

Dissertation

submitted to the

Combined Faculties for the Natural Sciences and for Mathematics
of the Ruperto-Carola University of Heidelberg, Germany

for the degree of

Doctor of Natural Sciences

Presented by

Dipl.-Inf. Jochen Knopf

born in Mannheim, Germany

Oral examination: November 9th, 2011

Development, Characterization and Operation of the DCDB,
the Front-End Readout Chip for the Pixel Vertex Detector
of the Future BELLE-II Experiment

Referees: Prof. Dr. Peter Fischer
Prof. Dr. Johanna Stachel

Zusammenfassung:

Entwicklung, Charakterisierung und Betrieb des DCDB, dem Auslesechip des Pixel Vertex Detektors (PXD) im geplanten BELLE-II Experiment

Der BELLE-II Detektor ist eine Weiterentwicklung des BELLE Detektors am Forschungszentrum KEK im japanischen Tsukuba. Mit Letzterem konnte in der Vergangenheit die Existenz CP-verletzender Zerfälle erfolgreich nachgewiesen werden. Der ebenfalls weiterentwickelte Teilchenbeschleuniger SuperKEKB erzeugt eine Luminosität von $8 \times 10^{35} \text{ cm}^{-2} \text{ s}^{-1}$. Die damit einhergehenden Sekundärereignisse führen zu einer erheblichen Auslastung des Detektors, insbesondere des innersten Pixel Vertex Detektors (PXD). Um die geforderten physikalischen Leistungsmerkmale in diesem Umfeld erfüllen zu können, müssen höchste Anforderungen an die jeweilige Ausleseelektronik gestellt werden.

Das PXD Pixel Detektor System basiert auf der sogenannten DEPFET Technologie. DEPFET Transistoren vereinen die Teilchendetektion und die Verstärkung des resultierenden Signals in einem Element. Der DCDB Chip wurde entwickelt, um die Signale dieser Transistoren den Bedingungen im BELLE-II Detektor entsprechend zu messen und zu digitalisieren. Die vorliegende Ausarbeitung beschreibt die Fähigkeiten dieses Chips sowie dessen Implementierungsprozess. Mit Hilfe eines eigens entwickelten Testaufbaus wurde der DCDB umfassend charakterisiert. Die entsprechenden Ergebnisse werden hier dargelegt. Die Einsetzbarkeit dieses Chips in einem teilchenphysikalischen Messinstrument wird anhand eines DEPFET Detektor Prototyp Systems eindrucksvoll unter Beweis gestellt. Die Höhepunkte sind die Messung eines Zerfallsspektrums von Cd-109 sowie die erfolgreiche Durchführung eines Teilchenstrahlexperiments am CERN.

Abstract:

Development, Characterization and Operation of the DCDB, the Front-End Readout Chip for the Pixel Vertex Detector of the Future BELLE-II Experiment

The BELLE-II detector is the upgrade of its predecessor named BELLE at KEK research centre in Tsukuba, Japan, which was successfully used in the past to find evidence for CP violating decays. The upgraded SuperKEKB accelerator is specified to produce a luminosity of $8 \times 10^{35} \text{ cm}^{-2} \text{ s}^{-1}$. Consequently, the BELLE-II detector and particularly the innermost pixel vertex detector (PXD) suffers from enormous occupancy due to background events. Coping with this harsh environment while providing the required physics performance results in tough specifications for the front-end readout electronics.

The PXD pixel detector system is based on the DEPFET technology. DEPFET transistors combine particle detection and signal amplification within one device. The DCDB chip is developed to sample and digitize signals from these transistors while complying with the specifications of BELLE-II. The presented work illustrates the chip's features and describes its implementation process. The device is comprehensively characterized using an individually developed test environment. The obtained results are presented. The DCDB's ability to serve as a readout device for particle physics applications is demonstrated by its successful operation within a DEPFET detector prototype system. Highlights are a decay spectrum measurement using Cd-109 and the successful operation in a beam test experiment at CERN.

Table of Contents

CHAPTER 1	<i>Introduction</i>	<i>1</i>
1.1	Motivation.....	2
1.2	The Challenge of SuperKEKB and BELLE-II	3
1.3	BELLE-II Experiment Overview.....	4
1.3.1	The SuperKEKB B-Factory	4
1.3.2	The Study Subjects.....	4
1.3.3	The BELLE-II Detector System.....	5
1.4	Focus of the Presented Work	7
CHAPTER 2	<i>The PXD Vertex Detector for BELLE-II</i>	<i>9</i>
2.1	The DEPFET Pixel Detector	10
2.1.1	The DEPFET Principle	10
2.1.2	The History of DEPFET Detectors	11
2.1.3	Reading DEPFET Pixel Matrices	12
2.2	The PXD Detector System for BELLE-II	16
2.2.1	Impact Parameter Resolution	16
2.2.2	Occupancy.....	16
2.2.3	Layer Radii.....	17
2.2.4	Frame Readout Time.....	17
2.2.5	Modules and Dimensions.....	17
2.2.6	Pixel Geometries	19
2.2.7	Detector Thinning	20
2.2.8	Front-End Readout System	21
2.2.9	Bump Bond Interconnection Technology	23
2.2.10	Higher Level Readout System	24
2.3	The SwitcherB Steering ASIC.....	25
2.3.1	Overview	25
2.3.2	Channel Boosting.....	26
2.3.3	Overlapping Gates.....	28
2.3.4	Operation Mode Details	29
2.4	DCDB - The Drain Current Digitizer for BELLE-II	31
2.5	DHP - The Data Handling Processor	33
CHAPTER 3	<i>The Analog Domain of the DCDB</i>	<i>37</i>
3.1	Overview	38
3.2	The Analog-To-Digital Conversion Principle.....	38
3.3	The Cyclic ADC Realization.....	40
3.4	Details of the Building Blocks.....	41

3.4.1	The Current Receiver	41
3.4.2	The Current Memory Cell	43
3.4.3	The Comparator	45
3.4.4	Pre-Sampling cell	46
3.4.5	Calibration Circuit	46
3.4.6	Offset Current Compensation	47
3.5	Configuration Summary	48
CHAPTER 4	<i>The Digital Domain of the DCDB</i>	51
4.1	General Considerations	52
4.1.1	Digital Tasks	52
4.1.2	From Full Custom to Synthesized Digital Logic	53
4.1.3	Revision History	53
4.2	Logic Development	54
4.2.1	Data Format Conversion	54
4.2.2	Output Serialization	55
4.2.3	Input Value Distribution for the Dynamic Offset Compensation	56
4.2.4	ADC Control Sequence Generation	56
4.2.5	Clocking and Resetting Scheme	57
4.2.6	JTAG Configuration and Debugging Interface	57
4.2.7	Digital Test Signal Injection	58
4.3	Verification	58
4.3.1	Digital-Only Functional Verification	59
4.3.2	Mixed-Mode Simulation	62
4.3.3	Concluding Remark	64
4.4	Standard Cell Library Development	64
4.4.1	Radiation Hard Standard Cell Library: First Approach	66
4.4.2	Standard Cell Library: Second Approach	67
4.4.3	The Standard Cell Library for the DCDB	69
4.5	Physical Implementation	70
4.5.1	Constraining the Design	71
4.5.2	Encounter: Standard vs. MMMC Flow	72
4.5.3	Synthesis	73
4.5.4	Floorplanning and Placement	75
4.5.5	Power Planning	76
4.5.6	Clock Tree Synthesis	77
4.5.7	Signal Routing	79
4.5.8	Timing Analysis	80
4.5.9	Finalizing the Design	81
4.5.10	Tape-Out and Design Transfer to the Virtuoso ADE	83
CHAPTER 5	<i>The DCDB Test Environment</i>	85
5.1	The Hardware Components	86
5.1.1	Electrically Interfacing the DCDB	86
5.1.2	The DCDB Test Board	87
5.1.3	General Purpose FPGA Board	88
5.1.4	The DCDB Test Environment Hardware Setup	90
5.2	The DCDB Test Firmware	90
5.2.1	Complexity Distribution: Software vs. Hardware	90
5.2.2	The Conceptual Structure	91

5.2.3	DCDB Communication and Data Processing Issues	92
5.3	The DCDB Test Software	93
CHAPTER 6	<i>The DCDB-based Detector Prototype</i>	95
6.1	The Hardware Platform.....	96
6.1.1	DEPFET Detector and Switcher Chip Selection.....	96
6.1.2	FPGA-based Controlling and Readout System.....	97
6.1.3	The Hybrid Board	97
6.1.4	Powering Scheme.....	99
6.2	The Readout Firmware.....	99
6.2.1	Overview	99
6.2.2	Trigger Processing	99
6.2.3	Trailing Frames	101
6.2.4	SwitcherB Controller	101
6.3	The Data Acquisition Software	102
CHAPTER 7	<i>DCDB Characterization</i>	103
7.1	Digital Functionality Checks.....	104
7.1.1	Power Consumption of the Digital Block	104
7.1.2	Clock Insertion Delay	105
7.1.3	Digital Test Signal Injection	106
7.1.4	Maximum Operation Speed	108
7.2	Detailed Analog Channel Measurements	109
7.2.1	The Current Memory Cell.....	109
7.2.2	The Analog-to-Digital Converter.....	113
7.2.3	The Analog-to-Digital Converter: Stability Measurements.....	115
7.2.4	The Analog-to-Digital Converter: Dynamic Behaviour	117
7.2.5	The Transimpedance Amplifier: DC Measurements	118
7.2.6	The Transimpedance Amplifier: Dynamic Measurements	120
7.2.7	Overall Channel Performance	127
7.2.8	Power Consumption of the Analog Channels	129
7.3	Multi-Channel Measurements.....	129
7.3.1	Bad ADCs	130
7.3.2	Offset Analysis.....	131
7.3.3	Gain Analysis	131
7.3.4	Noise Analysis	133
7.3.5	Integral Non-Linearity Analysis	133
7.4	Conclusions	134
CHAPTER 8	<i>The Detector Prototype Operation</i>	137
8.1	System Simulation.....	138
8.1.1	Motivation.....	138
8.1.2	Simulation Setup	138
8.1.3	Simulation Result	140
8.1.4	Concluding Remarks.....	140
8.2	Reducing Pedestal Fluctuations.....	141
8.2.1	Theoretical Benefit.....	141

8.2.2	Optimization Algorithm.....	143
8.2.3	Measurement Results	144
8.3	Detector Operations	144
8.3.1	First Imaging Measurement	144
8.3.2	Radioactive Source Measurement.....	145
8.3.3	Clear Efficiency Studies.....	147
8.3.4	Beam Test Period at CERN	147
CHAPTER 9	<i>Conclusion & Outlook</i>	149
9.1	Conclusion	150
9.2	Summary of Own Contributions	151
9.3	Outlook	152
	 <i>Bibliography</i>	 153
	 <i>Acknowledgements</i>	 159

Introduction

Abstract:

This introduction defines the presented work's relevance in the scientific world of experimental particle physics.

After a coarse description of particle physics experiments searching for evidence of symmetry violating processes in nature, the future BELLE-II experiment is brought into focus. In particular, the DEPFET vertex pixel detector sub-system of BELLE-II is of outstanding importance: within the DEPFET collaboration, this thesis emerged from the development of major parts of the readout electronic devices for this detector.

1.1 Motivation

In modern physics, studying symmetries in physical transformations has led to a deep understanding of natural phenomena. Searching, finding and understanding the preservation or violation of symmetries turned out to be a very powerful tool in discovering the secrets of nature. Some symmetries are quite obvious, others are very hard to find. Physicists have been making huge efforts for both, their theoretical description and their experimental verification, historically, often with surprising results.

The following three fundamental symmetries, as well as the combination of those, have been of very strong interest:

- Charge conjugation C : transformation of a particle into its antiparticle
- Parity P : transformation of the location $r \rightarrow -r$
- Time reversal T : transformation of the time $t \rightarrow -t$

Very general considerations about the principles of symmetries lead to the fact that there is no consistent theory which allows the violation of the combined symmetry $C \cdot P \cdot T$ in any transformation. However, there are transformations that violate a single one or a combination of two of those. A very important example is the combined symmetry of charge conjugation and parity $C \cdot P$. For a long time, physicists assumed that any system would be invariant under the $C \cdot P$ symmetry. But in 1964, James Cronin, Val Fitch et al. found a violation of this symmetry in a kaon decay experiment [1]. This discovery not only led to the Nobel Prize of Physics in 1980, but also to an enormous excitement in the physics community. At that time, none of the present theories could explain this observation. The first theoretical description of $C \cdot P$ violating transformations was proposed in 1972 by Makoto Kobayashi and Toshihide Maskawa [2]. Their model extended the quantum field theory, so that finally the experiment of Cronin and Fitch could be explained.

Five years later, in 1977, S. W. Herb et al. [3] found evidence for the existence of the b-quark and therefore B-mesons. The Kobayashi-Maskawa model predicted that neutral B-mesons would have the same $C \cdot P$ properties as kaons, but with a significantly higher cross section. This fact gave the chance for an experimental verification of this theory and therefore, in the following years B-factories and appropriate detector systems were developed. The most important ones were the BABAR experiment at SLAC at Stanford, USA, and the BELLE experiment at KEK at Tsukuba, Japan. The independent measurements of the two collaborations confirmed the model with impressive agreement [4]. Nowadays, the Kobayashi-Maskawa model and hence the $C \cdot P$ violation is well established in the Standard Model.

However, there must be more to discover! It was back in 1964 when Andrei Sakharov claimed that a $C \cdot P$ violation must be the reason for the inequality of matter and antimatter in the universe [5]. The effect of $C \cdot P$ violation described by the Kobayashi-Maskawa model, however, is not strong enough [6]. There must be other, yet undiscovered sources of $C \cdot P$ violations in nature. Obviously, the state-of-the-art high energy physics experiments are either not precise enough, or simply do not reach the necessary energy level. Current developments of future particle accelerators and detectors are aiming to cope with both insufficiencies. The Large Hadron Collider at CERN with its experiments ATLAS and CMS try to push the energy frontier up to a

centre-of-mass energy of 14TeV . Others, like the LHCb (also LHC, CERN) for example, are designed for enhancing the precision of the measurements and thereby focusing on rare events. The precision measurement experiment SuperB at INFN in Italy is currently being planned.

Another representative of the latter group will be the BELLE-II experiment at the planned SuperKEKB accelerator at KEK in Japan. The BELLE-II experiment is currently under development and is going to be an upgrade of the predecessor experiment named BELLE. The new high precision detector system is intended to reveal unknown deviations from the Standard Model in strongly suppressed processes.

The development of the BELLE-II experiment started in December 2008 by forming the BELLE-II collaboration. As one of its members, the DEPFET collaboration is going to provide the inner layer vertex sub-detector (PXD) system. The presented dissertation emerged from the design and the development of the front-end readout electronics for this sub-detector system.

1.2 The Challenge of SuperKEKB and BELLE-II

The success of BELLE was not only the excellent experimental verification of the $C \cdot P$ violation that is predicted by the Kobayashi-Maskawa model. Beyond that, it also indicated, that there would be more to find. Indeed, results exist that are - conservatively speaking - hard to explain with the present understanding of nature. Numerous examples are given in [7]. They may hint to deviations from the Standard Model. In order to prove this it is necessary to have a closer look. Technically, this means to produce the physically interesting and mostly very rare events with a significantly higher rate than before, because in some cases actual measurement errors are still limited by experimental statistics. Increasing the relevant data set may eventually allow for well-grounded statements of New Physics. If so, mankind would have proceeded to identify yet unknown structures beyond those currently explained by the Standard Model. The key is to increase the overall number of produced and captured interactions enormously. This goal is addressed by BELLE-II and SuperKEKB.

From the accelerator's point of view, this requirement translates into an increase of luminosity. Since the luminosity is proportional to the beam current and inversely proportional to the beam's cross section, there are mainly two ways to go. The first one is to increase the number of particles in the beam. The second one is to narrow the beam by applying better focusing. Both options have been discussed by the SuperKEKB accelerator designers. Currently, a compromise with focus on the latter one, the so called *nano-beam* option, is the favourite [9].

For the detector system, however, the increased luminosity of the accelerator is fairly challenging, since there is not only an increase in those events that are intended to be studied. There is also a significant increase in second order events, the so called *background*. Particles produced by background events hit the detector just like those produced by the main events do, with considerable effect on the various sub-detectors. Depending on the type of sub-detector, they are causing at least higher occupancy and therefore a higher load on the readout devices, or, in worst case, even higher dead time and performance degradation.

