b. The MicroTCA shelf is the same one shown in figure 4.2. In accordance with the name *Pocket DAQ*, given by the KEK DAQ group to the scaled-down data acquisition system used on the same occasions, we refer to

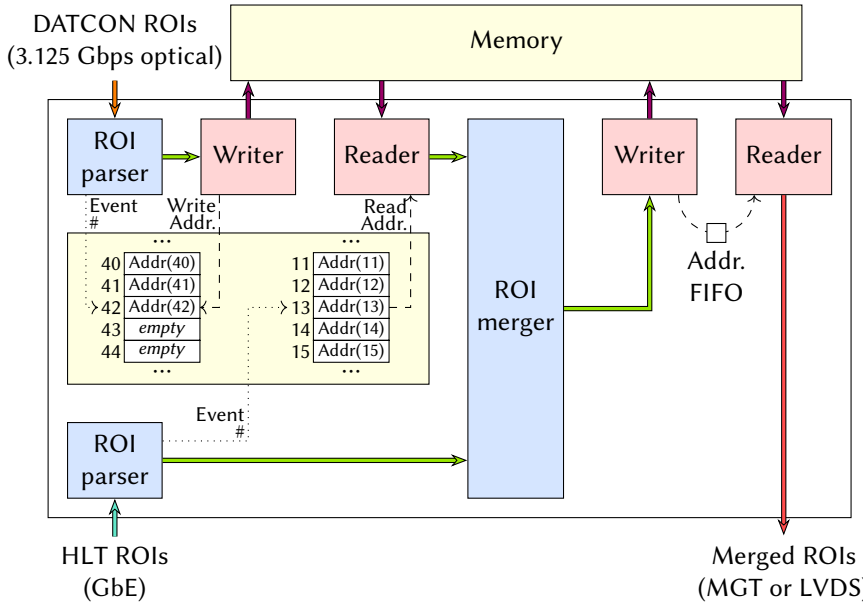


Figure 4.7: Simplified schematic view of the Merger node's FPGA firmware. Not shown: logic related to monitoring, slow control, and the management of free memory regions.

this system as *Pocket ONSEN*.

A different test setup is shown in figure 4.6c. Here, a Merger and one or more Selectors are placed in the same CNCB, and the hard-wired AMC interconnects are used to transfer the merged ROIs. The firmware configurations used for the alternative setups is mostly identical to that of the final ONSEN system, with the exception of the I/O interfaces (MGTs instead of LVDS links). Most of the logic needed for the operation of the Switch FPGA (like the event-number based ROI distribution) has already been tested by integrating it directly in the Merger and Selector nodes in one of the alternative setups.

The Merger node

Figure 4.7 shows how the Merger node processes the two inbound ROI data streams: After Belle II's GDL has issued a level-1 trigger, the DATCON system receives the SVD data for the event and produces ROIs based on reconstructed tracks, all on a timescale of about 10 microseconds. The Merger receives the ROI packet over an optical link, verifies its data integrity with a *cyclic redundancy check* (CRC), and extracts event metainformation like the trigger number. A writer core, constantly supplied with addresses from a pool of free memory

locations, writes the DATCON packet to a buffer in the DDR2 memory. The address is then stored in a lookup table that is indexed by the previously extracted event information. The size of the address lookup-table and number and size of the memory buffers are important parameters that depend on the available memory, the HLT's maximum processing time, and the size of the data packets. The same mechanism is used for the buffering of the much larger pixel data packets by the Selectors. The parameters are therefore discussed in section 4.3 below, after the description of the Selector node.

After the HLT has determined the trigger decision and ROIs for an event, it sends a packet to the Merger node over an Ethernet connection. It is important to note that not only is the delay of the HLT ROIs unpredictable, the packets can also appear in any order. It is, however, guaranteed that the HLT generates a packet for every event, possibly containing only the information that the event has been rejected. Even in this case, the Merger processes the event as usual, since the Selectors must be informed about the HLT decision for every event.

If DATCON ROIs for the same event are stored in memory—under normal conditions, this should always be the case—, their location is passed to a reader, which proceeds to retrieve them from the memory and afterwards return the address to the pool of free buffers. In cases where DATCON ROIs for the event are not available, a "dummy packet" is read instead from a special address, where it is prepared during the initialization of the system. The DATCON and HLT packets for the same event are then merged into a single packet. This packet is once again written to memory and read back as soon as the output to the Selector nodes signals that it can accept data[4]. As mentioned above, the Merger in the final ATCA-based system would then send the merged ROIs to its Switch FPGA over LVDS links, and the Switch FPGA would distribute them to the relevant Selectors for each event. In the development systems, this functionality is integrated into the Merger card itself, and direct MGT links are used.

The Selector node

Figure 4.8 shows a schematic view a Selector node's FPGA firmware. The dataflow is very similar to that of the Merger node—in fact, the Selector firmware was created first, and later adapted to perform the functions required from the Merger—, but the data processing steps are far more complicated.

[4]This mechanism does not change the content or order of the data packets, but it acts as a large elastic buffer: If one of the Selectors temporarily blocks the reception of ROIs, the Merger must halt its ROI distribution, so that this back pressure does not immediately propagate to the HLT packet input and block it.

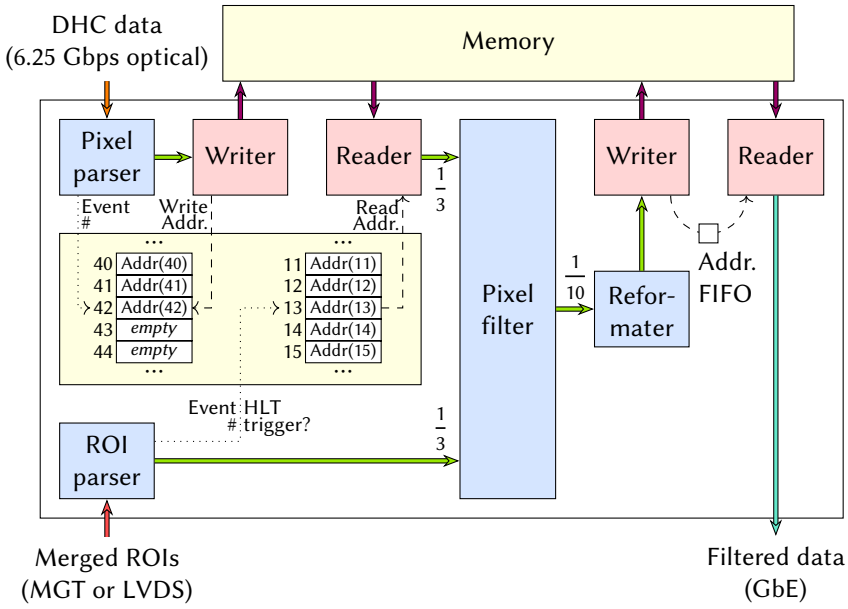


Figure 4.8: Simplified schematic view of a Selector node's FPGA firmware. Not shown: logic related to monitoring, slow control, and the management of free memory regions.

Again, data packets from two different sources must be received and later matched according to the event information. The DHC connected to each Selector sends pixel data shortly after the level-1 trigger for every fourth event. The data is segmented into individual frames (see section C.1). A parser extracts the event information from the data and combines all frames belonging to the same event into one large packet that can later be accessed from a single memory address. After the packet is written to memory, the address is stored in a lookup table in a mechanism identical to that of the Merger node.

The lookup and read-back processes are also similar to the Merger, but the Selector performs a first data-reduction step at this point: It receives the merged ROIs via the Switch FPGA (or a direct connection in the development system) and extracts from them the event information. The address for the corresponding pixel data is then retrieved from the lookup table, but the data are only read back from memory if the HLT has accepted the event. If the event was rejected, the used memory regions are freed and the ROI packet is discarded. As only one in three events will be accepted, this reduces the pixel data rate to 1/3.

In the next step, the merged ROIs for accepted events and the corresponding

pixel data (split again into individual frames after being read form the memory) are processed by a logic block referred to here as *pixel filter*. This entity is sometimes known as *ROI selection core* or roi_frame_handler. The pixel filter discards all PXD hits that are not inside an ROI from either HLT or DATCON. An additional data-reduction factor of 10 is achieved by this mechanism. The processed data are then reformatted, written to memory, and read back as soon as the event-building system is ready to accept data. As in the Merger's case, this step constitutes a large output buffer for the outbound data stream. It is also required to produce the ONSEN output format (see section C.4).

The part of the FPGA firmware developed during the work on this thesis encompasses the data input and output with GbE and MGT links, the parsing of data streams, extraction of event information, and checksum verification, the pixel frame handling and reformatting, and the memory writing and reading cores. The logic responsible for ROI merging, pixel filtering, and memory address management (including the address lookup) was created by other developers from our group. The filtering mechanism, in particular, is described in detail in another PhD thesis connected to the ONSEN project [103]. The design of the overall system architecture was a collaborative effort.

The next section gives insight into some general aspects of the firmware-design process. A closer look at the contributions to the ONSEN system from this thesis follows in section 4.4. Detailed information about the structure of the Merger and Selector nodes in terms of the individual logic cores can be found in appendix section A.

4.3 Design Aspects

Embedded system workflow

The firmware of an FPGA-based data-processing system must provide methods to monitor and control the logic during operation. During the development stage, this can be achieved with the help of in-system debugging tools accessed over a JTAG connection. For a complex, running system, this approach is unfeasible, and a more practical method is required. One such method makes use of firmware designs that include CPUs, so-called embedded systems. For the ONSEN firmware development, we use a workflow based on the Xilinx *Embedded Development Kit* (EDK) [104].

EDK provides a graphical user interface (GUI) that allows the designer to create FPGA firmwares built around a CPU. The CPU can either be a hard-core one (like the PowerPC in the Virtex-4 and Virtex-5 devices used in the ONSEN system) or a soft-core processor built from FPGA resources (like the Xilinx Microblaze architecture). Hard-core CPUs are device-dependent and take up a

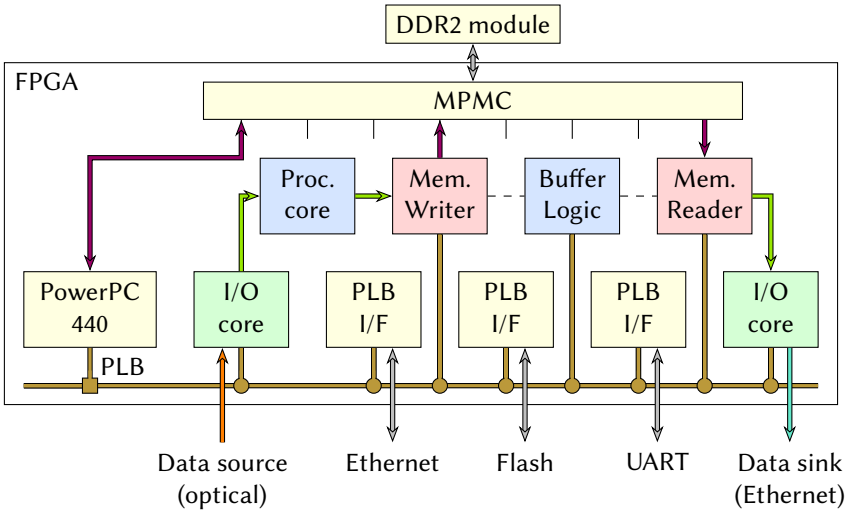


Figure 4.9: Example of the interconnection of various IP cores in an FPGA firmware project based on the Xilinx EDK workflow

part of the FPGA's silicon area, but they can also be clocked much faster than soft-core CPUs and do not consume additional resources during the firmware design.

Figure 4.9 shows an exemplary schematic of an embedded system running on an FPGA. The logic is divided into distinct blocks referred to as *intellectual property cores* (IP cores), built around the PowerPC processor. A bus system, built from FPGA resources and based on IBM's *Processor Local Bus* (PLB) standard [105], connects the CPU to several of the IP cores. The CPU is a bus master—it can send read and write requests to any address—while the connected cores are bus slaves that answer to requests to defined address regions. PLB-slave cores in EDK are known as *peripherals*. IP cores can also be made completely passive, not providing a bus interface. It should be noted that this example gives a simplified picture of the system architecture that is made more complex by including additional buses, interrupts, cores with *direct memory access* (DMA), and other features.

In their simplest form, peripheral cores expose one or more slave registers to the PLB. These 32-bit words can usually be read by the PowerPC and the core's HDL logic and written by one of the two sides. Registers written by the logic and read by the CPU can be used to make status information about the core available through software. Registers written by the CPU and read by the logic allow to control the FPGA logic during operation.

Custom cores used in the EDK workflow must be created in a special format.

In addition to the HDL files containing the core's data processing logic, information must be supplied that defines how the core appears in the GUI, how its outward interfaces connect to standard bus signals, and which parameters can be set for individual instances of the core. For the passing of payload data between IP cores, most cores created for the ONSEN system provide a simple, 32-bit point-to-point interface based on the Xilinx LocalLink protocol [106].

Besides the cores specially created for the ONSEN system, the Merger and Selector nodes also include Xilinx IP cores that allow the PowerPC to communicate with external devices through standard I/O interfaces. These interface cores are used for Ethernet connections via the RJ45 connector on the front of every xFP card and for UART connections, either via the front USB connector or via the AMC connector and carrier board. This makes it possible to access monitoring or control programs with a serial terminal or via a network.

Memory management

All accesses to the DDR2 memory are managed by a Xilinx multi-port memory controller (MPMC) [107]. The MPMC controls the DDR2 signals and arbitrates read and write requests to the memory between up to eight individual ports. In a Virtex-5 system, one of the ports is reserved for access by the PowerPC. The others can be used by user cores that implement one of the supported MPMC interface types. The ONSEN cores handling the writing and reading of the payload data streams use the fastest, most low-level *native port interface* (NPI). Every reader and writer therefore occupies one of the eight MPMC ports. The cores managing free memory addresses, the address FIFOs, and the address-lookup cores also require access to the memory. All of them use PLB interfaces and share access to a single MPMC port through a second PLB bus (independent from the main system bus controlled by the PowerPC).

The PowerPC CPUs used in the ONSEN system are based on a 32-bit architecture. Their address space has a size of 4 GiB. This space must be shared between the RAM accessible by the processor and the address ranges provided by the cores connected to the system PLB. Table 4.1 shows which address region is allocated to which purpose.

A 1.5 GiB range is reserved for the buffering of DATCON ROIs and merged ROIs on the Merger and for raw and processed pixel data on the Selector. This space is divided into 1.5 Mi = 1.5×2^{20} = 1 572 864 buffers with a size of 1 KiB. Every data packet written to memory (i.e., all ROIs or pixel data belonging to the same event) consumes at least two of these buffers: Any number of buffers can be concatenated in a linked-list fashion to accommodate arbitrary payload sizes. After all payload data of a packet have been written to memory, an additional index buffer is created and inserted in front of the payload buffers.

Table 4.1: Allocation of the PowerPC's address space on the Merger and Selector nodes. Some of the assigned regions are larger than the currently used space; the unused space is reserved for each region.

	Range	Size	Description
2 GiB DDR2	0x00000000 –0x0FFFFFFF	256 MiB	Linux O/S
	0x10000000 –0x6FFFFFFF	1.5 GiB	Data buffers 1.5 Mi × 1 KiB
	0x70000000 –0x77FFFFFF	128 MiB	Address lookup table 1 Mi × 8 B used
	0x78000000 –0x7BFFFFFF	64 MiB	List of free buffer addresses 1.5 Mi × 4 B used
	0x7C000000 –0x7FFFFFFF	64 MiB	Buffers addresses in FIFO 1.5 Mi × 4 B used
	0x80000000 –0xBFFFFFFF	1 GiB	*Reserved*
	0xC0000000 –0xDFFFFFFF	512 MiB	IP cores Slave registers and address spaces
	0xE0000000 –0xEFFFFFFF	256 MiB	Flash 64 MiB used
	0xF0000000 –0xFFFFFFFF	256 MiB	Block RAM **Last** 64 KiB used

For the pixel data, this buffer contains the number and size of the individual frames belonging to the event. This information is needed to split the frames up before they are passed to the pixel-filter core. It is also prepended to the output going to the EB2, where it is needed to discern event boundaries in the Ethernet data stream. The data formats used for this scheme are explained in appendix section C.3.

With this information, we can make a realistic estimate for the maximum data retention time of the ONSEN system. As discussed in section 3.4, the pixel data rate at each DHC output link—and therefore each Selector input link—amounts to 562.5 MB/s. The DHH load-balancing mechanism reduces the event rate at the Selector inputs by a factor of 4. Each Selector then receives events with a rate of 7.5 kHz and an average event size of up to 75 kB. With 1 572 864 buffers sized 1 KiB we can therefore store around 20 000 events, corresponding to a time of 2.8 s—much more than the average HLT processing time.

During this time, the memory addresses of the stored pixel data for all events

must be kept in the address lookup table. The table's size is directly connected to the required retention time: Every entry is 8 bytes wide—4 for the actual address and 4 to store the trigger number for a cross-check after the lookup—, and the number of entries depends on the bit width of the table index. It is clear that we cannot use the complete trigger number, let alone the run number, as the index, as this would create a table with a size of $8 \times 2^{32}\,\mathrm{B} = 32\,\mathrm{GiB}$. We must use a subset of the trigger number's bits that is guaranteed not to repeat in at least 3 seconds. In addition, we want to include bits from the run number in the lookup. This avoids an error condition experienced in past system-integration tests, where the trigger number is reset because of a run-number change without a reset of the ONSEN system.

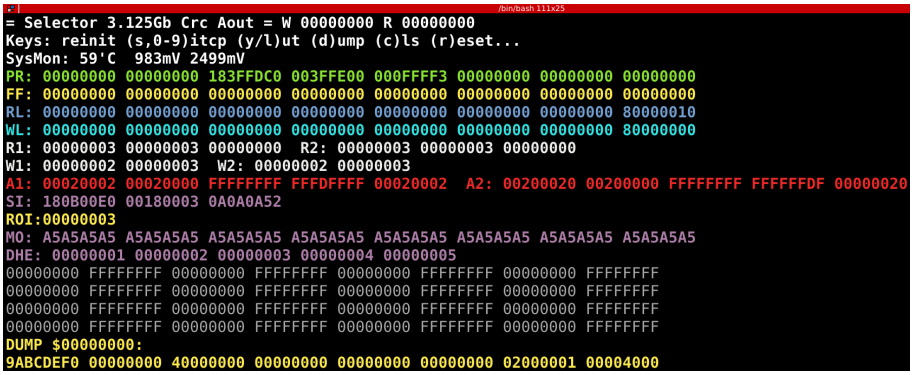
As the level-1 trigger logic issues consecutive trigger numbers, the lowest n bits repeat every $2^n/(30\,\mathrm{kHz})$ seconds. As the current design value, we use a table index of 20 bits, including the 18 least significant bits from the trigger number and the 2 least significant bits from the run number. This pattern is unique within any time window of at least 8.7 s. The address lookup table in this case has a size of 8 MiB. If required, this value can be easily increased by at least a factor of 16.

Additional RAM regions are required for the list of free memory-buffer addresses and for the address FIFO used in the elastic-buffer scheme for outbound data streams of the Merger and Selector. Both ranges must be large enough to accommodate all 1.5 Mi addresses. With an address width of 32 bits, the required space is 6 MiB in both cases.

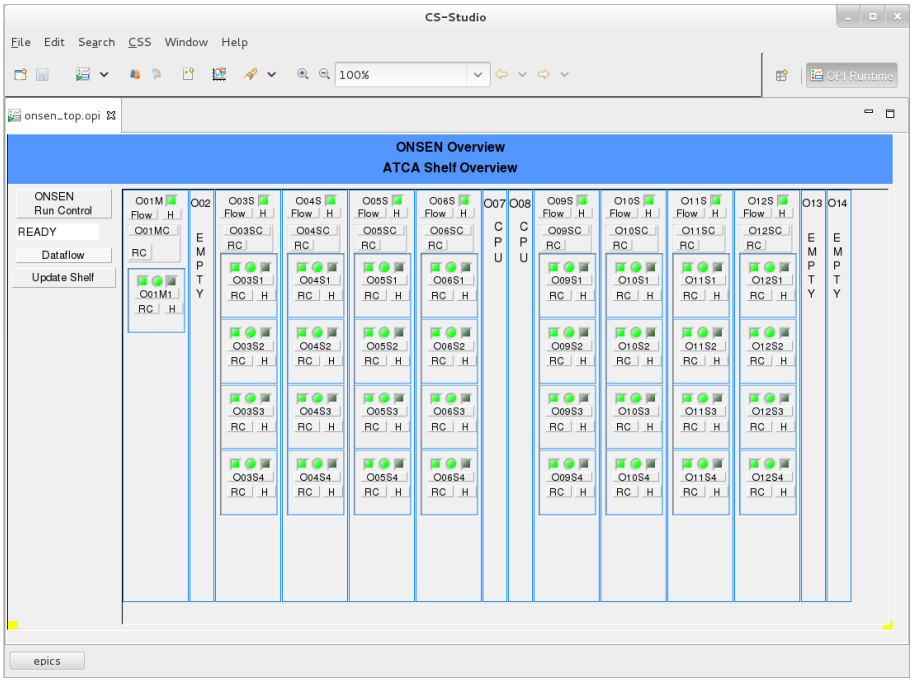
Software and slow control

Programs running on the embedded CPU in an EDK project can be written and compiled with the Xilinx *Software Development Kit* (SDK), which also provides software libraries for the various Xilinx I/O cores. This allows the rapid development of basic *standalone* programs (not requiring an underlying operating system) with serial and even Ethernet interfaces. During the development of the ONSEN system, such programs were used for the monitoring the FPGA logic during operation. Figure 4.10a shows the output of such a low-level monitoring application.

The interpretation of the rather cryptic output in the example requires deep knowledge of the ONSEN system and its components. For a complex experiment like Belle II, it is obviously not feasible to use individual slow-control mechanisms for every subsystem that only experts can understand. Belle II has therefore adopted the *Experimental Physics and Industrial Control System* (EPICS) [108] for slow control. EPICS allows subsystems to expose parameters for monitoring and control as so-called *process variables* (PVs) on a network.



(a)



(b)

Figure 4.10: Monitoring and control programs for the ONSEN system. **(a)** Console-based view of IP-core registers used in developments systems. **(b)** EPICS/CSS-based view of the health of all ONSEN modules in an ATCA shelf (work in progress; picture courtesy of B. Spruck, Uni Mainz).

GUIs like Control System Studio (CSS) can be used to visualize all PVs in the system in a central way, allowing the user to display information with different levels of verbosity. For example, a traffic-light health status can be provided to shift personnel, while experts can access special displays that allow an in-depth diagnosis for their subsystems.

PVs for the ONSEN system are provided by an *Input/Output Controller* application (IOC) running on the PowerPC. The IOC is not a standalone program, but requires a Linux operating system to run. Using a cross-compiler and an SDK-generated device-tree file (containing information about IP cores used by the FPGA project and their PLB addresses), a recent Linux kernel can be generated for the PowerPC. Drivers for standard Xilinx IP cores (in particular Ethernet and UART interfaces) are included in current stock Linux sources. The first 256 MiB of the DDR2 RAM are reserved for use by the operating system.

In addition to the monitoring and control of the FPGA logic, the hardware status of the Compute Nodes and ATCA shelf must be monitored. For this to work, an IPMC for the CNCB, an MMC for the xFP, and the respective microcontroller firmwares are required. A special EPICS IOC, running on a monitoring PC, communicates with the Shelf Manager and makes all relevant information available to the EPICS system. The information must then be combined with the firmware PVs, so that the status of the FPGA logic and the board it runs on can be displayed side-by-side. These issues are currently worked on by the Belle II group at the University of Mainz. Figure 4.10b shows what a high-level health-status display for the ONSEN ATCA shelf could look like.

Data input and output

The ONSEN system uses MGT-based high-speed serial links for the reception of ROIs from the DATCON, the reception of pixel data from the DHH, and the distribution of ROIs over the ATCA backplane. Such links require a special encoding scheme for the serial data stream that guarantees frequent transitions between 0 and 1. Besides many other advantages, this makes it possible to use different clock sources for the sender and receiver (clock recovery) and to use capacitive signal coupling for the links. The most common such encoding is the 8b/10b line code, where and 8-bit data word is transferred as a sequence of 10 bits on the serial link.

Not all possible 10-bit words are valid 8b/10b characters, but some words that are not used to encode data bytes are defined as so-called K-characters. They can be used for special purposes, like the marking of word or frame boundaries in the data stream. The exact interpretation of the serial data depends on the link-layer protocol. In the ONSEN system, all MGT links use the Xilinx Aurora

protocol [109] with serial data rates of either 3.125 Gbps or 6.25 Gbps. Aurora allows the use of special *native flow-control* sequences that make it possible for the receiver to request a throttling or stop of the dataflow from the sender.

Due to the 8b/10b encoding, the line rate is 10 times higher than the parallel data rate, so that a 6.25 Gbps link corresponds to a payload transfer of 625 MB/s, minus a minimal overhead from inserted sequences for framing, flow-control, and clock correction. The MGT SerDes provides a 32-bit user-logic interface for data transmission and reception, so that the words must be processed with at least 156.25 MHz.

As mentioned earlier, the LVDS-based links between the Switch FPGA and the FPGAs on inserted xFPs are not yet integrated into the system. Clock recovery cannot be used for these links, as they do not provide the special analog circuits that MGT have for this purpose. A common (system-synchronous) clock must be therefore used for the sender and receiver. The fabric clock pins of the AMC connectors, sourced by a clock fan-out chip with an input from the Switch FPGA, are used for this purpose. The maximum serial clock the Switch FPGA can generate from its 100 MHz input is 300 MHz, corresponding to a line rate (using double data-rate (DDR)) of 600 Mbps. The four links to each AMC card will therefore achieve a data rate of 240 MB/s. The Aurora protocol will likely also be used for these links.

The connections to the HLT and EB2 are Ethernet-based. There are two choices for the transport-layer protocol of these connections: the *User Datagram Protocol* (UDP), where the sender transmits data packets to the network without verification; and the *Transmission Control Protocol* (TCP), where a tunnel is established between two link partners, and each transmitted packet must be acknowledged by the receiving side. While UDP is much simpler and can be faster than TCP, it is inherently error-prone. Bit errors can be detected with a checksum in the UDP frame, but the receiver has no defined way of requesting a retransmission of a packet in case of a problem. On a switched network, the order of UDP packets can become jumbled, and complete packets can be lost without notice. All of this is prevented by TCP, since the sender retransmits packets that are not acknowledged and the receiver can reorder data packets based on a sequence number.

Implementing TCP on an FPGA is complicated. A sender must retain all transmitted packages until they are acknowledged by the receiver, and a previously sent packet must be correctly retransmitted in case of an error. The receiver must be able to accept out-of-order packets and rearrange them in the correct sequence. These processes require many logic and memory resources. We use the commercial SiTCP [110] IP core from Bee Beans Technologies Co., Ltd, for this purpose. The Merger uses one SiTCP core for data reception from the HLT, and each Selector uses one for data transmission to the EB2. In both

cases, the link partner is required to initiate the TCP connection to the SiTCP core running on the FPGA. Licenses for SiTCP are issued per hardware address, of which 33 are needed in the ONSEN system.

FPGA initialization

During system development, we used a PC and a JTAG programmer, connected via USB, to download bitstreams to the FPGAs after system power-up. This process has to be initiated by hand, and each CNCB has its own JTAG chain, requiring either a USB programmer for each board or a JTAG-multiplexing mechanism. This mechanism is obviously not feasible for a production system. The initialization of programs running on the PowerPC poses another problem. Programs that fit in the CPU-accessible block RAM (64 KiB) can be integrated into the bitstream. When the FPGA is configured, the PowerPC executes instructions from the end of its memory range, where the block RAM is located, and the program starts automatically. This works for small, console-based monitoring programs, but not for the much larger Linux kernel. Large programs can be copied to the RAM on a running FPGA via JTAG and executed by hand, but this is equally unsuited for the final system.

For an automated bitstream download, we therefore use the slave-serial mechanism described in section 4.1: All FPGAs on a CNCB equipped with xFPs are placed in a configuration daisy-chain, and a combined bitstream file is located in the Switch FPGA's Flash memory. Upon power-up, the CPLD on the CNCB reads this file from the Flash and pushes it to the FPGAs. The start-up of large PowerPC programs is achieved using the Flash memory of each FPGA. A Linux kernel is placed at a defined address in the Flash, and the FPGA bitstreams include a small boot-loader program in the block RAM. Upon power-up, the boot loader is executed. It copies the Linux kernel from Flash to RAM and executes it from there. When power is first supplied to a CNCB, all FPGAs should be configured and the required software started automatically in about 30 seconds.

Besides monitoring and control functions, the software running on the PowerPC is responsible for several necessary initialization steps that must be executed before the logic can operate as required. These steps include the activation of IP cores by write operations to PLB registers, the setting of addresses for the Ethernet interfaces, and the initialization of the memory regions belonging to address lookup tables and buffer-address lists. The complete logic can be reset to a defined state by repeating this initialization procedure without having to reconfigure the FPGA.

4.4 Contributions from the Work on this Thesis

As previously mentioned, the design of the ONSEN system and the development of the firmware for the two modules was a collaborative effort of the Belle II group at the University of Gießen. Nevertheless, the two firmware projects for the Merger and Selector can be broken down into individual IP cores created by different developers. Five IP cores, in particular, are products from the work on this thesis and are used in the current firmware versions. This section gives an overview of the function and operation of these cores. The place of each cores in the ONSEN dataflow, as shown in the simplified diagrams of figures 4.7 and 4.8, will be indicated in the core descriptions given below. A complete dissection of each project in terms of all involved IP cores is too technical for the context of this chapter; it is provided in appendix A. A more comprehensive documentation of the cores, including interface and register definitions, can be found in appendix B.

xFP Aurora wrapper

High-speed serial links using the Xilinx Aurora protocol are used for the optical inputs for DATCON ROIs and DHC data, as well as the transfer of merged ROIs between Merger and Selector. They are indicated as inbound and outbound arrows in the dataflow diagrams above. The HDL code for the implementation of the Aurora protocol belongs to a Xilinx IP core that handles the transmission and reception of control sequences, the channel initialization between link partners, data framing, and other features. The developed core is a wrapper that adapts the Xilinx code for the use with the xFP cards in the ONSEN design flow.

The Aurora wrapper provides the user with a simple interface, allowing to chose one or more MGT links based on logical names (like “SFP 0” or “AMC port 12”) and handling the abstraction to the hardware layer. For each chosen interface, the core automatically instantiates the Xilinx Aurora code, the required MGT transceiver cores, and FIFOs for the input side, output side, or both (depending on the chosen dataflow direction). It generates all necessary location and timing constraints, ensures the required routing of MGT clocks (required for certain configurations with unused MGTs), and sets the correct clocking parameters based on the line rate chosen by the user. The activated links are presented as LocalLink buses in the EDK GUI and can be easily connected to other cores in the design.

For the incorporation into the Belle II data-acquisition system, it is important that the ONSEN system can handle back pressure from the downstream side and, if necessary, relay it to the upstream side: If, for example, the EB2 rejects

the output from a Selector node, the Selector's memory will begin to fill up. If the dataflow remains blocked (as could be the case of a broken link to an event-builder PC), the Selector must be able to notify the DHC of this condition, so that it stops sending data. For this purpose, the Aurora core allows the insertion of *native flow-control* sequences in the back-channel link. The wrapper core implements this feature by requesting the sender to stop data transmission as soon as the receiving FIFO is more than half full.

The Aurora wrapper is a PLB slave core. It contains status registers, relaying the state of Aurora channels and LocalLink interfaces, and control registers to steer the input and output dataflow. In addition, a custom PLB-bridge core was developed that allows software access to the dynamic reconfiguration ports (DRPs) of all instantiated MGTs from the PowerPC. This interface can be used for online link tuning through the adjustment of hardware parameters like preemphasis and equalization.

SiTCP wrapper

SiTCP links are used for the reception of HLT ROIs and the transmission of processed data to the EB2. They are indicated as inbound and outbound arrows in the dataflow diagrams above. Similar to the Aurora core, this core is a wrapper that instantiates the pre-synthesized netlist of the proprietary SiTCP logic. It can be used either with a GMII interface to an Ethernet PHY, to use the xFP's RJ45 socket, or a 1000BASE-X interface, using one of the MGTs connected to the board's SFP+ ports. In each case, the wrapper core provides the required interface logic, using a Xilinx PCS/PMA core for MGT links. As for the Aurora core, the SiTCP wrapper has LocalLink buses for connections to other cores, instantiates necessary FIFOs, and provides a PLB slave interface to monitor and control the links status and access the DRP.

Data transmitted over a TCP connection is sent as a stream of bytes that does not, per se, contain any framing information[5]. Inside the ONSEN system, data is handled in the form of frames, exchanged between cores through LocalLink interfaces that provide frame-delimiter flags. An ROI packet received with SiTCP must be formatted as a frame before it is presented on the LocalLink interface. To that end, a framing mechanism was developed for the wrapper core. It captures the first four bytes received from the TCP link-partner and interprets them as the frame size n. Then it forwards the next n received bytes to the LocalLink output, setting the correct delimiter flags. After the end of the

[5]While it is true that two network interfaces exchange data in the form of packets, this is only a transport mechanism. The size of each packet is usually transparent to the application and unrelated to any logical grouping of the bytes.

frame, it expects the next frame size. The TCP sender (in this case the HLT) is required to prepend the correct length to each data frame.

An additional necessity for the SiTCP core is the inclusion of a memory block that contains a license file required for startup. SiTCP licenses are issued per MAC address, meaning that one license is required for every core in the same network. The license file contains the MAC address in an encrypted form. On boards specifically designed to use SiTCP, the file is stored in an EEPROM chip connected to the FPGA. After a reset, the SiTCP core accesses the EEPROM and reads out the license information. This mechanism was not foreseen when the xFP was designed. As a replacement, an EEPROM interface is emulated with FPGA logic and connected to one port of a dual-port block RAM primitive. The second port is connected to the wrapper core's PLB interface. With this configuration, an SiTCP license file can be written to the block RAM from the PowerPC. If the SiTCP core is reset afterwards, it reads the file through the bridge interface and starts up as if were accessing an actual EEPROM chip.

Belle II format handler

All data streams entering the ONSEN system first pass through a core that parses the data and extracts information like the trigger number, run number, and, in the case of an HLT ROI frame, the trigger decision. This format-handler core also performs an integrity check by verifying the checksum appended to each data frame. It can be configured to digest either the DHC output format or the ROI format (see appendix C). In the dataflow diagrams above, the core appears in various places labeled as *pixel parser*, *ROI parser*, and *reformater*, depending on the specific purpose of each instance. The format handler forwards the extracted event information and the result of the integrity check to the address lookup-table core.

In the case of the pixel data, the format handler has an additional purpose: It receives the multiple incoming data frames belonging to one event and fuses them into a single, long frame before they are written to memory. The core buffers the length of all processed input frames and writes them to an index frame appended to the end of the event. The index frame is needed when the frame is read back from memory later. It is either used to split the event data back up into individual frames before passing them to the pixel filter, or it is kept and prepended in front of the pixel data at the output of the Selector. This creates the ONSEN output format, required by the EB2 to determine event and frame boundaries in the output data stream (hence the name *reformater* above).

NPI writer

The NPI writer core is used in both projects to transfer payload data (DATCON ROIs, merged ROIs, raw pixel data, and processed pixel data) to the DDR2 memory with high speed, using one of the eight MPMC ports. It was developed as a replacement for a previously used PLB memory I/O core that did not provide the required throughput. The core appears under the label "Writer" in the dataflow diagrams above.

The writer receives data frames via a LocalLink interface and writes each frame to a linked-list memory buffer, using the format shown in appendix section C.3. It is supplied with addresses to empty memory buffers by a buffer-provider core via a dedicated pointer-bus interface. After all payload data for an event have been written to memory, the index frame received from a format-handler core is optionally prepended to the data by writing it to the first buffer in the chain. Afterwards, a pointer to the first buffer is forwarded to the address lookup table core or an address FIFO.

NPI reader

The NPI reader is the counterpart to previously described writer core. In the firmware projects for the ONSEN system, the two always appear pairwise, each taking up one MPMC port.