Four main sources for background effects have been identified in the previous KEKB/BELLE setup [8]: beam gas scattering, Touschek scattering, synchrotron radiation and radiative Bhabha scattering. A lot of simulation effort has been put in already in order to find out how to extrapolate from these individual background contribution towards a realistic estimation of the background levels for BELLE-II and there are still some unanswered questions about it. Nevertheless, a conservative pre-estimation results in an increase of background hits in the detector by a factor of 20, while the rate of main events will increase by a factor of 50 [9]. Due to this harsh environment, the primary requirement for the BELLE-II detector system will be to at least maintain the performance of BELLE while coping the higher background.

1.3 BELLE-II Experiment Overview

1.3.1 The SuperKEKB B-Factory

Like the predecessor KEKB, the SuperKEKB is an asymmetric positron-electron collider. Electrons and positrons are accelerated in two separate rings and there is a single crossing point, the so called *Interaction Region (IR)*, where the two beams collide.

As already mentioned in section 1.1, the main purpose of the SuperKEKB is to produce B-mesons. First of all, using an electron-positron collider is a quite reasonable choice, since the acceleration of those particles to a certain energy level can be done with very high accuracy. This is necessary in order to exactly find the $\Upsilon(4S)$ resonance in the centre-of-mass energy (10.58GeV), at which the B-meson pair production (mass: $m_{B^\pm} = 5279.15 \pm 0.31\text{MeV}/c^2$ $m_{B^0} = 5279.53 \pm 0.33\text{MeV}/c^2$ [13]) is very likely. Secondly, using asymmetrically accelerated particles for the collisions is very useful, because in that case the centre-of-mass is moving relatively to the detector (Lorentz boost). Therefore, the various decays in the B-meson decay chain are locally separated. The decay products are detected at different space points which simplifies the event reconstruction.

The current design of the SuperKEKB plans that electrons are accelerated in the high energy ring (HER) to $7\text{GeV}/c$ and positrons are accelerated in the low energy ring (LER) to $4\text{GeV}/c$. The crossing angle of the two beams is 92mrad and the target luminosity at the point of interaction is calculated to be $8 \times 10^{35}\text{cm}^{-2}\text{s}^{-1}$, which is an improvement by a factor of almost 38 compared to KEKB. A full set of machine parameters can be found in [9].

1.3.2 The Study Subjects

Once the B-meson are created via the $\Upsilon(4S)$ resonance, they decay into lighter particles after a very short period of time. Measuring these decay vertices, in terms of quantity and position for example, is the key to determine their inherent $C \cdot P$ violation parameters.

A very prominent example for such a decay is illustrated in figure 1-1. Here, a pair of neutral B-mesons ($B^0\bar{B}^0$) is created, where the one decays into a D and a charged lepton (tag side), while the other decays basically into a J/Ψ (CP side). A speciality of this

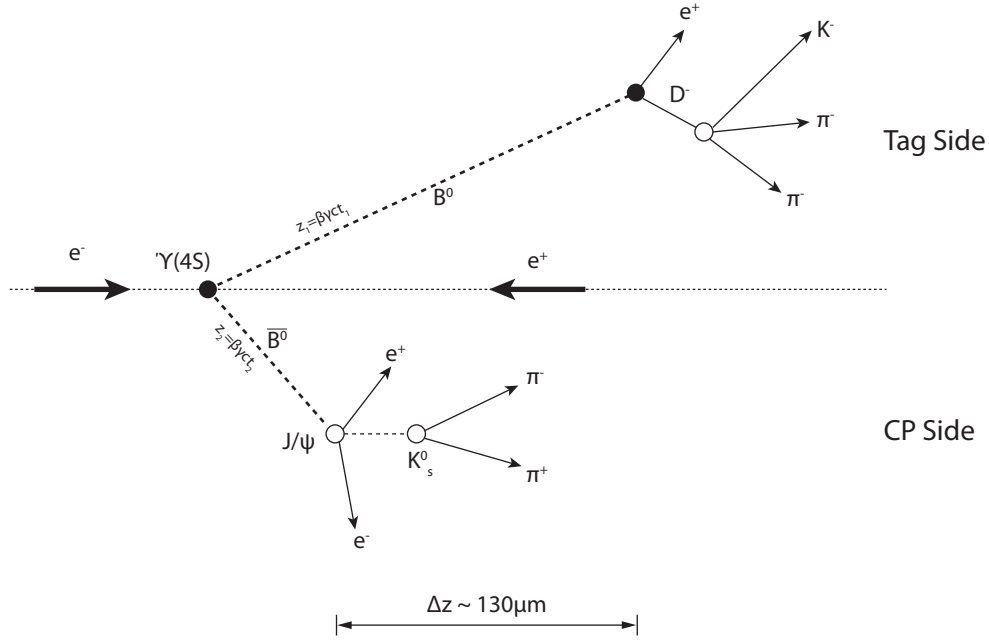


Figure 1-1 Exemplary decay mode of neutral B-mesons to be studied with the BELLE-II detector [10].

decay mode is that both mesons, B^0 and \bar{B}^0 , can go both ways and the only chance to clarify the situation is by determining the charge of the lepton on the tag side [11]. The $C \cdot P$ violation parameters in this case are obtained from the time difference $\Delta t = t_1 - t_2$ of these decays, which is typically in the order of a picosecond. As explained above, the measuring process is simplified by using the trick of an asymmetric collider. For the given machine parameters, the created particles are boosted by a factor of $\beta\gamma = 0.283$, which translates into a spatial separation of about $85 \mu\text{m}$ per picosecond. Thus, the time difference is actually measured via the spatial separation of the vertices, which are reconstructed using the signals introduced by further decay products in the surrounding detector.

1.3.3 The BELLE-II Detector System

The BELLE-II detector system is located around the Interaction Region. It is built up using different types of sub-detectors, arranged in the typical barrel fashion. A superconducting solenoid with an inner radius of 1.7m is providing a magnetic field of 1.5T in order to allow for particle identification. A corresponding schematic drawing is presented in figure 1-2.

The innermost detector sub-system is a vertex detector. It provides an excellent spatial resolution and therefore allows for the reconstruction of B-meson decay vertices. Moreover, high precision vertex reconstruction has been proven to be a powerful tool to identify and reject background events [12]. The vertex detector is built up using six layers of solid state detectors. The four outer layers are double-sided silicon strip detectors (SVD), while the two inner ones are pixel detectors based on the DEPFET principle (PXD). A lot more details about the PXD in particular can be found throughout

this work, since it emerged from the development of the readout electronics for this detector.

The vertex detector is surrounded by the central drift chamber (CDC). It is not only used for the reconstruction of charged tracks and precise momenta measurements, but also as a trigger source. Additionally, it can be used as particle identification device for low momentum tracks that lose all their energy within the chamber's gas volume.

Outside of the central drift chamber, a particle identification system is located (PID). Its main capability is the separation of kaons and pions at a nominal energy of $4\text{GeV}/c$ [14]. To this end, it uses quartz radiators that allow for the emittance of photons by crossing particles due to the Cherenkov effect. Particle information is then derived from time and location of the photon detection. In addition, a particle identification system based on the ring-imaging Cherenkov detector principle is placed into the forward end cap.

The PID is enclosed by an array of electromagnetic calorimeter elements for precise energy measurements. They are mounted in a barrel shape around the beam pipe. This barrel is closed by more of those elements in both end caps. The calorimeter elements themselves are built up using a tower structure of CsI scintillator crystals. Each of them will be tilted individually in order to project directly to the interaction region.

The outermost detector sub-systems is a K_L and muon detector, located outside of the superconducting solenoid. This detector is built up using a sandwich structure. There are arrays of massive iron plates as stopping material with glass-electrode resistive plate chambers in-between.

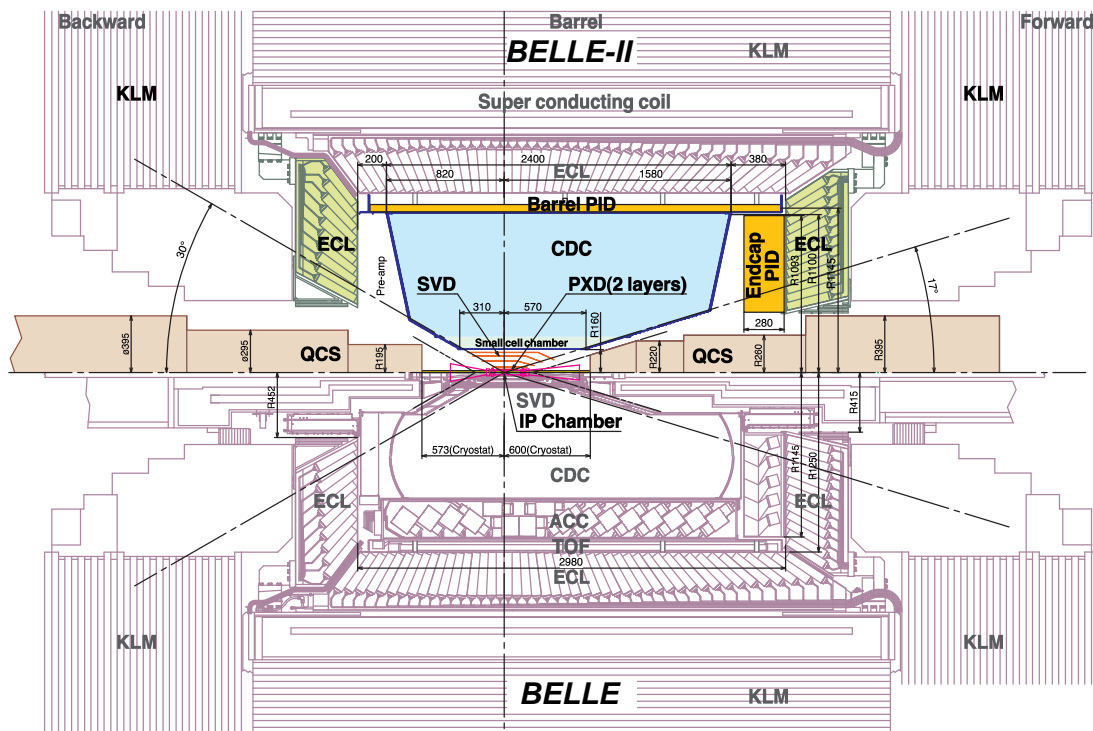


Figure 1-2 Schematic of the BELLE-II detector system (upper half) in comparison to its predecessor BELLE (lower half) [9].

1.4 Focus of the Presented Work

The main focus of this work is placed on the electronic devices of the front-end readout chain for the pixel vertex detector (PXD) of BELLE-II.

Chapter 2 begins with a presentation of the DEPFET transistor, the actual detection device of the PXD. Afterwards, the technical specification parameters of the detector and its readout devices are derived from the physics aspects in the BELLE-II experiment. The front-end readout ASICs¹ of the detector system are introduced. One of these, the *Drain Current Digitizer for BELLE-II* (DCDB), is the major subject of the presented work. It is used to convert the analog electrical signals of the detector into digital data.

The DCDB is developed in a team effort by Dr. Ivan Peric and Jochen Knopf (the author) at the Chair of Circuit Design, Heidelberg University. The chip's analog domain, developed by Dr. Ivan Peric, is described in chapter 3. Its digital domain is the contribution of Jochen Knopf and is comprehensively explained in chapter 4.

Beyond the participation in the development of the DCDB, its testing, characterization and operation are further major contributions of the author and are therefore outlined in this work. The chapters 5 and 6 describe the development of a chip testing setup and a DCDB-based DEPFET prototype system respectively. Using these environments, the DCDB is characterized and the results are presented in chapter 7. Finally, chapter 8 presents evidence for the DCDB's successful operation together with a DEPFET detector. The highlights are results from a radioactive source measurement as well as the system's operation in a beam test at CERN.

Since the DCDB is a crucial element in the readout chain of the PXD detector for BELLE-II, its proper functionality and the achievement of major quality parameters are of outstanding importance for the entire project. In that context, this work aims for both, providing the chip itself and proving its adequateness for the target job. Thus, it is of great relevance for both, the DEPFET and the BELLE-II collaborations.

1. Application-Specific Integrated Circuit

The PXD Vertex Detector for BELLE-II

Abstract:

The second chapter of this thesis focuses on the inner layer vertex detector setup for BELLE-II. Starting with an explanation of the DEPFET transistor principle, the assembly of the detector's half ladder building block is presented. Major specifications for both, detector and readout electronics, are derived from physics aspects of the BELLE-II experiment. Finally, the readout ASICs used are introduced. These are the SwitcherB, the DCDB and the DHP, which were developed exclusively for this project.

2.1 The DEPFET Pixel Detector

2.1.1 The DEPFET Principle

The DEPFET¹ principle describes the combination of radiation detection and signal amplification in a single transistor. A MOSFET transistor (JFET is also possible) is integrated onto a n-doped silicon detector substrate. Figure 2-1 provides an example schematic drawing of such a device. The substrate is fully depleted by means of sidewall depletion, a technique that is well known from silicon drift chambers. While the substrate is kept at a constant potential, negative voltages (relative to the substrate potential) are applied to the top and the bottom of the substrate using p^+ contacts, in order to deplete it from both sides. Once the voltages are low enough, the substrate is fully depleted while still having a tiny horizontal layer of „high“ bulk potential where the two depletion volumes meet. By relatively varying these voltages, the depth of the bulk potential layer inside the substrate can be influenced. For the DEPFET operation, it is shifted right below the top surface of the detector substrate, that is where the transistor is located. Additionally, at the same depth as the bulk potential layer, a n^+ -doped region is implanted into the substrate just underneath the gate of the transistor. Since this n^+ -doped region is depleted as well, only the atomic cores are left and therefore form the potential minimum for electrons inside the substrate.

Once such a device is hit by a particle with sufficient energy, electron-hole pairs are created. Due to the electric field inside the substrate caused by the depletion voltages, the

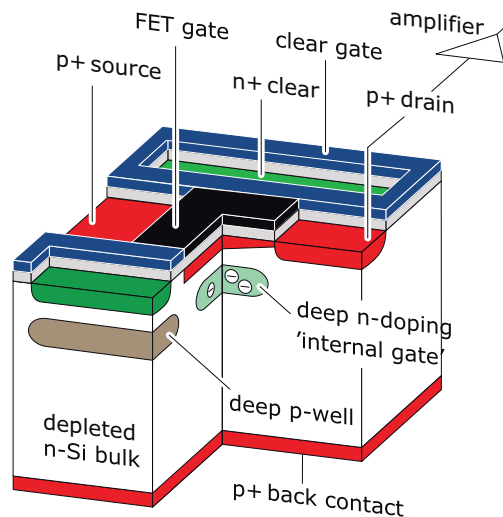


Figure 2-1 Schematic drawing of a DEPFET transistor [9]. The transistor structure is integrated onto the detector substrate. A deep n^+ -doped region underneath the FET gate is acting as the potential minimum for electrons. Signal electrons get trapped there and modulate the transistor current. There are clear contacts and clear gates on both sides of the transistor structure. Deep p-doped regions underneath the clear contacts prevent signal electrons from getting attracted by the clear contact rather than the internal gate.

1. DEpleted P-channel Field Effect Transistor

free electrons and the holes drift apart and hence cannot recombine. While the holes drift towards the depletion voltage contacts, where they can recombine, the free electrons accumulate at the potential minimum underneath the transistor gate. Once there is a p-channel established between the source and the drain contact of the transistor by applying an appropriate voltage to its gate, the p-channel is modulated also by the electric field of the electrons residing in the potential minimum. In other words, the potential minimum is acting as an „internal gate“. That means first of all, electrons created by incidental particles are measurable due to their influence on the p-channel of the transistor and therefore on the current flowing through it. Secondly, because of the transistor effect, the signal is amplified simultaneously. Thirdly, the signal electrons are measured indirectly, so the measurement is neither destructive, nor is there any charge transfer necessary. Fourthly, the fully depleted substrate leads to a very low input capacitance, so the measurement can be performed with very low noise, even at room temperature.

Having a non-destructive measurement scheme, however, requires for some kind of mechanism to take signal electrons away from the internal gate once the measurement is finished. Therefore, another n-contact, the *clear* contact, is introduced to the substrate right next to the transistor. By applying a sufficiently high positive voltage to that contact, a punch-through is established, which removes the electrons from the internal gate. The only drawback of this solution is that special care is necessary in order to prevent signal electrons from drifting to the clear contact rather than to the internal gate, since this would cause signal loss. The approved way to cope with this issue is to shield the n-doped region of the clear contact by a deep p-doped region underneath. Additionally, another gate structure is introduced, the so-called *clear gate*, which is located just above the gap between clear contact and internal gate. By means of the clear-gate, the electric fields inside the substrate can be influenced and therefore, the optimal working conditions can be adjusted.