The reader core initiates a read operation whenever it receives a memory address via its pointer-bus interface. It reads the buffer header and determines from it the number of bytes to be read and the address for the next buffer in the linked list. An null pointer indicates that the current buffer is the last one. For data beginning with an index frame, the reader can be configured to either split the data back up into individual frames or prepend the index frame for the output to the EB2. Each buffer is freed (i.e., returned to the free-buffer provider core) after its contents have been read completely.

The writer and reader cores always use two ports of the same MPMC. The arbitration of memory accesses between the two can lead to run conditions that have caused data corruption in the past. One such condition can occur if a reader schedules a read access from a short memory buffer, then frees that buffer before all data have appeared on the memory output. In that case, a writer can pick up the freed address, schedule a write operation, and be given priority *before* the reader. The buffer is then overwritten before it is read. This condition is now avoided by monitoring the number of bytes on the output of the reader's memory interface, and only freeing a pointer after all bytes requested from the corresponding buffer have appeared on the output.

The opposite condition is more difficult to prevent: There is now way to

determine when a write operation, scheduled at the MPMC, has actually taken place. If a writer passes on a memory pointer after scheduling a write request to it, a request to the same address from a following reader can be granted *before* the write operation has taken place, resulting in the read of invalid data. To avoid this, the writer initializes each buffer by writing a guard word at its start before the payload data. After all other data have been written to the buffer, the guard word is overwritten with the actual header, and the pointer to the buffer is passed on. If a following reader accesses the address before the correct buffer has been written, it encounters the guard word, causing it to repeat the read process until it sees a valid header.

Other firmware-related work

Much of the groundwork that has facilitated the currently used design flow for the ONSEN system is a result of the work on this thesis. This includes the implementation of the LocalLink interfaces for inter-core communication and the adaption of all hardware revisions of the xFP and CNCB into the EDK base system builder, allowing the GUI-based creation of new projects for the boards. In addition, an HDL library was created—including functions like FIFO generators and checksum calculators—that is also used by IP cores from other designers.

4.5 Remaining Issues

As mentioned above, the most crucial uncompleted task is the migration of the ONSEN test system to the full ATCA-based architecture, including ROI distribution via the CNCBs and the ATCA backplane. The HDL logic necessary for this setup and the verification of the required hardware functionality are mostly completed (see also the link tests described in the next chapter). LVDS-based IP cores, based on the Aurora protocol, will be created as as drop-in replacement for the MGT-based cores used in the current projects.

A possible future upgrade is the integration of the second DDR2 module into the firmware for the xFP cards. This would double the amount of available RAM, making a longer data retention, higher data rate, or the addition of new features possible. The main reason for the omission of the second module is the amount of available FPGA resources: The number of slices occupied by an MPMC is very large, and it scales steeply with the number of ports used. In order to make room for a second memory controller, resources would have to be freed in other places.

A possible candidate is the SiTCP sender connected to the EB2. SiTCP is one of the largest cores in the system, and it is required on every Selector

node. A currently discussed upgrade path foresees an offload of the sending functionality of the processed pixel data from the Selector to another point in the system. There are two possibilities:

In the first option, the SiTCP cores connected to the EB2 are moved to the eight Selector Switch-FPGAs. Instead of transmitting the processed data via Ethernet, the Selectors would send them to the Switch FPGA on the same links used for receiving the merged ROIs. The Switch FPGA would collect the data from all four Selectors on the board and send them out via a single SiTCP core connected to the CNCB's backplane Ethernet channel. In section 3.4, the output data rate for each Selector was estimated to be at most 18.75 MB/s (disregarding full-frame read-outs, that can be scaled down if necessary). The rate for each Switch FPGA would therefore be 75 MB/s, still well beyond the limit for GbE. The data would be routed through a base-channel Ethernet switch in the shelf's first hub slot. The total output data rate is 600 MB/s. By using a switch with a 10GbE uplink port, a single connection would suffice to transfer all output data from the ONSEN system to the EB2.

The second option requires an additional CNCB and several additional xFPs, acting as *Concentrator* nodes. Again, each Selector sends its output to its Switch FPGA, but in this scheme, all Switch FPGAs send their combined output to the concentrator board via the ATCA backplane fabric. The receiving Switch FPGA forwards the data to the Concentrator cards via the LVDS links. At least three Concentrator cards are required, since each link is limited to 240 MB/s. The Concentrators offer the possibility to introduce further data processing or event-building steps before they send the data on to the EB2. Again, there are multiple possibilities: Either each Concentrator sends out its data via two SiTCP cores; or a final *Sender* node receives the data through the AMC interconnects and sends them out on one or more high-speed serial links with an optical interface. This would require the addition of a add-on card to the EB2 that provides the receiving side for this link and puts the data into the event-building system.

Both options are only conceptual, and their feasibility is not yet verified. For the purposes of this thesis, the current baseline option with one SiTCP output link for every Selector node is assumed.

Chapter 5

Test Results

In this chapter, I present results from various tests that demonstrate the performance of the ONSEN system and its capability to fulfill all requirements for a successful integration into the Belle II experiment. The tests include both small-scale laboratory setups, used to determine various benchmark parameters, and system-integration tests that verify the logic functionality and the interaction with other Belle II subsystems. In the last part, I establish the feasibility of the migration from the current test setups to the full-scale system for Belle II, based on the ATCA carrier boards, with the current hardware.

5.1 Laboratory and Benchmark Tests

The largest part of the work on this thesis was concerned with the development of the data input and output mechanisms for the ONSEN system, as well as the logic for writing buffered data to memory and reading them back later. The data output from the PXD and the Belle II data-acquisition and trigger systems impose stringent requirements on the performance of these functions. This section shows test results for individual system components, confirming their ability to operate within the boundary conditions defined by the other systems.

Bit-error rate for xFP Aurora links

The defining quality for a digital link between a sender and receiver is the expected fraction of incorrectly received bits, the so-called *bit-error rate* (BER). Knowing the BER for a link, it is possible to predict the number of transmission errors in a given time interval. Link integrity is an especially critical issue for high-speed serial links. On the xFP cards in the ONSEN system, such links

are used for the reception of data with optical fibers as well as board-to-board transmissions via the carrier board in tests systems. Occasional bit errors do not lead to failures of any part of the ONSEN system, since they are detected with checksums present in all received and transmitted packets. Nevertheless, they usually make the affected packet unusable and lead to data loss: If a checksum error is encountered in the header or payload data of a frame received from the DHC, the entire pixel data for the event is discarded and replaced by a dummy packet. The number of bit errors should therefore be kept as low as possible.

To get an approximation for the BER of both types of high-speed links in the ONSEN system, a test setup with a CNCB v3.3 equipped with four xFP v4.0 was prepared. The two SFP+ links on each xFP card were used with optical cables to establish pairwise connections between the cards. In addition, the full-mesh interconnections provided by the CNCB (three links per board) were included in the test. All links were tested with the Aurora transceiver cores used in the ONSEN system. Each Aurora core was connected to a pattern generator and checker, sending consecutive counter values to the link and expecting the same on the receiver side. The number of transmitted bytes, received bytes, and detected errors, as well as the data rate, could be monitored with a slow-control program. The connections were tested with the design line-rates of 6.25 Gbps for the optical links and 3.125 Gbps for the xFP interconnects.

The total number of transferred bits was $8 \times 1.5 \times 10^{15}$ for the optical links, with a payload data-rate of 621.6 MB/s, and $12 \times 7.5 \times 10^{14}$ for the xFP interconnects, with a payload data-rate of 310.8 MB/s. No bit errors were observed. From a BER test with no errors and n transmitted bits, an upper limit for the actual BER can be calculated as $-\ln(0.05)/n \approx 3/n$ with a confidence level of 95 % [111]. We can therefore claim a BER better than 2.5×10^{-16} for the optical links, which will be used for the reception of the PXD data in the ONSEN system. Consequently, we expect an average of less than one bit error for every incoming pixel-data link during 4×10^{15} received bits, corresponding to a run time of 178 hours or 1.9×10^{10} events.

The numbers given above should be taken with a grain of salt. First, the connection of the optical links was tested between two xFP cards, and not, as in Belle II, between a DHC and xFP module. Second, the performed test was not a BER test in the strict sense. The deserialized data words were not checked directly, but first processed by an 8b/10b decoder and Aurora core. In a thorough BER test, pseudo-random binary sequences and stress patterns would be transmitted over the links and verified on the receiving side. Nevertheless, all sent payload data words were correctly transferred in the test, and the Aurora core did not issue a soft-error signal, which would almost certainly happen in case of a single-bit error.

Data rate for TCP interfaces

In its current design, the ONSEN system uses Ethernet interfaces based on the SiTCP core for the reception of ROI packets from the HLT and the transmission of processed data to the EB2. Both links do not require a very large bandwidth: The ROI data rate will be well below 10 MB/s (see appendix section C.2.5) and the output data rate for each selector will be around 30 MB/s (see appendix section C.4.4). Nevertheless, the maximum I/O rate for each link should be determined, especially in the light of the possible transition to a data output scheme with only one outbound TCP connection per carrier board. This change would quadruple the rate on each link.

The rate test was performed similarly to the aforementioned Aurora BER test: An SiTCP core was instantiated, and its transmitter and receiver were connected to a pattern generator and tester. The generated packets were sent to a PC and stored in a memory file while the rate was monitored on the FPGA. The file was then sent back to the FPGA through the same connection, and the data consistency was checked. No long-term link-integrity test was performed, as the physical links used are the same as those used for the much faster Aurora links that were characterized earlier. The resulting rates were 118.3 MB/s for data sent by SiTCP to a PC and 118.7 MB/s for the opposite direction. Both numbers surpass the system requirements by far and are close to the GbE hard limit of 125 MB/s imposed by the line rate and reduced by protocol overhead.

Memory Bandwidth

An important benchmark parameter for the data processing by the ONSEN system is the bandwidth with which packets can be written to and retrieved from memory. All memory accesses go via ports of the 8-port MPMC. On the Selector node, accesses must be arbitrated between: (1) the PowerPC, including the Linux operating system, (2) writing of raw pixel data for every event, (3) reading of raw pixel data for selected events, (4) writing of processed pixel data for selected events, (5) reading of processed pixel data for selected events, (6) a write and read access to the address lookup-table for every event, (7) swapping of pointers to free memory buffers, and (8) swapping of pointers in the address FIFO. The last three items share an MPMC port through a PLB bus, so that six ports are required in total. For the writing of the raw data, about 600 MB/s are required (see section 3.4). The read-back of selected data (every third event) then amounts to up to 200 MB/s. The processed data (reduced by a factor of 10) are once again buffered, requiring another 20 MB/s for both writing and reading. Other data rates are negligible in comparison, so the total bandwidth requirement is below 1 GB/s.

The MPMC uses a 200 MHz clock. The outward interface runs at double data rate with a 64-bit interface. The theoretical maximum for the total memory bandwidth is therefore 400 MHz × 8 B = 3200 MB/s. While this number surpasses the necessary throughput by far, it is significantly reduced by inefficiencies of the arbitration mechanism and by small (non-burst) memory transfers. An appropriate test must verify that the required bandwidth can indeed be reached.

Once again, a pattern generator and checker was used to provide test data and monitor the bandwidth. The test was performed with the same memory writer and reader cores used in the ONSEN system, writing the generated data to memory and immediately reading them back. Three situations were evaluated: a single writer-reader pair using two MPMC ports; two parallel writer-reader pairs using four MPMC ports; and three parallel writer-reader pairs using six MPMC ports. The ONSEN memory management logic was used to provide free memory addresses and pass pointers between the cores.

The resulting data with one reader and writer (two ports) saturated at the maximum output of the pattern generator core, 800 MB/s per port, or a total throughput of 1600 MB/s. With two readers and writers (four ports), the cumulative bandwidth rose to 1679 MB/s. A similar configuration (with an additional PLB port for the low-throughput applications) is used in the ONSEN system. We can therefore claim that The required total bandwidth is exceeded by more than a factor of 1.5[1]. Adding a third reader and writer (six ports) only had a slight impact on the total memory throughput, increasing it to 1683 MB/s.

Resource utilization

As mentioned in section 4.5, the resource utilization of the FPGA firmwares for the Merger and, in particular, the Selector nodes are already quite large and do not leave a lot of room for future extensions of the system. This section quantifies this claim by showing the relevant numbers for current versions of both firmware projects. The systems correspond to those shown in appendix A. Both projects were successfully implemented with a timing score of 0.

Table 5.1 shows the device utilization of the complete projects and the largest cores, listed in descending order of occupied slices. Only logic elements are listed, as the occupancy of other components (like block RAM) is well below the limit. Both projects cover most of the FPGA area, using more than 90 % of all slices, while only about half of the available registers and LUTs are needed. This means that there is still some room for the addition of upgrades and new features—even a slice coverage of 100 % does not mean that more logic cannot

[1] While the average rate per port (420 MB/s) is below the data rate of the inbound pixel data, the test showed that individual ports can still reach much higher throughput if other ports use a smaller fraction of the total bandwidth.

Table 5.1: Resource utilization of ONSEN FPGA projects. For multiple instances of one core, the average utilization per core is given. The reported number of slices used for each cores cannot be added up to the total number of occupied slices, since different cores can use elements from the same slice.

(a) Merger node

Component	Used slices		Registers		LUTs	
MPMC (6-port)	2 987	(26.7 %)	5 181	(11.6 %)	3 144	(7.0 %)
SiTCP	2 471	(22.1 %)	5 165	(11.5 %)	4 307	(9.6 %)
PPC Ethernet	1 098	(9.8 %)	1 807	(4.0 %)	1 684	(3.8 %)
Address FIFO	801	(7.2 %)	979	(2.2 %)	1 102	(2.5 %)
Aurora (×2)	727	(6.5 %)	1 010	(2.3 %)	888	(2.0 %)
Buffer provider	671	(6.0 %)	876	(2.0 %)	1 011	(2.3 %)
NPI Reader (×2)	621	(5.5 %)	978	(2.2 %)	743	(1.7 %)
Addr. LUT writer	602	(5.4 %)	846	(1.9 %)	916	(2.0 %)
Addr. LUT lookup	552	(4.9 %)	835	(1.9 %)	855	(1.9 %)
NPI Writer (×2)	417	(3.7 %)	786	(1.8 %)	597	(1.3 %)
Others		–	2 403	(5.4 %)	2 698	(6.0 %)
Total	10 013	(89.4 %)	23 639	(52.8 %)	20 172	(45.0 %)
Available	11 200		44 800		44 800	

(b) Selector node

Component	Used slices		Registers		LUTs	
MPMC (6-port)	2 678	(23.9 %)	5 017	(11.2 %)	3 091	(6.9 %)
SiTCP	2 256	(20.1 %)	4 949	(11.0 %)	4 129	(9.2 %)
Pixel filter	1 339	(12.0 %)	2 564	(5.7 %)	1 942	(4.3 %)
PPC Ethernet	1 045	(9.3 %)	1 807	(4.0 %)	1 679	(3.7 %)
Address FIFO	754	(6.7 %)	979	(2.2 %)	1 110	(2.5 %)
Aurora (×2)	664	(6.0 %)	1 046	(2.3 %)	887	(2.0 %)
Buffer provider	641	(5.7 %)	877	(2.0 %)	1 026	(2.3 %)
Addr. LUT writer	588	(5.3 %)	846	(1.9 %)	913	(2.0 %)
Addr. LUT lookup	546	(4.9 %)	860	(1.9 %)	866	(1.9 %)
NPI reader (×2)	540	(4.8 %)	993	(2.2 %)	691	(1.5 %)
NPI writer out	402	(3.6 %)	799	(1.8 %)	596	(1.3 %)
Others		–	2 394	(5.3 %)	2 756	(6.2 %)
Total	10 286	(91.8 %)	25 968	(58.0 %)	21 838	(48.7 %)
Available	11 200		44 800		44 800	

be added, as the implementation tools tend to use much of the available area to facilitate the routing—, but additional logic makes the design denser and impedes the achievement of good timing values. A reduction of the device usage is therefore desirable in order to keep the system ready for possibly required changes.

As expected, the most resource-intensive core in both cases is the memory controller. Its utilization scales strongly with the number of used ports, but this number cannot be made smaller with the current design-flow concepts. The second-largest component of both projects is the SiTCP transceiver, using logic from more than 20 % of all slices. As mentioned in section 4.5, some currently discussed upgrade options foresee the offloading of the TCP output to the Switch FPGAs. In that case, the SiTCP core would be replaced by an additional Aurora or similar core, requiring only about a fifth or resources in terms of both registers and LUTs.

Another possible leverage point is the Xilinx Ethernet MAC core, which the PowerPC uses to connect to other PCs over a network. In the ONSEN system, this core is used for the EPICS-based slow-control functions. It is currently configured with large FIFOs and options for the offloading of checksum calculations to the hardware. Removing these options significantly reduces the core's logic utilization, but might make the Ethernet connection too slow to handle the traffic induced by the EPICS system. This is a point that remains to be checked when the slow-control system is finalized.

5.2 System Integration Tests

The DESY test beam facility

After isolated unit tests, an important milestone for the individual parts of a larger system are the first integration tests with the surrounding components. The the case of the ONSEN system, two such tests were performed at the test-beam facility of the *Deutsches Elektronen-Synchrotron* (DESY): First, a test of the PXD data-acquisition chain in May 2013 to verify connectivity and basic dataflow mechanisms; and second, a more comprehensive test with modules from both PXD and SVD in January 2014 that provided the first opportunity to exercise the ROI-generation and data reduction mechanisms in a realistic environment.

DESY offers researchers from international institutes the possibility to test detector setups with electron or positron beams. Beam energies of up to 6 GeV and rates of up to 6 kHz can be individually selected in the three independent test beam areas 21, 22, and 24. A fourth "parasitic" area, 24/1, uses the beam after it has passed through area 24. The 1 T superconducting solenoid PCMAG is

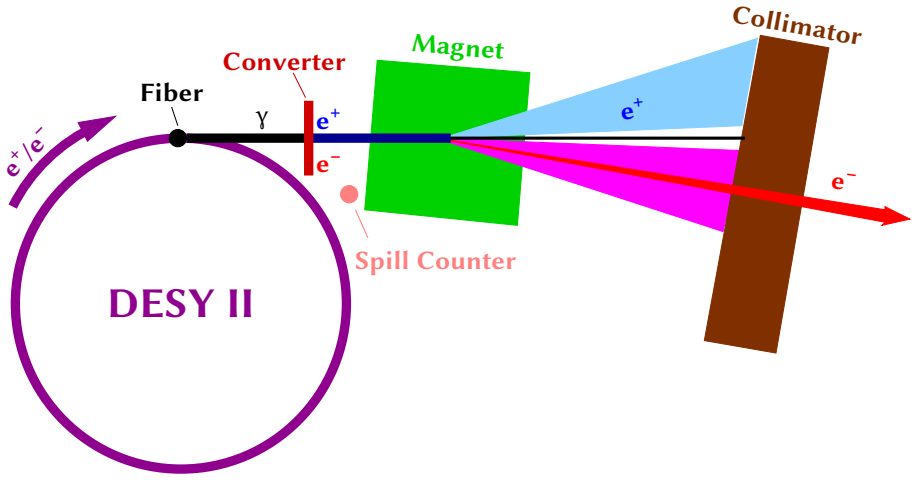


Figure 5.1: Beam generation for the DESY test beam areas (picture adapted from the DESY test beam website [112])

available in area 24/1. All four areas have separate beam shutters and interlock systems.

The mechanism for the generation of the test beams it illustrated in figure 5.1. It makes parasitic use of the beam stored in the DESY II ring. DESY II is a synchrotron with a circumference of almost 300 m. It accelerates electrons or positrons, injected with 450 MeV from DESY's LINAC II, to a final energy of up to 7 GeV. The particle bunches accelerated by DESY II are extracted and injected into the DORIS and PETRA III synchrotron-radiation sources. By placing carbon fiber targets with a thickness of several micrometers at three locations within the DESY II beam trajectory, bremsstrahlung photons are extracted without significantly disturbing the beam.

The bremsstrahlung photons are fired on a metal target, where they produce electrons and positrons through pair production. The energy spectrum of the generated particles depends on the thickness and material of the target, which the test-beam user can select from several options. This is shown in figure 5.2 for test-beam area 24. The electrons and positrons then pass through a vertical magnetic field, causing them to separate and fan out in the horizontal plane. The different angular regions of the fanned-out beam correspond to energy bands of the particles, and a final collimator and shutter allow only a small slice to propagate to the test area. By varying the magnet current and shutter opening, the test-beam user can select the mean energy, energy spread, and rate of the beam.

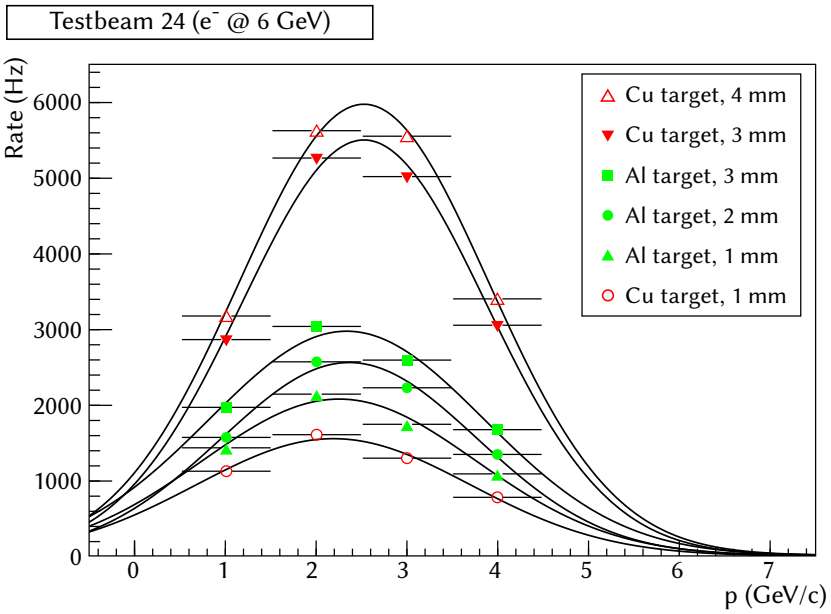


Figure 5.2: Correlation of beam energy and rate for different conversion targets at DESY test-beam area 24 (picture adapted from the DESY test beam website [112])

PXD test in May 2013

The first in-beam test of a Belle II PXD module together with the ONSEN system took place in May 2013 at DESY test beam area 21. The device under test was a DEPFET Hybrid 5.0 board with a small PXD matrix and a single SWITCHER, DCD, and DHP. The Hybrid is a PCB designed for testing DEPFET sensors during the development stage, as a substitute for the final arrangement explained in section 3.3, where all inputs and output will go via a Kapton cable. The PXD matrix and ASICs are wire-bonded to the Hybrid PCB, which provides connections to power supplies, I/O through InfiniBand sockets, and fastening positions for cooling appliances. The mounted sensor was a PXD6 with 16 × 128 pixels sized 50 µm × 75 µm. For Belle II, the newer PXD9 chip design will be used.

From the data-acquisition side, the goal of this test was to establish the hardware-side of the full PXD data-acquisition chain for the first time. This chain includes the DEPFET modules, the DCD and DHP ASICS, and the DHH and ONSEN systems. Connections to other parts of the Belle II data-acquisition system, like the high-level trigger and event builders, were explicitly not included in this test. Triggers were generated for coincident signals of plastic

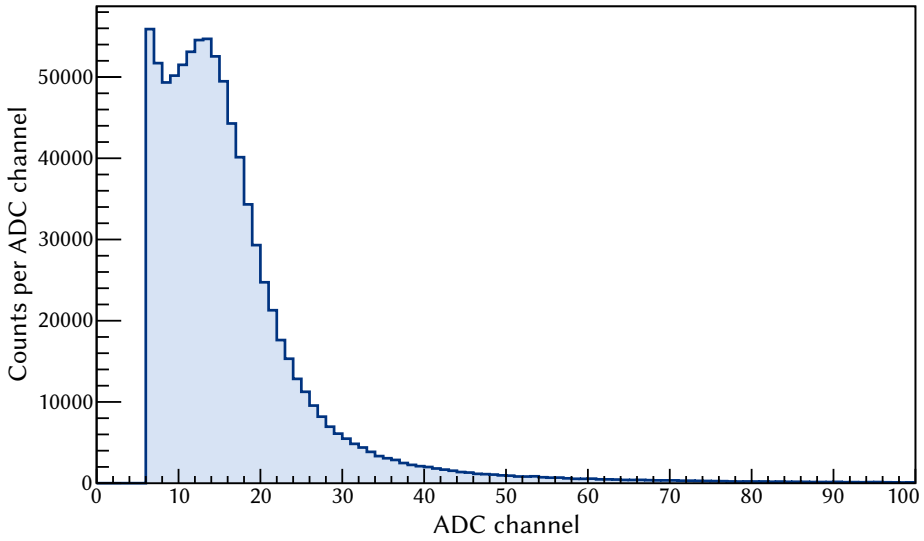


Figure 5.3: ADC histogram with more that 870 000 PXD hits from a run during the test beam campaign in May 2013, recorded with the ONSEN system. Six noisy pixels were masked out.

scintillators placed in the beam line. For every trigger, a single DHE module sent the zero-suppressed data to an ONSEN module, which forwarded it to a PC without performing any data-reduction steps. The recorded hits were combined with the output from the six-layer DATURA pixel telescope provided by DESY for alignment and verification purposes.

The ONSEN module used in this test was a single xFP v2.0 operated in a MicroTCA shelf. Since ROI inputs were not required, this module did not correspond to an ONSEN Selector board. Instead, it simply wrote all received events to the DDR2 memory, put the write address to an address FIFO, read the data back as soon as the output side was ready, and sent them to the read-out PC. This mechanism is used in a similar form during the buffering and reformatting of the output data in the Merger and Selector modules. The memory layout and buffer management were similar to those used in the current systems, but single, large buffers (4096 × 256 KiB) were used instead of the newer linked-list scheme. Another difference to the final system was the lower bit rate of the optical input links for the pixel data, which were operated at 3.125 Gbps instead of 6.25 Gbps.

A beam energy of 3 GeV at a rate of around 1.6 kHz was selected during most of the test. Because of the unknown trigger delay, the DHE read out 8 DHP frames for every trigger. This was possible because of the reduced frame size and read-out time of the small DEPFET sensor. The output data rate varied

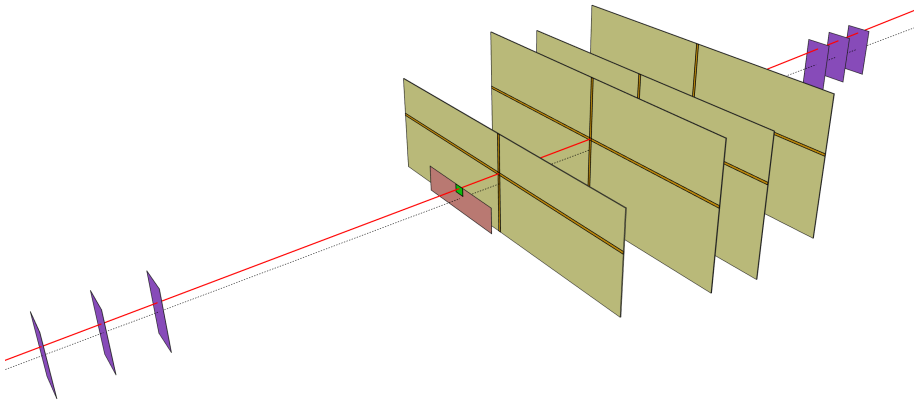


Figure 5.4: Illustration of the detector setup and ROI-generation mechanism during the DESY test in January 2014. Only active detector surfaces are shown. The visible sensors, from left to right, are: three EUDET planes, one PXD plane, four SVD planes, and three more EUDET planes.

between 200 and 300 kB/s. More than 50 million events were recorded in this configuration. Apart from minor errors in the structure and headers of some events (not caused by the ONSEN module), the data was successfully transmitted to the read-out PC, where it could be unpacked and processed. Figure 5.3 shows the histogram of observed ADC values for all pixels of the DEPFET matrix (excluding six noisy pixels) collected with the ONSEN system during one run. The shape is similar to the Landau distribution expected for the energy deposition of charged particles in a thin layer.

From the side of the ONSEN system, the test was very successful. The FPGA firmware ran reliably without requiring changes or reconfiguration. Data links in both directions were established and shown to be stable throughout the experiment. While the data-reduction mechanism could not be exercised in this test, it demonstrated the capability of the memory I/O and buffer-management systems. The test was also a success for SiTCP, which, at this time, competed with an alternative, UDP-based option for Ethernet I/O.

VXD test in January 2014

The first combined test of the two components that make up Belle II's vertex detector, the PXD and the SVD, took place in January 2014. The detector setup, which was placed in DESY test-beam area 24/1, is sketched in figure 5.4: One PXD layer and four SVD layers were arranged along the electron beam line, and three additional layers of the EUDET/AIDA pixel telescope [113] were

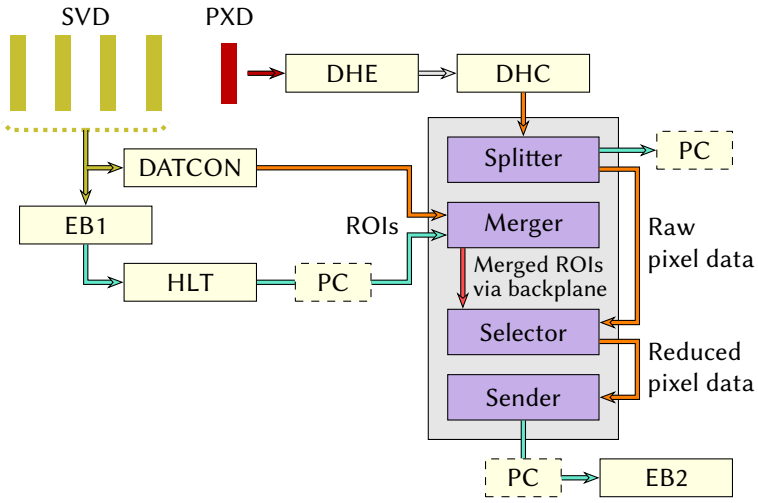
placed at each end of the setup. All sensors were located inside the field of the PCMAG solenoid provided by DESY, such that the magnetic field was oriented perpendicular to the beam axis in the horizontal plane, bending the electron beam upwards. With this geometry, it was possible to emulate a charged particle propagating from the Belle II interaction point outward through the VXD planes, leaving a hit in each layer[2] (cf. figure 3.11). The arrangement allowed to test not only the combined data acquisition of the two systems, but also the ROI-generation and data-reduction mechanisms.

Compared to the PXD test in May 2013, the second DESY test was much more complex, both in terms of hardware and data acquisition, and many more groups were involved. A large DEPFET matrix with 480 × 192 pixels sized 50 µm × 75 µm was bonded on a Hybrid 6 board. For the read-out of this sensor, four SWITCHERs, three DCDs, and three DHPs were required. The read-out of the SVD modules was performed with the FADC and COPPER boards that will also be used for Belle II. The data-acquisition group from KEK provided a scaled-down *Pocket DAQ* system with a working EB1, HLT, and EB2. As in the previous DESY test, trigger signals were generated from plastic-scintillator signals. The triggers were then fed into the FTSW system and distributed to SVD read-out boards and a DHC. ROIs were generated online by the HLT and a DATCON board, and a Pocket ONSEN system performed the ROI merging and pixel-data reduction. All major components of the planned Belle II data-acquisition system for the VXD were therefore present, and their interplay could be tested.

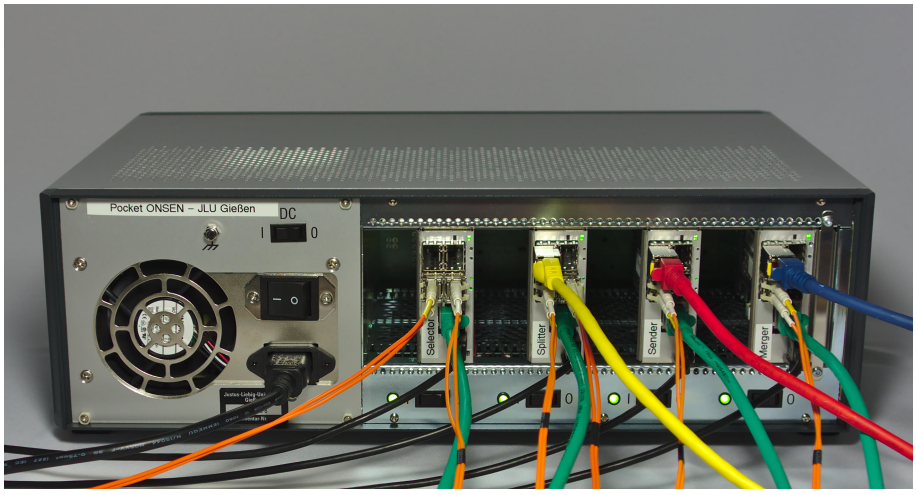
The part of the dataflow relevant for the data-reduction mechanism is shown in figure 5.5a: Following a trigger, a DHE read out the data from the PXD module and passed them to a DHC, which reformatted the packet and sent it to the ONSEN system on a 3.125 Gbps optical link. (The reduced bit rate, compared to the design value of 6.25 Gbps, was used because the link stability at full speed for the boards used in this test had not been validated at that point.) The load-balancing functionality of the DHC could not be tested due to the absence of the second PXD module. On the ONSEN side, the pixel data were received by a *Splitter* node designed specifically for this test. The Splitter node forwarded the data to the Selector, but also transmitted a copy to a PC through an additional Ethernet connection. This allowed the offline verification of the ONSEN system's operation, as the input before and after processing could be compared.

The SVD data from the four sensors for each event were collected and combined by the EB1. The HLT performed an online track reconstruction on

[2]Two PXD modules were prepared in order to make the arrangement as similar as possible for that in Belle II, but only one of them was ready in time for the test.



(a)



(b)

Figure 5.5: Setup of the ONSEN system during the test at DESY in January 2014. **(a)** Simplified dataflow diagram, showing all components relevant for the ONSEN system. **(b)** Photograph of a Pocket ONSEN shelf with the boards used during the test (reproduced from a previous publication [86], © 2015, IEEE).

the SVD hits, determined ROIs, and sent them to the ONSEN system. For this test, the HLT only calculated ROIs but did not reject complete events. During some runs, fixed ROI patterns were used instead of the calculated ROIs. Data taken during these runs were especially useful for the evaluation of the ROI-selection process. An additional PC could be inserted in the Ethernet data path between HLT and ONSEN. As with the Splitter node, this made it possible to record the data on the ONSEN input-side for later cross checks.

A single DATCON module received a copy of the SVD data, determined a second set of ROIs for every event, and sent them to the ONSEN system over a 1.5625 Gbps optical link. A Merger node received the ROIs from both sources and combined them, as described in section 4.2. The Merger firmware used in the test was mostly identical to the current version, barring some changes to the data formats and differences in the memory management introduced since then.

A Selector node received the raw pixel data from the Splitter via an optical link and the merged ROIs from the Merger via the MicroTCA backplane. It performed the filtering of the pixel hits with the ROIs for each event, as described in section 4.2. The firmware project used during this test was built with various debugging features, like integrated logic-analyzer cores, and versions of the ONSEN IP cores whose resource utilization has been reduced since then. As the usage of FPGA resources was close to the device limit, a complete Selector project with a perfect timing score could not be produced in time for the test. The SiTCP core was therefore offloaded to an additional *Sender* node. The Selector used an optical link to forward the processed pixel data to the Sender, which transmitted them to the EB2 using SiTCP. Since then, the Selector's resource utilization has been reduced by removing some of the debugging functionality that was required for the test. In addition, the buffering of incoming ROIs in memory was removed, as it was found to be unnecessary. This freed two MPMC ports, reducing the device occupancy significantly and allowing the reintegration of the SiTCP core into the Selector. Apart from these changes, only minor alterations were made, as in the case of the Merger node.

A photograph of a Pocket ONSEN system like the one used during the DESY test is shown in figure 5.5b. Four xFP cards from hardware revision v3.0 were were used in the test. This version still had four SFP cages, which were, however, not rated for the line rate of 6.25 Gbps that is required for Belle II.