2.1.2 The History of DEPFET Detectors

It was back in the late 1980s, when the two scientists J. Kemmer (Technische Universität München) and G. Lutz (Max-Planck-Institut für Physik und Astrophysik, München) were engaged in studying the innovative potential of silicon drift chambers. With the microelectronic technology present at that time, they found a way to combine the detection of radiation and the amplification of the induced electronic signal not only on the same chip, but also within a single transistor. Their idea was published in 1988 [15], the birth of the DEPFET detectors. The success of an experimental proof of principle was reported in 1990 [16].

During the following years, the idea was improved continuously. In co-operation with the Universities of Bonn and Dortmund, suitable electronics for steering DEPFET transistor prototypes as well as measuring their signal were developed. In 1997, the success of single pixel measurements [17] justified the development and production of larger arrays of DEPFET transistors. Finally, three years later, P. Fischer published a paper on the first successful operation of a 32×32 pixel imaging matrix based on DEPFET transistors using 60keV gamma rays and an IR laser [18].

Since that time, the DEPFET technology was ready for real use. Prototype systems for a variety of applications were developed in order to show the performance of the new

detector type. Not only biomedical devices [19], but also prototypes for X-ray astronomy missions like XEUS, BepiColombo and others were presented. Beside that, however, in 2003, the three partner institutes, MPI, University of Bonn and University of Mannheim formed the DEPFET collaboration. The goal was to develop DEPFET-based systems for high energy physics experiments. The first one in a series of high energy physics prototypes (working title: PXD4) was designed in 2003 for the TESLA experiment at DESY, Germany [20]. In 2006, the DEPFET collaboration decided to aim for a vertex detector system for the planned International Linear Collider (ILC) [21]. A new prototype detector device (working title: PXD5) was developed, customised for the ILC requirements.

Since 2005 the DEPFET collaboration has been triggering the interest of other groups working the field of high energy physics. Some of them, in particular groups from Prague, Karlsruhe, Valencia, Göttingen, Munich, Krakow and Giessen, joined the collaboration and became inherent parts of it. Nowadays, the collaboration is focused on the development of a vertex detector system for the future BELLE-II experiment at KEK, Japan. A new prototype (working title: PXD6) suitable for the BELLE-II requirements is currently being developed. The presented work is a part of this new prototype system and therefore continuing the history of DEPFET detectors.

In parallel to the work of the DEPFET collaboration, the DEPFET detector technology is still being used for X-ray experiments. The latest one is the so-called XFEL project, which is going to be a free electron laser experiment at DESY, Germany, started in the year 2008 [22].

2.1.3 Reading DEPFET Pixel Matrices

The straightforward way to realize a particle detector device based on DEPFET transistors is to arrange a number of transistors in a matrix structure, regarding each transistor as a pixel of an imaging frame. In order to capture a frame, each pixel of the matrix needs to be read. Reading in this context means the determination of the amount of signal electrons in the transistor's internal gate. As described in section 2.1.1, the signal electrons residing in the internal gate modulate the current from source to drain. This fact, however, causes the requirement for two separate measurements. Beside the measurement of the potentially modulated current, the offset current needs to be determined as well. That is the current flowing through the transistor while the transistor is switched on, but the internal gate is empty (known as pedestal current). Hence, the difference of both measurements is regarded as the signal.

Reading a Single Pixel

The need for two measurements for signal determination leads to two main strategies of reading DEPFET pixels. The first one is the so-called *double correlated sampling*. That means, whenever a pixel is addressed, two separate measurements are performed. The first measurement samples the pedestal current together with a possible modulation due to signal electrons. The second measurement is performed directly after the signal is erased by clearing. Afterwards, the difference of the two measurements is calculated in order to determine the signal.

The second strategy is called *single sampling*. Compared to the double correlated sampling, the second measurement is skipped here. The signal is calculated using a stored pedestal value. Thus, the pedestal current needs to be determined and stored before, during a dedicated pedestal measurement. The single sampling is almost twice as fast as double correlated sampling, since only half of the measurements are performed. The time consumption of the clearing can be neglected compared to that of the sampling. However, the quality of the signal determination is directly dependent on the quality of the stored pedestal value. Variations among pixels need to be considered as well as variations in time due to temperature, radiation and other effects.

Combining Multiple Pixels to a Matrix Structure

Beside the strategy for reading single pixels, the number of readout channels is the next parameter to be optimised. In principle, there are two extremes. On the one hand, only a single readout device can be sufficient. In that case, all the pixels of the matrix need to be multiplexed to the single readout device subsequently, which is obviously very time consuming. On the other hand, every pixel of the matrix could have its own readout device. In this way, the reading of the entire matrix was accelerated enormously. However, depending on the size of the matrix, a huge amount of not only readout devices but also interconnections would be necessary, which is hardly feasible for most realistic matrices.

The approved way is a compromise between the two extremes. To this end, the fact is used, that signal electrons residing in the internal gate of a DEPFET transistor are not able to set up a conducting channel between source and drain, but only to modulate a signal onto an existing channel that is switched on via the transistor's external gate. Groups of DEPFET transistors, typically arranged in a column of the matrix, share a single readout device by simply connecting all of them in parallel to the readout device's input. By external steering signals, it must be ensured that at any time only a single pixel of the group is switched on, while all others are switched off¹. The detector matrix is then built using several of these grouped columns in parallel. Pixels in the same row can then share the same pair of gate and clear steering signals, so that they are switched on/off and erased at the same time. This results in a number of readout devices that is equal to the amount of pixels in one dimension of the matrix.

Using this readout mode leads to the fact that pixels in the same row of the matrix are read out in parallel, while pixels in the same column are read out sequentially. This row-wise readout mode is known as *rolling shutter*.

Source Follower vs. Drain Readout

The way of connecting the grouped DEPFET transistors in parallel to a single readout device is not just a technical detail, but again an important design decision. In principle, there are two possibilities, the so called *source follower readout* or the *drain readout*. On the level of the DEPFET transistors, there is only a single difference between these two, namely whether the readout device is connected to a common source node (source

1. In principle it is also possible to have more than one pixel of the group being switched on. This method is called *ganged pixel readout*, but it is not considered here.

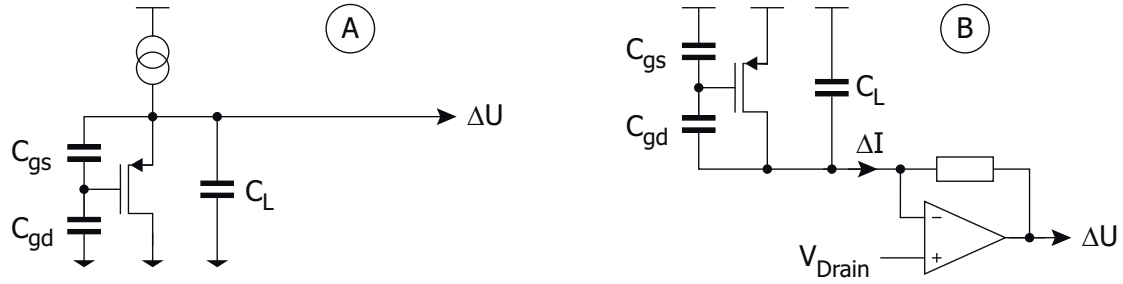


Figure 2-2 Generic and simplified schematic of a DEPFET transistor in source follower (A) and drain readout (B) mode [23].

follower) or a common drain node (drain readout). From the electrical point of view, these methods are completely different.

In the source follower configuration, as shown in figure 2-2 (A), a current source is forcing current through the DEPFET transistor, during which the voltage at the transistor's source node is measured. A variable conductivity of the DEPFET transistor induced by charge residing on its internal gate results in variable potential at this node.

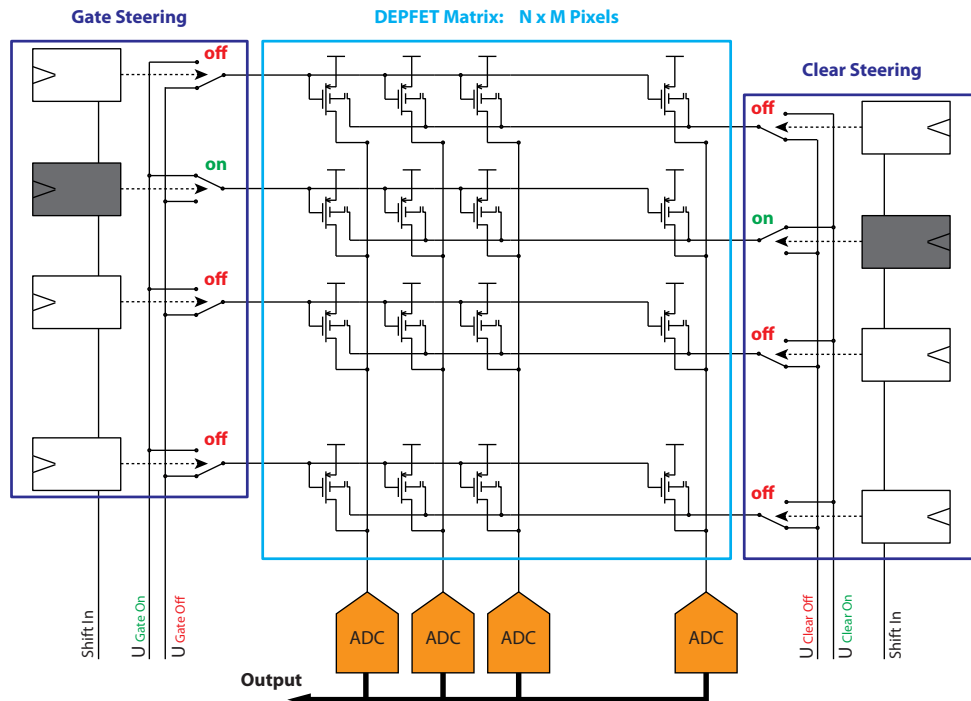


Figure 2-3 DEPFET pixel matrix readout arrangement (drain readout configuration). Pixels in the same column share a common readout drain line, while pixels in the same row are steered using the same gate and clear signals.

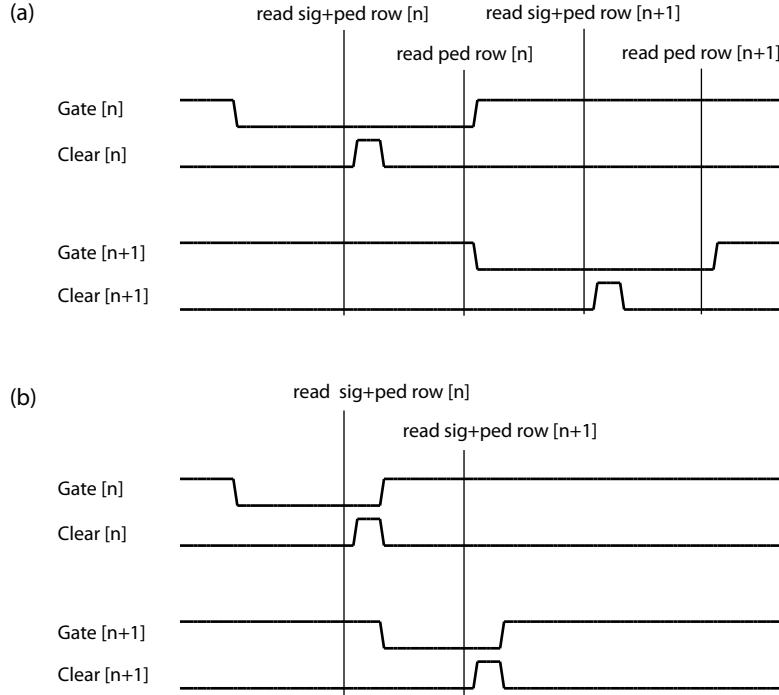


Figure 2-4 Example steering sequence for a DEPFET matrix. Gate and clear signals for two consecutive rows are shown. Note: DEPFET pixels are basically PMOS transistors, therefore, the gate signals are low-active. (a) Steering sequence in double correlated sampling mode. (b) Steering sequence in single sampling mode.

However, the signal rise time can be severely degraded by the line capacitance C_L of the common source node. It can be described by the following equation [23]:

$$t_r = \frac{C_L \left(1 + \frac{C_{gs}}{C_{gd}} \right) + C_{gs}}{g_m} \cdot 2.2$$

This settling time can easily reach several microseconds, so that the source follower configuration is not suitable for high speed applications.

In the drain readout configuration, the situation is completely different. The DEPFET transistor is operating as current source, and the signal is a change in its current. A simple amplifier can be connected to the common drain line, keeping its potential constant. Therefore, there is no need to charge and discharge the line capacitance C_L , so that it does not affect the speed anymore. The drawback of this configuration is, however, that fluctuations in the device thresholds and voltage drops on the source traces are amplified by the DEPFET transistor. As an effect for large matrix arrangements, fairly large current fluctuations among the pixels can appear.

An example DEPFET matrix readout arrangement using the drain readout configuration is illustrated in figure 2-3. Figure 2-4 shows the corresponding steering signal waveforms for single and double correlated sampling.

2.2 The PXD Detector System for BELLE-II

2.2.1 Impact Parameter Resolution

The major goal of the barrel-shaped inner layer pixel vertex detector of BELLE-II (PXD) is to enhance the tracking accuracy of the surrounding SVD detector by improving the impact parameter resolution. That is the measure of how good a decay position close to the interaction region can be reconstructed [58]. The target resolution is driven by physics aspects which demand a precision of about $20\mu m$ [9]. The equation for the impact parameter resolution σ in the case of a two layers detector, located at radii r_1 and r_2 , with an intrinsic spatial resolution¹ of σ_0 is made up of a geometric term and a multiple scattering term [25] [26]:

$$\sigma = \sqrt{\sigma_{geo}^2 + \sigma_{MS}^2}$$

$$\sigma_{geo} = \sqrt{\frac{r_2^2 + r_1^2}{(r_2 - r_1)^2}} \cdot \sigma_0$$

$$\sigma_{MS} = r_1 \cdot \frac{13.6 MeV}{\beta c p (\sin \theta)^k} \cdot z \cdot \sqrt{\frac{x}{X_0}} \cdot \left[1 + 0.038 \ln \left(\frac{x}{X_0} \right) \right]$$

For the latter, p is the momentum of the incident particle, θ is its track polar angle and z the charge. x is the thickness of the detector material and X_0 its radiation length. $k = 3/2$ for the resolution in $r \times \Phi$ and $k = 5/2$ for the z-projection.

The geometric term depends only on the distances of the detector layers to the interaction point and their spatial resolution. Beside the direct influence of σ_0 , the resolution is optimal for a minimized radius of the inner layer and a simultaneously maximized radius of the outer layer, which acts as lever arm. Considering the multiple scattering part of the equation, the detectors contribution to its minimization is again a close distance of the inner layer and a reduction of the detector's material. These are the major boundary conditions. Beyond that, extensive performance simulations yield to a set of design parameters which constrain the PXD detector development. The consequences of these simulations are discussed in the following.

2.2.2 Occupancy

The general issue that has to be handled in the context of parameter optimization is the detector's occupancy due to hits that are caused background events. The expected rate of background events is of course depending on the luminosity of the SuperKEKB accelerator. But this is certainly not a subject for optimizations here, so the luminosity

1. A simplifying assumption here is that the intrinsic spatial resolution is equal for both detector layers. This is not the case for the BELLE-II's PXD detector.

and thus the level of background events is considered given. The corresponding hits in the detector are of no interest for physical studies and only degrade the effective efficiency of reconstructing real events. So primarily, the basic question here is how much occupancy in a frame is allowed by the track reconstruction mechanisms in order for the PXD to improve the impact parameter resolution. The answer is given by Monte-Carlo simulations [33] (assuming baseline design parameters for the involved elements): about 1% per frame is fairly acceptable, for higher occupancies than 2.3% the PXD is not helping anymore. This result has to be respected by any optimization.

2.2.3 Layer Radii

The lower boundary for the radius of the detector's innermost layer is given by the existence of a beam pipe at the interaction region. Referring to its latest design [27], the beam pipe has an outer radius of 12mm . So by keeping a little safety distance, mounting the detector down to about 13mm radius is mechanically feasible. Simulations are made in order to find out how much the radius could be further increased while not unacceptably degrading the effective resolution, since the occupancy scales roughly with $1/r^2$ [9]. The result is that an increase to 14mm is only hardly degrading the effective resolution [28]. Although this is still a tough distance, it is decided to fix the radius of the BELLE-II's innermost pixel detector layer at that value. The position of the second layer is less critical¹ and is mainly driven by mechanical and mounting constraints. Its radius is set to 22mm .

2.2.4 Frame Readout Time

With the known radii of the detector layers, in particular that of the innermost one which is most affected by background hits, it is the job of the readout system to make sure that the occupancy per frame is kept within the defined limits. This is because the occupancy scales with the time required to read a frame.