Data was taken at different beam energies from 2 to 5 GeV and rates of several kHz. The magnetic field was turned on during later runs, after the initial alignment and calibration had been concluded. Some runs were taken with closed shutter (without beam) and artificial triggers to exercise the data-acquisition systems and record detector noise for pedestal calculations. More than 20 million events were recorded under stable run conditions with beam.

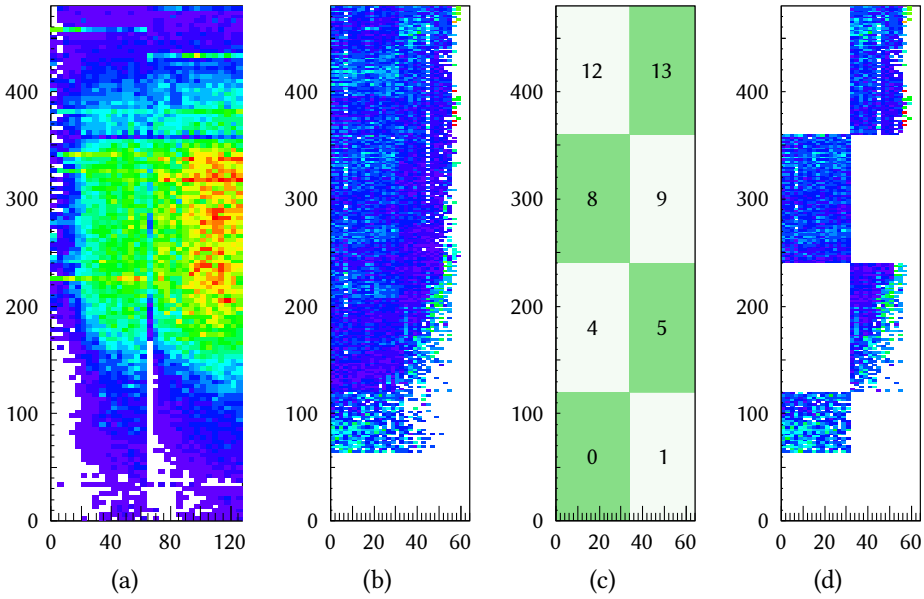


Figure 5.6: PXD hit maps from runs of the test at DESY in January 2014. **(a)** Data recorded with beam, with clearly visible beam spot. **(b)** Data from a "noise run", taken with uncalibrated pedestals and no beam. **(c)** Event-number dependent ROI pattern used for the noise run. **(d)** Data from noise run after event-number cut. (Figures reproduced from a previous publication [86], © 2015, IEEE.)

Figure 5.6 shows the hit distributions for two runs taken during the test[3]. The leftmost plot shows taken during a long run with active beam. The spread of the electron beam can be seen as a large blob in the hit map. The three plots on the right show the result of a test for the ONSEN system's data reduction mechanism. During this test, the beam shutter was closed, and PXD hits with uncalibrated pedestals were recorded. A predetermined ROI pattern was sent by the HLT, while the DATCON was disconnected. The result is the more-or-less uniform noise distribution shown in figure 5.6b. Figure 5.6c illustrates the ROI pattern that was used: Rectangular sections were selected for each event, varying with the event number. They can be made visible in the output data by plotting only hits with event numbers belonging to certain ROI areas (see figure 5.6d).

An event-wise analysis of the output from this test showed that all pixels on

[3]Note that all three DHPs were not read out during these runs. For cases where one DHP was read out, the data for 64 columns was recorded; for cases where two DHPs were read out, the data for 128 columns was recorded.

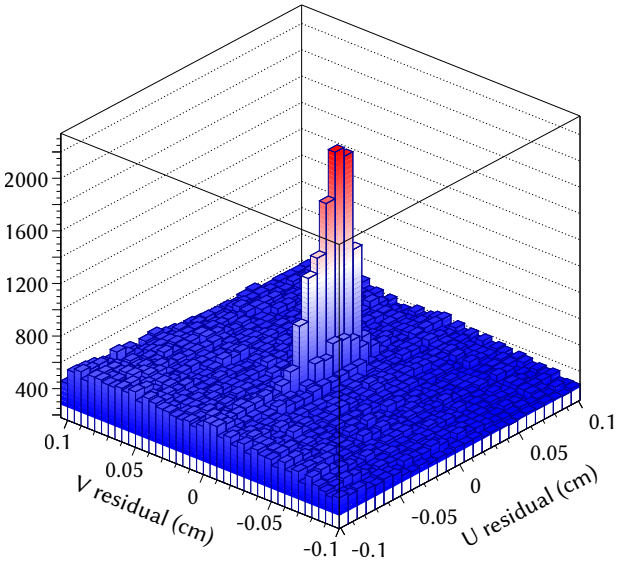


Figure 5.7: Residual plot with calibrated ROIs, from the DESY test in January 2014. (Figure adapted from reference [114], © 2015, IEEE.)

the output of the ONSEN system were inside the ROI generated for the event. This test proved the "selectivity" of the data reduction mechanism, as no data rejected by the HLT was passed to the output. It could not, however, show that all selected pixels sent by the DHC had been preserved. For a complete verification of the data-reduction mechanism, data from a subset of all runs were used, for which the inbound data streams of the ONSEN inputs had been recorded. For more than one million events captured in this mode, the data output from the ONSEN system was cross-checked against the incoming pixel data and the ROIs received from the HLT. It was confirmed that the ONSEN system operated correctly, forwarding selected pixels and discarding rejected ones, in all cases.

An additional study was performed by the group responsible for the generation of ROIs on the HLT [114]. The quality of the ROIs was determined by plotting the distance between each PXD hit on the output of the ONSEN system and the calculated ROI center (i.e., the intercept position of tracks from the online event reconstruction with the PXD layers) in a two-dimensional histogram. This so-called *residual plot* should exhibit a peaking structure if the actual pixel hits belonging to each track are present in the output data. The width of the peak can be used to determine the minimum ROI size. If the peak is not centered at the origin, its position can be used to adjust the ROI position by a constant translation parameter. This method was used during the

last runs of the test-beam campaign. The final residuals after calibration are shown in figure 5.7. The clear, central peak is a testament to the working track-reconstruction, ROI-calculation, and data-acquisition mechanisms.

In conclusion, the DESY test in January 2014 was a success for all involved subsystems. The full data-acquisition chain for the PXD and SVD was established, including real-time track reconstruction on the HLT, the ROI-feedback mechanism, and event building with data from both detectors. It proved, for the first time, the feasibility of the pixel-data reduction as well as the overall Belle II data-acquisition scheme. The interaction between the various components in the test demonstrated their interoperability, but also led to important realizations. For instance, the observation of data mismatches during the final event-building step necessitated the adoption of a new data format between the ONSEN system and the EB2, with which similar issues can be avoided in the future.

5.3 Carrier Board Tests

For the reasons previously stated in section 4.2, the test setups of the ONSEN system used so far did not correspond to the final architecture that includes data transmissions via the CNCB and the ATCA backplane. We can therefore not make a complete statement about the functionality of the system that will be used for Belle II. The main purpose of the Switch FPGA is the routing of data between the inserted xFPs and the backplane. The connections used for this routing are the most critical possible error sources. This section demonstrates the feasibility to use these connections for their foreseen purpose in the ONSEN system, dispelling possible doubts about the hardware performance and stability.

Bit-error rate for ATCA backplane links

The integrity of the ATCA backplane fabric-channels between two carrier boards was verified in the same manner as the xFP's high-speed serial links: Aurora cores were instantiated for the tested ports, and a pattern generator and tester confirmed that data was correctly received on both ends of the connection. As only two CNCBs of the required hardware revision were available at the time of this test, only one backplane channel could be checked. We chose fabric channel 13, connecting the two CNCBs in the leftmost and rightmost slot of the ATCA shelf used in the test (corresponding to physical slots 1 and 14 or logical slots 13 and 14, respectively). The other 15 MGTs of the Switch FPGA were also activated (although unconnected) during the test. This allowed us to exclude an issue observed in previous CNCB hardware revisions, where the

FPGA power-supply was insufficient to power more than 8 MGTs at the same time.

During a 33.5-hour test running at 3.125 Gbps, no errors were observed in either direction of the link. The total number of transferred bits was $2 \times 3.75 \times 10^{14}$, allowing us to claim a BER better than 4×10^{-15}. We expect less than one bit error for the transmission of merged ROIs during 2.5×10^{14} bits, corresponding to a run time of 22.2 hours or 2.4×10^{9} events. Since the backplane connection between the two most distant slots in the shelf was tested, the other backplane channels are expected to have the same or a better link integrity.

Investigation of CNCB-to-xFP LVDS links

The second crucial connection for the Switch FPGAs in the ONSEN system is the channel to the four AMC bays. Each such channel consists of four bi-directional LVDS links, driven by general-purpose SerDes cores in the two connected FPGAs. To include these links in the ONSEN firmware projects, new logic cores will have to be designed that provide LocalLink interfaces for transmitting and receiving payload data (as the currently used Aurora cores do), encode the data in an 8b/10b-base protocol, and implement to connections to the physical layer. Each such core will replace an Aurora core from the current firmware and is expected to have similar resource requirements, so that the overall device utilization of the projects will not be affected. For the moment, only the physical link integrity and the possibility to transfer data over these links can be confirmed.

Unlike the MGT links used for optical and backplane connections, The general-purpose I/O pins used for the LVDS channels do not support clock recovery. Consequently, a common clock source must be used for sender and receiver. To that end, the Switch FPGA drives a clock-fan-out chip, which distributes a 300 MHz clock to the AMC fabric-clock ports. The sending of data from Switch FPGA to AMCs is therefore source-synchronous, as the data and clock signals are provided by the same device, and their propagation delays (due to PCB-trace lengths) is more-or-less equal. In such a case, the destination can use the received clock to capture the data pins without requiring additional adjustments.

In the case of the reverse channel, the situation is different, since the AMC-FPGAs send out data with the captured clock, and the Switch FPGA samples each bit with its original clock. The phase relation between clock and data is still constant, but not easily predictable in this case, so that the captured data can be in an unstable state (i.e., in the middle of a bit transition) at the time of the sampling. This case was tested with a setup using one CNCB v3.3 equipped with four xFP v4.0. Each FPGA sent out a constant 10-bit word on

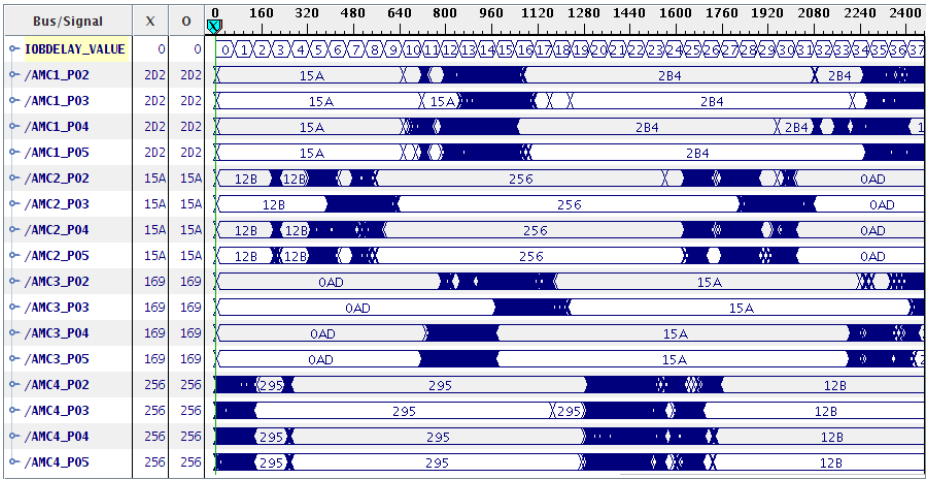


Figure 5.8: Waveforms of the deserialized LVDS data-streams, sent from the four xFP FPGAs to the Switch FPGA and sampled with varying input-delay values. The first row shows the current delay-tap value. Other rows show multiple data words, sampled at each delay value, for the four LVDS links from each xFP. For each link, valid-data windows with stable reception can be clearly discerned from the invalid phases. (Created with Xilinx Chipscope)

each of the LVDS links. On the Switch FPGA, additional variable delay elements were inserted in the data input paths. These elements could be used to shift the received bits in such a way that the capturing clock coincided with the valid-data phase of the signal. The line rate during the test was 600 Mbps, using the 300 MHz clock and double data-rate transmission.

Figure 5.8 shows a waveform, captured during this test on the Switch FPGA with a integrated logic-analyzer (ILA) core and Xilinx Chipscope [115]. The delays for all links were cycled between tap values of 0 and 63, each tap corresponding to a delay of 75 ps. The deserialized word was sampled 64 times for each value. For delays that put the clock edge inside the valid-data window, the captured word is expected to be identical for all 64 samples. Unsuitable delay values, on the other hand, should lead to an erratic output.

The captured waveform shows the delay value and the deserialized data words of all links during one delay cycle. The links are labeled with the AMC bay number and the AMC port used for the connection. The sampled words exhibit the expected behavior: For each link, a distinct valid-data window can be clearly discerned[4]. The windows are at least 10 taps wide, corresponding

[4]The captured words during the stable phases of all links are permutations of the same 10-bit word sent from the xFP-FPGAs. For a real data transmission, the next required step would be bit

to 750 ps or 45 % of the bit width (1.67 ns). The four ports of each AMC bay all show nearly identical window positions, reflecting the similar lengths of all PCB traces between the Switch FPGA and any one AMC bay. Consequently, a single delay value can be used to adjust all ports of an AMC bay. For bays 1 through 4, the extracted tap values are 25, 18, 25, and 11, respectively.

The observed delay values were stable between FPGA reconfigurations, the exchange of xFP cards, and firmware changes. To evaluate the link stability, we produced a new configuration for the Switch FPGA, fixing the delays to the extracted tap values. Again, constant 10-bit words were sent from both sides of the connection. With the correct delays, the captured words should be constant throughout the test. This was confirmed using a Chipscope ILA core, configured to trigger on any change of one of the deserialized bits.

During a 100-minute test, none of the 16 links on the Switch FPGA or the 4 links on any xFP-FPGA issued a trigger. We can therefore conclude that the LVDS links are working on the physical layer, requiring only a (constant) delay adjustment on the Switch FPGA's input links. Due to the shortness and method of the test, repeatedly transmitting the same data word, we cannot claim a BER. Since, however, the links run more than five times slower than the MGT-based xFP-interconnects (which have been confirmed as stable while going through two AMC connectors instead of one), link-integrity problems are very unlikely.

alignment, using special 8b/10b K-characters that are guaranteed to be unique within the data stream, so-called commas.

Chapter 6

Conclusion and Outlook

In the previous five chapters, I have presented the ONSEN system, responsible for the buffering and reduction of the output data from the Belle II pixel detector, and showed how it will achieve the required functionality and performance. The ONSEN system was developed by the Belle II group at the University of Gießen. It uses an FPGA-based hardware platform designed and built by the Trigger Lab at the IHEP in Beijing. The data processing with FPGAs is they key to the system's realization and therefore also a central topic of this thesis.

After a general overview of the thesis and a summary of the physics behind Belle II, I introduced the overall design of the experiment and its components, focusing on the special role of the pixel detector. I outlined the unique architecture of the Belle II data-acquisition system, which was built around the other subdetectors. The inclusion of the pixel detector into this system makes various extensions necessary in order to facilitate the processing and storage of the vast amount of background-dominated pixel data. In this context, I explained the necessity for an online data-reduction mechanism and specified the requirements for a system that is responsible for the buffering and real-time processing of the pixel data.

We have developed the ONSEN system to perform these tasks. In the main part of this thesis, I explained the ONSEN system in detail, showing how the created firmware leverages the FPGA platform to achieve the design performance. Among the biggest challenges for the system are the required data rates for the communication with external systems and the memory input and output for data buffering: A raw pixel-data stream of almost 600 MB/s must be received and processed by each module in the system. The buffering and retrieving of the data for all events takes up a memory bandwidth in the order of 1 GB/s. High-speed communication channels among the ONSEN modules must be established for data distribution, and Gigabit-Ethernet connections for

inbound and outbound data streams must be implemented based on the TCP protocol, which is particularly challenging for an FPGA-based system.

In the course of this thesis, I developed the parts of the ONSEN system's firmware that handle these functions. This includes memory I/O logic, realizing low-level access to a multi-port memory controller; the implementation of high-speed serial links, based on the Xilinx Aurora protocol; and the integration of the proprietary SiTCP core into the ONSEN system's dataflow. In the last chapter, I showed results from laboratory test setups that demonstrate the performance and stability of the created logic. The external interfaces performed without errors in long-term tests with a payload data-rate of more than 620 MB/s for the pixel-data reception and 118 MB/s for the TCP interfaces. Memory access worked reliably with a maintained throughput of around 1680 MB/s in a multi-port application, surpassing the requirements by far. For the 32 ONSEN modules responsible for pixel-data buffering and processing, this yields a total memory bandwidth of more than 53 GB/s, which must be compared to the raw data rate of less than 20 GB/s.

Other firmware-related work for this thesis concerned the processing of incoming data streams and the extraction of event metainformation. In addition, I worked on the commissioning and debugging of the hardware platform together with the developers from the IHEP. The current revisions of the boards used for the ONSEN system performed without problems in the long-term tests mentioned above. They were found to work reliably and can now be declared to be out the prototyping stage, so that the mass production of the ONSEN modules can commence.

From the side of the data processing (in particular the pixel-filtering logic based on regions of interest, developed for another thesis), the ONSEN system operates as required. To underline this point, I presented results from two system-integration tests, performed at the DESY test-beam facility in 2013 and 2014. During a test campaign including the Belle II pixel detector and silicon vertex-detector, more than 20 million events were processed by ONSEN modules and recorded with the Belle II Pocket DAQ system. An offline analysis, using a large part of the recorded data, proved the correct functionality of all involved systems. We conclude that the ONSEN system, in its current state, provides all required functionalities. The performance of all components relevant for the dataflow has been shown to surpass the requirements, and the hardware is in a mature state, needing no further design changes.

The firmware for the ONSEN system has not been completely finalized at the time of writing of this thesis. Most previous tests, including the DESY system-integration tests, have used small-scale setups that did not include the design dataflow through the ATCA carrier board. Further development of HDL code is required in order to integrate this board into the ONSEN firmware and

to establish the connections to and from the processing nodes. Nevertheless, the links required for the integration are usable and stable on the physical layer, as I have shown with the hardware tests presented in the latter part of the last chapter.

Several possibilities for a replacement of the current baseline mechanism for the processed-data output, which uses 32 TCP connections with the proprietary SiTCP core, are currently under investigation. Such a replacement is expected to free a large portion of the resources on the Selector nodes, providing reserves that may be needed for changes or upgrades in the future. One of the proposed options reduces the number of GbE output links from 32 to 8—more than enough for the output data rate of less than 600 MB/s—by collecting the output from four xFPs on their carrier board. The Switch FPGA would then transmit the combined data on a single link to an ATCA Ethernet switch with a 10GbE uplink. This solution will be tested in Gießen in the coming months, using multiple CNCBs, and possibly boards from previous Compute-Node versions, to measure the maximum throughput with a commercial 10GbE switch.

A second discussed option completely removes SiTCP form the output dataflow by collecting all processed events on a single node (requiring an additional CNCB and xFPs), and sending the combined data to a PC on one or more high-speed serial links. The feasibility of such a solution is currently being investigated in Gießen in the context of a Master's thesis for the PANDA experiment. The study uses an FPGA-based PCIe extension card, designed for the ALICE experiment, to feed the serial data into a PC system. An added benefit of a PC-setup with such a card lies in the possibility to send data from a PC to the ONSEN system with very high rates. Previous laboratory tests of the ONSEN firmware have relied on TCP connections for the transmission of generated pixel data to the Selector nodes. Such tests cannot ultimately prove that no unforeseen problems occur when the design values for data and trigger rates are reached. By using serial links with the design line-rate instead, realistic conditions can be tested and more meaningful predictions about the performance of the system can be reached.

The setup of the ONSEN system at the site of Belle II will begin in November 2015 with an ATCA shelf, two CNCBs, and eight xFP cards. Tests of the finalized firmware with this reduced setup will be used to confirm the complete dataflow scheme. The remaining hardware will then be delivered to KEK in 2016, and the system will be integrated with the Belle II data-acquisition system. The complete ONSEN system will be ready by the end of 2016.

The first PXD modules with the final versions of both sensor and ASICs will be assembled in late 2015. The next opportunity for tests of the ONSEN system together with real sensors and data-acquisition systems will be around April 2016, when a new experiment at the DESY test-beam facility is planned.

Two of the new PXD modules will be used in this test, together with four SVD planes, to establish a setup similar to the DESY test in January 2014. This will allow renewed tests of the complete VXD data-acquisition system, including ROI-based data reduction performed by the ONSEN. Changes and improvements introduced since the last test can be thoroughly evaluated, so that remaining open questions can be answered in time for the experiment's commissioning phase.

The current timeline for the construction of the PXD for Belle II, including all sensors and mechanics, is summer 2016. The detector will be commissioned and tested at the Max Planck Institute in Munich using cosmic muons, including data acquisition with an ONSEN setup. The assembly of the complete vertex detector at KEK will commence in summer 2017, in parallel to the second phase of the BEAST II experiment. BEAST II is a commissioning detector for SuperKEKB and Belle II that will help in the optimization of machine parameters and the understanding of the experiment's background. It will also include individual PXD sensors, read out with the help of a Pocket ONSEN system. The installation of the vertex detector for Belle II will take place after BEAST II has finished at the beginning of 2018. The schedule for the finalization and commissioning of the ONSEN system fits nicely into the timelines for the PXD and Belle II, so that everything will be in place and ready when the experiment starts its physics program in late 2018.

Appendix A

Node Architecture Details

A.1 Overview

This chapter shows the architecture of the ONSEN Merger and Selector nodes in terms of IP cores and their interconnection. The diagrams are simplified, omitting, in particular, some Xilinx cores required by the PowerPC. The omitted cores, in both cases, are:

- clock_generator, automatically instantiating clock primitives to generate required frequencies from the 100 MHz input clock,
- proc_sys_reset, generating reset signals for the PowerPC, buses, etc.,
- jtagppc_cntlr, providing the JTAG debugging interface for the PowerPC,
- xps_sysmon_adc, allowing read-out of system-monitor values (internal temperature and voltages) through the PLB,
- xps_intc, collecting interrupt signals from all cores and generating a single interrupt for the PowerPC,
- xps_bram_if_cntlr and bram_block, generating a block RAM region as a PLB-accessible address space,
- xps_uartlite, providing PLB access to the FPGA's UART interface,
- xps_mch_emc, providing PLB access to the FPGA's flash memory interface,
- xps_ll_temac, providing PLB access to the FPGA's Ethernet-PHY interface,
- user_reset, allowing software reset of ONSEN cores that do not provide a separate PLB interface, and
- ll_counter and plb_monitor, monitoring the throughput at the inputs and outputs of the Merger and Selector.

Of these, the last two were developed for the ONSEN system.

A.2 Merger Node Dataflow

Figure A.1 shows the IP cores relevant for the data processing by the Merger node. This section shortly describes the dataflow steps and function of all cores:

1. DATCON ROIs arrive via an xfp_mgt_aurora_ll.
2. A belle2_format_handler_ll parses them and extracts event information.
3. An npi_write_ll retrieves addresses of free buffers from a buffer_provider and writes the ROIs to memory.
4. A pix_write_lut creates a lookup-table entry from the event information and buffer address.
5. HLT ROIs arrive via an sitcp_ll.
6. A belle2_format_handler_ll parses them and extracts event information.
7. An hlt_lookup retrieves the memory address of corresponding DATCON ROIs from the lookup-table. A priority_addr_lock arbitrates the lookup-table access between pix_write_lut and hlt_lookup.
8. An npi_read_ll reads the DATCON ROIs back from memory and returns the now-unused addresses to the buffer_provider.
9. A hlt_ll_merger32 combines ROIs from HLT and DATCON into a single frame, and an ll_replace_checksum recalculates the CRC checksum.
10. An npi_write_ll retrieves addresses of free buffers from a buffer_provider and writes the merged ROI frame to memory.
11. A buffer_fifo forwards the buffer address to a reader as soon as the reader is ready to process it.
12. An npi_read_ll reads the merged ROI frame back from memory as soon as the output accepts data and returns the now-unused addresses to the buffer_provider.
13. The merged ROIs are sent to the Selector nodes via an xfp_mgt_aurora_ll (to be replaced with an LVDS SerDes core at a later point).

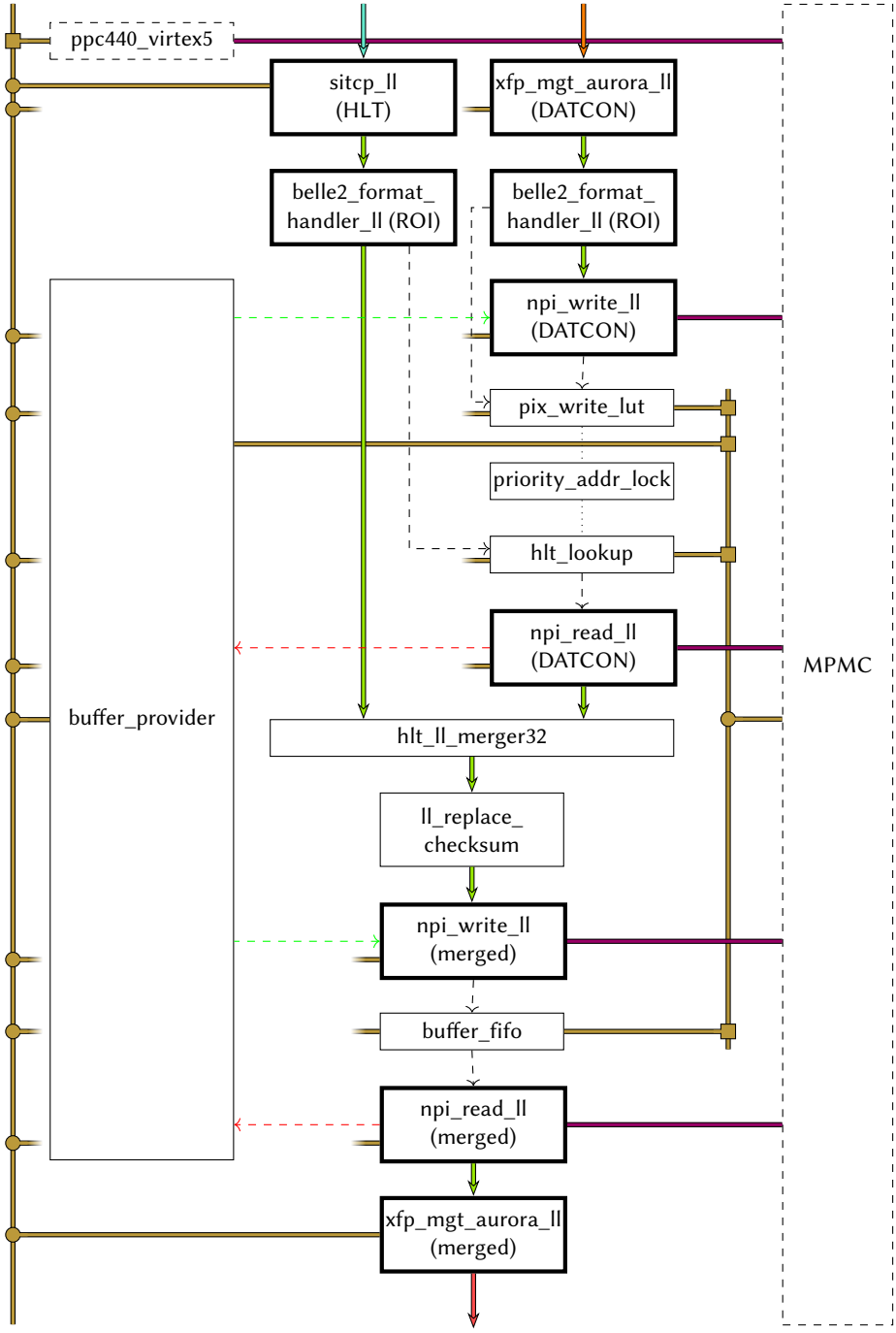


Figure A.1: Architecture of the Merger node in the XPS workflow. IP cores marked with bold lines are work from this thesis.

A.3 Selector Node Dataflow

Figure A.2 shows the IP cores relevant for the data processing by the Selector node. This section shortly describes the dataflow steps and function of all cores:

1. DHC pixel data arrive via an xfp_mgt_aurora_ll.
2. A belle2_format_handler_ll parses them and extracts event information. In addition, it combines all frames belonging to the same event into a single frame and appends an index frame with the frame lengths.
3. An npi_write_ll retrieves addresses of free buffers from a buffer_provider and writes the pixel data to memory. Then it prepends the index frame to the pixel data.
4. A pix_write_lut creates a lookup-table entry from the event information and buffer address.
5. Merged ROIs arrive via an xfp_mgt_aurora_ll (to be replaced with an LVDS SerDes core at a later point).
6. A belle2_format_handler_ll parses them and extracts event information.
7. An hlt_lookup retrieves the memory address of corresponding pixel data from the lookup-table. A priority_addr_lock arbitrates the lookup-table access between pix_write_lut and hlt_lookup.
8. An npi_read_ll reads the pixel data back from memory and returns the now-unused addresses to the buffer_provider. It uses the prepended index frame to split the event data back up into individual frames.
9. A roi_frame_handler_ll filters the pixel data according to the merged ROIs and calculates new checksums for processed frames.
10. A belle2_format_handler_ll parses the processed pixel data and extracts event information. In addition, it combines all frames belonging to the same event into a single frame and appends an index frame with the frame lengths.
11. An npi_write_ll retrieves addresses of free buffers from a buffer_provider and writes the processed pixel data to memory. Then it prepends the index frame to the pixel data.
12. A buffer_fifo forwards the buffer address to a reader as soon as the reader is ready to process it.
13. An npi_read_ll reads the processed pixel data back from memory as soon as the output accepts data and returns the now-unused addresses to the buffer_provider. It leaves the index frame prepended to the data.
14. The processed pixel data are sent to the EB2 via an sitcp_ll.

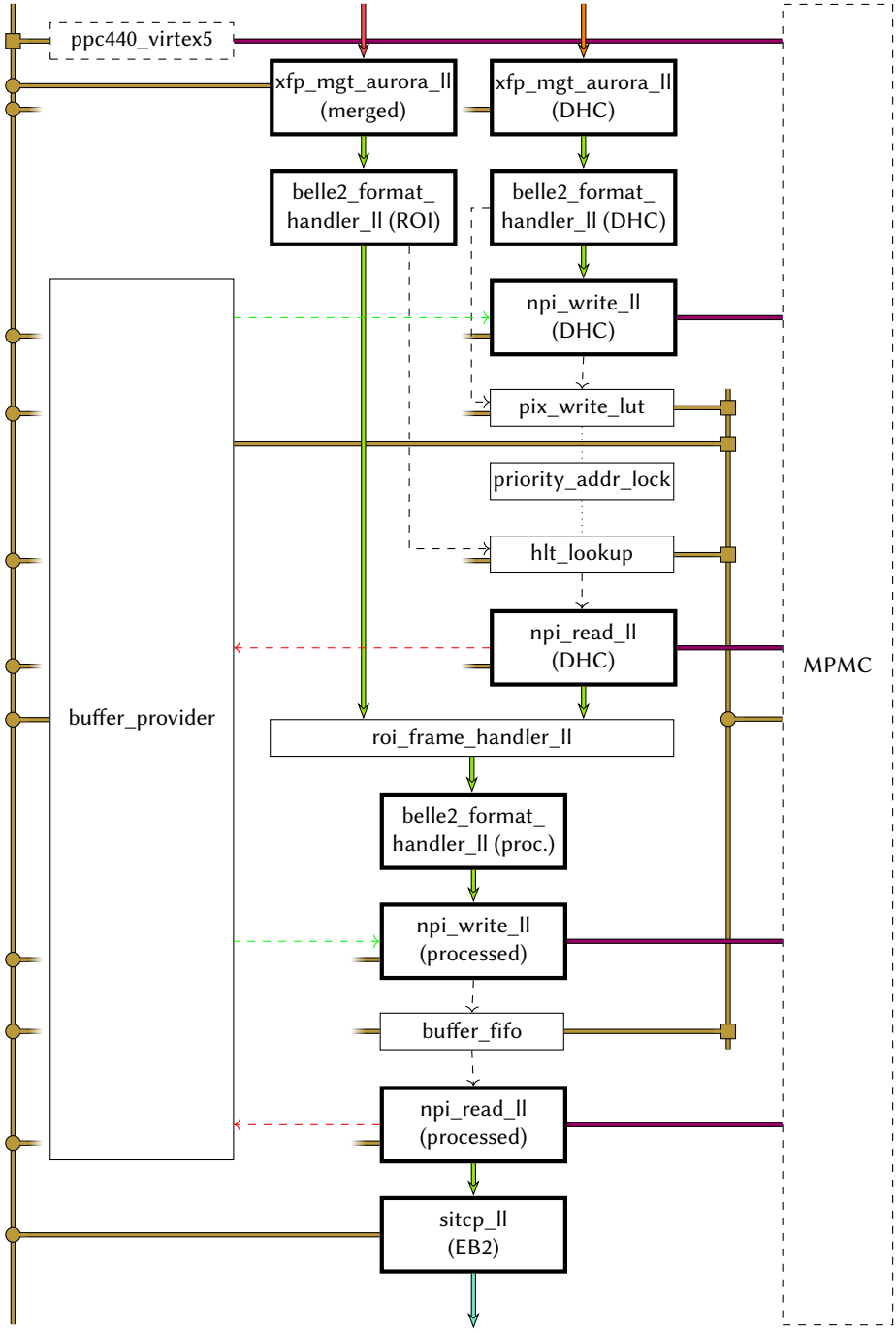


Figure A.2: Architecture of the Selector node in the XPS workflow. IP cores marked with bold lines are work from this thesis.

Appendix **B**

IP Cores

The FPGA design flow used for the ONSEN system is based on a hierarchical structure build of interconnected logic blocks, commonly referred to as *intellectual property cores* (IP cores). The system uses proprietary IP cores from Xilinx for common applications connected with the embedded CPU, such as the Ethernet and serial management interfaces. The cores that are involved in the steering, parsing, and processing of data, on the other hand, are custom developments. This chapter serves as a documentation of the cores that were created as part of this thesis and is therefore rather technical in nature.

B.1 Common Features

The development of FPGA projects using the embedded PowerPC CPU of the Virtex-5 FPGA requires the use of the Xilinx *Embedded Development Kit* (EDK) design flow [116]. IP cores created for the use in this design flow must follow the Xilinx Platform Specification Format [117]. The IP cores documented here were created in this format. They fall into one of two categories: *standalone* cores that contain only data-processing logic, and *peripheral* cores that, in addition, can be monitored and controlled by a CPU program over a bus interface.

B.1.1 PLB Slave Interface

The creation of standalone cores is straightforward, as they consist only of HDL code and a few definition files. Peripheral cores are more complicated: The *Create and Import Peripheral Wizard* is used to generate an HDL template with all ports and parameters that are necessary for connecting the core to a PLB bus; refer to the PLB specification [105] for the definition of the bus signals.

The template automatically instantiates the Xilinx LogiCORE IP PLBv46 Slave Single (v1.01a) [118]—a core that handles all communication with the PLB and conducts the relevant commands and requests to the user logic, which is encapsulated in a separate file. This mechanism allows the communication between the core logic and a CPU program. Various features of the PLB slave core can be enabled during generation. The ones that are relevant for the cores used in the ONSEN system are explained below.

B.1.2 Slave Registers

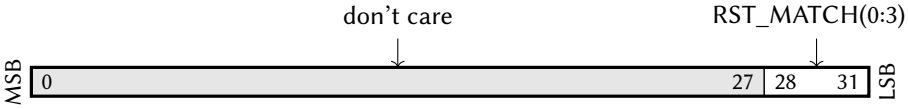
The simplest method of controlling and monitoring the core logic is by means of slave registers that are implemented as simple 32-bit signals in the HDL code. A CPU program can access these registers using requests to certain memory-mapped addresses on the PLB. The exact address for each register depends on offsets defined in the core configuration.

From the CPU's point of view, a register is usually either both readable and writable, used to set control and configuration bits of the logic, or only readable, used to query the core's status information. The peripheral wizard initially creates example code for a configurable number of read-/writable registers.

B.1.3 Software Reset

The PLB issues a reset signal to all connected peripherals when the FPGA is configured or the CPU is reset. In addition, the soft_reset core from the Xilinx proc_common library (v3.00.a) allows the user to reset an individual core by a write to the its soft-reset register. The address of this register depends on the configuration of the individual core.

Reset register



Bits	Name	Access	Description
28–31	RST_MATCH	W	A reset is triggered whenever the nibble `0b1010=0xA` is written to RST_MATCH. All other bits are ignored.

B.1.4 Interrupt Logic

If a peripheral is configured to issue interrupts to the CPU, the Xilinx LogiCORE IP Interrupt Control (v2.01a) is responsible for the generation of the interrupt. Using an interrupt-event signal with up to 32-bits, the core logic can request the controller to send an interrupt. The interrupt controller can be configured to interpret this signal in different ways; for instance, it can generate an interrupt when it detects a rising edge on one of the bits. After the CPU has registered the interrupt, it must query the interrupt controller to clear the interrupt and to find out which of the bits from the core logic was responsible for its generation.

The interrupt controller has PLB-accessible registers for the enabling, read-out, and clearing of interrupts. The base address of these registers depends on the configuration of each individual core. Their offsets and functionalities are explained in the interrupt controller's data sheet [119].

B.1.5 Dynamic Reconfiguration Port Access

Some FPGA primitives provide a *dynamic reconfiguration port* (DRP): Their parameters can be modified during run time by accessing their configuration register file through special signals. Several cores described in this chapter use the DRP of GTX_DUAL primitives (the Virtex-5 multi-gigabit transceivers). This allows them to tune, among other parameters, the transceiver's preemphasis and equalization settings, possibly fixing link integrity problems. For details, refer to GTX user guide [120, p. 117].

The GTX_DUAL DRP uses 16-bit configuration words addressed with 7 bits. The complete register therefore consists of 256 bytes. These are mapped into the core's PLB address space as a *user memory space*, so that they can be accessed like any other memory region. PLB requests to this region are translated to DRP requests by a custom bridge interface. The start address of each DRP memory region depends on the configuration of the individual core and the number of DRP ports it can access.

The mapping of PLB addresses to DRP addresses is not trivial because PLB addresses refer to bytes while DRP addresses refer to 16-bit words. This is made even more complicated by the fact that PLB data words usually use ascending bit order (the leftmost, most significant bit having bit index 0), while DRP configuration words use descending bit order (the rightmost, least significant bit having bit index 0). The left byte (15:8) of the DRP word at address DRP_ADDR is mapped to PLB address $2 \times$ DRP_ADDR while the right byte (7:0) is mapped to $(2 \times \text{DRP\_ADDR}) + 1$. Refer to figure B.1 for an illustration.

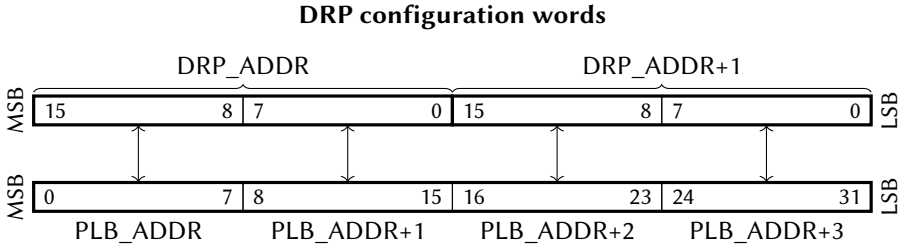


Figure B.1: Mapping between DRP and PLB addresses. DRP_ADDR is the address of a DRP configuration word. PLB_ADDR is the PLB address offset from the base address of the DRP memory region for the transceiver.

B.1.6 LocalLink Interfaces

For the transfer of large data packets from one core to another, the cores described in this chapter use the Xilinx LocalLink interface standard. LocalLink defines a number of signals for a point-to-point transmission—from *source* to *destination*—of data frames consisting of an arbitrary number of bytes. The bytes are transferred as words of 2^n bytes per word. All LocalLink interfaces used in the ONSEN system use 32-bit data words. For details about the LocalLink signal definitions refer to the protocol specification [106].

When the source transmits a data word, it defines how many of its bytes are valid and whether the word starts or ends a data frame. During any clock cycle, the destination can choose not to accept the transmitted data word, compelling the source to send it again in the next clock cycle. A possible reason for this could be a full input FIFO. This mechanism generates *back pressure* that can propagate through multiple cores up to the initial data source.

B.1.7 Pointer Bus and Return Interfaces

The PTR_BUS and PTR_RET interfaces were devised for the distribution of memory pointers in the ONSEN system by the buffer_provider core (see section B.7). For data transmission, they rely on a simple handshaking scheme, where every transferred word must be acknowledged by the opposite side. The interfaces are therefore not suited for high-throughput data transmission, but sufficient for the request-and-grant mechanism used for single pointers.

The main purpose of the PTR_BUS interface is the transmission of single 32-bit data words from a single master to multiple slaves; the slaves can request data and the master arbitrates its output between them. An example is the

Table B.1: Signals of the PTR_BUS interface

Signal	**Width**	**Direction** Master—Slave	**Description**
REQUEST	1	⟵⊣	Asserted by the slave to request new data; deasserted when the slave has received VALID and sampled DATA; one master can have REQUEST inputs from multiple slaves
DATA	32	⊢⟶	Data output from master; can be connected to multiple slaves
VALID	1	⊢⟶	Asserted by the master, upon a REQUEST from a slave, to inform the slave that it may sample DATA; deasserted when the slave has deasserted REQUEST; one master can have VALID outputs to multiple slaves, but no more than one may be asserted at each point in time

distribution of pointers to free memory regions from a buffer-provider core to several independent memory-writer cores. See table B.1 for the signal definitions.

The main purpose of the PTR_RET interface is the reception of single 32-bit data words by a single master from multiple slaves; the slaves can offer data and the master arbitrates its input between them. An example is the collection of pointers to disused memory regions from several independent memory-reader cores by a buffer provider core. See table B.2 for the signal definitions.

Table B.2: Signals of the PTR_RET interface

Signal	**Width**	**Direction** Master—Slave	**Description**
REQUEST	1	⟵⊣	Asserted by a slave to inform the master that the slave's DATA output is valid; deasserted when the slave has received GRANT; one master can have REQUEST inputs from multiple slaves
DATA	32	⟵⊣	Data output from slave; one master can have DATA inputs from multiple slaves
GRANT	1	⊢⟶	Asserted by the master, upon a REQUEST from a slave, to inform the slave that DATA was sampled; deasserted when the slave has deasserted REQUEST; one master can have GRANT outputs to multiple slaves

B.2 xFP Aurora Wrapper

B.2.1 Core properties

- **Name:** xfp_mgt_aurora_ll
- **Current version:** v1.05.a
- **Type:** PLB peripheral

B.2.2 Overview

The *xFP Aurora wrapper* provides a convenient interface for the use of the multi-gigabit transceivers in the xFP's Virtex-5 FPGA with the Aurora protocol. The current version was created for the xFP v3.0, but is also compatible with v4.0. The wrapper automatically instantiates GTX transceivers and Aurora cores for the ports selected by the user and maps the LocalLink input and output interfaces to logical names. The core augments the functionality of the Xilinx Aurora implementation by providing

- parameterizable asynchronous RX and TX FIFOs, allowing the use of arbitrary LocalLink clocks for each port;
- adjustable bit rate through automatic parameter setting of the GTX and digital clock manager cores;
- automatic back pressure (Aurora native flow control), inhibiting or reactivating the data transmission from a connected Aurora channel partner based on the RX FIFO occupancy;
- a PLB slave interface, providing
  - status and control registers,
  - interrupt generation, and
  - software reset of the user logic;
- access to the dynamic reconfiguration port (DRP) of the instantiated GTX_DUAL transceivers, accessible as a PLB user memory space; and
- dynamic generation of all necessary location and timing constraints.

Aurora is a point-to-point link-layer protocol devised by Xilinx for high-speed serial connections. The version of the protocol used in this wrapper core makes use of an 8b/10b encoding scheme [121] for the serial data stream. For details about the protocol, see the specification [109].

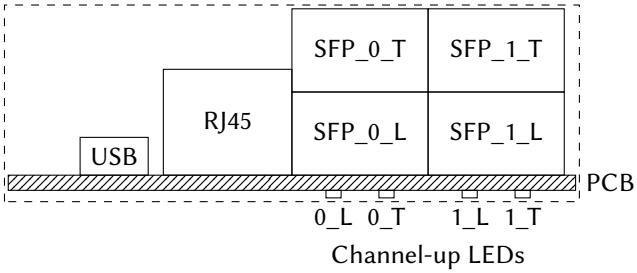


Figure B.2: Placement of connectors on the xFP v3.0. The view is on the board's front panel. For the xFP v4.0, only SFP+ cages 0_T and 1_T are present.

For the Aurora logic, the wrapper uses HDL code created with the Xilinx core-generator tool. A Xilinx Aurora 8B/10B core, version 5.3 [122], was generated with the following parameters:

- **Aurora lanes:** 1
- **Lane width:** 4 bytes
- **Line rate:** 3.125 Gbps (adjustable in the wrapper)
- **GT REFCLK:** 156.25 MHz (adjustable in the wrapper)
- **Dataflow mode:** Duplex
- **Interface:** Framing
- **Flow control:** Immediate NFC

The generated Aurora logic only allows the simultaneous use of multiple serial links as bonded lanes of a single Aurora channel. As a consequence, it is not possible to use the two individual transceivers of a single GTX_DUAL tile for independent connections to separate channel partners. This functionality is, however, necessary for the wrapper core since it is required for the simultaneous use of multiple of the xFP's AMC and SFP ports. The generated code was therefore modified by extracting the instantiation of the GTX_DUAL primitive and bringing it up in the instantiation hierarchy above the Aurora cores. This makes it possible to use both transceivers of a dual tile for independent Aurora channels.

Based on the Aurora ports selected by the user, the wrapper core automatically instantiates all required transceivers. It also allows the instantiation of additional unused transceivers for reference clock forwarding (see section B.2.4). All instantiated ports must use the same reference clock and bit rate, but multiple instances of the wrapper can be used if different bit rates are required.

The ten Aurora ports that are available are listed in table B.3. The seemingly odd numbering scheme is a consequence of the ascending bit numbering of PLB data words; it provides a consistent mapping of port numbers to bit positions

Table B.3: Aurora Port numbering for the wrapper

Port	Designator	Description	GTX_DUAL	GTX#
6	AMC_P20	AMC Port 20	X0Y5	0
7	AMC_P19	AMC Port 19	X0Y5	1
8	AMC_P18	AMC Port 18	X0Y6	0
9	AMC_P12	AMC Port 12	X0Y1	1
10	AMC_P10	AMC Port 10	X0Y1	0
11	AMC_P06	AMC Port 6	X0Y2	0
12	SFP_1_T	Upper right SFP cage	X0Y3	0
13	SFP_1_L	Lower right SFP cage	X0Y3	1
14	SFP_0_T	Upper left SFP cage	X0Y4	0
15	SFP_0_L	Lower left SFP cage	X0Y4	1

in PLB slave registers and interrupt vectors. The naming of the ports that are connected to SFP transceivers is not immediately obvious. The placement of the SFP cages, along with the according port names, is therefore depicted in figure B.2.

B.2.3 Ports and Buses

Figure B.3 shows the I/O ports and bus interfaces that the core provides. Each activated Aurora port provides a LocalLink input, output, or both, depending on the configuration, and serial I/O signals that must be connected to top-level FPGA ports. Channel-up signals for the SFP cages are automatically routed to the corresponding LEDs. For the clocking options, see section B.2.4 below.

B.2.4 Clock Routing and Transceiver Instantiation

The reference clock for the Virtex-5 GTX multi-gigabit transceivers can be sourced either from a dedicated oscillator connected with the FPGA or from a usual fabric clock signal. The core parameter C_REFCLK_SRC determines which kind of clock input is used. This section describes the different clocking options.

MGTREFCLK

For bit rates above 1 Gbps, the transceiver's reference clock must be sourced by a dedicated external oscillator connected to the pins MGT_REFCLK_P and MGT_REFCLK_N. If MGTREFCLK is selected as the reference clock source,

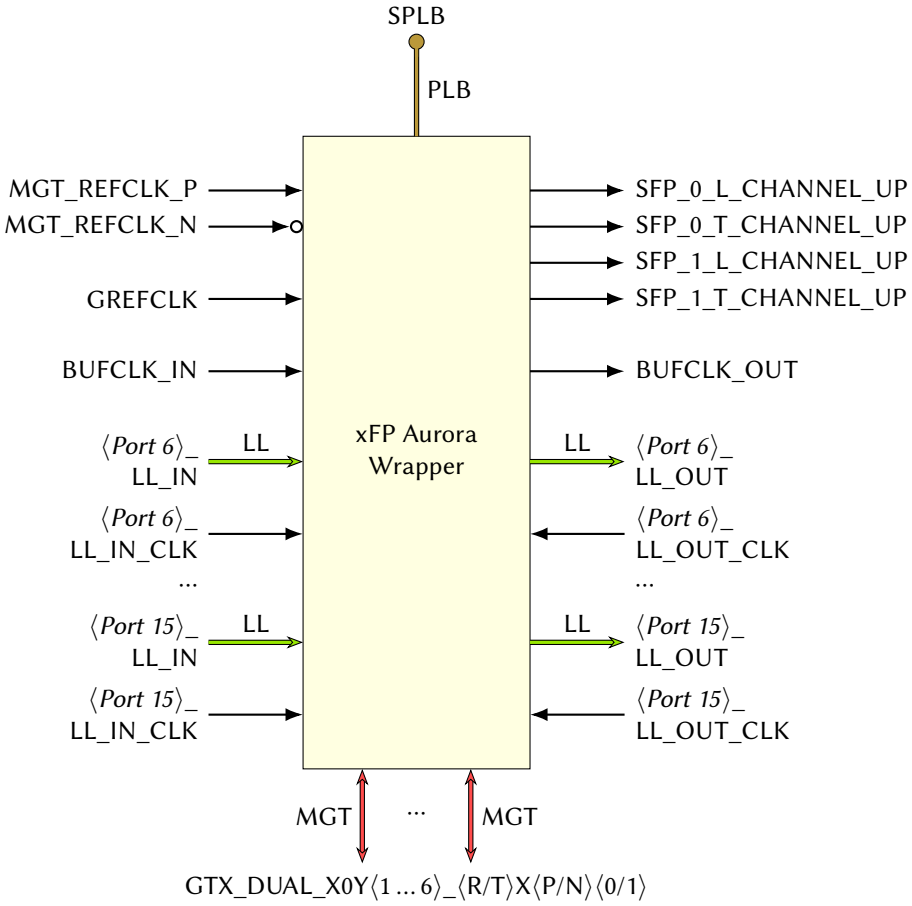


Figure B.3: I/O ports and buses of the xFP Aurora wrapper

these ports become valid and must be routed to external ports in the FPGA project's top entity. The wrapper core automatically instantiates a differential input buffer for the clock inputs and creates the location constraints to connect them to the correct pins, based on the C_MGTREFCLK_LOC setting.

If MGTREFCLK is used, the GTX reference clock routing scheme requires that *all* transceivers between the MGTREFCLK input and transceivers using the clock be instantiated and powered. If, for example, BUFDS_X0Y4 is selected as the C_MGTREFCLK_LOC setting and AMC Port 10 is the only activated Aurora port, the wrapper core will automatically instantiate the transceiver at location X0Y1. In order for the reference clock to reach this transceiver, the ones at locations X0Y2 through X0Y4 must also be instantiated. This can be achieved either by forcing the instantiation with the C_FORCE_GTX_DUAL_X0Y⟨*n*⟩

parameters, or by instantiating the transceivers in a separate core, such as a second Aurora wrapper or an SiTCP wrapper. Failing to provide a valid reference clock routing path will lead to an error in the mapping phase of the design.

Buffered input

If two cores use the same MGTREFCLK input, they cannot both be configured to instantiate an input buffer for the reference clock pins, since only one such buffer for each clock input may be present in the complete design. For that case, each wrapper core exposes its buffered reference clock signal through the BUFCLK_OUT port. A second core using the same clock source can be configured to use a *buffered input* for its reference clock. In that case, the port BUFCLK_IN becomes available, to which the buffered clock output from another core can be connected. The same restrictions for reference clock routing as for the MGTREFCLK setting apply.

GREFCLK

For bit rates up to 1 Gbps, a simple fabric clock signal (sourced, for example, from a digital clock manager) can be used. It is connected to the GREFCLK port of the transceiver. This mode does not have any restrictions with regard to the clock routing.

B.2.5 PLB Slave Interface

The core uses two parameters to define the address regions that are associated with functions that can be accessed via PLB:

1. The address region starting at the PLB slave base address (C_BASEADDR) contains status and control registers, the software reset register, and the interrupt registers. The memory ranges of the individual regions, given as offsets from C_BASEADDR, are shown in the following table:

Name	Description	Range	
		Start	End
USER_SLV_BASEADDR	Slave registers	`+0x000`	`+0x0FF`
RST_BASEADDR	Reset register	`+0x100`	`+0x1FF`
INTR_BASEADDR	Interrupt control	`+0x200`	`+0x2FF`

   The slave registers for this core are listed in section B.2.6. For an explanation of the soft-reset and interrupt-generation mechanisms, refer to

sections B.1.3 and B.1.4, respectively. The definition of the interrupt signals generated by this core is given in section B.2.7.

2. The DRP base address (C_MEM0_BASEADDR) allows access to the dynamic reconfiguration ports of the GTX transceivers (see section B.2.8).

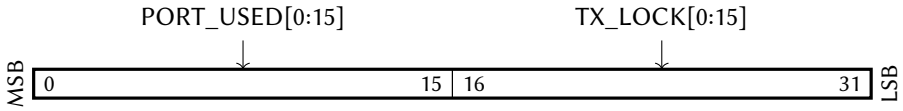
B.2.6 Status and Control Registers

The following table lists the available status and control registers, along with their respective offsets from the slave register base address:

Description	Name	USER_SLV_ BASEADDR Offset	R/W
Aurora Status Register 0	ASTA0	+0x00	R
Aurora Status Register 1	ASTA1	+0x04	R
Source-not-ready Register	SRDN	+0x08	R
Destination-not-ready Register	DRDN	+0x0C	R
LocalLink-Enable Register	LLEN	+0x10	RW

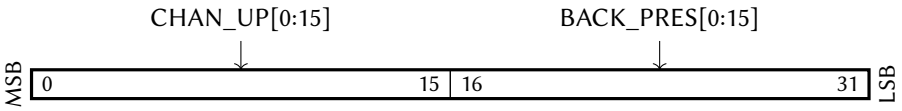
This section describes the function of each register and its bits. Note that many of the registers described here are divided into words of 16 bits. In that case, each bit is usually associated with one Aurora port; the bit index denotes the port number as defined in table B.3.

Aurora Status Register 0 (ASTA0)



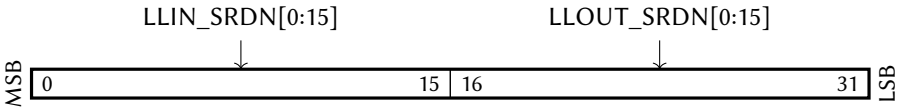
Bits	Name	R/W	Initial	Description
0–15	PORT_USED	R	0	Indicates that the port is configured to be active as Aurora receiver, transmitter, or both.
16–31	TX_LOCK	R	0	Indicates that the transceiver associated with the port has achieved TX lock. A 0 for an activated Aurora port indicates a reference clock problem.

Aurora Status Register 1 (ASTA1)



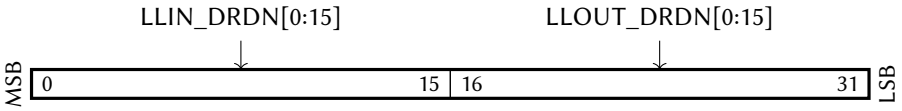
Bits	Name	R/W	Initial	Description
0–15	CHAN_UP	R	0	Indicates that the port has an active Aurora channel with a link partner.
16–31	BACK_PRES	R	0	Indicates that the port's RX FIFO occupancy is above the NFC threshold. In this case, back pressure is applied to the Aurora channel partner (see section B.2.10).

Source-not-ready Register (SRDN)



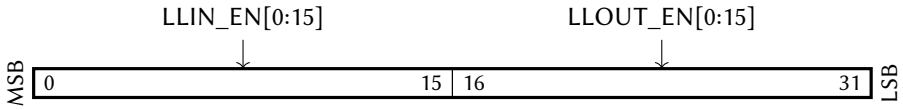
Bits	Name	R/W	Initial	Description
0–15	LLIN_SRDN	R	0	SRC_RDY_N status for the port's LocalLink input; if 1, the LocalLink source is not offering data
16–31	LLOUT_SRDN	R	0	SRC_RDY_N status for the port's LocalLink output; if 1, the TX FIFO is empty

Destination-not-ready Register (DRDN)



Bits	Name	R/W	Initial	Description
0–15	LLIN_DRDN	R	0	DST_RDY_N status for the port's LocalLink input; if 1, the RX FIFO is full
16–31	LLOUT_DRDN	R	0	DST_RDY_N status for the port's LocalLink output; if 1, the LocalLink destination is not accepting data

LocalLink-Enable Register (LLEN)



Bits	Name	R/W	Initial	Description
0–15	LLIN_EN	R	0	Enable dataflow through the port's LocalLink input interface
16–31	LLOUT_EN	R	0	Enable dataflow through the port's LocalLink output interface

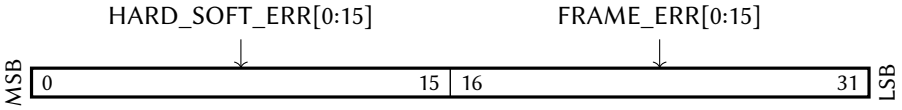
B.2.7 Interrupts

The interrupt controller was generated with the following parameters:

- **Use device ISC:** no
  (C_INCLUDE_DEV_ISC = false)
- **Use device ISC priority encoder service:** no
  (C_INCLUDE_DEV_PENCODER = false)
- **Number of interrupts generated by user logic:** 32
  (Length of C_IP_INTR_MODE_ARRAY: 32)
- **Capture mode:** rising-edge detect
  (Values of C_IP_INTR_MODE_ARRAY: 5)

The user logic generates interrupts for each port if the connected Aurora logic produces an error. For the definition of Aurora hard, soft, and frame errors, refer to the Aurora user guide [122, p. 64].

User-logic interrupt vector (IP2Bus_IntrEvent)



Bits	Name	Description
0–15	HARD_SOFT_ERR	Asserted when the Aurora logic for the port issues either a soft error or a hard error
16–31	FRAME_ERR	Asserted when the Aurora logic for the port issues a frame error

B.2.8 DRP Access

Refer to section B.1.5 for an explanation of the DRP-to-PLB mapping. Since the Aurora wrapper can instantiate more than one GTX_DUAL primitive, the offset of the DRP memory region for each transceiver depends on the GTX_DUAL location as well as the base address parameter C_MEM0_BASEADDR: The region belonging to GTX_DUAL at X0Yn starts at C_MEM0_BASEADDR + (n × `0x100`) and extends up to the region belonging to the next transceiver. Refer to table B.3 for the location of the GTX_DUAL associated with each Aurora port.

Besides parameters related to the link integrity, it is, in principle, possible to modify the parameters that determine the link bit rate online. This would, however, require changes of the user clock frequency, too. In a future version of the core, DRP access to the DCM for the user clock may be added to allow this.

B.2.9 Bit-Rate Selection

The Aurora logic used for this core was originally generated for a fixed bit rate and reference clock frequency. The parameter changes needed for other bit rates were determined by a systematic comparison of core generator outputs created with different settings, and the relevant parameters are used as generics of the instantiated Aurora cores. This makes it possible to change between bit rates and reference clocks without the need to instantiate different versions of the Aurora logic.

The wrapper core determines the values for the relevant parameters automatically based on the reference clock frequency and bit rate selected by the user. It also creates the correct clock period and location constraints. The bit rate settings are defined in a CSV file in the core's devl/ directory. Support for

new frequencies can be added by generating an Aurora core with the required settings and appending the relevant parameters to this file.

B.2.10 Flow Control

The core employs the Aurora *native flow control* mechanism to generate back pressure from an Aurora receiver to its channel partner if the receiver is not able to process the received data quickly enough. This can be the case when the LocalLink destination on an LL_OUT interface blocks the data flow, or if the LocalLink clock runs slower than the Aurora user clock.

The occupancy of a port's RX FIFO determines when the core sends a flow control command to the Aurora channel partner: When the FIFO is at least half full, the Aurora logic inserts an XOFF symbol into the serial data stream. When the channel partner receives this symbol, it immediately ceases its data output; in the case of a Xilinx Aurora core with NFC enabled, this means that it will apply back pressure ("destination-not-ready") to its own LocalLink input. The RX FIFO must be large enough, so that all data that the channel partner sends before it has received and processed the XOFF command can still be stored.

As soon as the port's LocalLink output destination has accepted enough data and the RX FIFO occupancy drops below the half-full mark, the Aurora logic sends the XON symbol to the channel partner, enabling its data output again. The initialization or reset of the complete core also triggers the sending of an XON symbol, in case the channel partner was disabled during a previous connection.

B.3 SiTCP Wrapper

B.3.1 Core properties

- **Name:** sitcp_ll
- **Current version:** v1.10.a
- **Type:** PLB peripheral

B.3.2 Overview

The *SiTCP wrapper* instantiates the proprietary SiTCP core from Bee Beans Technologies Co., Ltd. [110] and adapts it to the EDK design flow used by the ONSEN system. SiTCP is an FPGA implementation of the TCP/IP protocol for data transmission over GbE. It listens for TCP connections on a configurable IP address and port. After a connection has been established by a link partner, bytes written into the SiTCP TX FIFO are automatically packed into TCP packets and sent over the connection. Conversely, bytes received over the connection are presented by SiTCP on an RX FIFO interface. Details about the core can be found in the manual [123], available from the SiTCP forum [124].

The wrapper augments the functionality of the SiTCP core by providing

- parameterizable asynchronous RX and TX FIFOs, allowing the use of arbitrary LocalLink clocks;
- either a GMII interface for a connection via a PHY (e.g., to an 1000BASE-T RJ45 socket), or an automatically instantiated PCS/PMA core for a 1000BASE-X connection using an MGT (e.g., via an SFP port);
- a PLB slave interface, providing
  - status and control registers,
  - interrupt generation, and
  - software reset of the user logic;
- access to the dynamic reconfiguration port (DRP) of the instantiated GTX_DUAL transceivers, accessible as a PLB user memory space;
- emulation of an EEPROM for the SiTCP license file, accessible as a PLB user memory space; and
- dynamic generation of all necessary location and timing constraints.

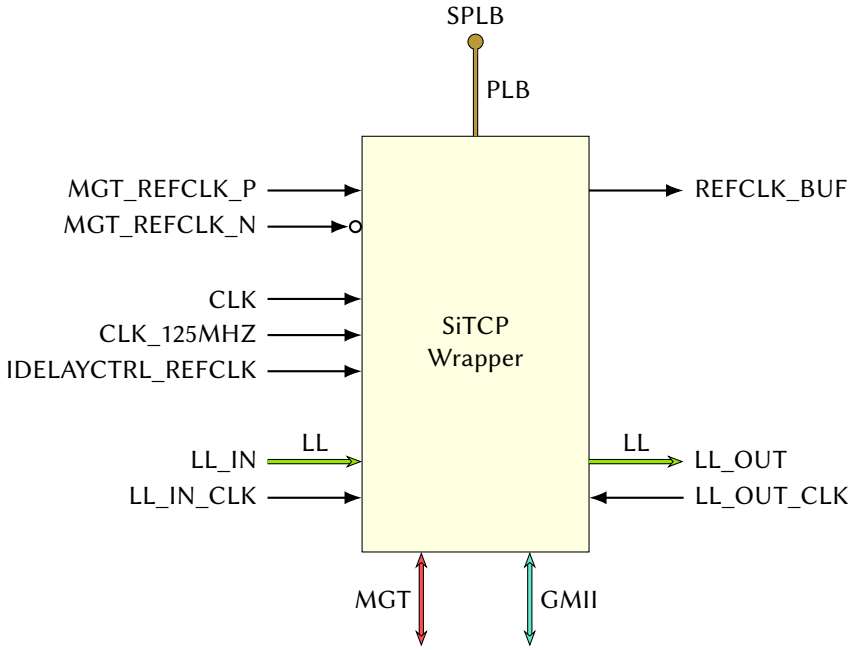


Figure B.4: I/O ports and buses of the SiTCP wrapper

B.3.3 Ports and Buses

Figure B.4 shows the I/O ports and bus interfaces that the wrapper provides. Data sent to a LocalLink input is passed to the SiTCP TX FIFO, and data from the RX FIFO is presented on a LocalLink output. I/O ports are available depending on the chosen interface type, and must be connected to FPGA top-level ports. SiTCP requires a core clock running with at least 130 MHz to be connected to the CLK input. The 125 MHz Ethernet clock is either provided to CLK_125MHZ, or an external MGT reference clock running at that frequency must be supplied. If the GMII interface is used with input delays, an additional 200 MHz must be connected to IDELAYCTRL_REFCLK. If the MGT interface is used, rules for transceiver instantiation and clock routing must be observed (see section B.2.4).

B.3.4 PLB Slave Interface

The core uses three parameters to define the address regions that are associated with functions that can be accessed via PLB:

1. The address region starting at the PLB slave base address (C_BASEADDR) contains status and control registers, the software reset register, and the

interrupt registers. The memory ranges of the individual regions, given as offsets from C_BASEADDR, are shown in the following table:

Name	Description	Range	
		Start	End
USER_SLV_BASEADDR	Slave registers	+0x000	+0x0FF
RST_BASEADDR	Reset register	+0x100	+0x1FF
INTR_BASEADDR	Interrupt control	+0x200	+0x2FF

The slave registers for this core are listed in section B.3.5. For an explanation of the soft-reset and interrupt-generation mechanisms, refer to sections B.1.3 and B.1.4, respectively. The definition of the interrupt signals generated by this core is given in section B.3.6.

2. The DRP base address (C_MEM0_BASEADDR) allows access to the dynamic reconfiguration ports of the GTX transceivers (see section B.1.5).

3. The EEPROM base address (C_MEM1_BASEADDR) allows to read and write the memory block connected to the EEPROM bridge containing the SiTCP license (see section B.3.7).

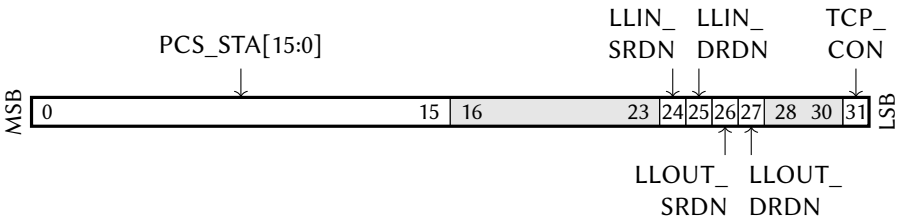
B.3.5 Status and Control Registers

The following table lists the available status and control registers, along with their respective offsets from the slave register base address:

Description	Name	USER_SLV_ BASEADDR Offset	R/W
Status Register	STA	+0x00	R
Control Register 0	CTRL0	+0x04	RW
Control Register 1	CTRL1	+0x08	RW

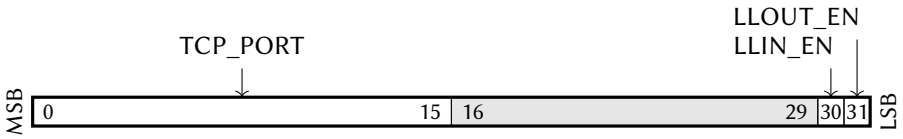
This section describes the function of each register and its bits.

Status Register (STA)



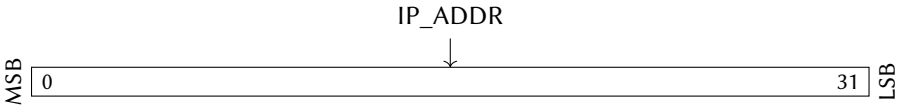
Bits	Name	R/W	Initial	Description
0–15	PCS_STA	R	0	pcs_status_vector of the PCS PMA core (see section B.3.8)
24	LLIN_SRDN	R	0	SRC_RDY_N status for the LocalLink input; if 1, the LocalLink source is not offering data
25	LLIN_DRDN	R	0	DST_RDY_N status for the LocalLink input; if 1, the RX FIFO is full
26	LLOUT_SRDN	R	0	SRC_RDY_N status for the LocalLink output; if 1, the TX FIFO is empty
27	LLOUT_DRDN	R	0	DST_RDY_N status for the LocalLink output; if 1, the LocalLink destination is not accepting data
31	TCP_CON	R	0	MAIN_OPEN_ACK from SiTCP; 1 if a TCP connection is active

Control Register 0 (CTRL0)



Bits	Name	R/W	Initial	Description
0–15	TCP_PORT	RW	C_DEFAULT_PORT	TCP port that SiTCP listens on
30	LLIN_EN	RW	0	Enable dataflow through the LocalLink input interface
31	LLOUT_EN	RW	0	Enable dataflow through the LocalLink output interface

Control Register 1 (CTRL1)



Bits	Name	R/W	Initial	Description
0–31	IP_ADDR	RW	C_DEFAULT_IP	IP address of the SiTCP core

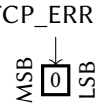
B.3.6 Interrupts

The interrupt controller was generated with the following parameters:

- **Use device ISC:** no
  (C_INCLUDE_DEV_ISC = false)
- **Use device ISC priority encoder service:** no
  (C_INCLUDE_DEV_PENCODER = false)
- **Number of interrupts generated by user logic:** 1
  (Length of C_IP_INTR_MODE_ARRAY: 1)
- **Capture mode:** rising-edge detect
  (Values of C_IP_INTR_MODE_ARRAY: 5)

The user logic generates an interrupt in case of an SiTCP TCP error signal. For the definition this signal, refer to the SiTCP manual [123].

User-logic interrupt vector (IP2Bus_IntrEvent)



Bits	Name	Description
0	TCP_ERR	TCP_OPEN_ERROR signal from the SiTCP core

B.3.7 Licensing and EEPROM Access

The SiTCP core provides ports for a connection to a 1024-bit Atmel AT93C46D EEPROM [125] with a serial three-wire (Microwire-compatible) interface. The wrapper emulates this EEPROM with a custom bridge interface and a block RAM unit that is also accessible over the PLB.

The EEPROM stores configuration variables like TCP time-out values, but also the MAC address and an encrypted license key. The mapping of the EEPROM's 256 bytes into SiTCP's internal address space is shown in the SiTCP manual, section *Internal register map*. In order for SiTCP to start up correctly, the 6 bytes of the MAC address must be written to the EEPROM address space at offset `0x12`. The remaining 16 bytes must be written to offset `0x40`, which is designated *access-forbidden area* in the SiTCP manual. The wrapper automatically initializes the block RAM with the correct values, if a hexadecimal representation of the SiTCP license file (MCS) is passed to the parameter C_MPC_HEX_STRING. This representation can, for example, be generated with the following command:

```
echo 0x$(xxd -p <LICENSE_FILE>.mpc)
```

B.3.8 1000BASE-X PCS/PMA

If the wrapper is configured to use the MGT interface, it automatically instantiates a *Xilinx LogiCORE IP Ethernet 1000BASE-X PCS/PMA or SGMII v11.5* [126]. This core was produced with the Xilinx core generator, using the following options:

- **Select standard:** 1000BASE-X
- **Physical interface:** Device-specific transceiver
- **MDIO management interface:** No
- **Auto negotiation:** Yes
- **Transceiver tile selection** Both transceivers

The wrapper sets the following configuration constants for the PCS/PMA core:

- **configuration_vector[4:0]:** `0b1_0000`
  (activates auto-negotiation)
- **link_timer_value[8:0]:** `0b1_0011_1101` = 317
  (sets the auto-negotiation link timer to about 10.39 ms)
- **an_adv_config_vector[15:0]:** `0b0000_0001_1010_0000`
  (sets the values advertised during auto-negotiation)

Refer to the PCS/PMA manual for the exact definition of each bit. The HDL code from the core generator was modified to allow the independent use of the transceivers from a GTX_DUAL, the optional inversion of the serial MGT pins, and access to the MGT's dynamic reconfiguration port.