Another general requirement for the readout time is derived from system aspects. The bunch circle time at the SuperKEKB accelerator is $10.06\mu\text{s}$. From the data analysis's point of view it is desirable to read the frames synchronously to that.

With respect to what is assumed to be feasible for the readout system, a frame readout time of $20.12\mu\text{s}$ is set as the baseline of the PXD detector. Simulations showed that the resulting expected occupancy due to background hits is about 0.4% [29] and thus fairly within the limits.

2.2.5 Modules and Dimensions

The cylindric shape of an ideal barrel detector is approximated by a polyangular arrangement of planar modules, the so-called *ladders*, as illustrated in figure 2-5. Eight ladders form the inner layer, while twelve of them are used for the outer one. The ladders are mounted onto solid support structures that are placed at both ends of the barrel. These

1. Since SVD data is used together with that from the PXD, the second layer's data is not primarily used as lever arm for the track reconstruction.

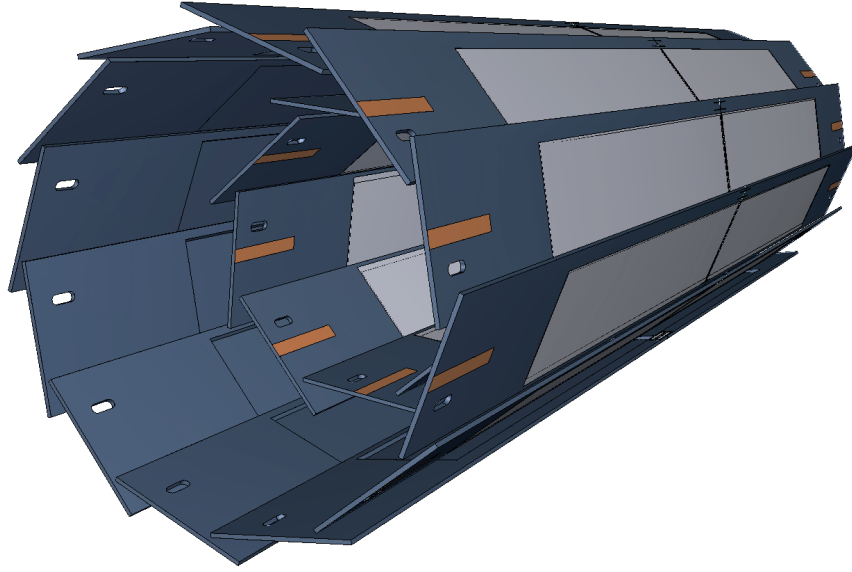


Figure 2-5 Drawing of the DEPFET vertex detector arrangement around the beam pipe at the interaction region [9]. A cylindric shape is approximated using flat ladder modules. The innermost layer comprises eight ladders at a radius of 14mm . The second layer is made of twelve elements resulting in a 22mm radius.



Figure 2-6 Picture of a DEPFET vertex detector assembly model. It shows a dummy beam pipe with the DEPFET vertex detector's mechanical support structures and ladder demonstrators mounted onto it. A can is used as a size reference [31].

support structures themselves are fixed to the beam pipe. A model of that assembly is presented in figure 2-6. The ladders are composed of two *half-ladder* modules, the major building block of the PXD detector, by gluing their top edges together. This is a self-supporting all-silicon module that serves as a substrate for the sensitive pixel area,

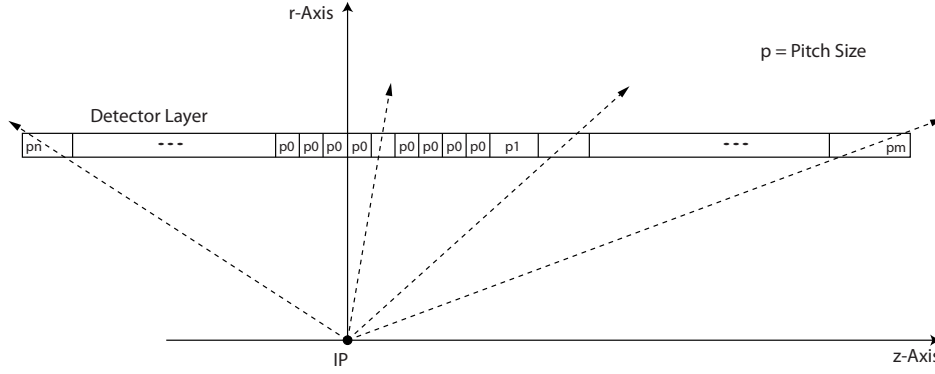


Figure 2-7 Illustration of the variable pixel pitch strategy [32].

chip housing as well as the interconnection platform at the same time. The width of the sensitive area is geometrically given by the polyangular arrangement and the aim of 100% coverage in $r \times \Phi$ plane. It is defined as 12.5mm for both layers. Its length along the beam pipe is primarily derived from the polar angular acceptance range of the SVD detector: $17^\circ < \theta < 150^\circ$ [9]. By additionally taking the layout and the geometric issues into account, the lengths are set to 44.8mm for the inner layer and 61.44mm for the outer one. The overall sizes of the half ladders are $15.4\text{mm} \times 67.975\text{mm}$ and $15.4\text{mm} \times 84.975\text{mm}$ respectively. The ladder designs are not fixed yet, so the numbers given here are the latest but may still be subject to slight changes.

2.2.6 Pixel Geometries

Having the radii and the sensitive area dimensions of the two innermost detector layers fixed, the pixel geometries are the next parameters to consider. Here, the constraints are of course given by the physics aspects, but also the achievable performance of the readout electronics comes into play. First of all, the detector's intrinsic spatial resolution is approximated by $\sigma_0 \approx P/(S/N)$, where P is the pixel pitch and S/N is the signal to noise ratio [30]. The former is obviously a property of the detector layout, while the latter is a quality parameter of the readout electronics. From that, a first order estimation of the pixel geometry is derived: by assuming an achievable signal to noise ratio of 10 to 20, a rectangular shape with a pitch of about $50\mu\text{m}$ is reasonable.

Using this number as starting point, detailed simulation studies have been performed in order to further optimise the pixel geometries. The most promising strategy is to use variable pixel pitches in the z -axis that are adjusted to the incidence angle of the particles as illustrated in figure 2-7. This idea is beneficial in two ways. First, increasing the pixel size towards the ends of a ladder reduces the overall number of pixels and thus relaxes the readout time per pixel for a given frame readout time. Second, the charge induced by hits at the ends is not spread over too many pixels, which is in principle advantageous for the track reconstruction accuracy. Simulations show that there is indeed an improvement, but not such that an excessive use would justify extra complications in track reconstruction algorithms [32].

In addition to the physics-driven considerations, more boundary conditions come from the readout electronics development, such as the available number of input and output channels of the various devices. Taking all this into account, leads to a segmentation of

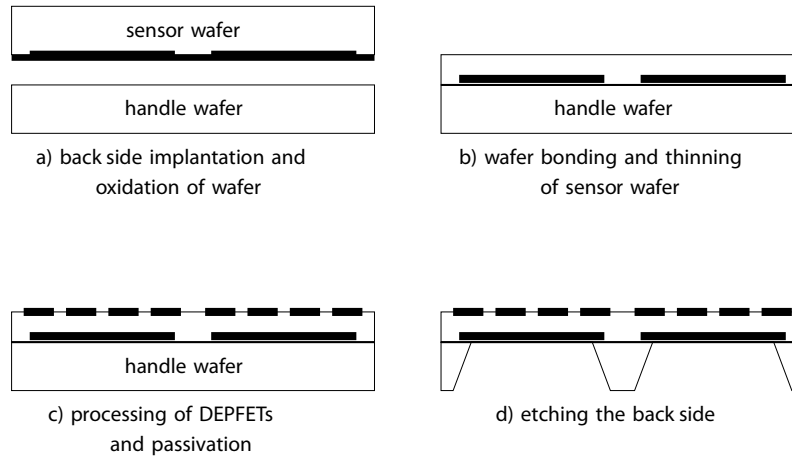


Figure 2-8 Processing steps for thinned wafers with both sides being structured [36].

the sensitive areas into pixels with the following sizes¹[34]. The sensitive area of the half-ladder modules for the innermost layer is divided into two regions with pixels of different pitch in the z-axis. The 256 rows² nearest to the interaction point (IP) are separated by $55\mu m$, the remaining area contains 512 rows of $60\mu m$ pitch. So there are 768 rows in total. Along the short side of the sensitive area the pitch of $50\mu m$ is kept, resulting in 250 columns. The same numbers of rows and columns hold for the modules of the outer layer, too. For the columns this is straightforward, since the modules are of the same width. With regard to the rows, an unequal number would result in different readout speeds per pixel for the two layers, which is not desirable for numerous reasons. So consequently, the pixel pitch along the z-axis must be adopted. Here there are currently two alternatives, either keeping the two regions approach using $256 \times 70\mu m$ and $512 \times 85\mu m$ or dismissing it by using $80\mu m$ pitch uniformly. A final decision about these options and values has not been made yet.

2.2.7 Detector Thinning

The optimization studies discussed so far focused only on the geometrical aspects of the detector design's influence on the effective impact parameter resolution. But as mentioned in section 2.2.1, the thickness of the detector material must be considered as well due to multiple scattering effects.

In order to tackle this issue, the technologists at the MPI Semiconductor Laboratory (Munich) have been acquiring know-how in wafer thinning [35]. Indeed, wafer thinning is a common technique in the semiconductor industry. In fact, for silicon detectors based on the DEPFT principle it is mandatory to structure both sides of a wafer, this however,

1. The given numbers are the latest, but they may still be changed.

2. A line of pixels along the short side of the sensitive area forms a row, a line along the long side (z-axis) forms a column respectively. The terminology of „row“ and „column“ is used in accordance to the meaning as defined in section 2.1.3. That means the readout procedure is defined such that the pixels are addressed consecutively along the z-axis. This scheme comes very naturally from area and material constraints of the used electronic devices and their required interconnections.

makes the processing extraordinary challenging. To this end, a special procedure has been developed. The relevant steps are illustrated in figure 2-8.

The main idea is to build the silicon device from two wafers rather than just a single one. In the first step both wafers, the so-called *sensor wafer* and the *handle wafer*, are oxidized and the back side structure of the final sensor device is realized on the back side of the sensor wafer. After that the two wafers are merged by means of direct wafer bonding. The unstructured top side of the sensor wafer can then be thinned down to the target thickness using conventional equipment. The third step is to build the DEPFET structures on the new polished top side of the sensor wafer and to deposit passivation material at the back side of the handling wafer. Finally, the openings in that back side passivation define the areas where the bulk of the handling wafer is removed by deep anisotropic wet etching. The etch process is stopped by the oxid layer in-between the two original wafers and thus uncovers the back side structure of the sensor wafer. A proof of principle showed that silicon membranes of only $50\mu m$ thickness can be produced this way.

By using this technique, the PXD detector modules can be thinned and thus the effects of multiple scattering are reduced, in particular the degradation of the achievable impact parameter resolution. In fact, the silicon bulk underneath the active area is thinned, while the remaining parts of the all-silicon modules keep their original thickness in order to provide sufficient mechanical stability. However, the thickness of the detector is also affecting the intrinsic spatial resolution of the detector itself. The thinner the detector, the smaller is the diffusion volume and the less energy is deposited by incident particles. Both reduces the effective resolution. It has been discovered by simulations that for the defined pixel sizes there is an optimum at $75\mu m$ [37][38]. Together with a thickness of the unthinned silicon parts of $420\mu m$ and including the contributions of the relevant ASICs, the total average thickness is equivalent to $< 0.2\%X_0$ [9].

2.2.8 Front-End Readout System

In the previous sections several assumptions and definitions concerning the achievable performance of the front-end readout system are made. They can be summarised as follows. On the half-ladder scale it is assumed possible to read a detector frame of 768×250 pixels within a time period of $20.12\mu s$ and a signal to noise ratio of 10 to 20. These performance assumptions turned into specifications!

Resolution, Dynamic Range and Noise Performance

The energy deposited in the detector by a minimum ionizing particle (MIP) for the defined geometries can be calculated using the Bethe-Bloch formula. It turns out that the energy is sufficient to generate 6000 electron-hole pairs in the silicon bulk of the detector. By expecting a DEPFET transistor gain of $500pA/e^-$ [9], the target signal-to-noise ratio of 20 demands for an input referred noise of the readout device of $150nA$. Moreover, it requires for a signal resolution of at least five bit.

The considerations about the readout device's dynamic input range is driven by mainly two facts. First of all, due to the Landau fluctuations of incident particles the input range must be extended. $8\mu A$ is considered sufficient. In addition to that the readout device needs to cope with fluctuations on the pedestal currents of the various DEPFET

transistors, which is about $12\mu A$ for an unirradiated prototype device [58]. In order to realize this extended dynamic range while keeping the effective signal resolution, the number of bits for the analog-to-digital conversion has to be increased accordingly. An eight bit representation is reasonable.

Single Pixel Readout Period

In the context of the DEPFET readout procedure as introduced in section 2.1.3, there is no doubt that the realizable performance is limited by the drain current measuring device rather than by that for steering the gate and clear lines. The number of required readout channels is depending on the actual organisation of rows and columns in the steering and drain lines. But in any case it is at least equal to the number of columns (250), so a fairly large amount of parallelly working channels is required. Consequently, the chip area is a limiting factor as well, which restricts the possible implementation alternatives. But nevertheless there is a good chance to meet these tough requirements, as already shown by a previous multi-channel microchip designs [42]. Derived from that experience, for the required accuracy a single pixel readout period of $100ns$ seems feasible with the existing concept.

Matrix Organisation

In order to achieve the required frame period of roughly $20\mu s$ while a single readout channel is able to read a pixel every $100ns$, the pixels of a half-ladder must be organised in up to 200 groups. The various groups are addressed for readout one after the other. All pixels within a group must be read in parallel. Taking the physical pixel arrangement into account (768×250 pixels), this structure can be mapped to the matrix in the following way. A group of pixels comprises four entire rows. This results in 192 groups of 1000 pixels each. Expressed in electronic terms, this structure translates into 192 gate and clear lines as well as 1000 readout channels per half-ladder module, which is illustrated in figure 2-9. Consequently, the effectively required readout period for a single pixel is slightly extended to roughly $104ns$.

The Readout ASICs

The ASICs required for reading the matrix are placed directly onto the all-silicon half-ladder surrounding the sensitive pixel area as shown on figure 2-9. The steering of the matrix is controlled by the *SwitcherB* ASICs. According to the applied readout scheme it is able to sequentially switch on/off and clear the matrix's quad-rows. The *SwitcherB* provides 32 output channels and can be daisy-chained. Using six of these along the half-ladder module allows for steering the entire matrix. For several reasons a multi-chip solution is preferred over a single chip design. Foremost, the tough area constraints for the balcony along the sensitive area in z-axis restricts the available routing space. More detailed information about the *SwitcherB* is presented in section 2.3.

The drain lines of the matrix are routed to the outer end of the half-ladder where the readout ASICs are located. As indicated in figure 2-9, two different sets of readout ASICs are used here, the *DCDB* and the *DHP*. The *DCDB* (Drain Current Digitizer for BELLE-II) provides 256 analog-to-digital conversion channels¹ on a single chip. Four of them are required to connect all the drain lines. More detailed information about the

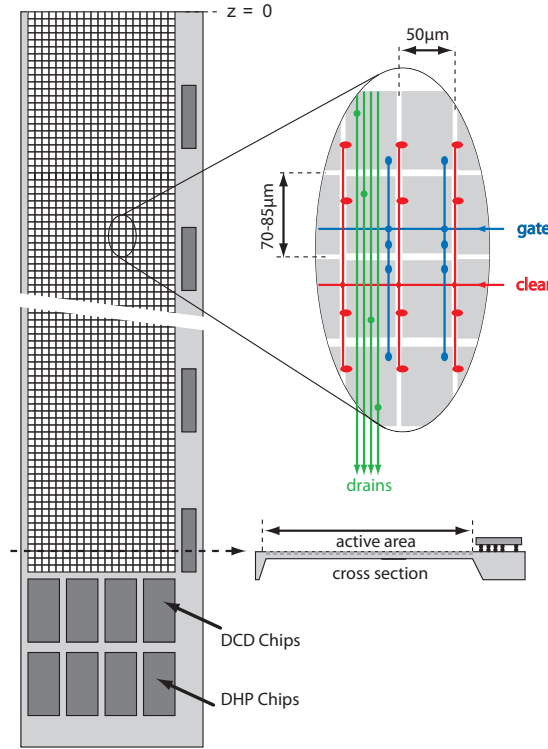


Figure 2-9 Schematic drawing of a half-ladder module. It shows the sensitive pixel area and the balconies for the SwitcherB steering chips and the DCDB / DHP readout ASICs. A zoomed section of the sensitive area provides some detailed information about the pixel interconnection. The pixel dimensions refer to the second layer half-ladder modules [9].