B.4 Belle II Format Handler

B.4.1 Core properties

- **Name:** `belle2_format_handler_ll`
- **Current version:** v1.12.a
- **Type:** Standalone

B.4.2 Overview

The Belle II Format Handler core processes a LocalLink data stream that contains Belle II data in one of several configurable data formats. It parses the data, extracts metainformation like trigger and run number, and outputs the extracted information on a PTR_RET interface. In addition, it optionally reformats the data stream by packing all incoming frames belonging to the same event into a single outgoing frame and appending a index frame that contains the length information of all bundled sub-frames. This step is needed before writing pixel data to memory.

The core can be configured to parse Belle II ROI frames or PXD frames in one of several container formats. It is needed at multiple points in the ONSEN system to

- parse incoming ROI data and extract the event information;
- parse incoming PXD data, extract the event information, and bundle subsequent frames belonging to the same event—the last step is needed before writing the PXD data to memory, so that all information belonging to the same event can be read back from a single location for the ROI filtering step; and
- generate the ONSEN output format (see section C.4), which consists of the filtered PXD data to which the index frame is prepended as a header—the swapping of the data and index frames is performed during the writing of the processed data to memory (see the NPI writer documentation, section B.5).

For a detailed description of the data formats, refer to appendix C.

B.4.3 Ports and Buses

Figure B.5 shows the I/O ports and bus interfaces that the core provides. The LocalLink input and output are both synchronous to CLK. As the core does not provide a PLB interface, it has an additional reset input that is internally synchronized.

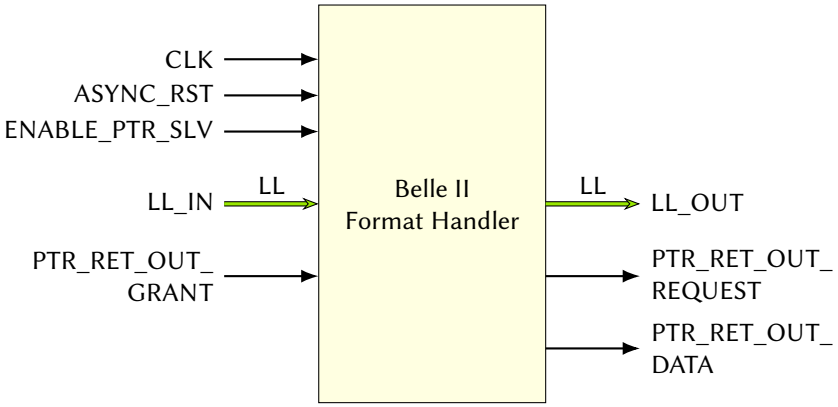


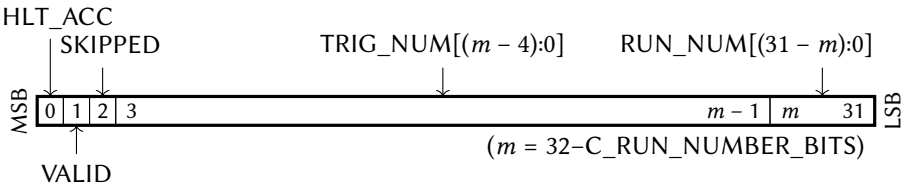
Figure B.5: I/O ports and buses of the Aurora LocalLink Wrapper

B.4.4 The ROI Parser

When C_DATA_TYPE is set for ROI data, the core forwards every frame arriving at the LocalLink input interface directly to the LocalLink output. Words that appear outside of a frame—this is an error condition—are dropped. Since only one ROI frame is allowed per event, the ROI parser does not bundle frames together. After a forwarded frame ends, the parser temporarily blocks the LocalLink input and inserts a index frame in the format shown in section C.3.2. If the index frame is not required, it can be disabled with the parameter C_ROI_NO_LENGTH_FRAME.

The parser extracts the trigger number, run number, and HLT trigger decision from the ROI frame. It checks the frame's header and CRC and outputs the obtained metainformation as a single 32-bit word over the PTR_RET interface.

ROI metadata word



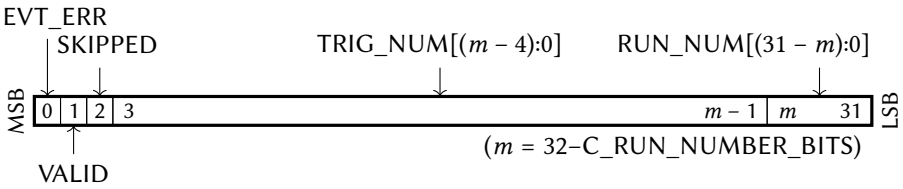
Bits	Name	Description
0	HLT_ACC	HLT decision for the event; 1 if the event is accepted
1	VALID	Data validity; 1 if both CRC and header of the frame are OK
2	SKIPPED	Indicates that a frame error occurred *before* this frame, and data was skipped. Does *not* indicate an error in the current frame.
3–(m – 1)	TRIG_NUM	The (m – 3) least significant trigger number bits extracted from the frame, where m = 32–C_RUN_NUMBER_BITS
m–31	RUN_NUM	The C_RUN_NUMBER_BITS least significant run number bits extracted from the frame

B.4.5 The PXD Parser

When C_DATA_TYPE is set for PXD data, the core forwards data arriving at the LocalLink input interface to the LocalLink output, fusing frames that belong to the same event into a single, long frame. The beginning of a new event is detected by a DHC start frame. Words that appear outside of a frame, as well as frames that appear outside of an event,—these are error conditions—are dropped After an event ends (indicated by a DHC end frame), the parser temporarily blocks the LocalLink input and inserts a index frame in the format shown in section C.3.2.

The parser extracts the trigger number and run number from the DHC start frame. It checks the CRC of all frames in the event and outputs the obtained metainformation for the whole event as a single 32-bit word over the PTR_RET interface.

PXD metadata word



Bits	Name	Description
0	EVT_ERR	Event error; 1 if an error occurred during the current event, including a LocalLink frame error, a new DHC start frame appearing before the DHC end frame for the current event, and too many frames in the event
1	VALID	Data validity; 1 only if *all* frames in the current event have the correct CRC
2	SKIPPED	Indicates that an error occurred *before* this event, and data was skipped. Does *not* indicate an error in the current event.
3–(m – 1)	TRIG_NUM	The (m – 3) least significant trigger number bits extracted from the frame, where m = 32–C_RUN_NUMBER_BITS
m–31	RUN_NUM	The C_RUN_NUMBER_BITS least significant run number bits extracted from the frame

B.5 NPI Writer

B.5.1 Core properties

- **Name:** npi_write_ll
- **Current version:** v1.11.a
- **Type:** PLB peripheral

B.5.2 Overview

The *NPI writer* writes data received over a LocalLink input to memory, using a low-level interface to a Xilinx multi-port memory controller. It also provides:

- a parameterizable asynchronous input FIFO, allowing the use of an arbitrary LocalLink clock; and
- a PLB slave interface, providing
  - status and control registers,
  - interrupt generation, and
  - software reset of the user logic.

B.5.3 Ports and Buses

Figure B.6 shows the I/O ports and bus interfaces that the core provides. Frames received over the LocalLink input are written to memory. Addresses of free buffers are received via a PTR_BUS interface; addresses of buffers that have been written to are passed on via a PTR_RET interface. The MPMC_PIM port connects to one port of an MPMC configured with a *native port interface* (NPI) [107, p. 185]. MPMC_Clk0 must be the same 200 MHz clock that is connected to the MPMC.

B.5.4 Operation

The NPI writer uses a finite-state machine for writing data to an memory FIFO and requesting write transfers from the MPMC. It requires a constant supply of pointers to free memory buffers. In the default configuration, the state machine cycles through a series of steps that can be simplified like this:

1. Initialize the first buffer with a guard word and a pointer to the second buffer; go to the second buffer
2. Initialize the buffer header with a guard word
3. Push data received over LocalLink to the buffer

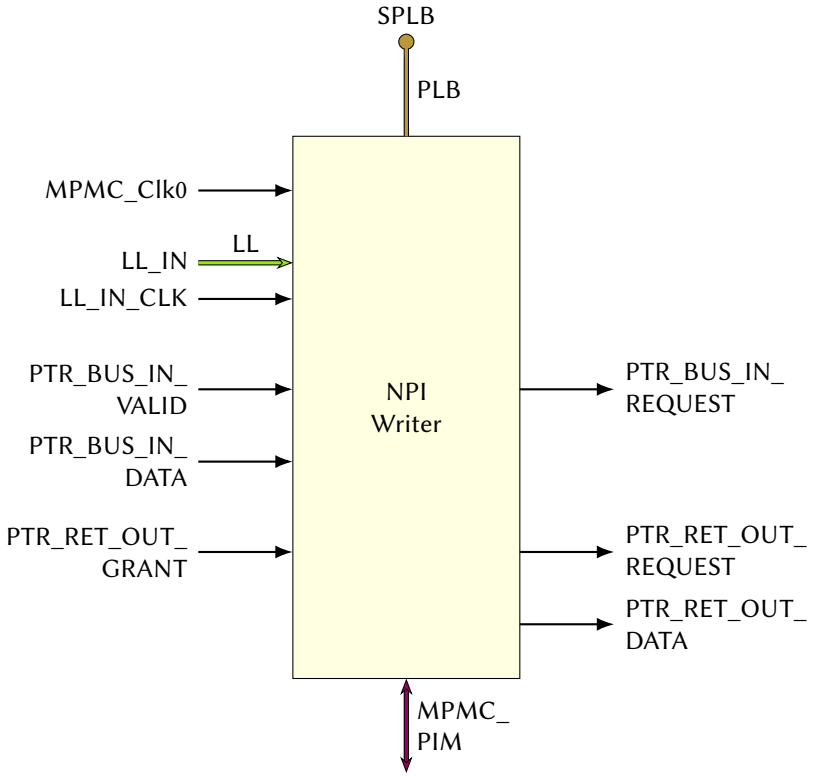


Figure B.6: I/O ports and buses of the NPI writer

4. If the buffer is full, write to its header the correct magic word, payload length, and a pointer to the next buffer; go to the next buffer and repeat from 2
5. When the input frame ends, write to the current buffer's header the correct magic word and payload length
6. Write the next frame input frame—this is required to be a index frame—to the payload of the first buffer
7. Write the correct magic word and length to the header of the first buffer and return its pointer on the PTR_RET interface, then repeat from 1

The relevant data formats are shown in section C.3. The buffer size is a configurable parameter of the writer. The distance of the supplied pointers must be at least as large as the buffer size, and the index frame cannot be larger than one buffer.

B.5.5 PLB Slave Interface

The address region starting at the PLB slave base address (C_BASEADDR) contains status and control registers, the software reset register, and the interrupt registers. The memory ranges of the individual regions, given as offsets from C_BASEADDR, are shown in the following table:

Name	Description	Range	
		Start	End
USER_SLV_BASEADDR	Slave registers	+0x000	+0x0FF
RST_BASEADDR	Reset register	+0x100	+0x1FF
INTR_BASEADDR	Interrupt control	+0x200	+0x2FF

The slave registers for this core are listed in section B.5.6. For an explanation of the soft-reset and interrupt-generation mechanisms, refer to sections B.1.3 and B.1.4, respectively. The definition of the interrupt signals generated by this core is given in section B.5.7.

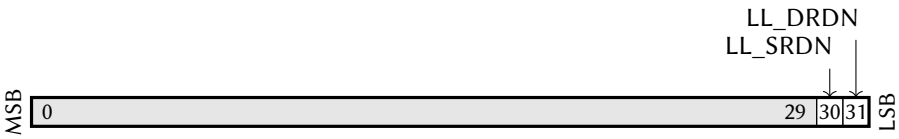
B.5.6 Status and Control Registers

The following table lists the available status and control registers, along with their respective offsets from the slave register base address:

Description	Name	USER_SLV_ BASEADDR Offset	R/W
Status Register	STA	+0x00	R
Control Register	CTRL	+0x04	RW

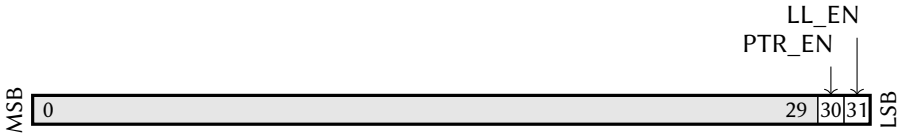
This section describes the function of each register and its bits.

Status Register (STA)



Bits	Name	R/W	Initial	Description
30	LL_SRDN	R	0	SRC_RDY_N status for the LocalLink input; if 1, the LocalLink source is not offering data
31	LL_DRDN	R	0	DST_RDY_N status for the LocalLink input; if 1, the input FIFO is full

Control Register (CTRL)



Bits	Name	R/W	Initial	Description
30	PTR_EN	R	0	Enable the pointer-bus interface
31	LL_EN	R	0	Enable dataflow through the LocalLink input interface

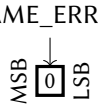
B.5.7 Interrupts

The interrupt controller was generated with the following parameters:

- **Use device ISC:** no
  (C_INCLUDE_DEV_ISC = false)
- **Use device ISC priority encoder service:** no
  (C_INCLUDE_DEV_PENCODER = false)
- **Number of interrupts generated by user logic:** 1
  (Length of C_IP_INTR_MODE_ARRAY: 1)
- **Capture mode:** rising-edge detect
  (Values of C_IP_INTR_MODE_ARRAY: 5)

The user logic generates an interrupt in case of a LocalLink frame error in the incoming data.

User-logic interrupt vector (IP2Bus_IntrEvent)



Bits	Name	Description
0	FRAME_ERR	Asserted when a LocalLink framing error is detected in the incoming data

B.6 NPI Reader

B.6.1 Core properties

- **Name:** npi_read_ll
- **Current version:** v1.11.a
- **Type:** PLB peripheral

B.6.2 Overview

The *NPI reader* reads data from memory using a low-level interface to a Xilinx multi-port memory controller and forwards it to a LocalLink output. It also provides:

- a parameterizable asynchronous output FIFO, allowing the use of an arbitrary LocalLink clock; and
- a PLB slave interface, providing
  - status and control registers,
  - interrupt generation, and
  - software reset of the user logic.

B.6.3 Ports and Buses

Figure B.7 shows the I/O ports and bus interfaces that the core provides. Addresses of memory buffers are received over a PTR_BUS interface. The buffer payload is read back from memory and passed to a LocalLink output. After all data have been read from a buffer, its address is passed on via a PTR_RET interface. The MPMC_PIM port connects to one port of an MPMC configured with a *native port interface* (NPI) [107, p. 185]. MPMC_Clk0 must be the same 200 MHz clock that is connected to the MPMC.

B.6.4 Operation

The NPI reader uses a combination of two finite-state machines for requesting reads from the MPMC and handling the output data. A new read process is initialized when the reader is idle and receives an address on its PTR_BUS interface.

The request state machine goes through the following steps:

1. Get the pointer to the first buffer of a linked list

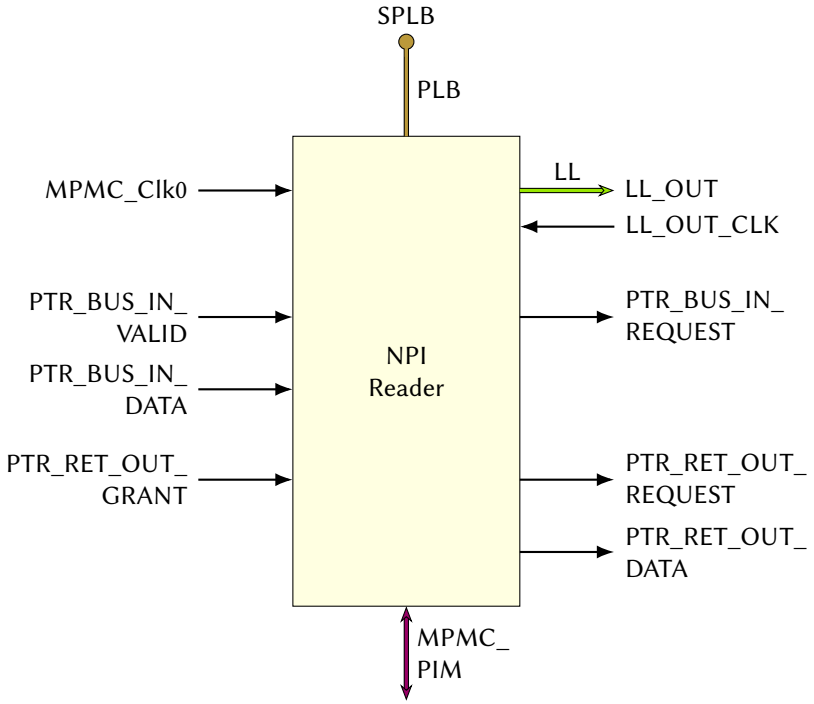


Figure B.7: I/O ports and buses of the NPI reader

2. Request a read-out of the buffer header
3. Wait for decoded header info from the output state machine
4. Request a read-out of buffer payload data, up to a maximum of 256 bytes
5. If there is unrequested data in the buffer, request the next read-out
6. If all data in the buffer have been requested, check whether it is the last buffer in a linked list; if no, go back to 2; if yes, go back to 1

Whenever the request state machine schedules a read operation from the MPMC, it appends information about the request to a short internal message queue, including the size of the request, the number bytes that should be sent to the output, and whether or not a buffer header was requested. The queue has only three entries, and requests are only sent to the MPMC when the queue is not full. Its output is processed by the output state machine that performs the following steps:

1. Get the info about a scheduled MPMC read from the message queue

2. Read words from the MPMC read FIFO and process them according to the request type: decode the header and send header information to the request state machine; or forward payload data to the LocalLink output
3. Repeat 2 until all requested words have been processed
4. If all words from a linked list of buffers have been read and processed, return the address of the first buffer on the PTR_RET interface
5. Go back to 1

The separation of the request and data-processing steps maximizes the reader's data throughput. The short message queue ensures that the MPMC read FIFO cannot overflow. This FIFO has a capacity of 1024 bytes. A read request has a maximum size of 256 bytes, so that three scheduled read operations cannot amount to more that 768 bytes. When the output state machine processes the output data from the first request, a slot in the message queue is freed, and a fourth request can bring the FIFO occupancy to its maximum. The next item is, however, not read from the queue until all words from the first request have been read from the FIFO, so that its occupancy can never be greater than 1024.

B.6.5 PLB Slave Interface

The address region starting at the PLB slave base address (C_BASEADDR) contains status and control registers, the software reset register, and the interrupt registers. The memory ranges of the individual regions, given as offsets from C_BASEADDR, are shown in the following table:

Name	Description	Range	
		Start	End
USER_SLV_BASEADDR	Slave registers	`+0x000`	`+0x0FF`
RST_BASEADDR	Reset register	`+0x100`	`+0x1FF`
INTR_BASEADDR	Interrupt control	`+0x200`	`+0x2FF`

The slave registers for this core are listed in section B.6.6. For an explanation of the soft-reset and interrupt-generation mechanisms, refer to sections B.1.3 and B.1.4, respectively. The definition of the interrupt signals generated by this core is given in section B.6.7.

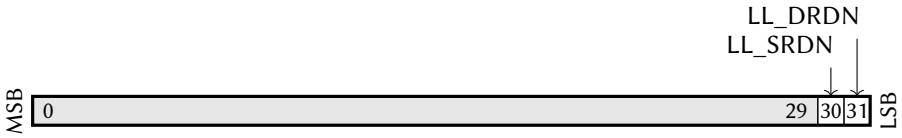
B.6.6 Status and Control Registers

The following table lists the available status and control registers, along with their respective offsets from the slave register base address:

Description	Name	USER_SLV_ BASEADDR Offset	R/W
Status Register	STA	+0x00	R
Control Register	CTRL	+0x04	RW
Error Pointer Register	EPTR	+0x08	R

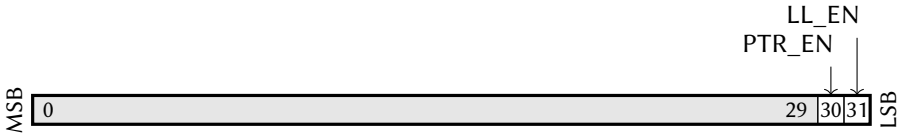
This section describes the function of each register and its bits.

Status Register (STA)



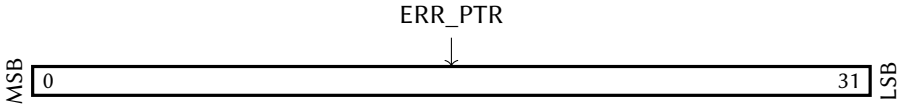
Bits	Name	R/W	Initial	Description
30	LL_SRDN	R	0	SRC_RDY_N status for the LocalLink output; if 1, the output FIFO is empty
31	LL_DRDN	R	0	DST_RDY_N status for the LocalLink output; if 1, the LocalLink destination is not accepting data

Control Register (CTRL)



Bits	Name	R/W	Initial	Description
30	PTR_EN	R	0	Enable the pointer-return interface
31	LL_EN	R	0	Enable dataflow through the LocalLink output interface

Error Pointer Register (EPTR)



Bits	Name	R/W	Initial	Description
0–31	ERR_PTR	R	0	Address of the buffer for which a header error was thrown

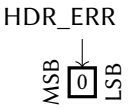
B.6.7 Interrupts

The interrupt controller was generated with the following parameters:

- **Use device ISC:** no
  (C_INCLUDE_DEV_ISC = false)
- **Use device ISC priority encoder service:** no
  (C_INCLUDE_DEV_PENCODER = false)
- **Number of interrupts generated by user logic:** 1
  (Length of C_IP_INTR_MODE_ARRAY: 1)
- **Capture mode:** rising-edge detect
  (Values of C_IP_INTR_MODE_ARRAY: 5)

The user logic generates an interrupt in case of an error in the buffer header read from memory. The first word in a buffer should be the magic word `0x600DB10C` or the guard word `0xBAADB10C`, in which case a reread takes place. If any other word is encountered, an interrupt is generated and the buffer address is written to the EPTR register.

User-logic interrupt vector (IP2Bus_IntrEvent)



Bits	Name	Description
0	HDR_ERR	Asserted when a header error is detected in the read data

B.7 Other ONSEN IP Cores

IP cores that were developed for the ONSEN system outside of the context of this thesis are documented elsewhere: The pixel-filter core (roi_frame_handler) is discussed in another PhD thesis that has arisen from the ONSEN project [103]. Information about the remaining cores (mainly those pertaining to the buffer management) are, at the moment, only available in internal documents [127, 128].

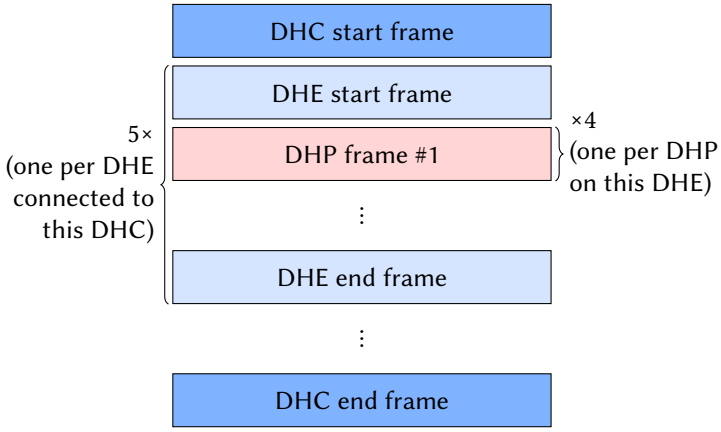
Appendix C

Data Formats

This chapter lists the various data formats that are relevant for the ONSEN system, including inbound and outbound data streams as well as internally used formats. In cases where formats are defined elsewhere, only a short overview is given and the relevant documents are referenced.

C.1 Pixel Data from DHH

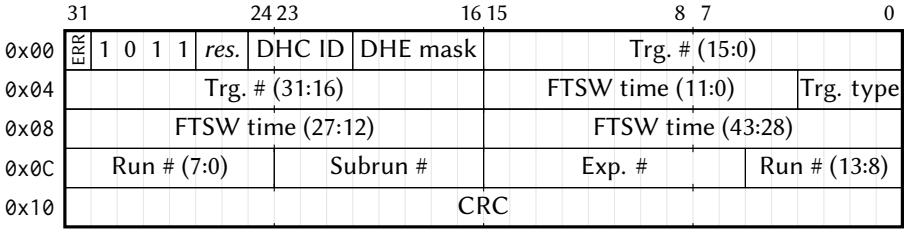
The pixel data stream sent by a DHC module to an ONSEN Selector module is divided into several Aurora frames for each event, containing event metainformation and pixel hits:



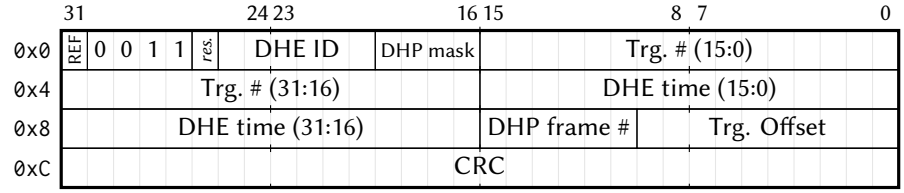
Each frame encompasses multiple 32-bit words. The first word has a 4-bit signature that uniquely identifies the type of frame. The last word is a CRC checksum. The CRC format is explained in section C.5 below. Only information

relevant for the processing by the ONSEN system is given here. For a more comprehensive description, refer to the data-format documentation from the DHH group [129].

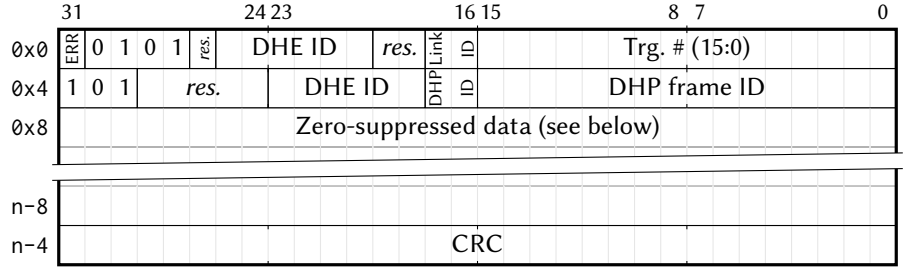
C.1.1 DHC start frame



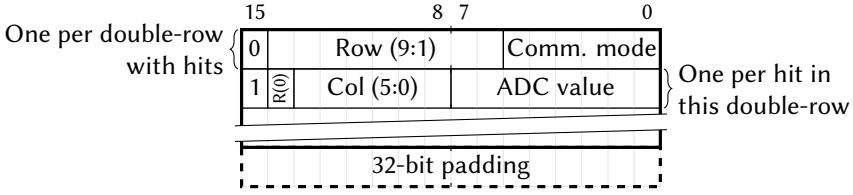
C.1.2 DHE start frame



C.1.3 DHP frame (zero-suppressed)

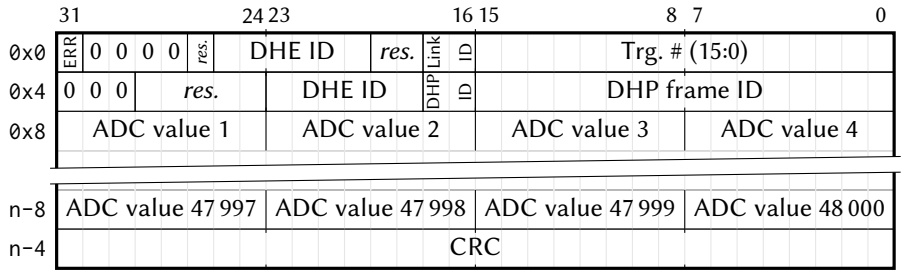


This frame contains the actual hit information for the event. The pixel-filter core processes the hits, discarding unselected ones, and adjusts the header and CRC of this frame. Hit data is encoded in the zero-suppressed format:



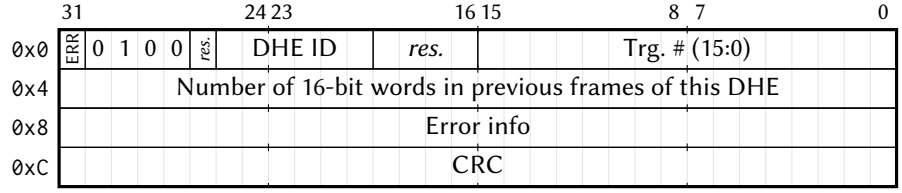
In this format, only entries for pixels that have above-threshold values after the subtraction of common mode and pedestals are listed. The hits are ordered with ascending row numbers, with rows grouped in pairs. For each double-row that contains at least one hit, a 16-bit row header appears, giving the 9 most significant row-number bits and the common-mode value that has been subtracted from the ADC values in these rows. Then, a 16-bit word follows for each hit in the double-row, giving the least significant row-number bit, the column number, and the ADC value for the hit. If the total number of row-header and hit-info words is odd, the last row-header word is repeated at the end to make the frame 32-bit aligned.

C.1.4 DHP frame (raw-data/full-frame)

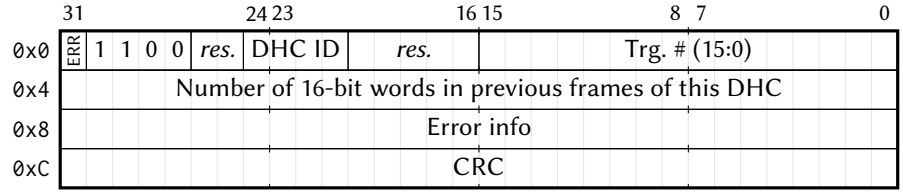


If a special trigger type is sent from the FTSW to the DHC, the DHEs switch all DHPs to raw-data read-out mode. For these events, the DHC sends raw-data frames instead of zero-suppressed ones, containing the unprocessed ADC values for all pixels. These frames are simply passed through by the ONSEN system to the EB2.

C.1.5 DHE end frame



C.1.6 DHC end frame



C.1.7 Data rate estimation

In normal operation, every input of an ONSEN Selector node will receive a sequence of 32 frames for every fourth level-1 trigger, i.e., with 7.5 kHz: A DHC start frame, a DHC end frame, 5 DHE start frames, 5 DHE end frames, and 20 DHP frames. The DHC and DHE frames for every trigger have a total size of 196 bytes, contributing to the total data rate with less than 1.5 MB/s.

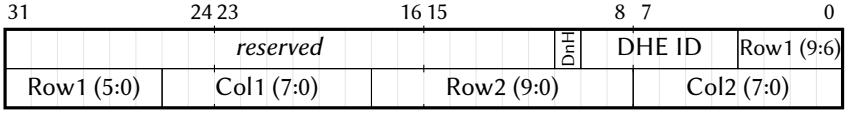
Each of the 20 zero-suppressed DHP frames adds another 12 bytes from header and checksum. The largest contribution comes from the actual hit information. With the format described in section C.1.3, two bytes are required for every hit, and two additional bytes for every double-row with at least one hit. Every DHP reads out 768 pixel rows and, one average, 62.5 columns. With the usual assumption of 3 % for the occupancy, we expect 1440 hits. In the worst case, all of the 384 double rows can therefore contain pixels with hits, yielding a data size of 3648 bytes, or approximately 2.5 bytes per hit. The total contribution to the data rate from the DHP frames then is $20 \times 3660\,\mathrm{B} \times 7.5\,\mathrm{kHz} = 549\,\mathrm{MB/s}$.

An additional major contribution comes from the raw-data read-out, required for PXD pedestal calculations, that will likely occur with 50 Hz (shortly before every injection into SuperKEKB). For these events, the zero-suppressed DHP frames are replaced by raw-data frames, each one 48 012 bytes in size (see section C.1.4 above). If we assume that every Selector receives the raw data for every fourth injection, it sees a rate of 12 MB/s. The total pixel-data input at each Selector is therefore below 600 MB/s.

C.2 ROI Data

C.2.1 ROI format

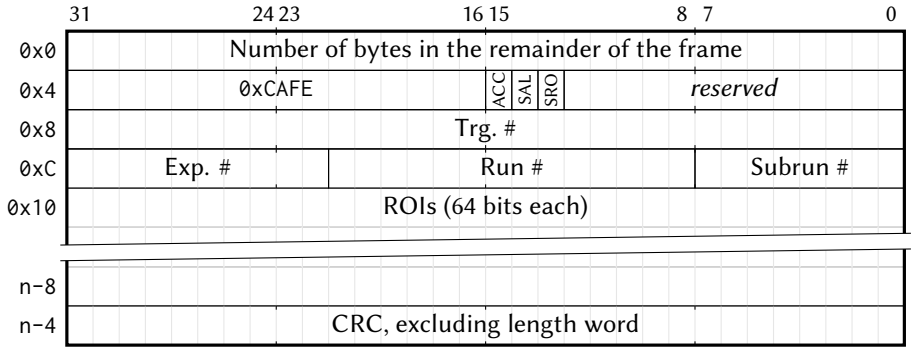
The ROIs are rectangular, defined by sensor-wide pixel coordinates using two corners: (Col1, Row1) and (Col2, Row2). The format requires that Col2 > Col1 and Row2 > Row1. It uses 64 bits for every ROI:



The 1-bit flag, here called DnH (DATCON/not HLT), specifies the source of each ROI.

C.2.2 HLT ROI packet

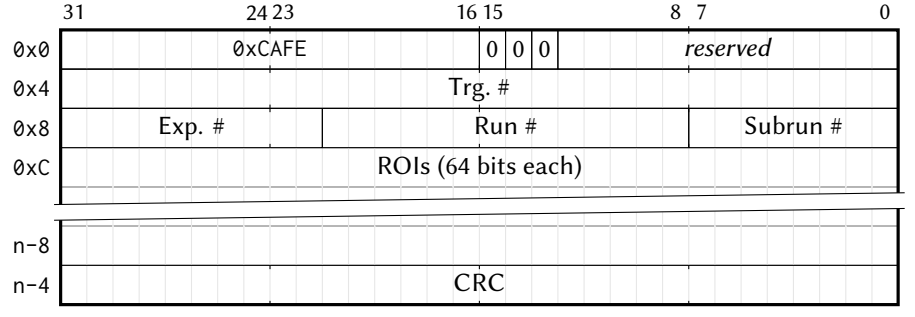
The ROI packet sent via TCP from the HLT must be prepended with a frame length, allowing the SiTCP wrapper to convert it into a LocalLink frame:



Three flags can be set by the HLT: ACC (Accept) specifies whether the event was accepted by the high-level trigger and should be sent to the EB2; SAL (send all) instructs the ONSEN Selectors to ignore ROIs and pass through the unprocessed pixel data; and SRO (send ROIs) instructs the ONSEN Selectors to append an ROI frame to the output of the event (see section C.4.3).

C.2.3 DATCON ROI packet

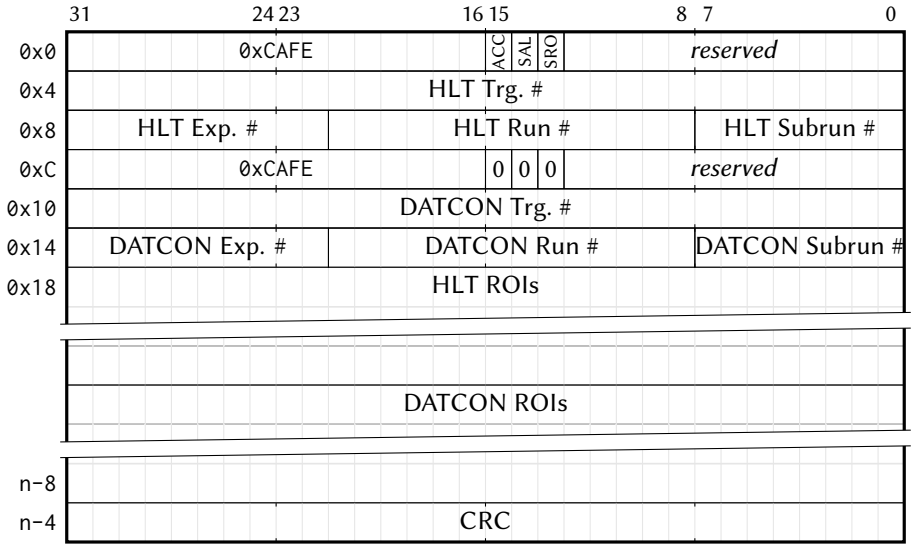
DATCON packets are received with the Aurora protocol, which provides a framing mechanism. The prepended length is therefore not required.