DCDB is presented throughout this work starting with section 2.4. In order to cope with the huge amount of data produced by the DCDB, the DHP (Data Handling Processor) is used for early data analysis and reduction. It is discussed very briefly in section 2.5.

2.2.9 Bump Bond Interconnection Technology

As the amount of detector material has to be minimized in order to reduce multiple scattering effects, the insensitive area of the half ladder module is kept as small as possible. As a consequence, space restrictions for placing and interconnecting the readout chips arise very naturally. Electrically connecting the chip's signals and power lines by means of *wire bonding*, a very basic and well-established technique, is therefore impossible. A much more dense method to do all the required interconnections is called *flip-chip bonding*, where the chip is flipped and placed face-to-face onto the carrier. So the latter is used for interconnecting all the chips on a half-ladder module.

There exist several ways of doing flip-chip bonding, distinguished by procedural differences. An important restriction to the choice of the flip-chip bonding method for

1. The number of channels is derived from chip layout constraints and thus does not match exactly an integer fraction of the drain count.

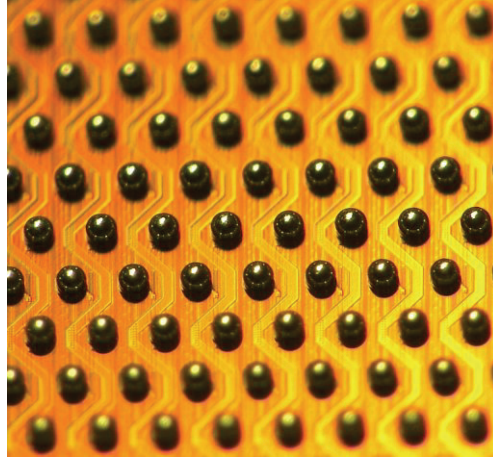


Figure 2-10 Picture of the DCDB solder bumps: $100\mu m$ diameter, $200\mu m$ horizontal pitch, $180\mu m$ vertical pitch [36].

the PXD project is due to the fact that the carrier, which is the half-ladder module in this case, has a very fragile structure. Thus, the use of any compressing forces is prohibited. A suitable solution is to use solder bumping. Here, the chip pads are assembled with solder balls. The electrical interconnection is established by simply placing the chip onto the carrier, heating it up and making the solder melt. Another big advantage of solder bumping over other existing technologies is the easy rework procedure as the solder can simply be remelted in case of any chip failure. This is of particular importance as there are 14 chips in total residing on a single half-ladder module, each with a limited production yield. In order to achieve an acceptable overall production yield on half-ladder level, an easy rework procedure is indispensable.

The DCDB and the DHP chip are assembled with solder balls on wafer level during production. However, for the SwitcherB such an option is not available, so the solder balls must be assembled afterwards. An additional complication arises from the fact that the SwitcherB's pads are made of aluminium, which is not wettable by solder. To this end, gold studs must be placed on the SwitcherB's pads prior to the assembly of the solder balls. Gold studs provide good mechanical and electrical connectivity to the aluminium pads on the one hand and are wettable by solder on the other hand. The gold studs are placed using a modified wire bond process. The solder balls are assembled by means of a solder jetting technology. More detailed information about chip assembly for the BELLE-II PXD detector is available in [36].

2.2.10 Higher Level Readout System

The data that is produced by the two chips DCDB and DHP must be transferred to the higher level readout system for further processing. Unfortunately, there is only very little room available around the inner layer detector, which has a large impact on its connection scheme. The baseline design is shown in figure 2-11.

A FLEX capton cable of about $20cm$ connects the half-ladder module to a patch panel. Besides power filtering and impedance matching, the patch panel is mainly acting as repeater within the electrical connection of the half-ladder module to the so-called *Data*

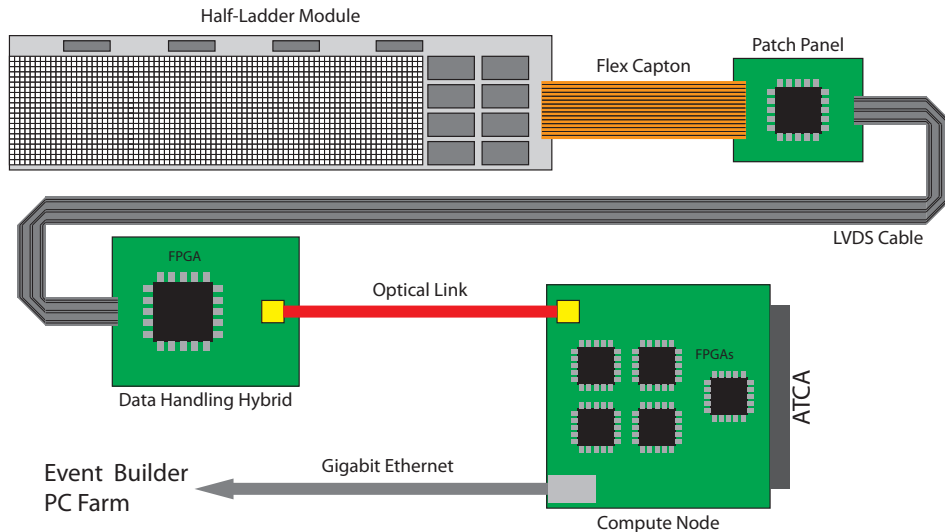


Figure 2-11 Architecture of the PXD readout system [9].

Handling Hybrid (DHH), a FPGA-based readout board. There is one DHH per half-ladder module, responsible for its interconnection to the outside world. First of all, it provides the clock signal as derived from the BELLE-II environment. Secondly, it acts as slow control master for all the ASICs. And thirdly, the data coming from the DHPs via electrical connections is multiplexed onto a single optical link. Via this optical link, the data is further transferred to the so-called *Compute Nodes* (CN). These CNs are FPGA-based devices and compatible to the ATCA¹ standard. There is one CN per DHH, resulting in the total number of 40 CNs for the entire PXD detector system. However, all the CNs are interconnected using a system level network, thus each CN is potentially able to access the data of every DHH. Together with tracking information coming from the SVD, data processing algorithms running on the CNs define regions of interest within the PXD data. These regions of interest are then further analysed and provided to the BELLE-II event builder farm, where they are combined with associated data from the rest of the BELLE-II sub-detectors.

2.3 The SwitcherB Steering ASIC

2.3.1 Overview

The SwitcherB ASIC [39] is a steering chip for DEPFET pixel matrices that is particularly designed for the application in the BELLE-II inner layer vertex detector system. It is the latest version in a series of DEPFET steering ASICs and is now in a close-to-final state.

The SwitcherB provides the total number of 32 output channels, each consisting of a gate line and a clear line driver. All gate output drivers share the two common supply inputs

1. Advanced Telecommunication Computing Architecture

G_{Hi} and G_{Lo} that determine their two output voltage levels. So at any time (apart from transients) every gate output is either on G_{Hi} or G_{Lo} level. Accordingly, the clear drivers share the two common supply inputs C_{Hi} and C_{Lo} for defining the voltage levels of the clear lines. Due to the fact that there are relatively high voltages¹ necessary for steering DEPFET pixels, especially for clearing the internal gates, the SwitcherB is designed using a special high voltage semiconductor technology with a minimum structure size of $350nm$, provided by Austria Microsystems². This technology in combination with special design techniques allow for a maximum output voltage swing of $50V$.

The controlling of the SwitcherB's output channels is adapted to the use case of the BELLE-II PXD. First of all, as mentioned in section 2.1.3, the rolling shutter readout mode is applied. In the context of the SwitcherB's mode of operation, that means the channel pairs must be activated one after the other. Hence, a random channel access is not necessary and the controlling mechanism can be kept simple. A 32 bit deep shift register with each bit representing a single output channel is sufficient for managing the activation/deactivation of the channels. Two additional common strobe signals are then used for steering the outputs of the activated channel(s). More information about the operation details can be found in section 2.3.4. Secondly, gate and clear channels are not operating in the same way, as their ON and OFF states have different meanings. Due to the fact that DEPFET pixels are basically PMOS transistors, a matrix row is regarded as switched-on, if the corresponding gate signal is at G_{Lo} level. Accordingly, it is regarded as switched-off with the gate signal being at G_{Hi} potential. For the clear channels, it is just the other way round. That means the pixels in the activated row are cleared if C_{Hi} potential is applied to the clear signal.

Although the SwitcherB has only 32 output channels, the chip can be used for steering much larger DEPFET matrices. Its design allows for a very simple series connection scheme of multiple SwitcherB chips, letting them act as one single device with respectively more output channels.

The SwitcherB ASIC is equipped with a JTAG compatible interface. It is used for configuration and debugging purpose. On the one hand, internal registers influencing the chip's operation mode can be accessed using this interface. On the other hand, a boundary scan chain is available which allows for verifying the off-chip interconnections. The latter is very important for the chip's application in the BELLE-II PXD detector system. Since there is only very little space for chip interconnection available on the half-ladder, wire bond connection are simply not feasible and so bump bond connections are used for all the chips that are placed onto it. Optical methods for verifying these connections are obviously not useful, hence, electrical tests using JTAG is a clever alternative.

2.3.2 Channel Boosting

Since one of the requirements for the BELLE-II operation is a constant low temperature environment, power dissipation is a concern, in particular for the innermost sub-detector system. To this end, power saving is a high priority issue for the applied microchips. In

1. Test matrices have been operated using up to $7V$ swing on the gate lines and about $10-20V$ swing on the clear lines.

2. Changing the technology to $180nm$ minimal feature size is currently considered.

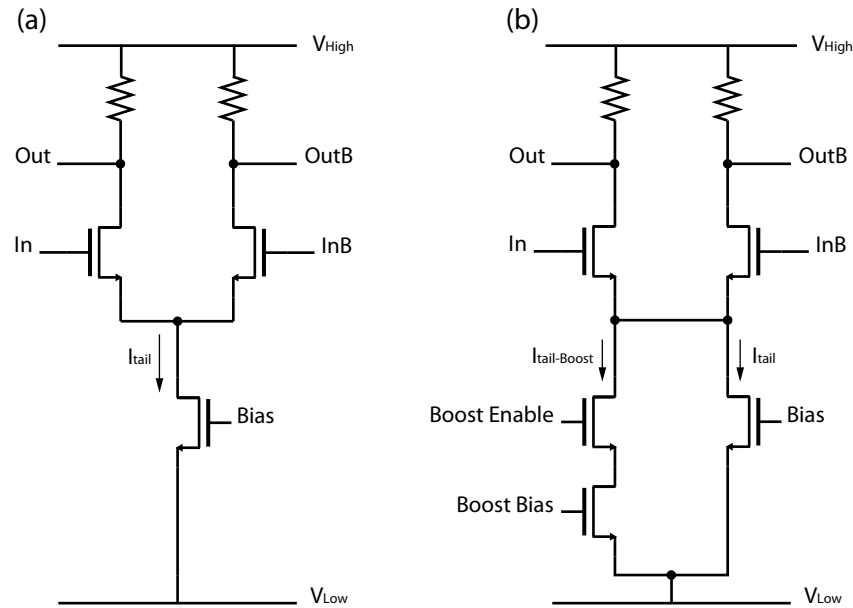


Figure 2-12 (a) Basic principle of a differential pair. The tail current is regulated by a bias transistor. A trade-off needs to be found between low power (low bias) and high speed (high bias). (b) Improved version of the differential pair, similar to the one that is used in the SwitcherB. A minimum tail current is kept by the regular bias transistor. A second bias transistor is added that provides a much higher tail current and therefore a much higher speed. It is enabled only when it is needed using the `Boost Enable` switch in order to save power.

the context of the SwitcherB ASIC, simulations and measurements with earlier chip versions revealed a rather simple power dissipation characteristic. First of all, in the normal rolling shutter operation mode, only one channel is active, while all others do not change their state. Secondly, the main power consuming part of the SwitcherB is identified to be a level shifter unit¹, that is residing in every output channel. It is used to transfer controlling information from the low voltage digital domain to the high voltage analog output stages. The level shifter is implemented using a differential pair. The basic principle of such a differential pair is illustrated in figure 2-12(a). Obviously, there are two competing design goals here. On the one hand, low power consumption is required by the application environment. On the other hand, a high current flowing through the differential pair is desirable for high operation speed.

The compromise between the two design goals is shown in figure 2-12(b): the differential pair is implemented using two different options for generating its tail current. The regular bias transistor provides a low current, which is sufficient for the circuit to keep its state. It can be configured via the JTAG interface in the range of $1 - 6.6\mu A$. The second bias transistor is switchable and provides a much higher tail current. It is configurable as well, namely in the range of $30 - 210\mu A$. So for normal operation, when

1. Actually, the main power consumption in the context of the SwitcherB is caused by charging and discharging external capacitances, which are the gate and the clear lines of the detector. This, however, is unavoidable and cannot be influenced by the SwitcherB design.

the majority of the channels has a constant output, their boosting second bias transistor is switched off resulting in a very low current consumption in these channels. Only the level shifters of the activated channel¹ are boosted, allowing for high speed data transmission.

2.3.3 Overlapping Gates

The SwitcherB is the first version in the series of Switcher chips that offers the feature of overlapping gates. Instead of switching on the gate signals of a DEPFET matrix strictly one after the other, it is now possible to have a certain overlap in the order of nanoseconds during the change of the activated row. That means, the next row is already switched on, while the currently activated one is not yet switched off. Figure 2-14 illustrates this behaviour.

It is believed that this feature could be useful in mainly two ways. First of all, this technique could increase the maximum frequency, at which DEPFET transistors matrices could potentially be read out, by hiding the early signal transition part of the transistors in the reading phase of the previous ones. But obviously the overlap must be kept small enough to avoid interferences between the consecutive measurements. Secondly, if a matrix row is not switched on before the previous row is switched off, the common readout line is significantly disturbed by the temporarily missing offset current. Since the offset current is typically much higher than the signal to be measured, there might be a significant impact on the readout device as well. This might make the readout device being indisposed for a short period of time, which, in turn, extends the overall readout time of the matrix.

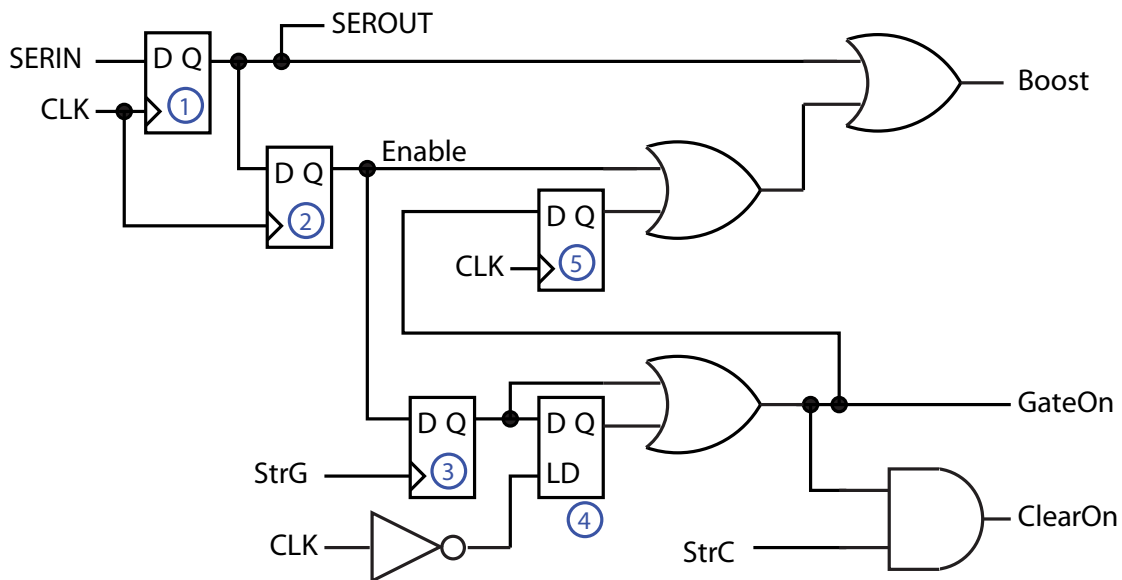


Figure 2-13 Schematic of a digital logic block for controlling an output channel pair [39]. The numbers in circles indicate pointers for referencing registers and latches in the text.

1. It is an implementation detail where by actually not only the activated channel is boosted, but also the previous and the following one.

2.3.4 Operation Mode Details

The 32 output channels of the SwitcherB are each controlled by an individual digital logic block. The schematic of these logic blocks is drawn in figure 2-13.

The two strobe signals, `StrG` for enabling the gate output and `StrC` for enabling the clear output, as well as the clock signal `CLK` are common to all control blocks. `SERIN` and `SEROUT`, the input and output of register #1, are used to interconnect the various control blocks, forming a 32 bit deep shift register. The `SERIN` signal of the first channel's control block as well as the `SEROUT` signal of the last channel's control block are connected to chip pads. The off-chip availability of the last `SEROUT` allows for the concatenation of multiple SwitcherB chips. The `GateOn`, `ClearOn` and `Boost` signals are directly connected to the corresponding analog output drivers. `GateOn` and `ClearOn` determine the output voltage level, that means whether the gate output has `GHi` or `GLo` potential, and whether the clear output has `CHi` or `CLo` polarity respectively.

Once a logical '1' is stored into the register #1 of such a control block, the corresponding output drivers are boosted, that is activated. At the following clock edge, the logical '1' is transferred from register #1 to register #2. Besides keeping the boosted state of the drivers, the control block is now sensitive to actions on the `StrG` strobe signal: A rising edge on `StrG` makes register #3 storing the logical '1' of register #2 and thereby forwarding it to `GateOn`, which brings the corresponding gate output driver to `GLo`¹ potential. Once the `GateOn` control signal is high, the `StrC` strobe is directly forwarded to the corresponding clear output driver. Finally, the logical state of `GateOn` is stored in register #5 at the next rising clock edge. This leads to a delay in releasing the boosted state by a single clock cycle, so that an output driver is boosted until the corresponding output has returned to the idle potential.

The SwitcherB's special feature of overlapping gates is realized using latch #4 and the relation of the `StrG` and `CLK` signals. The trick is that the change of the output value of register #3, which is triggered by a rising edge on `StrG`, occurs either during the transparent or during the holding state of latch #4. In the former case, that is when `CLK` is low, the latch simply does not have any effect. The `GateOn` signal is exclusively controlled by register #3. In the latter case, a state change in register #3 does not immediately influence the output of latch #4. Since `GateOn` is generated by a logical OR of the output states of register #3 and latch #4, this behaviour affects only the switching-off of `GateOn`. In fact, it introduces a delay that keeps `GateOn` high until the latch #4 is transparent again. Regarding subsequent control blocks, this delay results in overlapping gate output signals as shown in figure 2-14.

Measurement results illustrating the performance of the SwitcherB are published in [36].

1. DEPFET transistors have got a PMOS structure, so they have got a low-active gate.

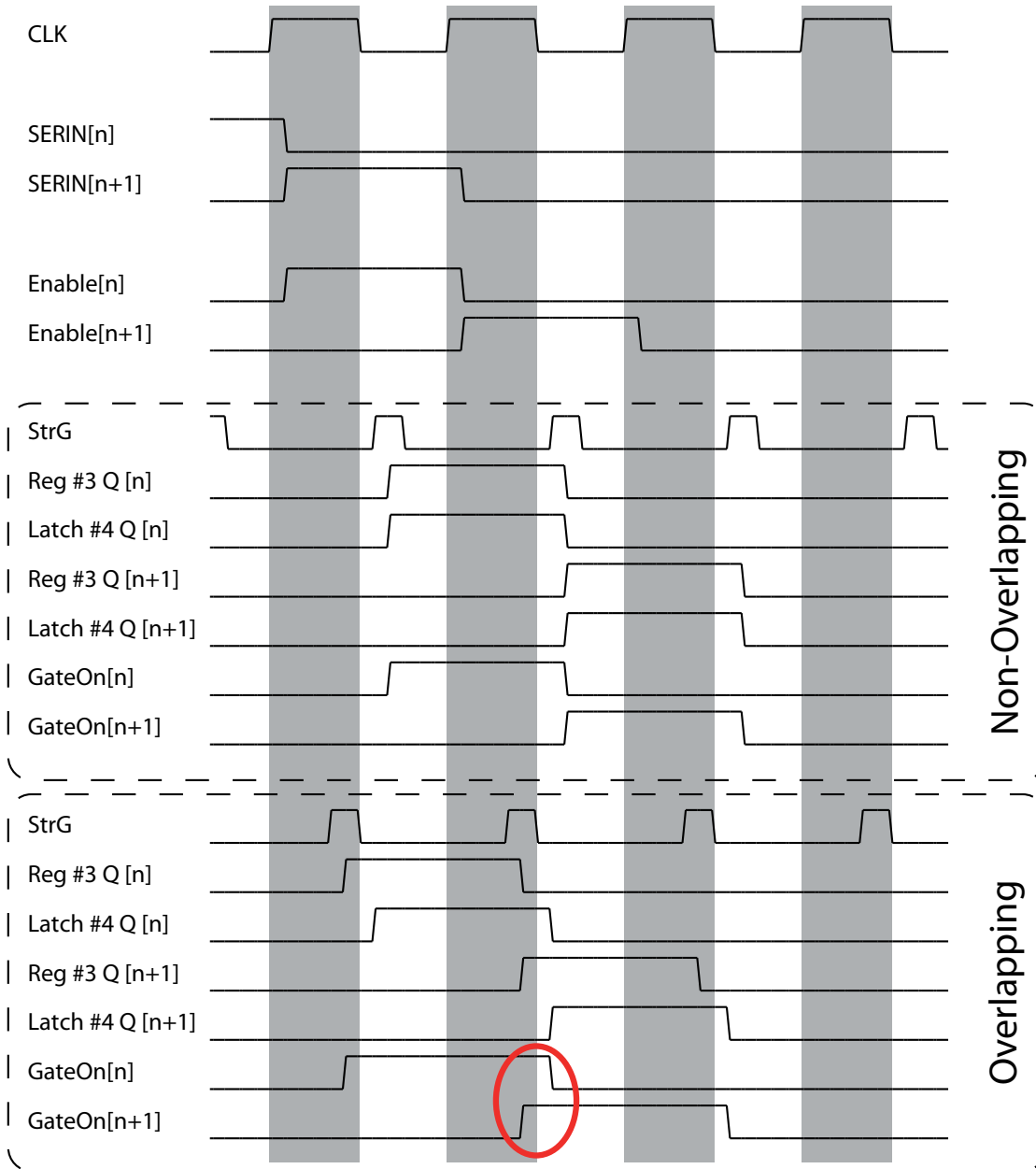


Figure 2-14 Timing diagram for illustrating the SwitcherB control logic operation. The top level control structure is the 32 bit deep shift register. Its functionality is indicated by the two consecutive SERIN signals of stage n and n+1. The relation of the CLK and StrG determine the relation of neighbouring gate output channels. For the two modes, overlapping and non-overlapping, the StrG signal together with the intermediate output signals of register #3 and latch #4 as well as the resulting GateOn signals are drawn.

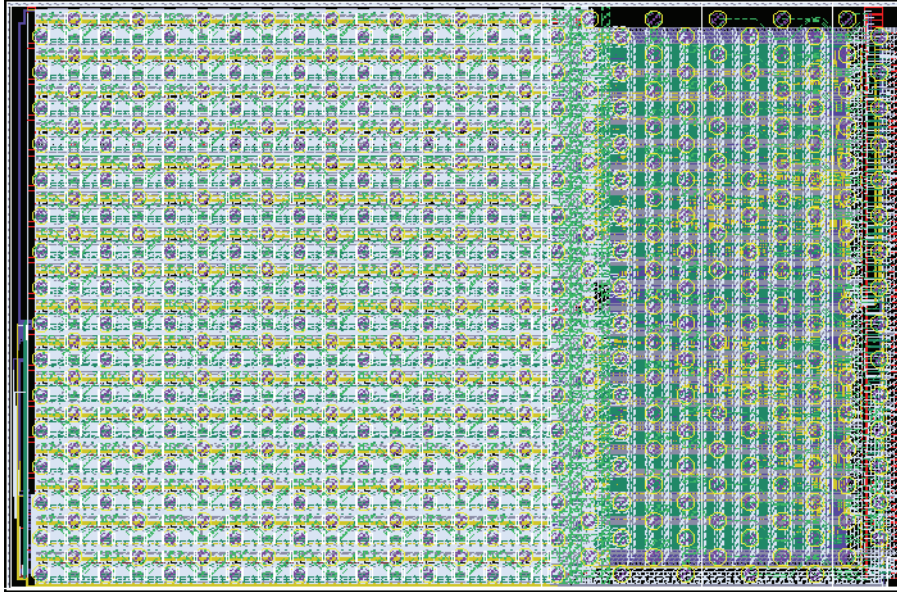


Figure 2-15 Picture of the DCDB layout. The 16x16 ADC channel matrix is located on the left-hand side, the digital logic is placed on the right-hand side.

2.4 DCDB - The Drain Current Digitizer for BELLE-II

The Drain Current Digitizer for BELLE-II (abbreviated DCDB) is the ASIC used for measuring the DEPFET signals in the BELLE-II application. Like the SwitcherB for the steering chips, the DCDB is the latest release in a series of DEPFET readout devices. The design is close-to-final, meaning that minor implementation changes might still be necessary, while the total number of channels is rather fixed due to dependencies of other developments within the project. The DCDB development was first presented at the TWEPP 2010 conference in Aachen, Germany, and later on published via JINST [40].

The DCDB is a highly optimised ASIC providing analog-to-digital conversion capabilities, especially designed for the needs of the DEPFET inner layer vertex detector of BELLE-II. It provides the total number of 256 analog-to-digital conversion (ADC) channels on a single chip. Each of these channels is able to capture measurements at a sampling period of only $100ns$ as it is required by the BELLE-II environment. It is designed to meet the specifications as defined in section 2.2.8.

On the analog side, the nominal dynamic range of the ADCs is designed to be $16\mu A$. This is more than enough for the expected signal bandwidth but in fact not sufficient for coping with the pedestal current fluctuations among the DEPFET transistors of the detector. To this end, an additional compensation mechanism is implemented to compress the effective pedestal spread. The constant fraction of the pedestal current is subtracted by adequate current sources. Furthermore, a transimpedance amplifier-based current receiver is used at every channel's input. It is used primarily to actively regulate the input potentials, which is mandatory for the target readout speed. Since the current

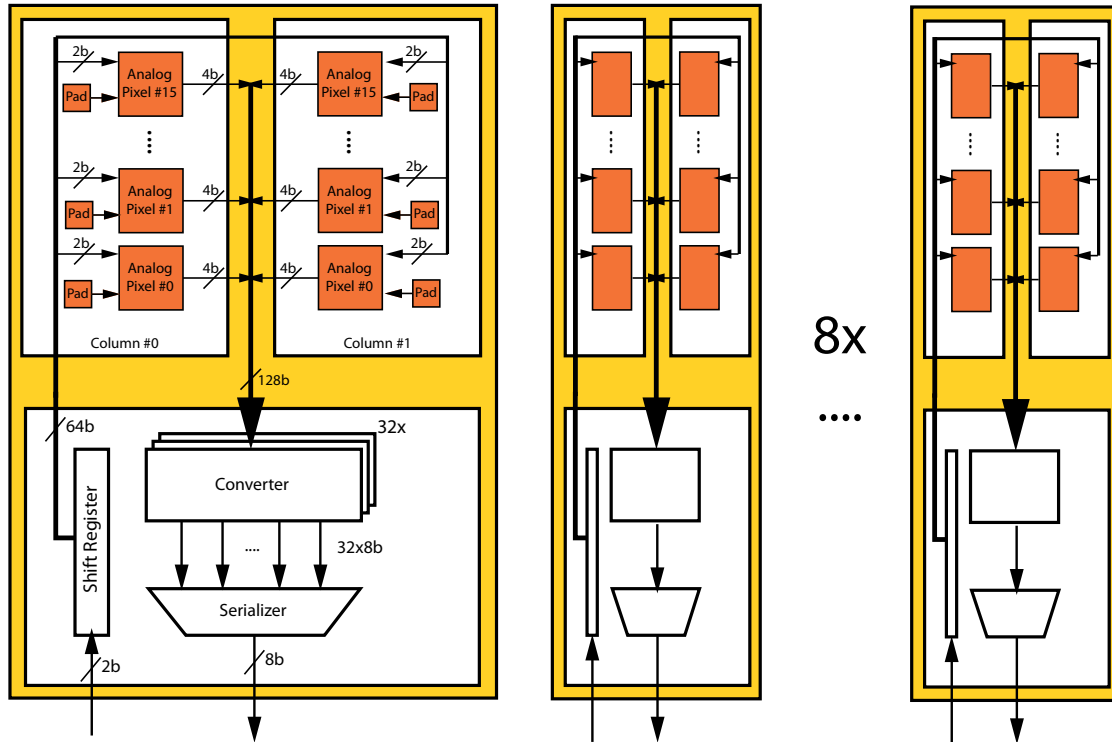


Figure 2-16 DCDB overview schematic. The DCDB's analog channels are arranged in groups of 32 channels. Each group has its own digital converter block and serialization unit. Shift registers with a width of two bits and a depth of 32 steps are used for the deserialization of the input values for the dynamic pedestal fluctuation compensation.

receiver is expected not to be the dominant source of noise in the system, its optional signal amplification feature can be used to additionally reduce the effective noise¹.

On the digital side, each measured signal is digitized to an eight bit value. Digital serialization logic is used to multiplex the conversion results of all 256 input channels onto eight output channels. Each of these channels is an eight bit wide bus, operating at a clock frequency of 320MHz . So when operating at full speed, the DCDB produces 2.56GB of data per second.

Like the SwitcherB, the DCDB provides a JTAG compatible interface for configuration and debugging purpose. It is primarily the ADC channel design that requires a lot of configuration, such as bias voltages and currents, but also switches manipulating the mode of operation. Some of these settings are global ones, affecting all the ADC channels on the chip. Others are set individually for each channel. All these internal configuration registers are interconnected, building two shift registers that are accessible via the JTAG interface. Beside that, all the DCDB's digital signal pads are included in a boundary scan chain, that is connected to the JTAG interface as well. Using this boundary scan technique, electrical connectivity checks for off-chip connections can be performed, which is going to play an important role later on during the quality control of the final BELLE-II half-ladder modules.

1. This feature is applicable only in case the actual operation conditions allow for the reduction of dynamic input range that comes along with it.

The DCDB is built using a $180nm$ CMOS technology with six metal layers, provided by UMC via a EuroPractice multi-project wafer run. An additional metal layer is used offering signal redistribution capabilities and bump bonding pads. These are mandatory in order to be able to flip the chip directly onto the half-ladder module. The chip occupies an area of $3240 \times 4969 \mu m^2$. The ADC channels are arranged in a 16×16 matrix, each $200 \times 180 \mu m^2$ in size.

Figure 2-15 shows a picture of the final layout, figure 2-16 provides its overview schematic. The details of the DCDB's implementation are examined much closer in the following chapters.

2.5 DHP - The Data Handling Processor

With its eight output buses operating at a nominal frequency of $320MHz$, the DCDB produces a data rate of $2.56GB/s$. For the entire PXD detector system, which uses 160 DCDB chips in total, the data rate adds up to $409.6GB/s$. Both, transferring and processing such huge amounts of data is hardly feasible using off-the-shelf products. Even with FPGA-based processing nodes it would be a challenging task. A much more suitable solution, however, is to employ an application specific chip performing first order data reduction, right next to the DCDB on the half-ladder module. Within the PXD detector system, the so-called *Data Handling Processor*, abbreviated DHP, is used for this job [9]. A one-to-one mapping is applied, meaning that every DCDB has its own DHP chip for reducing its data output.

The obvious question here is why not to combine the two designs into one chip. This is because the advantage of the following two aspects is weighted higher than the additional routing effort on the half-ladder module as well as the risks that come along with it, like the yield of the production and the assembly¹. First of all, the development of the designs can be done completely separate, which allows to easily spread the job over several groups of the collaboration. Secondly, different microelectronics technologies can be used to better fit the individual needs of the predominantly analog DCDB design and the almost pure digital DHP.

The DHP is being developed by the group of Prof. Dr. Wermes at the University of Bonn. Since such a kind of chip was not planned for earlier projects of the DEPFET collaboration, like ILC for instance, the DHP is currently in a testing phase. A first test chip of $2mm \times 4mm$ size supporting half of the DCDB output buses has already been produced. The second version, a fully featured chip with $3.2mm \times 4mm$ size, is under way [44]. Its operation principle was first published in 2010 [43]. Figure 2-17 provides an appropriate block diagram. The Data Handling Processor is implemented using a $90nm$ CMOS technology provided by IBM².