The DATCON is not allowed to set the HLT control flags. The corresponding bits should be set to 0.

C.2.4 Merged ROI packet

The Merger simply concatenates the headers and ROIs from the two sources and calculates a new checksum:



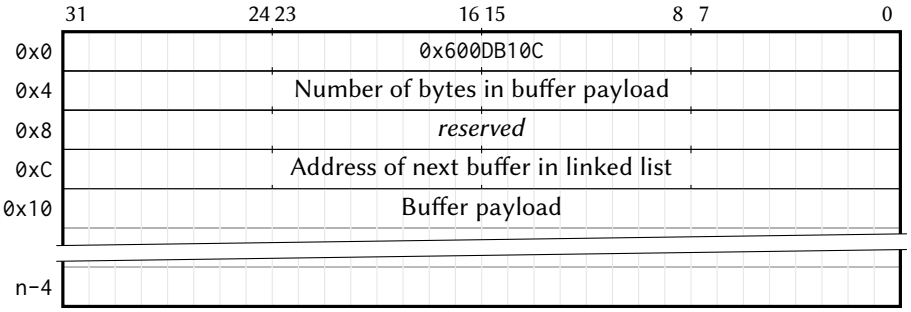
C.2.5 Data rate estimation

The bottleneck for the ROI reception is the GbE input used for HLT ROIs. The DATCON connection and internal distribution of merged packets use the much faster Aurora links. We therefore only consider the HLT packet size.

In any sane event, the number of reconstructed tracks should be well below 50. We therefore use this as an upper limit, and assume that each track induces two ROIs (one on a PXD sensor of each layer), so that the total number is 100. The HLT sends ROI packets to the Merger with 30 kHz. Header and checksum add up to 20 bytes, or 600 kB/s. Selected events arrive with 10 kHz and contain the additional ROI information: $100 \times 8\,\mathrm{B} \times 10\,\mathrm{kHz} = 8\,\mathrm{MB/s}$. The total rate is therefore below 10 MB/s.

C.3 ONSEN Memory Management

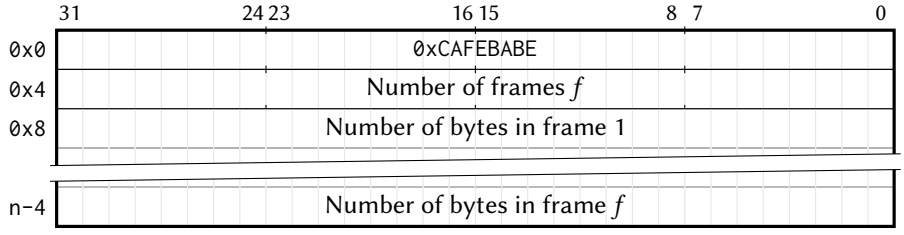
C.3.1 Buffer format



Valid memory buffers begin with the "magic word" `0x600DB10C`. The first operation an NPI writer performs on a buffer is to write the guard word `0xBAADB10C` to its start. The last operation before the pointer is passed on is to write the correct header. This can be used by the reader to prevent a possible run condition, caused by the MPMC arbitration.

Multiple buffers are concatenated by writing the address of the next buffer to the current buffer header. A null-pointer indicates that the current buffer is the last one.

C.3.2 Index frame



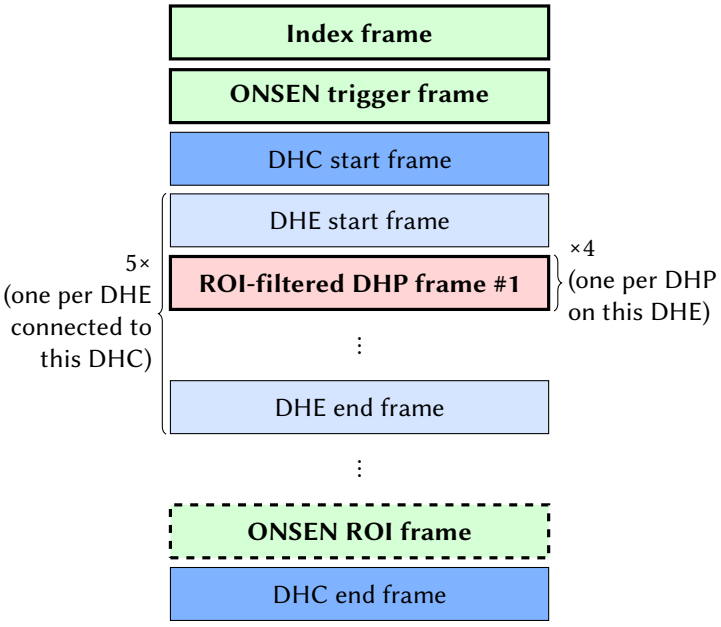
If an index frame is used, it is the *only* data in the first buffer of a linked list, and at least one more buffer is required. The individual sub-frames indicated in the index frame are concatenated and written to the payload of the following buffers. The start of all sub-frames must be 32-bit aligned. If any frame (except the last) has a length that is not a multiple of 4 bytes, the gap up to the next frame is filled with void data. With the formats currently used for the ONSEN system, this should never be necessary.

The buffer header requires 16 bytes, and the index frame has a size of $8 + 4 \times n_{\text{frames}}$ bytes. With a buffer size of 1024 bytes, the maximum number of subframes is therefore 250.

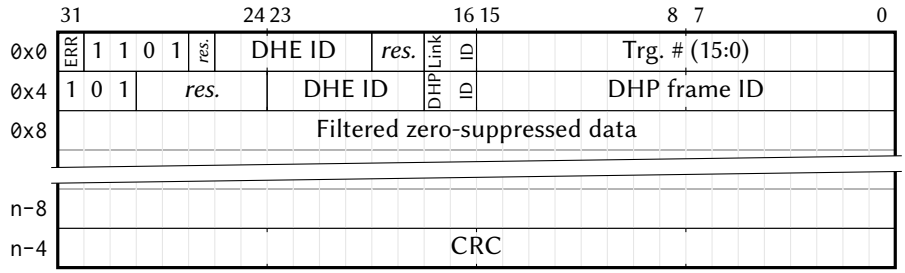
C.4 ONSEN Output Data

The TCP output data-stream from the ONSEN system consists of multiple frames that must be disentangled again on the EB2. To that end, the previously mentioned index frame is prepended to the data. In order to provide a reliable source for the event metainformation, a special trigger frame, containing the headers of the HLT and DATCON ROI packets, is sent next. (The DHC start frame is not considered reliable, as it can be replaced by a dummy frame if the DHC should not provide data for an event.)

The following diagram indicates frames that are processed or generated by the ONSEN system with a bold font. All other frames are simply passed through from the DHC:

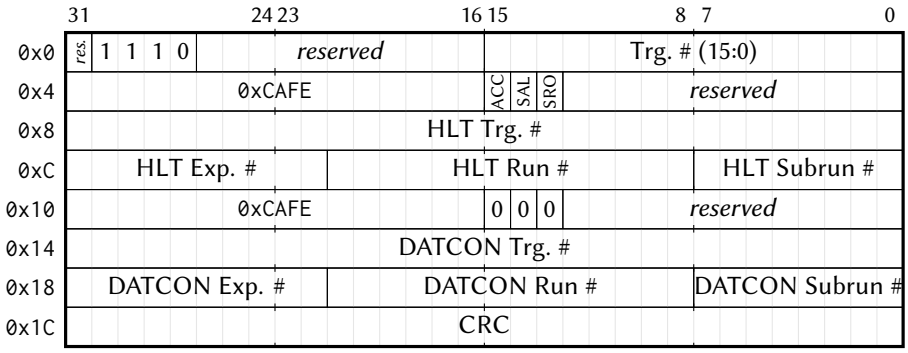


C.4.1 ROI-filtered DHP frame (zero-suppressed)



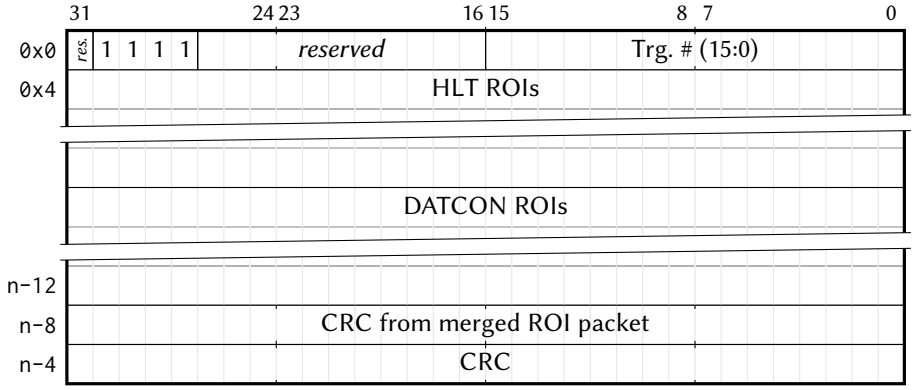
C.4.2 Trigger frame

The trigger frame contains the event metainformation from the HLT at a constant offset. This information is necessarily present for every event, as the the output from ONSEN to EB2 is triggered by the arrival of the HLT ROI packet.



C.4.3 ROI frame

The ONSEN ROI frame is included in the output data of events for which the HLT has set the SRO flag. It provides the only possibility to forward DATCON ROIs, which are otherwise lost, to the EB2.



C.4.4 Data rate estimation

We infer the reduced data rate from the input data rate calculated in section C.1.7. Applying the reduction factor of 30 (3 from the rejection of complete events by the HLT and 10 form the ROI-based data reduction in the remaining events) to the data rate of about 550 MB/s, we arrive at a 18.3 MB/s. The rate from the raw-data read-out (12 MB/s) is not reduced, so that the total required bandwidth is around 30 MB/s.

C.5 Checksum Format

The CRC checksum in all frames is a CRC-32 code, using the polynomial `0x04C11DB7`, an initial value of `0x00000000` and no reflection or inversion of either input or output. On the PC-side, such a checksum can be calculated with the functions used in the following C++-code snippet:

```
#include <boost/crc.hpp>
...
boost::crc_optimal<32,0x04C11DB7,0,0,false,false> crc;
crc.process_byte(0x42);
```

It uses the CRC methods from the Boost C++ Libraries, documented at: `http://www.boost.org/doc/libs/1_58_0/libs/crc/crc.html`

On the ONSEN system, Virtex-5 CRC32 primitives are used for the checksum calculation (see the Virtex-5 GTX manual [120]). By default, they generate an Ethernet checksum. To obtain "PXD checksums", reflections and inversion must be reverted.

Appendix D

Hardware Details

This chapter lists details about the Compute Node boards that did not fit in the main text, including information about equipped components, MGT connections, and FPGA pin-outs in the form of UCF files.

D.1 xFP v4.0

D.1.1 Components

The **flash memory** is implemented with two Numonyx StrataFlash Embedded Memory (P33) modules [130] (PC28F256P33B85). Each module has a 16-bit data bus and 256 Mbit capacity. The two chips are used with common address and control inputs and separate data lines, appearing to the FPGA as a single 64 MiB block with a 32-bit data bus.

The **Ethernet PHY** is a Marvell 88E1111 [131] (88E1111-BAB1).

For our use of the **DDR2 memory**, any 200-pin SO-DIMM module from DDR2-400 upwards with a CAS latency no higher than CL5 should work. We have successfully used the following 2 GiB dual-rank devices:

- SK Hynix HYMP125S64CP8-S6
- Corsair VS2GSDS800D2
- Kingston ValueRAM KVR800D2S6/2G

As **SFP transceivers**, plugged into the xFP's SPF+ cages, we have successfully used the Finisar FCLF-8521-3 [132] as 1000BASE-X-to-1000BASE-T (optical-to-RJ45) transceiver and the Finisar FTLF8528P3BCV [133] as SFP+ multi-gigabit optical transceiver at up to 6.25 Gbps.

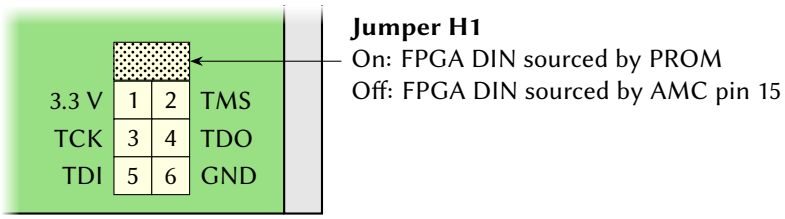


Figure D.1: JTAG header pin-out and jumper on the xFP v4.0

D.1.2 AMC connector

AMC ports 2, 3, 4, 5, 7, 8, 9, 11, 13, 14, 15, and 17 are connected to LVDS-capable I/O pins of the FPGA. The fabric clock and telecom clocks are also connected to differential I/O pairs. For the exact pin-to-port mapping, refer to the UCF file in section D.3.

AMC ports 6, 10, 12, 18, 19, and 20 are connected to GTX-transceiver pins of the FPGA. For the transceiver-to-port mapping, refer to section D.1.4.

The AMC JTAG pins are connected to the JTAG chain on the board. Refer to section D.1.3 for details.

The pins assigned to AMC ports 0 and 1 are not connected to differential transceivers. Instead, they are used for UART and programming. The following pins are affected:

AMC connector pin	FPGA pin
11	INIT
12	DONE
14	CCLK
15	DIN
20	DOUT
21	UART TX (xFP FPGA ⟶ AMC connector)
23	PROG
24	UART RX (xFP FPGA ⟵ AMC connector)

This custom assignment can lead to problems when the card is used in a MicroTCA shelf instead of the CNCB. In that case, the resistors R6 through R13 should be removed in order to decouple the signals from the AMC connector.

D.1.3 JTAG and programming

A bitstream can be downloaded to the FPGA on the xFP in four different ways:

1. with a JTAG programmer connected to the JTAG header on the board; the pin-out of the header is shown in figure D.1;
2. with a JTAG programmer via the AMC JTAG pins, from a carrier board or MicroTCA shelf;
3. from the Xilinx PROM on the xFP; this mode is explained below; and
4. via the serial daisy chain, using the programming pins on the AMC connector; this mode is explained in section D.2.5.

The JTAG chain on the xFP encompasses the FPGA and the 32 Mbit Xilinx Platform Flash configuration PROM (XCF32PVOG48C) [134]. The JTAG signals are simultaneously connected to the header on the xFP and to the AMC connector. When the card is plugged into a MicroTCA shelf with JTAG functionality, using the header pins can lead to an erroneous JTAG connection due to the dangling ends on the AMC connector side.

Of modes 3 and 4, only one can be active at the same time. PROM programming works by writing an FPGA bitstream, converted to the MCS format [135, p. 21], to the PROM via JTAG. When the FPGA is placed in master-serial configuration mode, it automatically loads the bitstream from the PROM upon power-up. For this mode to be active, resistor R20 must be placed on the xFP, and resistor R19 must be removed. In addition, jumper H1 must be on (see figure D.1), so that the FPGA's DIN pin is connected to the PROM's data output.

When the FPGA is placed in slave-serial configuration mode (R19 on and R20 off) and DIN is sourced by the AMC connector (H1 off), the daisy-chain programming mode of the CNCB can be used.

D.1.4 Multi-gigabit transceivers

The Virtex-5 FX70T has one *column* with 8 GTX_DUAL primitives. Each GTX_DUAL consists of two GTX transceivers with shared clock resources. Figure D.2 shows how the transceivers connect to AMC ports and SFP cages. Each GTX_DUAL has a clock input, and two of them are connected to oscillator outputs on the xFP. When using GTX transceivers, clock routing limitations must be observed [120, p. 100]: An MGTREFCLK can only be used for transceivers no more than three positions above or below the clock input, and all transceivers between an MGTREFCLK and a GTX_DUAL using this clock must be instantiated.

D.1.5 Sensors

The xFP v4.0 provides the following sensors on the sensor I^2C bus accessible from the MMC:

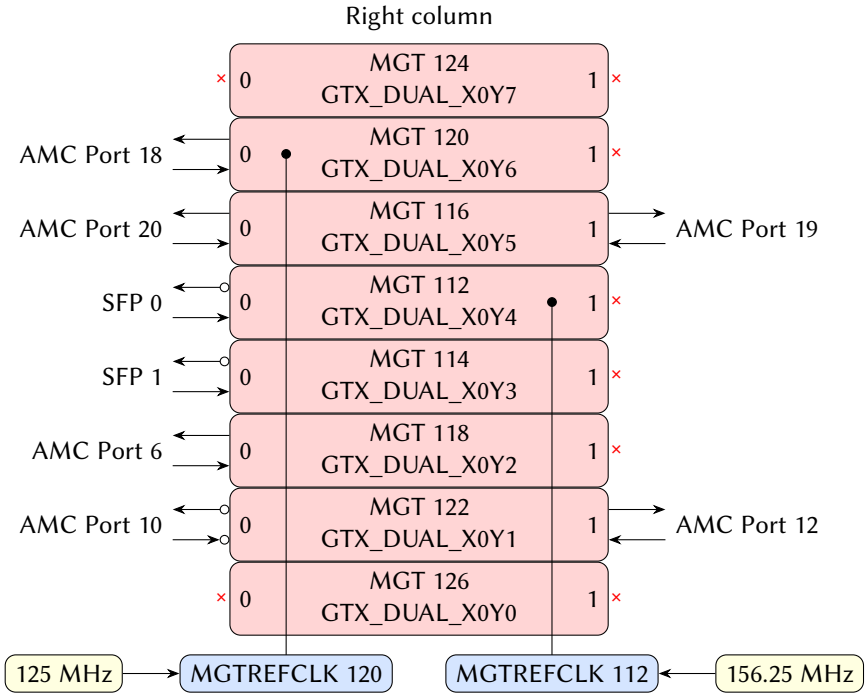


Figure D.2: External connections of the MGTs of the xFP v4.0, showing transceiver designators, tile locations, and AMC-port or SFP names. The two transceivers in each GTX_DUAL are shown on the two sides of the figure. Note that some MGTs have inverted signals for their receiver inputs (marked with →○) or their transmitter outputs (marked with ○→).

- A Maxim MAX1239EEE+ 12-channel, 12-bit ADC [136] with slave address 0110101. It measures the following voltages:
  - AIN0: VTT0.9V
  - AIN1: VCC1.8V
  - AIN2: 1.0VINT
  - AIN3: VCC1.2V
  - AIN4: VCC2.5 ×1/(10 + 1)
  - AIN5: VCC3.3V ×1/(10 + 1)
  - AIN6: VCC5V ×1/(10 + 1)
  - AIN7: 1.0VMGT
  - AIN8: 1.2VPLL
  - AIN9: 1.2VTT
  - AIN10: VCC12V ×1/(10 + 1)
  - AIN11: n/c

Voltages higher that 2 V are scaled down by voltage dividers to adapt them to the ADC's dynamic range.

- A Maxim MAX1617AMEE+ temperature sensor [137] with slave address 0011000. It measures its own temperature (located on the opposite side about 1 cm from the FPGA) and the FPGA-internal temperature-sensing diode pins.
- A Maxim MAX6626PMUT temperature sensor [138] with a pin misconfiguration. This sensor is not placed on new boards.

	a	b	c	d	e	f	g	h
1	AMC 1 Port 15				AMC 3 Port 15			
2	AMC 1 Port 17				AMC 3 Port 17			
3	AMC 1 Port 18				AMC 3 Port 18			
4	AMC 1 Port 19				AMC 3 Port 19			
5	AMC 1 Port 20				AMC 3 Port 20			
6	AMC 2 Port 15				AMC 4 Port 15			
7	AMC 2 Port 17				AMC 4 Port 17			
8	AMC 2 Port 18				AMC 4 Port 18			
9	AMC 2 Port 19				AMC 4 Port 19			
10	AMC 2 Port 20				AMC 4 Port 20			

J31

a	b	c	d	e	f	g	h
Switch LVDS 15				Switch LVDS 16			
Switch LVDS 13				Switch LVDS 14			
Switch LVDS 11				Switch LVDS 12			
Switch LVDS 9				Switch LVDS 10			
Switch LVDS 7				Switch LVDS 8			
Switch LVDS 5				Switch LVDS 6			
Switch LVDS 3				Switch LVDS 4			
Switch LVDS 1				Switch LVDS 2			
M2+	M2-	TMS	TDO	M3+	M3-	TCK	TDI
M0+	M0-	D_N	VBUS	M1+	M1-	D_P	AGND

J32

Figure D.3: Pin-out for the RTM connectors J31 and J32 on the CNCB v3.3. The Mn± pins correspond to the MDI[n]± physical-layer interface of the RTM PHY.

D.2 CNCB v3.3

D.2.1 Components

For the **Flash memory**, both **Ethernet PHYs**, and the single **DDR2 memory** memory module, the CNCB uses the same components as the xFP v4.0 (see section D.1.1).

D.2.2 ATCA backplane connector

On the ATCA backplane connector, port 0 of fabric channels 1 through 15 as well as the update channel connect to a multi-gigabit transceiver of the FPGA. The exact mapping is described in section D.2.6. In addition, the physical-layer side of one of the two Ethernet PHYs is connected to base channel 1.

In order to determine how the fabric channels of two boards in an ATCA shelf are connected, two documents must be consulted: the routing-assignment table for the backplane type (e.g., full-mesh) from the ATCA specification [87]; and the mapping between logical and physical slots from the shelf manual (e.g., our Schroff 14-slot shelf [90]). In any ATCA shelf, base channels 1 of all non-hub slots are connected in a star topology to logical slot 1 (the first hub slot, usually in the center of the shelf).

D.2.3 RTM connector

The RTM connector has connections to AMC ports 15, 17, 18, 19, and 20 of each AMC bay, as well as 16 bi-directional LVDS links to the Switch FPGA

Table D.1: Mapping of AMC ports between the four AMC bays on the CNCB v3.3.

Source port	Target bay-port			
	Source bay 1	Source bay 2	Source bay 3	Source bay 4
6	4-12	1-12	2-12	3-12
7	4-13	1-13	2-13	3-13
8	4-14	1-14	2-14	3-14
9	3-11	4-11	1-11	2-11
10	3-10	4-10	1-10	2-10
11	3-9	4-9	1-9	2-9
12	2-6	3-6	4-6	1-6
13	2-7	3-7	4-7	1-7
14	2-8	3-8	4-8	1-8

and signals for physical Ethernet, USB, and JTAG connectors. The mapping is shown in figure D.3.

D.2.4 AMC bay interconnection

AMC ports 6 through 14 of the four AMC bays provide interconnections between the cards. Of these, ports 6, 10, and 12 are connected to MGTs on the xFP. Table D.1 shows the exact mapping.

D.2.5 JTAG and programming

A bitstream can be downloaded to the Switch FPGA and the FPGAs on plugged-in xFPs in two different ways:

1. with a JTAG programmer connected to the JTAG header on the CNCB or, if used, the RTM; the JTAG chains of all xFPs are concatenated into a single JTAG chain, from which missing boards are automatically decoupled; and
2. automatically upon power-up via the slave-serial daisy chain of the Switch FPGA and all xFPs, from which missing boards are automatically decoupled.

The second mode requires all present xFPs to be set up in slave-serial mode, as described in section D.1.3. This configuration mode is described in the Virtex-5 configuration user guide [135, p. 40]. All FPGAs then are configuration slaves, and the CPLD on the CNCB is the configuration master: It provides the

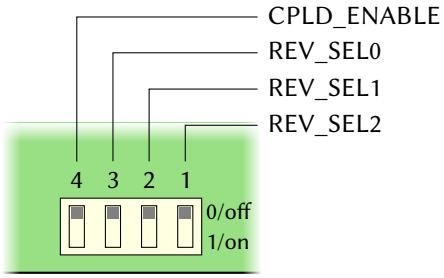


Figure D.4: DIP switch controlling the automatic programming by the CPLD on the CNCB. Note that the signal names are swapped on the PCB's assembly print.

configuration data and controls the configuration clock and PROGRAM signals of the FPGAs.

The configuration file for the slave-serial mode can be generated from the individual bitstreams for the Switch FPGA and xFP-FPGAs with the following command line:

```
promgen -b -p bin -u 0 <sw_fpga>.bit \
  <xfp_1>.bit <xfp_2>.bit <xfp_3>.bit <xfp_4>.bit -o <combined>.bin
```

At least as many bitstreams must be combined as FPGAs are present in the daisy chain. If fewer FPGAs are present, the configuration succeeds and unused configuration data at the end is discarded.

The combined bitstream must be written to the flash memory, which is connected to the FPGA and CPLD on the CNCB. Currently, the only way to do this is by accessing the flash with software running on the PowerPC, for which an EDK project with a PLB interface for the flash memory (xps_mch_emc) is required. The flash can then be accessed as a block device by a Linux system with the correct drivers. Alternatively, the Program Flash option from Xilinx SDK can be used to automatically download and execute a standalone flash-programmer binary on the PowerPC. A possible add-on the the CPLD's functionality, allowing bitstream writing over IPMI, is under development.

The CPLD initiates the automatic configuration process upon power-up or upon a command sent by the IPMC, but only if the CPLD_ENABLE bit of the CNCB's DIP switch is active (see figure D.4). The CPLD reads the combined bitstream from a specific offset in the 64 MiB flash memory. Currently, this offset is fixed to 48 MiB. The system design foresees that the offset is configurable with the DIP switch's REV_SEL bits, so that a backup bitstream can be loaded in case the default one is overwritten by a non-working version. As the combined bitstream has a size of almost 16 MiB, a reasonable offset for the

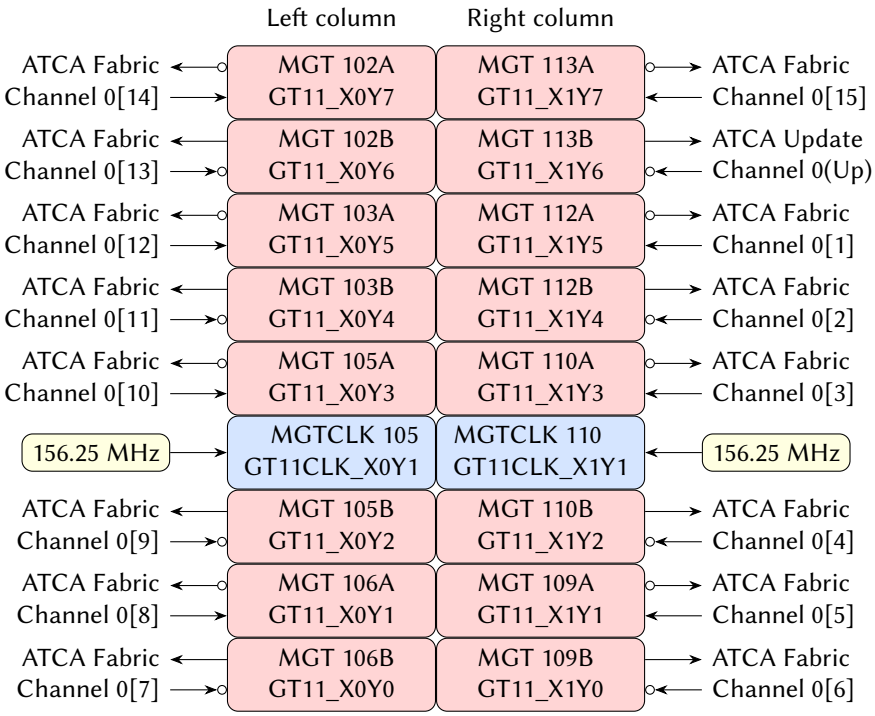


Figure D.5: External connections of the MGTs of the CNCB v3.3, showing transceiver designators, tile locations, and ATCA-channel names. Note that some MGTs have inverted signals for their receiver inputs (marked with →○) or their transmitter outputs (marked with ○→).

backup bitstream would be 32 MiB, with the flash's lower 32 MiB reserved for Linux kernels and storage.

D.2.6 Multi-gigabit transceivers

The Virtex-4 FX60 has two *columns* with 8 GT11 primitives in each column. Two neighboring transceivers (designated *A* and *B*) use shared clock resources. Figure D.5 shows how the transceivers connect to ATCA backplane ports. One clock input of each column is used. In contrast to the Virtex-5 GTX transceivers, there are no limitations on clock-routing distance or required instantiations.

D.2.7 Sensors

The CNCB v3.3 provides the following sensors on the sensor I^2C bus accessible from the IPMC:

- A Maxim MAX1239EEE+ 12-channel, 12-bit ADC [136] with slave address 0110101. It measures the following voltages:
  - AIN0: VCC1V2
  - AIN1: VCC1V8
  - AIN2: VCC2V5 ×3.3/(3.3 + 10)
  - AIN3: VCC3V3 ×3.3/(3.3 + 10)
  - AIN4: VCC5V ×3.3/(3.3 + 10)
  - AIN5: V_{shunt} × 1/(1 + 10)
  - AIN6: VCC12V_TOP ×1/(1 + 10)
  - AIN7: VCC12V_AMC1 ×1/(1 + 10)
  - AIN8: VCC12V_AMC2 ×1/(1 + 10)
  - AIN9: VCC12V_AMC3 ×1/(1 + 10)
  - AIN10: VCC12V_AMC4 ×1/(1 + 10)
  - AIN11: VCC12V_RTM ×1/(1 + 10)

  Voltages higher that 2 V are scaled down by voltage dividers to adapt them to the ADC's dynamic range.
- A Maxim MAX1617AMEE+ temperature sensor [137] with slave address 0011000. It measures its own temperature (located on the PCB near the FPGA) and the FPGA-internal temperature-sensing diode pins.

The voltage V_{shunt} measured by the ADC reflects the total power consumed by the CNCB and all plugged-in boards. The voltages used on the CNCB are derived from the 48 V input of the Zone 1 connector. The 48 V are converted to 12 V, from which all other voltages are generated. V_{shunt} is the voltage dropped over a 5 mΩ shunt resistor that is put in the 12 V path before the loads, magnified by a factor of 20 by an Analog Devices AD8210YRZ current monitor [139]. The AD8210 is operated in unidirectional mode with ground-referenced input. The power consumed by the board can be calculated as $P = 12\,\text{V} \times 10 V_{\text{shunt}} / 1\,\Omega$.