-
1. It is currently subject to discussions whether this decision is revised in the near future. Refer to section 9.3 for further information.
 2. The DHP used to be produced in a multi-project wafer run via MOSIS. Unfortunately, MOSIS recently ceased offering the respective 90nm technology node of IBM. Thus, the DHP will have to be rebuilt using an alternative technology. Changing from 90nm to 65nm technology in one go is currently discussed.

higher the occupancy, the fewer values can be discarded and thus the higher the necessary transmission bandwidth for transmitting information from the hit pixels.

Furthermore, since the DHP is able to interpret trigger signals from the BELLE-II environment, a so-called *triggered readout scheme* can be applied. Using a trigger, the system is able to distinguish between uninteresting background signals and those that are worth to have a detailed look at. This technique also holds a large data reduction potential. In fact, the effectiveness of data reduction based on triggered readout is not depending on the bare detector occupancy, but on the rate of interesting physics events. Both parameters are only estimations and are subject to statistical fluctuations. However, it is believed that zero suppression and triggered readout can reduce the data output of a single DCDB down to about 150MB/s . Hence, a single 1.25Gb/s differential output line per DHP should be sufficient.

Apart from the data processing tasks, the DHP is also providing some infrastructure for the other ASICs on the half-ladder module. Each DHP is equipped with a phase-locked loop (PLL) circuit, generating the clock signal for its partner DCDB chip. Together with the on-chip generated reset signal for the DCDB, the DHP synchronizes the DEPFET matrix readout with the rest of the BELLE-II environment. Additionally, one of the DHPs on a half-ladder module is generating the steering sequence for the SwitcherB chips. In conformity with the DCDB and the SwitcherB, the DHP implements a JTAG compatible slow control interface. Hence, all the ASICs residing on a half-ladder module can be connected in a large JTAG chain for configuration purposes.

The Analog Domain of the DCDB

Abstract:

While the DCDB is mentioned only very briefly in the previous chapter, its detailed description starts here with a focus on the analog domain. After the presentation of the analog channel's basic structure, the analog-to-digital conversion principle as well as its realization is explained. The main part of the chapter is then dedicated to the implementation details of the various building blocks.

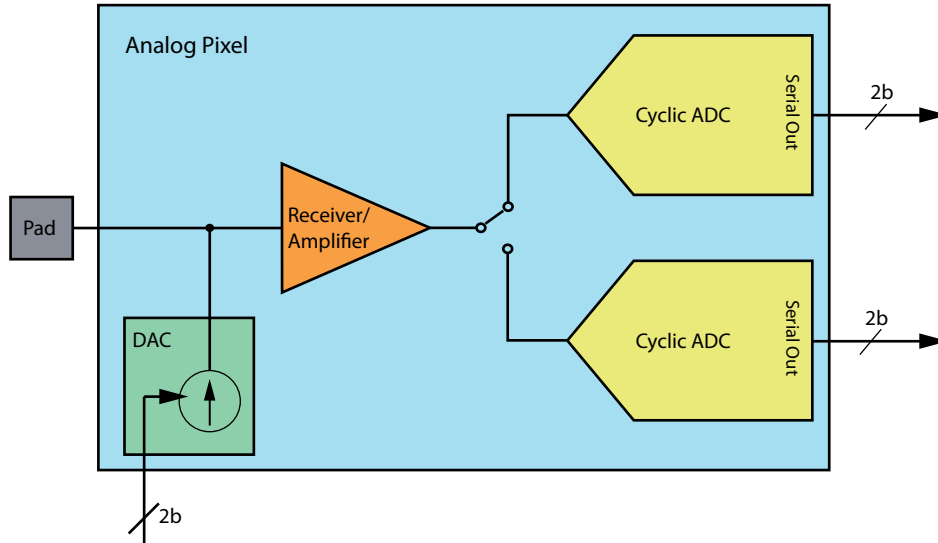


Figure 3-1 Simplified schematic of the DCDB's analog-to-digital conversion channel. The receiver keeps the input potential constant and amplifies the signal, which is then digitized alternately by two ADCs. The DAC is able to add current for offset compensation.

3.1 Overview

In the previous chapter the DCDB chip was introduced as one of the readout ASICs for the DEPFET detector matrices in the BELLE-II experiment. It is used for converting analog signals, namely the currents flowing through the matrix pixels, into digital data. There are 256 of those conversion channels working in parallel, taking new samples every 100ns , which makes the DCDB a fairly large design. In order to cope with this complexity, the design is split into two portions, the analog and the digital part, allowing a group of designers to work simultaneously on that project. In this context, the present chapter provides a description of the chip's analog domain.

A very simplified schematic of a DCDB analog-to-digital conversion channel is provided in figure 3-1. Its basic operation principle is described as follows. The input signal current is received by a current receiver circuit. A digitally controlled current source is able to dynamically add a current to the input node for offset compensation. The output of the receiver is digitized alternately by two cyclic analog-to-digital conversion (ADC) blocks. The resulting digital codes are passed to the DCDB's digital domain for further processing.

3.2 The Analog-To-Digital Conversion Principle

Generally, the variety of different existing ADC implementations is large, and so is the number of publications, such as [41] for example, dealing with this subject. The decision about which of these to use for implementing the DCDB is driven by a multidimensional

trade-off that has to be made in order to comply with the specifications of the PXD detector for the BELLE-II experiment. A very obvious restriction is the available silicon area. The DCDB is specified to provide 256 independent conversion channels on a chip area that is not only constrained by the available area on the half-ladder module. Power dissipation is also a primary concern, as there are only very limited possibilities of cooling the ASIC during operation in the vicinity of the beam pipe. Finally, reaching the target accuracy at a sampling period of 100ns is a must in order for the entire sub-detector to contribute to the BELLE-II performance enhancement. Taking all these requirements into account, the use of a cyclic ADC approach seems to be a fairly good choice. First of all, the design layout is small, compared to pipelined ADC implementations for example, since the same electronic parts are reused for every bit of the conversion result. This fact addresses not only the area, but also helps to reduce the power consumption. Secondly, simulations showed that the variant's drawback of reduced sampling rate can be overcome by using two of those in parallel, which is still small enough to meet the area constraints.

The cyclic ADCs use a redundant signed-digit (RSD) cyclic conversion algorithm, producing two digital output bits in every cycle. The algorithm starts comparing the input signal to two thresholds, an upper and a lower one. If the input signal is larger than the upper threshold, the pair of output bits is set to 10, meaning +1, and a reference current is subtracted. If the input signal is smaller than the lower threshold, the output code is set to 01 (-1) and a reference current is added. If the input signal value is between the thresholds, the output bits are set to 00 (0) and no arithmetical operation is carried out. The residue signal is then multiplied by two and the result undergoes the same operation for the determination of the next bits. It is clear that the proper selection of thresholds and reference current is essential for the correct operation of the ADC.

Theoretically, the more such cycles are performed on an input signal, the more accurate is the result. Assuming n conversion cycles, the result can then be translated to binary representation using the following equation:

$$D = 2^n b_n + 2^{n-1} b_{n-1} + \dots + 2^0 b_0, \quad b \in \{\pm 1, 0\}$$

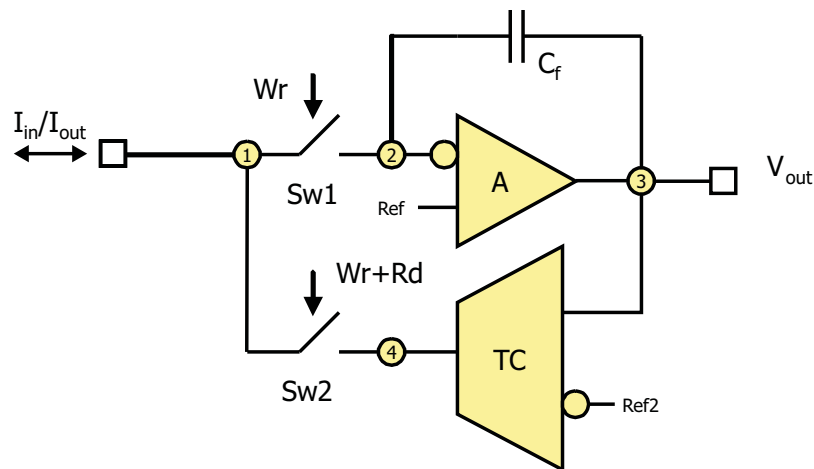


Figure 3-2 Simplified schematic of the current memory cell as it is used in the DCDB's ADC blocks [46].

Consequently, D can take values in the range of $\pm(2^n - 1)$ and there are $n + 1$ bits necessary to represent these values in standard binary format.

Practically, there are limitations due to the quality of the electronics, mainly in terms of noise that is distorting the signal. But also conversion time is a concern, which is obviously increasing with every cycle. Since the ADCs of the DCDB are specified to produce eight bit output codes (binary format), at least seven conversion cycles are necessary.

3.3 The Cyclic ADC Realization

A current-mode realization is chosen for implementing the ADC circuit for the DCDB. This is a quite natural decision, not only because the incoming signal is current-modulated. The arithmetic operations, that are required to implement the conversion, like addition, subtraction and multiplication by two, can be realized very easily in this case. Moreover, this part of the design can be reused from the previous version of the chip, where it is proven to work properly [42]. The basic building block of this implementation is the current memory cell as illustrated in figure 3-2. It is a very simple and clever design. A more detailed description of its implementation is given in section 3.4.2, extensive speed and noise considerations can be found in [42]. For describing the ADC's operation principle, however, a black box view is sufficient. In that sense the current memory cell (CMC) has a current port, where current can be written to

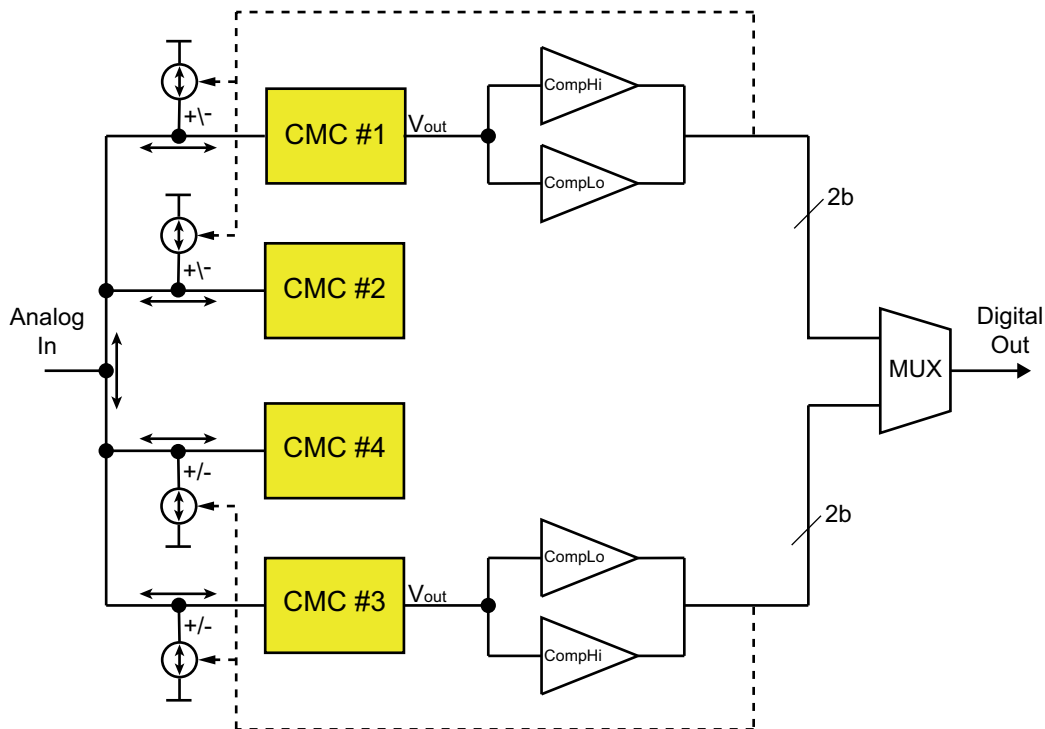


Figure 3-3 Block diagram of the cyclic ADC.

and read from the cell. In addition, there is a voltage output node, which is regulated by an internal amplifier to a potential that is proportional to the stored current.

Using this current memory cell as the main component, the cyclic ADC can be implemented as shown in figure 3-3. The algorithm is performed as follows. In the very first step, the input current introduced via the *Analog In* pin is stored into current memory cell #1 (CMC #1). Once the current is stored there, the current memory cell's current output is used to copy the input signal to CMC #2. Simultaneously, the current memory cell's voltage output is used to feed the recorded signal into two comparators. According to the conversion principle as it was described above, these comparators decide whether the input signal is larger than an upper threshold, smaller than a lower threshold or in-between these two. The result of these comparisons form the first two bit of the conversion result in RSD representation, in fact they are the most significant bits. They are forwarded to the Digital Out pin via the multiplexer MUX. In the next step of the algorithm, the sum of the stored currents of CMC #1 and CMC #2 is copied into CMC #3. Actually, this action performs a multiplication by two of the current stored in CMC #1, since CMC #2 contains simply a copy. Additionally, as instructed by the conversion principle, the comparison results from the step before are used internally for steering current sources at the input nodes of the two current memory cells #1 and #2, possibly resulting in an addition or subtraction of a reference current. From now on, the same procedure as before is carried out by the current memory cells #3 and #4 ending up in the determination of the next two bits of the conversion result. This ping-pong-like algorithm is executed until n comparison results are available, represented in n pairs of bits. After that, the next analog input signal is sampled and the algorithms starts again from the beginning.

3.4 Details of the Building Blocks

It is clear that the analog performance of the DCDB defines its worthiness for the target application. The determination of this performance is subject of later chapters. In order to understand the measurements and results presented there, a deeper understanding of the operation principles and implementation details is necessary. To this end, the major building blocks of the DCDB's analog channel are described in the following. Starting point is figure 3-4, which provides once again a schematic drawing of the analog channel, this time with a lot more details.

3.4.1 The Current Receiver

The current receiver is based on a transimpedance-amplifier circuit with output resistor. It is used to keep the input node of an ADC channel at a constant potential, that is independent from the current flowing into it, at least for a certain range.

Figure 3-5 gives a detailed schematic drawing of the two-stage transimpedance amplifier. Its input stage is implemented as cascode with two supplying current sources, *TCP* and *TCPL*. *TCP* provides a rather high current, that is adjustable in the range of $0.25 - 1\text{ mA}$ (simulated) allowing for a high amplification factor of the input transistor. The cascode transistor together with the *TCPL* current source effect a high output

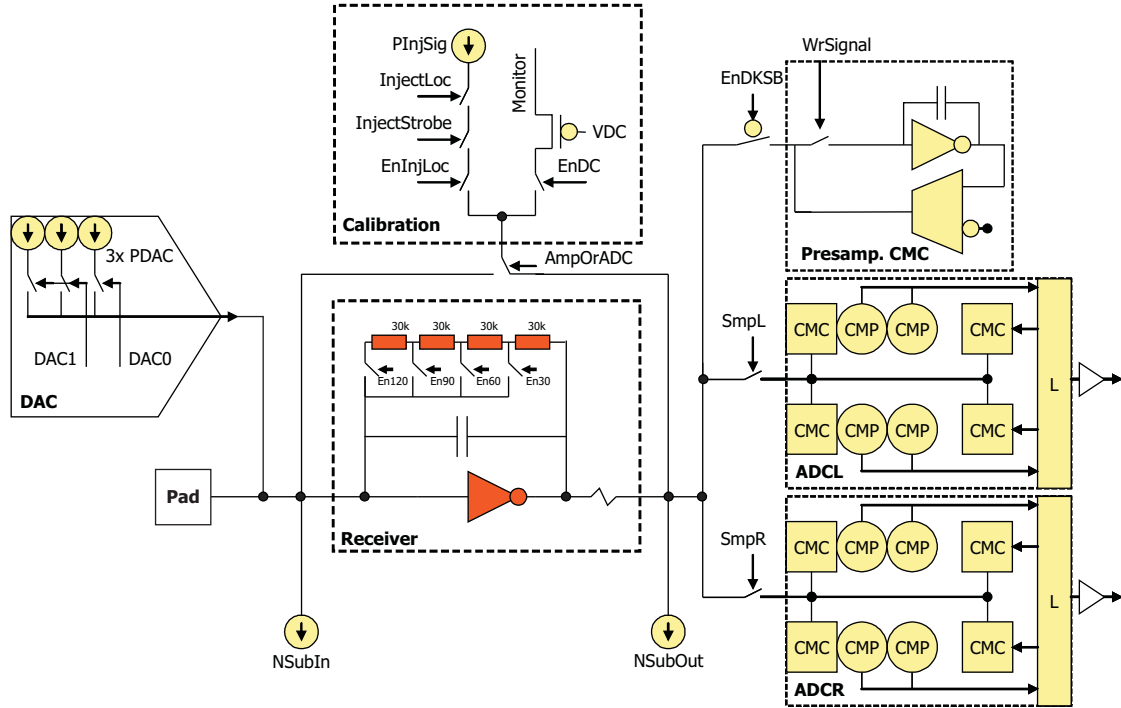


Figure 3-4 Detailed drawing of the DCDB's analog channel [46].

resistance of this first stage. In order to fine-tune the circuit, the TCP and TCPL current sources as well as the bias voltage of the cascode transistor are configurable. Additionally, a variable amount (0 to 4) of $30fF$ capacitors acting a low-pass filter can be attached to the output node, which can help to reduce the noise of the circuit. The *AmpLow* node, which acts as virtual ground node for the input stage, is powered by an extra supply, provided from off-chip. Consequently, its potential can be influenced as well.