D.3 UCF Files

Listing D.1: Pin-out for xFP v4.0 FPGA (XC5VFX70T-2FFG1136C)

```
NET "CLK_100" LOC=AF18 | IOSTANDARD=LVCMOS25;
NET "FCLKA_N" LOC=AH19 | IOSTANDARD=LVPECL_25;
NET "FCLKA_P" LOC=AH20 | IOSTANDARD=LVPECL_25;
NET "TCLKA_N" LOC=AH22 | IOSTANDARD=LVPECL_25;
NET "TCLKA_P" LOC=AG22 | IOSTANDARD=LVPECL_25;
NET "TCLKB_N" LOC=AF19 | IOSTANDARD=LVPECL_25;
NET "TCLKB_P" LOC=AG18 | IOSTANDARD=LVPECL_25;
NET "TCLKC_N" LOC=H13  | IOSTANDARD=LVPECL_25;
NET "TCLKC_P" LOC=J14  | IOSTANDARD=LVPECL_25;
NET "TCLKD_N" LOC=J21  | IOSTANDARD=LVPECL_25;
NET "TCLKD_P" LOC=J20  | IOSTANDARD=LVPECL_25;

NET "RESET_N" LOC=AH8 | IOSTANDARD=LVTTL;

NET "DDR2_A_A_0"     LOC=T26  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_1"     LOC=U26  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_2"     LOC=R27  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_3"     LOC=R26  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_4"     LOC=U28  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_5"     LOC=U27  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_6"     LOC=T29  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_7"     LOC=T28  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_8"     LOC=AE32 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_9"     LOC=AD32 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_10"    LOC=P30  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_11"    LOC=P31  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_12"    LOC=AP32 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_A_13"    LOC=U31  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_BA_0"    LOC=AA25 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_BA_1"    LOC=T25  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_BA_2"    LOC=U25  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_CAS_N"   LOC=AB27 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_CK_0"    LOC=K33  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_CK_0_N"  LOC=K32  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_CK_1"    LOC=M31  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_CK_1_N"  LOC=N30  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_CKE_0"   LOC=AC28 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_CKE_1"   LOC=AB28 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_DM_0"    LOC=C34  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DM_1"    LOC=G33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DM_2"    LOC=T34  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DM_3"    LOC=F29  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DM_4"    LOC=L30  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DM_5"    LOC=AM33 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DM_6"    LOC=Y33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DM_7"    LOC=AC33 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_0"    LOC=B32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_1"    LOC=A33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_2"    LOC=B33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_3"    LOC=C33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_4"    LOC=D32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_5"    LOC=C32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_6"    LOC=D34  | IOSTANDARD=SSTL18_II_DCI;
```

```
NET "DDR2_A_DQ_7"    LOC=G32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_8"    LOC=E32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_9"    LOC=E33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_10"   LOC=F33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_11"   LOC=E34  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_12"   LOC=F34  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_13"   LOC=L33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_14"   LOC=M32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_15"   LOC=P34  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_16"   LOC=P32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_17"   LOC=N32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_18"   LOC=T33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_19"   LOC=R34  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_20"   LOC=R32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_21"   LOC=R33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_22"   LOC=U33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_23"   LOC=U32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_24"   LOC=E29  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_25"   LOC=E31  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_26"   LOC=F30  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_27"   LOC=G30  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_28"   LOC=F31  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_29"   LOC=H29  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_30"   LOC=J29  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_31"   LOC=L29  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_32"   LOC=H30  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_33"   LOC=G31  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_34"   LOC=J30  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_35"   LOC=J31  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_36"   LOC=M30  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_37"   LOC=T31  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_38"   LOC=R31  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_39"   LOC=U30  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_40"   LOC=AJ32 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_41"   LOC=AK32 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_42"   LOC=AL34 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_43"   LOC=AL33 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_44"   LOC=AM32 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_45"   LOC=AN34 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_46"   LOC=AN33 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_47"   LOC=AN32 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_48"   LOC=Y32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_49"   LOC=Y34  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_50"   LOC=AA34 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_51"   LOC=AA33 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_52"   LOC=V34  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_53"   LOC=W34  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_54"   LOC=V33  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_55"   LOC=V32  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_56"   LOC=AB32 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_57"   LOC=AB33 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_58"   LOC=AC32 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_59"   LOC=AC34 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_60"   LOC=AD34 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_61"   LOC=AG32 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_62"   LOC=AK33 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQ_63"   LOC=AK34 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_A_DQS_0"   LOC=J32  | IOSTANDARD=DIFF_SSTL18_II_DCI;
```

```
NET "DDR2_A_DQS_0_N" LOC=H33  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_1"   LOC=H34  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_1_N" LOC=J34  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_2"   LOC=L34  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_2_N" LOC=K34  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_3"   LOC=N29  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_3_N" LOC=P29  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_4"   LOC=K31  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_4_N" LOC=L31  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_5"   LOC=AH34 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_5_N" LOC=AJ34 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_6"   LOC=AF33 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_6_N" LOC=AE33 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_7"   LOC=AF34 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_DQS_7_N" LOC=AE34 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_A_ODT_0"   LOC=AB25 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_ODT_1"   LOC=AB26 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_RAS_N"   LOC=AA26 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_S_0_N"   LOC=Y24  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_S_1_N"   LOC=AA24 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_A_SA_0"    LOC=M10  | IOSTANDARD=LVCMOS18;
NET "DDR2_A_SA_1"    LOC=L9   | IOSTANDARD=LVCMOS18;
NET "DDR2_A_SCL"     LOC=AG28 | IOSTANDARD=LVCMOS18;
NET "DDR2_A_SDA"     LOC=AA28 | IOSTANDARD=LVCMOS18;
NET "DDR2_A_WE_N"    LOC=AC27 | IOSTANDARD=SSTL18_I_DCI;

NET "DDR2_B_A_0"     LOC=T24  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_1"     LOC=E27  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_2"     LOC=E26  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_3"     LOC=AG30 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_4"     LOC=AA30 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_5"     LOC=AA29 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_6"     LOC=AE24 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_7"     LOC=AD24 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_8"     LOC=AD25 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_9"     LOC=AD26 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_10"    LOC=AC24 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_11"    LOC=AC25 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_12"    LOC=AJ26 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_A_13"    LOC=AH27 | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_BA_0"    LOC=D12  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_BA_1"    LOC=J11  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_BA_2"    LOC=K11  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_CAS_N"   LOC=H10  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_CK_0"    LOC=AK29 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_CK_0_N"  LOC=AJ29 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_CK_1"    LOC=AK28 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_CK_1_N"  LOC=AK27 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_CKE_0"   LOC=B13  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_CKE_1"   LOC=C13  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_DM_0"    LOC=AG27 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DM_1"    LOC=V30  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DM_2"    LOC=AD30 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DM_3"    LOC=AF29 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DM_4"    LOC=J24  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DM_5"    LOC=F25  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DM_6"    LOC=N24  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DM_7"    LOC=F10  | IOSTANDARD=SSTL18_II_DCI;
```

```
NET "DDR2_B_DQ_0"    LOC=AE28 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_1"    LOC=AF28 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_2"    LOC=AF24 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_3"    LOC=AG25 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_4"    LOC=AF25 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_5"    LOC=AF26 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_6"    LOC=AE27 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_7"    LOC=AE26 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_8"    LOC=Y27  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_9"    LOC=Y26  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_10"   LOC=V25  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_11"   LOC=V24  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_12"   LOC=W27  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_13"   LOC=W26  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_14"   LOC=W24  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_15"   LOC=W25  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_16"   LOC=V28  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_17"   LOC=V27  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_18"   LOC=W31  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_19"   LOC=Y31  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_20"   LOC=AF31 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_21"   LOC=AE29 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_22"   LOC=W29  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_23"   LOC=V29  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_24"   LOC=AC29 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_25"   LOC=AD29 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_26"   LOC=AJ31 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_27"   LOC=AK31 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_28"   LOC=AF30 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_29"   LOC=AJ30 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_30"   LOC=AH30 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_31"   LOC=AH29 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_32"   LOC=K24  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_33"   LOC=L24  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_34"   LOC=L25  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_35"   LOC=L26  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_36"   LOC=J25  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_37"   LOC=M25  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_38"   LOC=M26  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_39"   LOC=J27  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_40"   LOC=G25  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_41"   LOC=G26  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_42"   LOC=H25  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_43"   LOC=H24  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_44"   LOC=F26  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_45"   LOC=K28  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_46"   LOC=L28  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_47"   LOC=K27  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_48"   LOC=N25  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_49"   LOC=R24  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_50"   LOC=P26  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_51"   LOC=P27  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_52"   LOC=P24  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_53"   LOC=P25  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_54"   LOC=M28  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_55"   LOC=N28  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_56"   LOC=E9   | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_57"   LOC=E8   | IOSTANDARD=SSTL18_II_DCI;
```

```
NET "DDR2_B_DQ_58"   LOC=F9   | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_59"   LOC=F8   | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_60"   LOC=G10  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_61"   LOC=G8   | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_62"   LOC=H8   | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQ_63"   LOC=D11  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_B_DQS_0"   LOC=AK26 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_0_N" LOC=AJ27 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_1"   LOC=Y28  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_1_N" LOC=Y29  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_2"   LOC=AB31 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_2_N" LOC=AA31 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_3"   LOC=AB30 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_3_N" LOC=AC30 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_4"   LOC=G27  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_4_N" LOC=H27  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_5"   LOC=H28  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_5_N" LOC=G28  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_6"   LOC=E28  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_6_N" LOC=F28  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_7"   LOC=A13  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_DQS_7_N" LOC=B12  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_B_ODT_0"   LOC=K8   | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_ODT_1"   LOC=K9   | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_RAS_N"   LOC=C12  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_S_0_N"   LOC=J10  | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_S_1_N"   LOC=J9   | IOSTANDARD=SSTL18_I_DCI;
NET "DDR2_B_SA_0"    LOC=E12  | IOSTANDARD=LVCMOS18;
NET "DDR2_B_SA_1"    LOC=E13  | IOSTANDARD=LVCMOS18;
NET "DDR2_B_SCL"     LOC=M8   | IOSTANDARD=LVCMOS18;
NET "DDR2_B_SDA"     LOC=G12  | IOSTANDARD=LVCMOS18;
NET "DDR2_B_WE_N"    LOC=H9   | IOSTANDARD=SSTL18_I_DCI;

NET "FLASH_1_D_0"  LOC=AC8  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_1"  LOC=AD11 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_2"  LOC=AG11 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_3"  LOC=AA9  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_4"  LOC=AA8  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_5"  LOC=AK8  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_6"  LOC=AA10 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_7"  LOC=AG10 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_8"  LOC=AE11 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_9"  LOC=AE8  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_10" LOC=AB8  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_11" LOC=AC9  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_12" LOC=AM13 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_13" LOC=AK9  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_14" LOC=AH10 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_15" LOC=AJ10 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_0"  LOC=AB7  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_1"  LOC=AC7  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_2"  LOC=AF11 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_3"  LOC=AC5  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_4"  LOC=AB10 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_5"  LOC=AJ9  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_6"  LOC=AC10 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_7"  LOC=AG8  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_8"  LOC=AB6  | IOSTANDARD=LVTTL | PULLDOWN;
```

```
NET "FLASH_2_D_9"  LOC=AA5  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_10" LOC=AF9  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_11" LOC=AC4  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_12" LOC=AN13 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_13" LOC=AE9  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_14" LOC=AL10 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_15" LOC=AL11 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_A_1"    LOC=Y8   | IOSTANDARD=LVTTL;
NET "FLASH_A_2"    LOC=Y9   | IOSTANDARD=LVTTL;
NET "FLASH_A_3"    LOC=AB5  | IOSTANDARD=LVTTL;
NET "FLASH_A_4"    LOC=AF10 | IOSTANDARD=LVTTL;
NET "FLASH_A_5"    LOC=AD10 | IOSTANDARD=LVTTL;
NET "FLASH_A_6"    LOC=Y7   | IOSTANDARD=LVTTL;
NET "FLASH_A_7"    LOC=AA6  | IOSTANDARD=LVTTL;
NET "FLASH_A_8"    LOC=Y6   | IOSTANDARD=LVTTL;
NET "FLASH_A_9"    LOC=W6   | IOSTANDARD=LVTTL;
NET "FLASH_A_10"   LOC=V7   | IOSTANDARD=LVTTL;
NET "FLASH_A_11"   LOC=AD9  | IOSTANDARD=LVTTL;
NET "FLASH_A_12"   LOC=W7   | IOSTANDARD=LVTTL;
NET "FLASH_A_13"   LOC=W11  | IOSTANDARD=LVTTL;
NET "FLASH_A_14"   LOC=V8   | IOSTANDARD=LVTTL;
NET "FLASH_A_15"   LOC=U8   | IOSTANDARD=LVTTL;
NET "FLASH_A_16"   LOC=AN14 | IOSTANDARD=LVTTL;
NET "FLASH_A_17"   LOC=AM12 | IOSTANDARD=LVTTL;
NET "FLASH_A_18"   LOC=V10  | IOSTANDARD=LVTTL;
NET "FLASH_A_19"   LOC=V9   | IOSTANDARD=LVTTL;
NET "FLASH_A_20"   LOC=AP14 | IOSTANDARD=LVTTL;
NET "FLASH_A_21"   LOC=AP12 | IOSTANDARD=LVTTL;
NET "FLASH_A_22"   LOC=AN12 | IOSTANDARD=LVTTL;
NET "FLASH_A_23"   LOC=AF8  | IOSTANDARD=LVTTL;
NET "FLASH_A_24"   LOC=AK11 | IOSTANDARD=LVTTL;
NET "FLASH_A_25"   LOC=W10  | IOSTANDARD=LVTTL;
NET "FLASH_CE_N"   LOC=Y11  | IOSTANDARD=LVTTL;
NET "FLASH_OE_N"   LOC=AM11 | IOSTANDARD=LVTTL;
NET "FLASH_WAIT"   LOC=AH9  | IOSTANDARD=LVTTL;
NET "FLASH_WE_N"   LOC=AJ11 | IOSTANDARD=LVTTL;

NET "MMC_BUS_0" LOC=AG5 | IOSTANDARD=LVTTL;
NET "MMC_BUS_1" LOC=AF5 | IOSTANDARD=LVTTL;
NET "MMC_BUS_2" LOC=AH5 | IOSTANDARD=LVTTL;
NET "MMC_BUS_3" LOC=AG6 | IOSTANDARD=LVTTL;
NET "MMC_BUS_4" LOC=AH7 | IOSTANDARD=LVTTL;
NET "MMC_BUS_5" LOC=AG7 | IOSTANDARD=LVTTL;
NET "MMC_BUS_6" LOC=W9  | IOSTANDARD=LVTTL;
NET "MMC_BUS_7" LOC=AJ7 | IOSTANDARD=LVTTL;

NET "PHY_125CLK"  LOC=F6   | IOSTANDARD=LVCMOS25 |
    CLOCK_DEDICATED_ROUTE=FALSE;
NET "PHY_COL"     LOC=N8   | IOSTANDARD=LVCMOS25;
NET "PHY_CRS"     LOC=R8   | IOSTANDARD=LVCMOS25;
NET "PHY_GTX_CLK" LOC=H19  | IOSTANDARD=LVCMOS25;
NET "PHY_INT_N"   LOC=H5   | IOSTANDARD=LVCMOS25;
NET "PHY_MDC"     LOC=T9   | IOSTANDARD=LVCMOS25;
NET "PHY_MDIO"    LOC=G5   | IOSTANDARD=LVCMOS25;
NET "PHY_RESET_N" LOC=G7   | IOSTANDARD=LVCMOS25;
NET "PHY_RX_CLK"  LOC=H17  | IOSTANDARD=LVCMOS25;
NET "PHY_RXD_0"   LOC=R6   | IOSTANDARD=LVCMOS25;
NET "PHY_RXD_1"   LOC=N7   | IOSTANDARD=LVCMOS25;
```

```
NET "PHY_RXD_2"   LOC=P7   | IOSTANDARD=LVCMOS25;
NET "PHY_RXD_3"   LOC=R7   | IOSTANDARD=LVCMOS25;
NET "PHY_RXD_4"   LOC=T8   | IOSTANDARD=LVCMOS25;
NET "PHY_RXD_5"   LOC=P5   | IOSTANDARD=LVCMOS25;
NET "PHY_RXD_6"   LOC=T6   | IOSTANDARD=LVCMOS25;
NET "PHY_RXD_7"   LOC=P10  | IOSTANDARD=LVCMOS25;
NET "PHY_RX_DV"   LOC=N5   | IOSTANDARD=LVCMOS25;
NET "PHY_RX_ER"   LOC=P6   | IOSTANDARD=LVCMOS25;
NET "PHY_TX_CLK"  LOC=AH18 | IOSTANDARD=LVCMOS25;
NET "PHY_TXD_0"   LOC=L4   | IOSTANDARD=LVCMOS25;
NET "PHY_TXD_1"   LOC=K6   | IOSTANDARD=LVCMOS25;
NET "PHY_TXD_2"   LOC=M7   | IOSTANDARD=LVCMOS25;
NET "PHY_TXD_3"   LOC=J6   | IOSTANDARD=LVCMOS25;
NET "PHY_TXD_4"   LOC=L5   | IOSTANDARD=LVCMOS25;
NET "PHY_TXD_5"   LOC=K7   | IOSTANDARD=LVCMOS25;
NET "PHY_TXD_6"   LOC=J5   | IOSTANDARD=LVCMOS25;
NET "PHY_TXD_7"   LOC=G6   | IOSTANDARD=LVCMOS25;
NET "PHY_TX_EN"   LOC=M5   | IOSTANDARD=LVCMOS25;
NET "PHY_TX_ER"   LOC=L6   | IOSTANDARD=LVCMOS25;

NET "UART_USB_CTS" LOC=AK6 | IOSTANDARD=LVTTL;
NET "UART_USB_RTS" LOC=AK7 | IOSTANDARD=LVTTL;
NET "UART_USB_RXD" LOC=AD6 | IOSTANDARD=LVTTL;
NET "UART_USB_TXD" LOC=AE6 | IOSTANDARD=LVTTL;

NET "UART_AMC_RXD" LOC=H18 | IOSTANDARD=LVCMOS25;
NET "UART_AMC_TXD" LOC=H20 | IOSTANDARD=LVCMOS25;

NET "SFP_0_LED_A"   LOC=AG12 | IOSTANDARD=LVTTL;
NET "SFP_0_LED_B"   LOC=AF13 | IOSTANDARD=LVTTL;
NET "SFP_0_MOD_ABS" LOC=AE12 | IOSTANDARD=LVTTL;
NET "SFP_0_SCL"     LOC=AG23 | IOSTANDARD=LVTTL;
NET "SFP_0_SDA"     LOC=AF23 | IOSTANDARD=LVTTL;
NET "SFP_0_TX_DIS"  LOC=AE13 | IOSTANDARD=LVTTL;

NET "SFP_1_LED_A"   LOC=AE17 | IOSTANDARD=LVTTL;
NET "SFP_1_LED_B"   LOC=AF16 | IOSTANDARD=LVTTL;
NET "SFP_1_MOD_ABS" LOC=AE16 | IOSTANDARD=LVTTL;
NET "SFP_1_SCL"     LOC=AD20 | IOSTANDARD=LVTTL;
NET "SFP_1_SDA"     LOC=AE21 | IOSTANDARD=LVTTL;
NET "SFP_1_TX_DIS"  LOC=AF15 | IOSTANDARD=LVTTL;

NET "LVDS_AMC_P02_RX_N" LOC=L20  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P02_RX_P" LOC=L21  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P02_TX_N" LOC=L16  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P02_TX_P" LOC=L15  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P03_RX_N" LOC=H22  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P03_RX_P" LOC=G22  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P03_TX_N" LOC=K14  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P03_TX_P" LOC=L14  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P04_RX_N" LOC=H23  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P04_RX_P" LOC=G23  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P04_TX_N" LOC=K12  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P04_TX_P" LOC=K13  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P05_RX_N" LOC=J19  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P05_RX_P" LOC=K18  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P05_TX_N" LOC=G16  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P05_TX_P" LOC=G15  | IOSTANDARD=LVDS_25;
```

```
NET "LVDS_AMC_P07_RX_N" LOC=K19  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P07_RX_P" LOC=L19  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P07_TX_N" LOC=J17  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P07_TX_P" LOC=J16  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P08_RX_N" LOC=AH13 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P08_RX_P" LOC=AH14 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P08_TX_N" LOC=AG16 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P08_TX_P" LOC=AH17 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P09_RX_N" LOC=AG15 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P09_RX_P" LOC=AH15 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P09_TX_N" LOC=AG20 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P09_TX_P" LOC=AG21 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P11_RX_N" LOC=L18  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P11_RX_P" LOC=K17  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P11_TX_N" LOC=AG13 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P11_TX_P" LOC=AH12 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P13_RX_N" LOC=T11  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P13_RX_P" LOC=T10  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P13_TX_N" LOC=J7   | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P13_TX_P" LOC=H7   | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P14_RX_N" LOC=H15  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P14_RX_P" LOC=H14  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P14_TX_N" LOC=E7   | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P14_TX_P" LOC=E6   | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P15_RX_N" LOC=K22  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P15_RX_P" LOC=K23  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P15_TX_N" LOC=H12  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P15_TX_P" LOC=J12  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P17_RX_N" LOC=K21  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P17_RX_P" LOC=J22  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC_P17_TX_N" LOC=J15  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC_P17_TX_P" LOC=K16  | IOSTANDARD=LVDS_25;
```

Listing D.2: Pin-out for CNCB v3.3 FPGA (XC4VFX60-11FFG1152C)

```
NET "CLK_100" LOC=H17 | IOSTANDARD=LVCMOS25;

NET "FANOUT_CLK_OUT_N" LOC=H4 | IOSTANDARD=LVPECL_25;
NET "FANOUT_CLK_OUT_P" LOC=H5 | IOSTANDARD=LVPECL_25;

NET "RESET_N" LOC=AH18 | IOSTANDARD=LVTTL;

NET "DDR2_A_0"     LOC=G16 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_1"     LOC=G17 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_2"     LOC=H13 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_3"     LOC=H14 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_4"     LOC=F18 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_5"     LOC=G18 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_6"     LOC=M23 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_7"     LOC=N23 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_8"     LOC=N24 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_9"     LOC=P24 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_10"    LOC=N22 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_11"    LOC=P22 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_12"    LOC=L21 | IOSTANDARD=SSTL18_II;
NET "DDR2_A_13"    LOC=J20 | IOSTANDARD=SSTL18_II;
NET "DDR2_BA_0"    LOC=E17 | IOSTANDARD=SSTL18_II;
NET "DDR2_BA_1"    LOC=E18 | IOSTANDARD=SSTL18_II;
NET "DDR2_BA_2"    LOC=G15 | IOSTANDARD=SSTL18_II;
NET "DDR2_CAS_N"   LOC=F14 | IOSTANDARD=SSTL18_II;
NET "DDR2_CK_0"    LOC=E19 | IOSTANDARD=DIFF_SSTL18_II;
NET "DDR2_CK_0_N"  LOC=F19 | IOSTANDARD=DIFF_SSTL18_II;
NET "DDR2_CK_1"    LOC=K24 | IOSTANDARD=DIFF_SSTL18_II;
NET "DDR2_CK_1_N"  LOC=L24 | IOSTANDARD=DIFF_SSTL18_II;
NET "DDR2_CKE_0"   LOC=D6  | IOSTANDARD=SSTL18_II;
NET "DDR2_CKE_1"   LOC=C5  | IOSTANDARD=SSTL18_II;
NET "DDR2_DM_0"    LOC=D24 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DM_1"    LOC=J22 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DM_2"    LOC=K26 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DM_3"    LOC=H20 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DM_4"    LOC=E11 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DM_5"    LOC=G8  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DM_6"    LOC=J12 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DM_7"    LOC=C7  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_0"    LOC=H24 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_1"    LOC=J24 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_2"    LOC=E23 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_3"    LOC=F23 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_4"    LOC=E24 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_5"    LOC=F24 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_6"    LOC=G23 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_7"    LOC=C24 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_8"    LOC=C23 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_9"    LOC=C22 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_10"   LOC=J25 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_11"   LOC=H25 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_12"   LOC=D22 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_13"   LOC=E22 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_14"   LOC=D25 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_15"   LOC=C25 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_16"   LOC=G26 | IOSTANDARD=SSTL18_II_DCI;
```

```
NET "DDR2_DQ_17"   LOC=F26 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_18"   LOC=J21 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_19"   LOC=K21 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_20"   LOC=E26 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_21"   LOC=D26 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_22"   LOC=F21 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_23"   LOC=G21 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_24"   LOC=J26 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_25"   LOC=D21 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_26"   LOC=E21 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_27"   LOC=E27 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_28"   LOC=D27 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_29"   LOC=K23 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_30"   LOC=C28 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_31"   LOC=C27 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_32"   LOC=G10 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_33"   LOC=H10 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_34"   LOC=D10 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_35"   LOC=C10 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_36"   LOC=F10 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_37"   LOC=F9  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_38"   LOC=H9  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_39"   LOC=F11 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_40"   LOC=D9  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_41"   LOC=E9  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_42"   LOC=D12 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_43"   LOC=D11 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_44"   LOC=E8  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_45"   LOC=F8  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_46"   LOC=J11 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_47"   LOC=J10 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_48"   LOC=G12 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_49"   LOC=G11 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_50"   LOC=J7  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_51"   LOC=K7  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_52"   LOC=K11 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_53"   LOC=L11 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_54"   LOC=G7  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_55"   LOC=H7  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_56"   LOC=H12 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_57"   LOC=E7  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_58"   LOC=E6  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_59"   LOC=E13 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_60"   LOC=E12 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_61"   LOC=K9  | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_62"   LOC=E14 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQ_63"   LOC=D14 | IOSTANDARD=SSTL18_II_DCI;
NET "DDR2_DQS_0"   LOC=G22 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_0_N" LOC=H22 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_1"   LOC=G25 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_1_N" LOC=F25 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_2"   LOC=F20 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_2_N" LOC=G20 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_3"   LOC=F28 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_3_N" LOC=E28 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_4"   LOC=C9  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_4_N" LOC=C8  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_5"   LOC=C13 | IOSTANDARD=DIFF_SSTL18_II_DCI;
```

```
NET "DDR2_DQS_5_N" LOC=C12 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_6"   LOC=F6  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_6_N" LOC=G6  | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_7"   LOC=D16 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_DQS_7_N" LOC=D15 | IOSTANDARD=DIFF_SSTL18_II_DCI;
NET "DDR2_ODT_0"   LOC=G13 | IOSTANDARD=SSTL18_II;
NET "DDR2_ODT_1"   LOC=D7  | IOSTANDARD=SSTL18_II;
NET "DDR2_RAS_N"   LOC=F15 | IOSTANDARD=SSTL18_II;
NET "DDR2_S_0_N"   LOC=F16 | IOSTANDARD=SSTL18_II;
NET "DDR2_S_1_N"   LOC=F13 | IOSTANDARD=SSTL18_II;
NET "DDR2_SA_0"    LOC=M11 | IOSTANDARD=LVCMOS18;
NET "DDR2_SA_1"    LOC=M12 | IOSTANDARD=LVCMOS18;
NET "DDR2_SCL"     LOC=E4  | IOSTANDARD=LVCMOS18;
NET "DDR2_SDA"     LOC=E3  | IOSTANDARD=LVCMOS18;
NET "DDR2_WE_N"    LOC=E16 | IOSTANDARD=SSTL18_II;

NET "FLASH_1_D_0"  LOC=AL30 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_1"  LOC=AK31 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_2"  LOC=AL28 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_3"  LOC=AM27 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_4"  LOC=AM21 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_5"  LOC=AJ27 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_6"  LOC=AM23 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_7"  LOC=AK22 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_8"  LOC=AM32 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_9"  LOC=AL29 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_10" LOC=AK29 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_11" LOC=AK28 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_12" LOC=AL24 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_13" LOC=AM25 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_14" LOC=AK21 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_1_D_15" LOC=AL20 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_0"  LOC=AF15 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_1"  LOC=AJ16 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_2"  LOC=AH15 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_3"  LOC=AH14 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_4"  LOC=AH8  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_5"  LOC=AL13 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_6"  LOC=AH10 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_7"  LOC=AG11 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_8"  LOC=AL18 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_9"  LOC=AG15 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_10" LOC=AG13 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_11" LOC=AH13 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_12" LOC=AG12 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_13" LOC=AH12 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_14" LOC=AG10 | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_2_D_15" LOC=AH7  | IOSTANDARD=LVTTL | PULLDOWN;
NET "FLASH_A_1"    LOC=AK32 | IOSTANDARD=LVTTL;
NET "FLASH_A_2"    LOC=AM31 | IOSTANDARD=LVTTL;
NET "FLASH_A_3"    LOC=AL31 | IOSTANDARD=LVTTL;
NET "FLASH_A_4"    LOC=AM30 | IOSTANDARD=LVTTL;
NET "FLASH_A_5"    LOC=AK26 | IOSTANDARD=LVTTL;
NET "FLASH_A_6"    LOC=AM28 | IOSTANDARD=LVTTL;
NET "FLASH_A_7"    LOC=AJ29 | IOSTANDARD=LVTTL;
NET "FLASH_A_8"    LOC=AM26 | IOSTANDARD=LVTTL;
NET "FLASH_A_9"    LOC=AK27 | IOSTANDARD=LVTTL;
NET "FLASH_A_10"   LOC=AL26 | IOSTANDARD=LVTTL;
```

```
NET "FLASH_A_11"   LOC=AJ26 | IOSTANDARD=LVTTL;
NET "FLASH_A_12"   LOC=AJ24 | IOSTANDARD=LVTTL;
NET "FLASH_A_13"   LOC=AJ25 | IOSTANDARD=LVTTL;
NET "FLASH_A_14"   LOC=AK24 | IOSTANDARD=LVTTL;
NET "FLASH_A_15"   LOC=AL23 | IOSTANDARD=LVTTL;
NET "FLASH_A_16"   LOC=AL21 | IOSTANDARD=LVTTL;
NET "FLASH_A_17"   LOC=AK14 | IOSTANDARD=LVTTL;
NET "FLASH_A_18"   LOC=AK23 | IOSTANDARD=LVTTL;
NET "FLASH_A_19"   LOC=AJ15 | IOSTANDARD=LVTTL;
NET "FLASH_A_20"   LOC=AM20 | IOSTANDARD=LVTTL;
NET "FLASH_A_21"   LOC=AL19 | IOSTANDARD=LVTTL;
NET "FLASH_A_22"   LOC=AL14 | IOSTANDARD=LVTTL;
NET "FLASH_A_23"   LOC=AK16 | IOSTANDARD=LVTTL;
NET "FLASH_A_24"   LOC=AJ12 | IOSTANDARD=LVTTL;
NET "FLASH_A_25"   LOC=AM22 | IOSTANDARD=LVTTL;
NET "FLASH_CE_N"   LOC=AK13 | IOSTANDARD=LVTTL;
NET "FLASH_OE_N"   LOC=AM13 | IOSTANDARD=LVTTL;
NET "FLASH_WAIT"   LOC=AJ14 | IOSTANDARD=LVTTL;
NET "FLASH_WE_N"   LOC=AH9  | IOSTANDARD=LVTTL;

NET "CPLD_IO" LOC=AK19 | IOSTANDARD=LVTTL;

NET "RESERVED_IO_0" LOC=AG20 | IOSTANDARD=LVTTL;
NET "RESERVED_IO_1" LOC=AH20 | IOSTANDARD=LVTTL;
NET "RESERVED_IO_2" LOC=AH19 | IOSTANDARD=LVTTL;

NET "LED_BOT_BLUE1"           LOC=AF20 | IOSTANDARD=LVTTL;
NET "LED_BOT_BLUE2"           LOC=AF21 | IOSTANDARD=LVTTL;
NET "LED_BOT_GREEN"           LOC=AF18 | IOSTANDARD=LVTTL;
NET "LED_BOT_RED"             LOC=AF19 | IOSTANDARD=LVTTL;
NET "LED_FP_AMBER1"           LOC=AJ19 | IOSTANDARD=LVTTL;
NET "LED_RAM_ERROR"           LOC=D5   | IOSTANDARD=LVCMOS18;
NET "LED_RAM_INIT_DONE"       LOC=K12  | IOSTANDARD=LVCMOS18;
NET "LED_TOP_GREEN_FP_AMBER3" LOC=AG17 | IOSTANDARD=LVTTL;
NET "LED_TOP_RED_FP_RED1"     LOC=AG16 | IOSTANDARD=LVTTL;

NET "PHY_BP_125CLK"  LOC=K18  | IOSTANDARD=LVCMOS25;
NET "PHY_BP_COL"     LOC=AA29 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_CRS"     LOC=AA30 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_GTX_CLK" LOC=AE31 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_INT_N"   LOC=AE29 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_MDC"     LOC=AD29 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_MDIO"    LOC=AF29 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RESET_N" LOC=AC29 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RX_CLK"  LOC=J14  | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RXD_0"   LOC=AC32 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RXD_1"   LOC=AG31 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RXD_2"   LOC=AD30 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RXD_3"   LOC=AB31 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RXD_4"   LOC=AB32 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RXD_5"   LOC=AB30 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RXD_6"   LOC=AA31 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RXD_7"   LOC=Y32  | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RX_DV"   LOC=AD31 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_RX_ER"   LOC=AF31 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_TX_CLK"  LOC=K19  | IOSTANDARD=LVCMOS25;
NET "PHY_BP_TXD_0"   LOC=AE32 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_TXD_1"   LOC=AF30 | IOSTANDARD=LVCMOS25;
```

```
NET "PHY_BP_TXD_2"   LOC=AG32 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_TXD_3"   LOC=AJ30 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_TXD_4"   LOC=AJ32 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_TXD_5"   LOC=AG30 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_TXD_6"   LOC=AJ31 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_TXD_7"   LOC=AH30 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_TX_EN"   LOC=AC30 | IOSTANDARD=LVCMOS25;
NET "PHY_BP_TX_ER"   LOC=AD32 | IOSTANDARD=LVCMOS25;

NET "PHY_RJ_125CLK"  LOC=J16 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_COL"     LOC=AG8 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_CRS"     LOC=AG7 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_GTX_CLK" LOC=AL4 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_INT_N"   LOC=AF6 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_MDC"     LOC=AG3 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_MDIO"    LOC=AE6 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RESET_N" LOC=AH5 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RX_CLK"  LOC=L15 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RXD_0"   LOC=AJ6 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RXD_1"   LOC=AM3 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RXD_2"   LOC=AL3 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RXD_3"   LOC=AK6 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RXD_4"   LOC=AM5 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RXD_5"   LOC=AL5 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RXD_6"   LOC=AL6 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RXD_7"   LOC=AM6 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RX_DV"   LOC=AG6 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_RX_ER"   LOC=AJ4 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_TX_CLK"  LOC=H19 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_TXD_0"   LOC=AK4 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_TXD_1"   LOC=AG5 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_TXD_2"   LOC=AJ5 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_TXD_3"   LOC=AE4 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_TXD_4"   LOC=AF4 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_TXD_5"   LOC=AF5 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_TXD_6"   LOC=AF3 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_TXD_7"   LOC=AE3 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_TX_EN"   LOC=AK3 | IOSTANDARD=LVCMOS25;
NET "PHY_RJ_TX_ER"   LOC=AH3 | IOSTANDARD=LVCMOS25;

NET "UART_RXD" LOC=AF13 | IOSTANDARD=LVTTL;
NET "UART_TXD" LOC=AF14 | IOSTANDARD=LVTTL;

NET "LVDS_AMC1_P02_RX_N" LOC=M5 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC1_P02_RX_P" LOC=M6 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC1_P02_TX_N" LOC=P4 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC1_P02_TX_P" LOC=P5 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC1_P03_RX_N" LOC=T4 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC1_P03_RX_P" LOC=T5 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC1_P03_TX_N" LOC=T3 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC1_P03_TX_P" LOC=U3 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC1_P04_RX_N" LOC=R3 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC1_P04_RX_P" LOC=R4 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC1_P04_TX_N" LOC=N4 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC1_P04_TX_P" LOC=N5 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC1_P05_RX_N" LOC=N3 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC1_P05_RX_P" LOC=M3 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC1_P05_TX_N" LOC=L3 | IOSTANDARD=LVDS_25;
```

```
NET "LVDS_AMC1_P05_TX_P" LOC=L4 | IOSTANDARD=LVDS_25;

NET "LVDS_AMC2_P02_RX_N" LOC=W6  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC2_P02_RX_P" LOC=W7  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC2_P02_TX_N" LOC=AA4 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC2_P02_TX_P" LOC=AA5 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC2_P03_RX_N" LOC=Y3  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC2_P03_RX_P" LOC=Y4  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC2_P03_TX_N" LOC=W4  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC2_P03_TX_P" LOC=W5  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC2_P04_RX_N" LOC=V3  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC2_P04_RX_P" LOC=V4  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC2_P04_TX_N" LOC=T8  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC2_P04_TX_P" LOC=U8  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC2_P05_RX_N" LOC=U6  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC2_P05_RX_P" LOC=U7  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC2_P05_TX_N" LOC=P6  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC2_P05_TX_P" LOC=P7  | IOSTANDARD=LVDS_25;

NET "LVDS_AMC3_P02_RX_N" LOC=AB8 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC3_P02_RX_P" LOC=AC7 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC3_P02_TX_N" LOC=Y7  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC3_P02_TX_P" LOC=Y8  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC3_P03_RX_N" LOC=AD6 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC3_P03_RX_P" LOC=AD7 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC3_P03_TX_N" LOC=AD4 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC3_P03_TX_P" LOC=AD5 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC3_P04_RX_N" LOC=AC3 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC3_P04_RX_P" LOC=AC4 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC3_P04_TX_N" LOC=V7  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC3_P04_TX_P" LOC=V8  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC3_P05_RX_N" LOC=AB5 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC3_P05_RX_P" LOC=AC5 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC3_P05_TX_N" LOC=Y6  | IOSTANDARD=LVDS_25;
NET "LVDS_AMC3_P05_TX_P" LOC=AA6 | IOSTANDARD=LVDS_25;

NET "LVDS_AMC4_P02_RX_N" LOC=N9  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC4_P02_RX_P" LOC=N10 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC4_P02_TX_N" LOC=U27 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC4_P02_TX_P" LOC=U28 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC4_P03_RX_N" LOC=N32 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC4_P03_RX_P" LOC=P32 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC4_P03_TX_N" LOC=M10 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC4_P03_TX_P" LOC=L10 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC4_P04_RX_N" LOC=L8  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC4_P04_RX_P" LOC=L9  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC4_P04_TX_N" LOC=R27 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC4_P04_TX_P" LOC=R28 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC4_P05_RX_N" LOC=N7  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC4_P05_RX_P" LOC=N8  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_AMC4_P05_TX_N" LOC=R26 | IOSTANDARD=LVDS_25;
NET "LVDS_AMC4_P05_TX_P" LOC=T26 | IOSTANDARD=LVDS_25;

NET "LVDS_RTM_01_RX_N" LOC=J29 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_01_RX_P" LOC=K29 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_01_TX_N" LOC=N28 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_01_TX_P" LOC=N29 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_02_RX_N" LOC=T28 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
```

```
NET "LVDS_RTM_02_RX_P" LOC=T29 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_02_TX_N" LOC=U31 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_02_TX_P" LOC=U32 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_03_RX_N" LOC=P29 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_03_RX_P" LOC=R29 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_03_TX_N" LOC=V32 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_03_TX_P" LOC=W32 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_04_RX_N" LOC=W30 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_04_RX_P" LOC=W31 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_04_TX_N" LOC=L30 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_04_TX_P" LOC=L31 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_05_RX_N" LOC=L28 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_05_RX_P" LOC=L29 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_05_TX_N" LOC=R31 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_05_TX_P" LOC=R32 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_06_RX_N" LOC=U30 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_06_RX_P" LOC=V30 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_06_TX_N" LOC=M30 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_06_TX_P" LOC=N30 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_07_RX_N" LOC=F30 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_07_RX_P" LOC=G30 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_07_TX_N" LOC=M25 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_07_TX_P" LOC=M26 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_08_RX_N" LOC=T30 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_08_RX_P" LOC=T31 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_08_TX_N" LOC=J27 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_08_TX_P" LOC=K28 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_09_RX_N" LOC=J30 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_09_RX_P" LOC=J31 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_09_TX_N" LOC=H29 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_09_TX_P" LOC=H30 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_10_RX_N" LOC=M31 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_10_RX_P" LOC=M32 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_10_TX_N" LOC=K31 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_10_TX_P" LOC=K32 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_11_RX_N" LOC=C29 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_11_RX_P" LOC=D29 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_11_TX_N" LOC=D31 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_11_TX_P" LOC=E31 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_12_RX_N" LOC=H32 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_12_RX_P" LOC=J32 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_12_TX_N" LOC=G31 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_12_TX_P" LOC=G32 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_13_RX_N" LOC=E29 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_13_RX_P" LOC=F29 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_13_TX_N" LOC=F3  | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_13_TX_P" LOC=F4  | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_14_RX_N" LOC=F31 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_14_RX_P" LOC=E32 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_14_TX_N" LOC=C32 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_14_TX_P" LOC=D32 | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_15_RX_N" LOC=C30 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_15_RX_P" LOC=D30 | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_15_TX_N" LOC=H3  | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_15_TX_P" LOC=G3  | IOSTANDARD=LVDS_25;
NET "LVDS_RTM_16_RX_N" LOC=K3  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_16_RX_P" LOC=K4  | IOSTANDARD=LVDS_25 | DIFF_TERM=TRUE;
NET "LVDS_RTM_16_TX_N" LOC=L5  | IOSTANDARD=LVDS_25;
```

```
NET "LVDS_RTM_16_TX_P" LOC=L6  | IOSTANDARD=LVDS_25;
```

Listing D.3: Pin-out for CNCB v3.3 CPLD (XC95144XL-10TQG100C)

```
NET CLK_IN          LOC = P23 | PERIOD = 10 ns;
NET ASYNC_RST_N     LOC = P67;

NET ENABLE          LOC = P65;
NET REV_SEL<0>      LOC = P13;
NET REV_SEL<1>      LOC = P12;
NET REV_SEL<2>      LOC = P9;

NET FPGA_CCLK       LOC = P27;
NET FPGA_INIT_B     LOC = P56;
NET FPGA_PROGRAM_B  LOC = P58;
NET FPGA_DONE       LOC = P55;
NET FPGA_D_IN       LOC = P54;
NET FPGA_D_OUT      LOC = P86;
NET FPGA_CS_B       LOC = P59;
NET FPGA_M<0>       LOC = P14;
NET FPGA_M<1>       LOC = P61;
NET FPGA_M<2>       LOC = P60;

NET IPMC_IO<0>      LOC = P87;
NET IPMC_IO<1>      LOC = P89;
NET FPGA_IO_LED<0>  LOC = P68;

NET FLASH_A<1>      LOC = P17;
NET FLASH_A<2>      LOC = P18;
NET FLASH_A<3>      LOC = P19;
NET FLASH_A<4>      LOC = P20;
NET FLASH_A<5>      LOC = P24;
NET FLASH_A<6>      LOC = P25;
NET FLASH_A<7>      LOC = P8;
NET FLASH_A<8>      LOC = P7;
NET FLASH_A<9>      LOC = P6;
NET FLASH_A<10>     LOC = P4;
NET FLASH_A<11>     LOC = P3;
NET FLASH_A<12>     LOC = P2;
NET FLASH_A<13>     LOC = P1;
NET FLASH_A<14>     LOC = P74;
NET FLASH_A<15>     LOC = P73;
NET FLASH_A<16>     LOC = P72;
NET FLASH_A<17>     LOC = P53;
NET FLASH_A<18>     LOC = P52;
NET FLASH_A<19>     LOC = P71;
NET FLASH_A<20>     LOC = P10;
NET FLASH_A<21>     LOC = P11;
NET FLASH_A<22>     LOC = P15;
NET FLASH_A<23>     LOC = P16;
NET FLASH_A<24>     LOC = P63;
NET FLASH_0_DQ<0>   LOC = P81;
NET FLASH_0_DQ<1>   LOC = P80;
NET FLASH_0_DQ<2>   LOC = P76;
NET FLASH_0_DQ<3>   LOC = P77;
NET FLASH_0_DQ<4>   LOC = P78;
NET FLASH_0_DQ<5>   LOC = P79;
NET FLASH_0_DQ<6>   LOC = P82;
NET FLASH_0_DQ<7>   LOC = P85;
NET FLASH_0_DQ<8>   LOC = P91;
```

```
NET FLASH_0_DQ<9>  LOC = P92;
NET FLASH_0_DQ<10> LOC = P96;
NET FLASH_0_DQ<11> LOC = P97;
NET FLASH_0_DQ<12> LOC = P99;
NET FLASH_0_DQ<13> LOC = P95;
NET FLASH_0_DQ<14> LOC = P94;
NET FLASH_0_DQ<15> LOC = P93;
NET FLASH_1_DQ<0>  LOC = P36;
NET FLASH_1_DQ<1>  LOC = P33;
NET FLASH_1_DQ<2>  LOC = P32;
NET FLASH_1_DQ<3>  LOC = P29;
NET FLASH_1_DQ<4>  LOC = P28;
NET FLASH_1_DQ<5>  LOC = P30;
NET FLASH_1_DQ<6>  LOC = P34;
NET FLASH_1_DQ<7>  LOC = P35;
NET FLASH_1_DQ<8>  LOC = P41;
NET FLASH_1_DQ<9>  LOC = P42;
NET FLASH_1_DQ<10> LOC = P46;
NET FLASH_1_DQ<11> LOC = P43;
NET FLASH_1_DQ<12> LOC = P49;
NET FLASH_1_DQ<13> LOC = P50;
NET FLASH_1_DQ<14> LOC = P40;
NET FLASH_1_DQ<15> LOC = P39;
NET FLASH_ADV_N    LOC = P66;
NET FLASH_CE_N     LOC = P37;
NET FLASH_OE_N     LOC = P70;
NET FLASH_WE_N     LOC = P22;
NET FLASH_RST_N    LOC = P64;
NET FLASH_WAIT     LOC = P90;
```

Bibliography

[1] S. Glashow. "Partial-Symmetries of Weak Interactions". In: *Nucl. Phys.* **22** (1961), pp. 579–588. DOI: 10.1016/0029-5582(61)90469-2.

[2] S. Weinberg. "A Model of Leptons". In: *Phys. Rev. Lett.* **19** (1967), pp. 1264–1266. DOI: 10.1103/PhysRevLett.19.1264.

[3] A. Salam. "Weak and Electromagnetic Interactions". In: *Conf. Proc.* **C680519** (1968), pp. 367–377.

[4] F. Englert and R. Brout. "Broken Symmetry and the Mass of Gauge Vector Mesons". In: *Phys. Rev. Lett.* **13** (1964), pp. 321–323. DOI: 10.1103/PhysRevLett.13.321.

[5] P. W. Higgs. "Broken Symmetries and the Masses of Gauge Bosons". In: *Phys. Rev. Lett.* **13** (1964), pp. 508–509. DOI: 10.1103/PhysRevLett.13.508.

[6] G. Guralnik, C. Hagen, and T. Kibble. "Global Conservation Laws and Massless Particles". In: *Phys. Rev. Lett.* **13** (1964), pp. 585–587. DOI: 10.1103/PhysRevLett.13.585.

[7] Y. Fukuda et al. (Super-Kamiokande). "Evidence for oscillation of atmospheric neutrinos". In: *Phys. Rev. Lett.* **81** (1998), pp. 1562–1567. DOI: 10.1103/PhysRevLett.81.1562. arXiv: hep-ex/9807003 [hep-ex].

[8] S. Glashow, J. Iliopoulos, and L. Maiani. "Weak Interactions with Lepton-Hadron Symmetry". In: *Phys. Rev.* **D2** (1970), pp. 1285–1292. DOI: 10.1103/PhysRevD.2.1285.

[9] M. Thomson. *Modern Particle Physics*. 1st ed. Cambridge University Press, 2013.

[10] A. Sakharov. "Violation of *CP* Invariance, *C* Asymmetry, and Baryon Asymmetry of the Universe". In: *Pisma Zh. Eksp. Teor. Fiz.* **5** (1967), pp. 32–35. DOI: 10.1070/PU1991v034n05ABEH002497.

[11] A. Bevan et al. (BaBar, Belle). "The Physics of the *B* Factories". In: *Eur. Phys. J.* **C74**.11 (2014), p. 3026. DOI: 10.1140/epjc/s10052-014-3026-9. arXiv: 1406.6311 [hep-ex].

[12] C. Wu, E. Ambler, R. Hayward, et al. "Experimental Test of Parity Conservation in Beta Decay". In: *Phys. Rev.* **105** (1957), pp. 1413–1414. DOI: 10.1103/PhysRev.105.1413.

[13] R. Garwin, L. Lederman, and M. Weinrich. "Observations of the Failure of Conservation of Parity and Charge Conjugation in Meson Decays: The Magnetic Moment of the Free Muon". In: *Phys. Rev.* **105** (1957), pp. 1415–1417. DOI: 10.1103/PhysRev.105.1415.

[14] J. Christenson, J. Cronin, V. Fitch, et al. "Evidence for the 2 π Decay of the K_2^0 Meson". In: *Phys. Rev. Lett.* **13** (1964), pp. 138–140. DOI: 10.1103/PhysRevLett.13.138.

[15] N. Cabibbo. "Unitary Symmetry and Leptonic Decays". In: *Phys. Rev. Lett.* **10** (1963), pp. 531–533. DOI: 10.1103/PhysRevLett.10.531.

[16] M. Kobayashi and T. Maskawa. "*CP* Violation in the Renormalizable Theory of Weak Interaction". In: *Prog. Theor. Phys.* **49** (1973), pp. 652–657. DOI: 10.1143/PTP.49.652.

[17] K. Olive et al. (Particle Data Group). "Review of Particle Physics". In: *Chin. Phys.* **C38** (2014), p. 090001. DOI: 10.1088/1674-1137/38/9/090001.

[18] L.-L. Chau and W.-Y. Keung. "Comments on the Parametrization of the Kobayashi-Maskawa Matrix". In: *Phys. Rev. Lett.* **53** (1984), p. 1802. DOI: 10.1103/PhysRevLett.53.1802.

[19] L. Wolfenstein. "Parametrization of the Kobayashi-Maskawa Matrix". In: *Phys. Rev. Lett.* **51** (1983), p. 1945. DOI: 10.1103/PhysRevLett.51.1945.

[20] J. Aubert et al. (E598). "Experimental Observation of a Heavy Particle J". In: *Phys. Rev. Lett.* **33** (1974), pp. 1404–1406. DOI: 10.1103/PhysRevLett.33.1404.

[21] J. Augustin et al. (SLAC-SP-017). "Discovery of a Narrow Resonance in e^+e^- Annihilation". In: *Phys. Rev. Lett.* **33** (1974), pp. 1406–1408. DOI: 10.1103/PhysRevLett.33.1406.

[22] S. Herb, D. Hom, L. Lederman, et al. "Observation of a Dimuon Resonance at 9.5 GeV in 400-GeV Proton-Nucleus Collisions". In: *Phys. Rev. Lett.* **39** (1977), pp. 252–255. DOI: 10.1103/PhysRevLett.39.252.

[23] I. I. Bigi and A. Sanda. "Notes on the Observability of *CP* Violations in B Decays". In: *Nucl. Phys.* **B193** (1981), p. 85. DOI: 10.1016/0550-3213(81)90519-8.

[24] I. I. Bigi and A. I. Sanda. *CP Violation*. 2nd ed. Cambridge University Press, 2009. DOI: 10.1017/CBO9780511581014.

[25] H. Albrecht et al. (ARGUS). "Observation of B^0-$\bar{B}^0$ Mixing". In: *Phys. Lett.* **B192** (1987), pp. 245–252. DOI: 10.1016/0370-2693(87)91177-4.

[26] A. Abashian, K. Gotow, N. Morgan, et al. "The Belle Detector". In: *Nucl. Instrum. Meth.* **A479** (2002), pp. 117–232. DOI: 10.1016/S0168-9002(01)02013-7.

[27] B. Aubert et al. (BaBar). "The BaBar detector". In: *Nucl. Instrum. Meth.* **A479** (2002), pp. 1–116. DOI: 10.1016/S0168-9002(01)02012-5. arXiv: hep-ex/0105044 [hep-ex].

[28] K. Abe et al. (Belle). "Observation of large *CP* violation in the neutral *B* meson system". In: *Phys. Rev. Lett.* **87** (2001), p. 091802. DOI: 10.1103/PhysRevLett.87.091802. arXiv: hep-ex/0107061 [hep-ex].

[29] B. Aubert et al. (BaBar). "Observation of *CP* violation in the B^0 meson system". In: *Phys. Rev. Lett.* **87** (2001), p. 091801. DOI: 10.1103/PhysRevLett.87.091801. arXiv: hep-ex/0107013 [hep-ex].

[30] I. Adachi, H. Aihara, D. Asner, et al. "Precise measurement of the *CP* violation parameter $\sin 2\phi_1$ in $B^0 \to (c\bar{c})K^0$ decays". In: *Phys. Rev. Lett.* **108** (2012), p. 171802. DOI: 10.1103/PhysRevLett.108.171802. arXiv: 1201.4643 [hep-ex].

[31] Y. Chao et al. (Belle). "Evidence for direct *CP* violation in $B^0 \to K^+\pi^-$ decays". In: *Phys. Rev. Lett.* **93** (2004), p. 191802. DOI: 10.1103/PhysRevLett.93.191802. arXiv: hep-ex/0408100 [hep-ex].

[32] B. Aubert et al. (BaBar). "Direct *CP* Violating Asymmetry in $B^0 \to K^+\pi^-$ Decays". In: *Phys. Rev. Lett.* **93** (2004), p. 131801. DOI: 10.1103/PhysRevLett.93.131801. arXiv: hep-ex/0407057 [hep-ex].

[33] R. Aaij et al. (LHCb). "Evidence for *CP* violation in time-integrated $D^0 \to h^-h^+$ decay rates". In: *Phys. Rev. Lett.* **108** (2012), p. 111602. DOI: 10.1103/PhysRevLett.108.111602. arXiv: 1112.0938 [hep-ex].

[34] J. Charles et al. (CKMfitter Group). "CP violation and the CKM matrix: Assessing the impact of the asymmetric B factories". In: *Eur. Phys. J.* **C41** (2005), pp. 1–131. DOI: 10.1140/epjc/s2005-02169-1. arXiv: hep-ph/0406184 [hep-ph].

[35] D. Bernard (BaBar). "Results on conventional and exotic charmonium at BaBar". In: *PoS* **DIS2013** (2013), p. 179. arXiv: 1311.0968 [hep-ex].

[36] K. Abe et al. (Belle). "Observation of double $c\bar{c}$ production in e^+e^- annihilation at $\sqrt{s} \approx 10.6$ GeV". In: *Phys. Rev. Lett.* **89** (2002), p. 142001. DOI: 10.1103/PhysRevLett.89.142001. arXiv: hep-ex/0205104 [hep-ex].

[37] T. Barnes, S. Godfrey, and E. Swanson. "Higher charmonia". In: *Phys. Rev.* **D72** (2005), p. 054026. DOI: 10.1103/PhysRevD.72.054026. arXiv: hep-ph/0505002 [hep-ph].

[38] L. Liu et al. (Hadron Spectrum). "Excited and exotic charmonium spectroscopy from lattice QCD". In: *JHEP* **1207** (2012), p. 126. DOI: 10.1007/JHEP07(2012)126. arXiv: 1204.5425 [hep-ph].

[39] C. J. Morningstar and M. J. Peardon. "The Glueball spectrum from an anisotropic lattice study". In: *Phys. Rev.* **D60** (1999), p. 034509. DOI: 10.1103/PhysRevD.60.034509. arXiv: hep-lat/9901004 [hep-lat].

[40] S. Choi et al. (Belle). "Observation of a narrow charmoniumlike state in exclusive $B^{\pm} \to K^{\pm}\pi^+\pi^- J/\psi$ decays". In: *Phys. Rev. Lett.* **91** (2003), p. 262001. DOI: 10.1103/PhysRevLett.91.262001. arXiv: hep-ex/0309032 [hep-ex].

[41] R. Aaij et al. (LHCb). "Determination of the X(3872) meson quantum numbers". In: *Phys. Rev. Lett.* **110** (2013), p. 222001. DOI: 10.1103/PhysRevLett.110.222001. arXiv: 1302.6269 [hep-ex].

[42] B. Aubert et al. (BaBar). "Observation of a broad structure in the $\pi^+\pi^- J/\psi$ mass spectrum around 4.26 GeV/c^2". In: *Phys. Rev. Lett.* **95** (2005), p. 142001. DOI: 10.1103/PhysRevLett.95.142001. arXiv: hep-ex/0506081 [hep-ex].

[43] S. Choi et al. (Belle). "Observation of a resonancelike structure in the $\pi^{\pm}\psi'$ mass distribution in exclusive $B \to K\pi^{\pm}\psi'$ decays". In: *Phys. Rev. Lett.* **100** (2008), p. 142001. DOI: 10.1103/PhysRevLett.100.142001. arXiv: 0708.1790 [hep-ex].

[44] B. Aubert et al. (BaBar). "Search for the $Z(4430)^-$ at BABAR". In: *Phys. Rev.* **D79** (2009), p. 112001. DOI: 10.1103/PhysRevD.79.112001. arXiv: 0811.0564 [hep-ex].

[45] R. Aaij et al. (LHCb). "Observation of the resonant character of the $Z(4430)^-$ state". In: *Phys. Rev. Lett.* **112**.22 (2014), p. 222002. DOI: 10.1103/PhysRevLett.112.222002. arXiv: 1404.1903 [hep-ex].

[46] M. Ablikim et al. (BESIII). "Observation of a Charged Charmoniumlike Structure in $e^+e^- \to \pi^+\pi^-$ J/ψ at $\sqrt{s}$ =4.26 GeV". In: *Phys. Rev. Lett.* **110** (2013), p. 252001. DOI: 10.1103/PhysRevLett.110.252001. arXiv: 1303.5949 [hep-ex].

[47] Z. Q. Liu et al. (Belle). "Study of $e^+e^- \to \pi^+\pi^- J/\psi$ and Observation of a Charged Charmoniumlike State at Belle". In: *Phys. Rev. Lett.* **110** (2013), p. 252002. DOI: 10.1103/PhysRevLett.110.252002. arXiv: 1304.0121 [hep-ex].

[48] O. Eberhardt, G. Herbert, H. Lacker, et al. "Impact of a Higgs boson at a mass of 126 GeV on the standard model with three and four fermion generations". In: *Phys. Rev. Lett.* **109** (2012), p. 241802. DOI: 10.1103/PhysRevLett.109.241802. arXiv: 1209.1101 [hep-ph].

[49] G. Branco, P. Ferreira, L. Lavoura, et al. "Theory and phenomenology of two-Higgs-doublet models". In: *Phys. Rept.* **516** (2012), pp. 1–102. DOI: 10.1016/j.physrep.2012.02.002. arXiv: 1106.0034 [hep-ph].

[50] S. P. Martin. "A Supersymmetry primer". In: *Adv. Ser. Direct. High Energy Phys.* **21** (2010), pp. 1–153. DOI: 10.1142/9789814307505_0001. arXiv: hep-ph/9709356 [hep-ph].

[51] I. J. Aitchison. *Supersymmetry and the MSSM: An Elementary introduction.* 2005. arXiv: hep-ph/0505105 [hep-ph].

[52] L. Ewing. *Vectorized version of the Linux mascot, Tux.* 1996; vectorization 2006. URL: http://commons.wikimedia.org/wiki/File:Tux-simple.svg (visited on 06/03/2015).

[53] M. A. Shifman. *Foreword to ITEP lectures in particle physics.* 1995. arXiv: hep-ph/9510397 [hep-ph].

[54] M. Ciuchini, G. Degrassi, P. Gambino, et al. "Next-to-leading QCD corrections to $B \to X_s\gamma$ in supersymmetry". In: *Nucl. Phys.* **B534** (1998), pp. 3–20. DOI: 10.1016/S0550-3213(98)00516-1. arXiv: hep-ph/9806308 [hep-ph].

[55] K. Abe et al. (Belle). "A Measurement of the branching fraction for the inclusive $B \to X_s\gamma$ decays with the Belle detector". In: *Phys. Lett.* **B511** (2001), pp. 151–158. DOI: 10.1016/S0370-2693(01)00626-8. arXiv: hep-ex/0103042 [hep-ex].

[56] S. Fratina et al. (Belle). "Evidence for CP violation in $B^0 \to D^+D^-$ decays". In: *Phys. Rev. Lett.* **98** (2007), p. 221802. DOI: 10.1103/PhysRevLett.98.221802. arXiv: hep-ex/0702031 [HEP-EX].

[57] V. Khachatryan et al. (CMS, LHCb). "Observation of the rare $B_s^0 \longrightarrow \mu^+\mu^-$ decay from the combined analysis of CMS and LHCb data". In: *Nature* (2015). DOI: 10.1038/nature14474. arXiv: 1411.4413 [hep-ex].

[58] Z. Maki, M. Nakagawa, and S. Sakata. "Remarks on the unified model of elementary particles". In: *Prog. Theor. Phys.* **28** (1962), pp. 870–880. DOI: 10.1143/PTP.28.870.

[59] S. Petcov. "The Processes $\mu \longrightarrow e\gamma$, $\mu \longrightarrow ee\bar{e}$, $\nu' \longrightarrow \nu\gamma$ in the Weinberg-Salam Model with Neutrino Mixing". In: *Sov. J.Nucl. Phys.* **25** (1977), p. 340.

[60] A. Brignole and A. Rossi. "Anatomy and phenomenology of μ-τ lepton flavor violation in the MSSM". In: *Nucl. Phys.* **B701** (2004), pp. 3–53. DOI: 10.1016/j.nuclphysb.2004.08.037. arXiv: hep-ph/0404211 [hep-ph].

[61] K. Hayasaka et al. (Belle). "New search for $\tau \longrightarrow \mu\gamma$ and $\tau \longrightarrow e\gamma$ decays at Belle". In: *Phys. Lett.* **B666** (2008), pp. 16–22. DOI: 10.1016/j.physletb.2008.06.056. arXiv: 0705.0650 [hep-ex].

[62] T. Abe et al. (Belle II). *Belle II Technical Design Report*. 2010. arXiv: 1011.0352 [physics.ins-det].

[63] P. Raimondi. "New developments in Super B-factories". In: *Conf. Proc.* **C070625** (2007), p. 32.

[64] M. Friedl, T. Bergauer, P. Dolejschi, et al. "The Belle II Silicon Vertex Detector". In: *Phys. Procedia* **37** (2012), pp. 867–873. DOI: 10.1016/j.phpro.2012.02.428.

[65] N. Taniguchi, M. Ikeno, Y. Iwasaki, et al. "All-in-one readout electronics for the Belle-II Central Drift Chamber". In: *Nucl. Instrum. Meth.* **A732** (2013), pp. 540–542. DOI: 10.1016/j.nima.2013.06.096.

[66] J. Brodzicka et al. (Belle). "Physics Achievements from the Belle Experiment". In: *PTEP* **2012** (2012), p. 04D001. DOI: 10.1093/ptep/pts072. arXiv: 1212.5342 [hep-ex].

[67] K. Inami (Belle II PID Group). "TOP counter for particle identification at the Belle II experiment". In: *Nucl. Instrum. Meth.* **A766** (2014), pp. 5–8. DOI: 10.1016/j.nima.2014.07.006.

[68] M. Barrett (Belle II iTOP Group). "Particle identification with the iTOP detector at Belle-II". In: *Meeting of the APS Division of Particles and Fields (DPF 2013) Santa Cruz, California, USA, August 13-17, 2013*. 2013. arXiv: 1310.4542 [physics.ins-det].

[69] R. Pestotnik, I. Adachi, K. Hara, et al. "Aerogel RICH for forward PID at Belle II". In: *Nucl. Instrum. Meth.* **A732** (2013), pp. 371–374. DOI: 10.1016/j.nima.2013.06.080.

[70] Z. Dolezal, C. Kiesling, C. Lacasta, et al., eds. *The PXD Whitebook*. Version 0. 2012.

[71] J. Kemmer and G. Lutz. "New Detector Concepts". In: *Nucl. Instrum. Meth.* **A253** (1987), pp. 365–377. DOI: 10.1016/0168-9002(87)90518-3.

[72] R. H. Richter, L. Andricek, P. Fischer, et al. "Design and technology of DEPFET pixel sensors for linear collider applications". In: *Nucl. Instrum. Meth.* **A511** (2003), pp. 250–256. DOI: 10.1016/S0168-9002(03)01802-3.

[73] S. Rummel and L. Andricek (DEPFET). "The DEPFET active pixel sensor for vertexing at ILC and Super KEKB". In: *Nucl. Instrum. Meth.* **A623** (2010), pp. 189–191. DOI: 10.1016/j.nima.2010.02.191.

[74] H. Kruger (DEPFET). "Front-end electronics for DEPFET pixel detectors at SuperBelle (BELLE II)". In: *Nucl. Instrum. Meth.* **A617** (2010), pp. 337–341. DOI: 10.1016/j.nima.2009.10.042.

[75] M. Lemarenko, T. Hemperek, H. Krüger, et al. "Test results of the data handling processor for the DEPFET pixel vertex detector". In: *JINST* **8** (2013), p. C01032. DOI: 10.1088/1748-0221/8/01/C01032.

[76] J. Knopf, P. Fischer, C. Kreidl, et al. "A 256 channel 8-Bit current digitizer ASIC for the Belle-II PXD". In: *JINST* **6** (2011), p. C01085. DOI: 10.1088/1748-0221/6/01/C01085.

[77] M. Nakao. "Timing distribution for the Belle II data acquistion system". In: *JINST* **7** (2012), p. C01028. DOI: 10.1088/1748-0221/7/01/C01028.

[78] S. Suzuki, S. Yamada, R. Itoh, et al. "The Three-Level Event Building System for the Belle II Experiment". In: *IEEE Trans. Nucl. Sci.* **62**.3 (2015), pp. 1162–1168. DOI: 10.1109/TNS.2015.2422376.

[79] M. Nakao, T. Higuchi, R. Itoh, et al. "Data acquisition system for Belle II". In: *JINST* **5** (2010), p. C12004. DOI: 10.1088/1748-0221/5/12/C12004.

[80] T. Higuchi et al. "Development of a PCI based data acquisition platform for high intensity accelerator experiments". In: *eConf* **C0303241** (2003), TUGT004. arXiv: hep-ex/0305088 [hep-ex].

[81] R. Itoh, T. Higuchi, M. Nakao, et al. "Data flow and high level trigger of Belle II DAQ system". In: *IEEE Trans. Nucl. Sci.* **60**.5 (2013), pp. 3720–3724. DOI: 10.1109/TNS.2013.2273091.

[82] D. Levit, I. Konorov, and S. Paul. "FPGA based data read-out system of the Belle II pixel detector". In: *IEEE Trans. Nucl. Sci.* **62**.3 (2015), pp. 1033–1039. DOI: 10.1109/TNS.2015.2424713. arXiv: 1406.3864 [physics.ins-det].

[83] M. Schnell. "Development of an FPGA-based Data Reduction System for the Belle II DEPFET Pixel Detector". PhD thesis. Rheinische Friedrich-Wilhelms-Universität Bonn, 2015.

[84] B. Spruck, T. Geßler, W. Kühn, et al. "The Belle II Pixel Detector Data Acquisition and Reduction System". In: *IEEE Trans. Nucl. Sci.* **60**.5 (2013), pp. 3709–3713. DOI: 10.1109/TNS.2013.2281571.

[85] D. Münchow, J. Dingfelder, T. Gessler, et al. "The data acquisition system of the Belle II Pixel Detector". In: *JINST* **9** (2014), p. C08009. DOI: 10.1088/1748-0221/9/08/C08009.

[86] T. Geßler, W. Kühn, J. S. Lange, et al. "The ONSEN Data Reduction System for the Belle II Pixel Detector". In: *IEEE Trans. Nucl. Sci.* **62**.3 (2015), pp. 1149–1154. DOI: 10.1109/TNS.2015.2414713. arXiv: 1406.4028 [physics.ins-det].

[87] PICMG. *PICMG 3.0 R3.0, AdvancedTCA Base Specification*. 2008.

[88] PICMG. *AMC.0 R2.0, AdvancedMC Mezzanine Module Specification*. Version 2.0. 2006.

[89] PICMG. *MTCA.0 R1.0, MicroTCA Specification*. 2006.

[90] Schroff GmbH. *14-slot AdvancedTCA Shelf 11592-402, User's Manual*. 2006.

[91] Intel, Hewlett-Packard, NEC, et al. *Intelligent Platform Management Interface Specification v1.5*. Version 1.1. 2002.

[92] Schroff GmbH. *MicroTCA System 11850-013, User's Manual*. 2008.

[93] W. Kühn et al. (PANDA). "FPGA based compute nodes for high level triggering in PANDA". In: *J. Phys. Conf. Ser.* **119** (2008), p. 022027. DOI: 10.1088/1742-6596/119/2/022027.

[94] J. Zhao et al. "A general xTCA compliant and FPGA based data processing building blocks for trigger and data acquisition system". In: *Proceedings, 19th Real Time Conference (RT2014)*. 2014. DOI: 10.1109/RTC.2014.7097528.

[95] Xilinx, Inc. *DS100. Virtex-5 Family Overview*. Version 5.1. 2015. URL: http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf.

[96] Xilinx, Inc. *UG190. Virtex-5 FPGA User Guide*. Version 5.4. 2012. URL: http://www.xilinx.com/support/documentation/user_guides/ug190.pdf.

[97] Xilinx, Inc. *DS112. Virtex-4 Family Overview*. Version 3.1. 2010. URL: http://www.xilinx.com/support/documentation/data_sheets/ds112.pdf.

[98] Xilinx, Inc. *UG070. Virtex-4 FPGA User Guide*. Version 2.6. 2008. URL: http://www.xilinx.com/support/documentation/user_guides/ug070.pdf.

[99] Xilinx, Inc. *DS054. XC9500XL High-Performance CPLD Family Data Sheet*. Version 2.5. 2009. URL: http://www.xilinx.com/support/documentation/data_sheets/ds054.pdf.

[100] J. Lang. "Control and Management Unit for ATCA conform Trigger and Data Acquisition Processors in the PANDA Experiment". Diploma thesis. Justus-Liebig-Universität Gießen, 2008.

[101] T. Geßler. "An IPMI-Based Slow Control System for Data Acquisition in the PANDA Experiment". Master's thesis. Justus-Liebig-Universität Gießen, 2010.

[102] DESY. *MicroTCA.4 Module Management Controller Data Sheet*. 2014. URL: http://mtca.desy.de/sites/site_mtca/content/e174425/e199296/e213964/MMC_Datasheet_v1_eng.pdf.

[103] D. Münchow. "Development of the Online Data Reduction System and Feasibility Studies of 6-Layer Tracking for the Belle II Pixel Detector". PhD thesis. Justus-Liebig-Universität Gießen, 2015.

[104] Xilinx, Inc. *UG683. EDK Concepts, Tools, and Techniques*. Version 14.6. 2013. URL: http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/edk_ctt.pdf.

[105] IBM. *128-bit Processor Local Bus. Architecture Specifications*. Version 4.6. 2004. URL: http://www.xilinx.com/txpatches/pub/applications/misc/128bitplbbus_v4_6.zip.

[106] Xilinx, Inc. *SP006. LocalLink Interface Specification*. Version 2.0. 2005. URL: http://www.xilinx.com/aurora/aurora_member/sp006.pdf.

[107] Xilinx, Inc. *DS643. LogiCORE IP Multi-Port Memory Controller (MPMC) (v6.03.a) Product Specification*. Version 1.0. 2011. URL: http://www.xilinx.com/support/documentation/ip_documentation/mpmc.pdf.

[108] *Experimental Physics and Industrial Control System*. URL: http://www.aps.anl.gov/epics.

[109] Xilinx, Inc. *SP002. Aurora 8B/10B Protocol Specification*. Version 2.3. 2014. URL: http://www.xilinx.com/support/documentation/ip_documentation/aurora_8b10b_protocol_spec_sp002.pdf.

[110] T. Uchida. "Hardware-Based TCP Processor for Gigabit Ethernet". In: *IEEE Trans. Nucl. Sci.* **55**.3 (2008), pp. 1631–1637. DOI: 10.1109/TNS.2008.920264.

[111] J. Redd. "Calculating statistical confidence levels for error-probability estimates". In: *Lightwave Magazine* **21**.5 (2000), pp. 110–114. URL: http://www.lightwaveonline.com/articles/print/volume-17/issue-5/features/calculating-statistical-confidence-levels-for-error-probability-estimates-53462167.html.

[112] *Test Beams at DESY*. URL: http://testbeam.desy.de.

[113] I. Rubinskiy (EUDET, AIDA). "An EUDET/AIDA Pixel Beam Telescope for Detector Development". In: *Phys. Procedia* **37** (2012), pp. 923–931. DOI: 10.1016/j.phpro.2012.02.434.

[114] T. Bilka et al. "Demonstrator of the Belle II Online Tracking and Pixel Data Reduction on the High Level Trigger System". In: *IEEE Trans. Nucl. Sci.* **62**.3 (2015), pp. 1155–1161. DOI: 10.1109/TNS.2015.2419879. arXiv: 1406.4955 [physics.ins-det].

[115] Xilinx, Inc. *UG029. ChipScope Pro Software and Cores User Guide*. Version 14.3. 2012. URL: http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/chipscope_pro_sw_cores_ug029.pdf.

[116] Xilinx, Inc. *UG111. Embedded System Tools Reference Manual*. Version 14.6. 2013. URL: http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/est_rm.pdf.

[117] Xilinx, Inc. *UG642. Platform Specification Format Reference Manual*. Version 14.1. 2012. URL: http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/psf_rm.pdf.

[118] Xilinx, Inc. *DS561. PLBv46 Slave Single (v1.01a) Product Specification*. Version 1.3. 2010. URL: http://www.xilinx.com/support/documentation/ip_documentation/plbv46_slave_single.pdf.

[119] Xilinx, Inc. *DS516. LogiCORE IP Interrupt Control (v2.01a) Product Specification*. Version 3.1. 2012. URL: http://www.xilinx.com/support/documentation/ip_documentation/interrupt_control.pdf.

[120] Xilinx, Inc. *UG198. Virtex-5 FPGA RocketIO GTX Transceiver User Guide*. Version 3.0. 2009. URL: http://www.xilinx.com/support/documentation/user_guides/ug198.pdf.

[121] A. X. Widmer and P. A. Franaszek. "A DC-balanced, partitioned-block, 8B/10B transmission code". In: *IBM J. Res. Dev.* **27**.5 (1983), pp. 440–451. DOI: 10.1147/rd.275.0440.

[122] Xilinx, Inc. *UG353. LogiCORE IP Aurora 8B/10B v5.3 User Guide*. Version 5.3. 2012. URL: http://www.xilinx.com/support/documentation/ip_documentation/aurora_8b10b/v5_3/aurora_8b10b_ug353.pdf.

[123] Uchida, T. *SiTCP Manual*. Version 1.4. 2012.

[124] *SiTCP User Community*. URL: http://sitcp.bbtech.co.jp.

[125] Atmel Corporation. *AT93C46D*. Data sheet. 2015. URL: http://www.atmel.com/Images/Atmel-5193-SEEPROM-AT93C46D-Datasheet.pdf.

[126] Xilinx, Inc. *PG047. LogiCORE IP Ethernet 1000BASE-X PCS/PMA or SGMII v11.5 Product Guide*. Version 1.2. 2012. URL: http://www.xilinx.com/support/documentation/ip_documentation/gig_eth_pcs_pma/v11_5/pg047-gig-eth-pcs-pma.pdf.

[127] B. Spruck. *ONSEN (kind of documentation)*. Internal document. 2015.

[128] B. Spruck. *Dokumentation der IP Cores*. Internal document. 2015.

[129] D. Levit. *DHE/DHC Data Formats*. Internal document. Version 0.4.38. 2015.

[130] Numonyx. *StrataFlash Embedded Memory (P33)*. Data sheet. 2007. URL: http://www.micron.com/~/media/documents/products/data-sheet/nor-flash/parallel/p33/314749_p33_130nm_discrete_ds.pdf.

[131] Marvell Technology Group, Ltd. *88E1111*. Data sheet. 2013. URL: http://origin-www.marvell.com/transceivers/assets/Marvell-Alaska-Ultra-88E1111-GbE.pdf.

[132] Finisar Corporation. *FCLF-8520/8521-3*. Data sheet. 2014. URL: http://www.finisar.com/sites/default/files/downloads/fclf-8520-3_fclf-8521-3_1000base-t_copper_sfp_transceiver_spec_revd.pdf.

[133] Finisar Corporation. *FTLF8528P3BxV*. Data sheet. 2011. URL: http://www.finisar.com/sites/default/files/downloads/ftlf8528p3bxy_8.5gbs_short-wavelength_sfp-plus_transceiver_spec_reva.3.pdf.

[134] Xilinx, Inc. *DS123. Platform Flash In-System Programmable Configuration PROMs*. Version 2.18. 2010. URL: http://www.xilinx.com/support/documentation/data_sheets/ds123.pdf.

[135] Xilinx, Inc. *UG191. Virtex-5 FPGA Configuration User Guide*. Version 3.11. 2012. URL: http://www.xilinx.com/support/documentation/user_guides/ug191.pdf.

[136] Maxim Integrated. *MAX1236–MAX1239*. Data sheet. 2010. URL: http://datasheets.maximintegrated.com/en/ds/MAX1236-MAX1239.pdf.

[137] Maxim Integrated. *MAX1617A*. Data sheet. 2012. URL: http://datasheets.maximintegrated.com/en/ds/MAX1617A.pdf.

[138] Maxim Integrated. *MAX6625/MAX6626*. Data sheet. 2014. URL: http://datasheets.maximintegrated.com/en/ds/MAX6625-MAX6626.pdf.

[139] Analog Devices Inc. *AD8210*. Data sheet. 2013. URL: http://www.analog.com/media/en/technical-documentation/data-sheets/AD8210.pdf.