The output stage of the amplifier is implemented as source follower in order to provide sufficient driving strength for translating the signal back into a current by the output resistor R_s . There is a switch named *AmpSFON* that can be used to disconnect the output transistor from the supply. Since the sinking current source *TCFSN* is configurable as well, the current receiver can be de facto switched off for test purposes.

The amplifier's feedback path splits up into two parts, a capacitive and a resistive one. Both are configurable in order to influence the characteristic of the circuit. The default capacitance of the feedback path is $60fF$. In addition, up to four extra capacitors of $100fF$ each can be connected by means of programmable switches. The resistive feedback is adjusted using the four switches, *En30*, *En60*, *En90* and *En120*. The feedback resistance is selectable from $30k\Omega$, $60k\Omega$, $90k\Omega$ or $120k\Omega$. On transistor level, this fact results simply in a signal amplification factor (current gain G) of one to four because of the following equation:

$$G = \frac{R_f}{R_s}$$

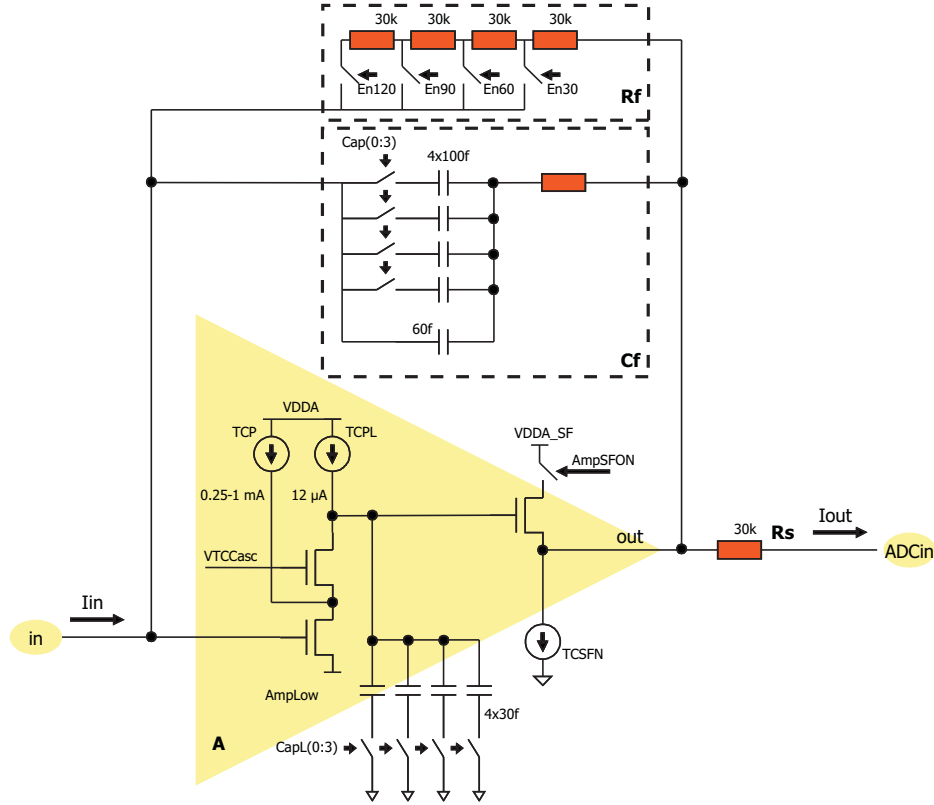


Figure 3-5 Detailed schematic drawing of the transimpedance amplifier [46].

However, there is a considerable effect even on system level as the effective resolution and the effective dynamic range of the analog-to-digital conversion are influenced by amplifying the input signal. The higher the amplification factor is, the higher the resolution gets, but the smaller the dynamic range becomes at the same time. Assuming that the major source of noise within the system is the ADC itself rather than the input current receiver, this feature can really help to improve the overall effective performance.

3.4.2 The Current Memory Cell

A detailed schematic drawing of the current memory cell used for implementing the cyclic ADC is shown in figure 3-6. Its purpose is to store a current in the range of $\pm 8\mu A$ and to keep the input node at a constant potential of 1 V.

Principle of Operation

Node 1 is used as input for the current to be stored, as well as output for the read current. In case of writing a current into the cell, switch Sw1 is closed, allowing the charge to be accumulated onto the capacitance C_f via node 2. The amplifier A generates a potential at its output node (3) that is proportional to the input current. By using this node as input to the transconductor TC, a negative feedback current is generated at node 4, which influences node 1 since Sw2 is closed as well. The storing process is finished once this

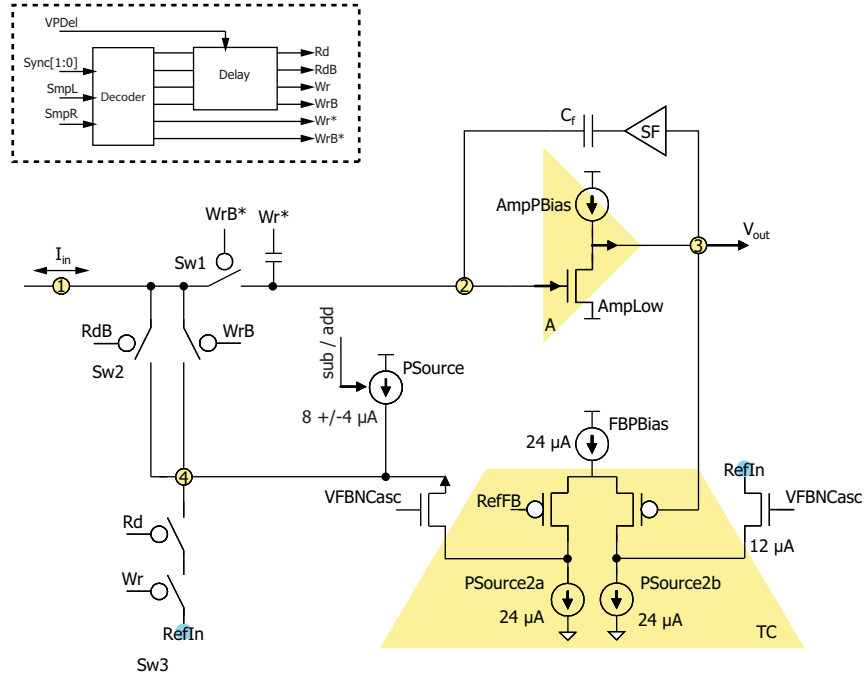


Figure 3-6 Detailed schematic drawing of the current memory cell used to implement the cyclic ADC [46].

system is in balance and the entire input current flows over node 4. The switches Sw1 and Sw2 are opened now, freezing the charge on C_f and thus the potential of node 3. In read state, only Sw2 is closed and the transconductor TC initiates a current through node 1 that is equal to the current that was stored before. If the cell is neither written nor read, Sw3 is used to dump the current through node 4 to the *RefIn* supply.

Implementation Details

The amplifier A is implemented as gain stage with configurable load current source *AmpPBias*. Its ground potential is defined by the *AmpLow* supply. The TC is implemented as a differential pair with one input being fixed (biased via *RefFB*) as well as configurable sourcing (*FBPBias*) and sinking (*PSource2*) current sources. It is intended to have a fixed (but adjustable) current of $24\mu A$ flowing through each of them. Cascode transistors with configurable gate potential are attached to the differential pair's outputs. While one of the outputs is kept at a fixed potential (*RefIn*), the other is fed back to the CMC's input node.

In accordance to the algorithmic principle of the ADC that is intended to be realized with this current memory cell, it is required to provide the capability to add and subtract a reference current in case the cell is in read state. This feature is realized by the *PSource* current source, that is attached to node 4. It provides a constant current of $8\mu A$ (adjustable). Control lines from the two associated comparators within the ADC manipulate this current by $\pm 4\mu A$.

There is a small decoder logic block translating the synchronization signals from the main sequencer into steering signals for the various switches of the current memory cell. In order to improve the cell's sampling accuracy, it is necessary to make sure that Sw1 is

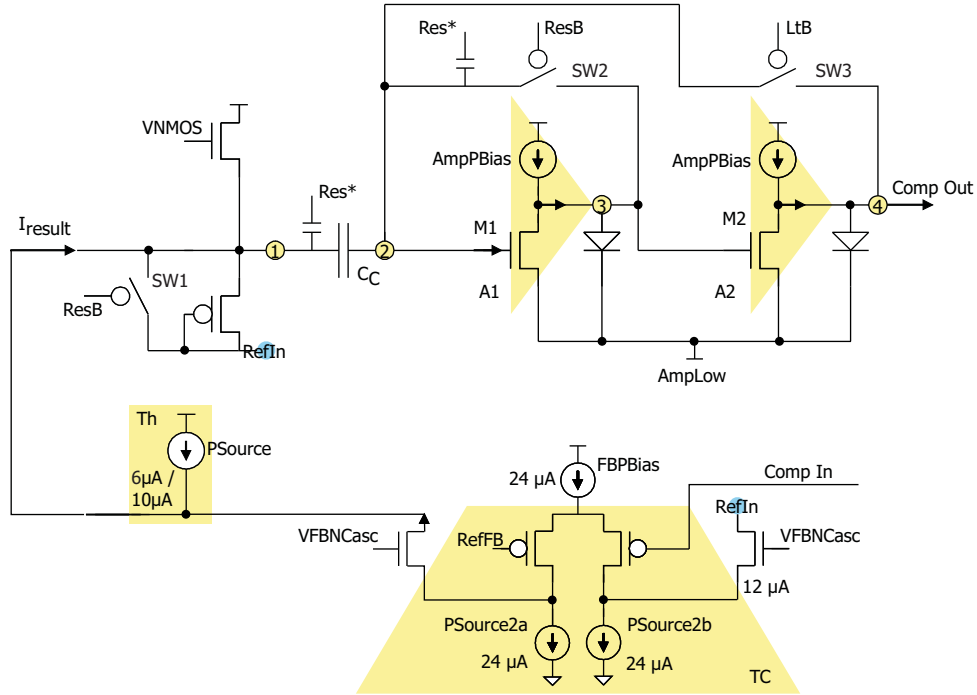


Figure 3-7 Detailed schematic drawing of the comparator cell used to implement the cyclic ADC [46].

opened prior to Sw2 when leaving the cell's write state. To this end, there is a delay element implemented right behind the decoder to delay all the steering signals but those for Sw1 by a few nanoseconds.

3.4.3 The Comparator

Comparators are necessary within the ADC in order to control the current manipulation source inside the memory cell while its stored current is read as well as to produce the analog-to-digital conversion result. Figure 3-7 provides a detailed schematic drawing of such a circuit.

The comparator circuit is built very similar to the current memory cell. In particular, the components of its input stage, which is mainly a transconductor with a current source at the output, are built and biased almost identically to those of the CMC design. The main difference, however, is the current of the PSource current source. Here it delivers a constant current of either $8\mu A - 2\mu A = 6\mu A$ or $8\mu A + 2\mu A = 10\mu A$, depending on whether it is a „too high“ or a „too low“ comparator. In fact, the trick is that by connecting the voltage output of the associated CMC within the ADC (node 3 in figure 3-6) to the input of the comparator's transconductor, the CMC's stored current is compared to the threshold of either $6\mu A$ or $10\mu A$. The outcome of the comparison is expressed by the sign of the resulting current I_{result} .

The heart of the comparator's output stage is a latch-like structure implemented using two cross-coupled amplifiers. It operates in three mutually exclusive states. The „Reset“ state and the „Latch“ state are indicated by the respective controlling signals „Res“/„ResB“ and „LtB“. If non of those two states is active, the circuit is in the „Evaluation“

state. In Reset state, I_{result} is dumped to RefIn via the switch SW1 and the amplifier A1 is kept in an undefined and unstable state by shorting its input and output. When the state is changed from Reset to Evaluate, I_{result} dynamically pushes A1 to either of the two directions. Finally, in the Latch state, this influence is evident via a positive feedback by A2.

3.4.4 Pre-Sampling cell

Although single sampling is decided to be the baseline readout technique for the DEPFET detectors, the DCDB is able to do double correlated sampling as well. This is because there is another current memory cell, the so-called *pre-sampling cell*, connected to the common input node of the two ADC blocks within a channel, as indicated in figure 3-4. In accordance to the double correlated sampling procedure (refer to section 2.1.3), the pre-sampling cell takes the first of the two correlated measurements. After clearing the addressed DEPFET pixel, the second measurement is taken and the subtraction is performed by simply setting the pre-sampling cell to the read state while the ADC is sampling.

3.4.5 Calibration Circuit

In order to be able to do system calibrations, performance tests or eventually debugging, a calibration circuitry is present in every channel. It comprises a current source *PInjSig* and a connection to a monitor bus. The latter is a global signal that is common to all channels. It is connected to a dedicated pad of the DCDB and thus externally accessible. By means of the three switches *EnInjLoc*, *EnDC* and *AmpOrADC* either of the two can be connected to either the input or the output node of the current receiver as shown in figure 3-4. In contrast to all the other configurations, these three switches can be set individually for every channel.

Beside the *EnInjLoc* switch, the *PInjSig* current source has two more activation switches, *InjectLoc* and *InjectStrobe*. While *InjectLoc* is simply another globally configured activation switch, *InjectStrobe* is the reason why this current source can be used for dynamic measurements. In contrast to the other switches, this one is not controlled by any configuration register, but it is connected to an input pad of the DCDB. By means of this the *PInjSig* source can be pulsed externally. Unfortunately, there is no such pad reserved in the footprint of the DCDB for this signal. Thus, the *InjectStrobe* shares a pad with another totally unrelated input signal, which is the TDI of the DCDB's JTAG interface.

The monitor bus can be used in various ways and therefore plays a central role in many DCDB test scenarios. First of all, an input signal from external can be distributed to all the channels by only a single electrical off-chip connection. That means there is no need to contact all the regular signal inputs of the channels, which simplifies the test setup enormously. Besides connecting an external signal source, the monitor bus can be used to probe internal nodes and to calibrate the internal signal sources by measuring the potential at or the current through the monitor bus. In addition to the *EnDC* switch, there is a PMOS transistor acting as cascode in order to optionally protect the internal nodes from the capacitance of the monitor bus. This is necessary, for example, to keep the receiver circuit or the current memory cell able to regulate their input nodes.

By using the monitor signal it is even possible to access the internal nodes of the analog-to-digital conversion circuits. In normal operation mode, the two switches *SmpR* and *SmpL* are used to alternately selecting the ADCs for signal digitization, controlled by steering signals coming from the main sequencer inside digital domain. However, for test purpose these switches can also be controlled statically by configuration registers, if the test mode is activated via the *Enable Test* configuration register. In the same way, the two bit wide Sync bus, that is used for steering the switches inside the ADC's current memory cell (refer figure 3-6), can be controlled, too. This provides insight into these circuits and thus great examination opportunities.

3.4.6 Offset Current Compensation

The ADC circuit used for the DCDB is designed to have an input current range of $\pm 8\mu A$, which is constrained primarily by the expected signal amplitude from the DEPFET detector. However, in order to operate the DEPFET transistors at their optimal working point, an offset current (so-called *pedestal current*) is required, which is expected to be in the order of about $100\mu A$. Consequently, it is an essential feature of the DCDB's analog channel to be able to subtract this offset current prior to any signal processing. In fact there are two mechanisms to cope with this issue.

Constant Offset Current Compensation

In the first order the pedestal current is considered constant and thus is statically subtracted by means of globally configurable current sources. Two of them are implemented in each of the channels, as illustrated in figure 3-4. They are called *NSubIn* and *NSubOut*. The former is the most important one, since it is attached directly to the channel's input node. It is a rather strong current source with a coarse grain adjustment. In an ideal situation, the entire constant pedestal current is subtracted here. If it turns out, however, that some fine-tuning is necessary, the much weaker *NSubOut* current source can be used in addition for that purpose.

Dynamic Offset Current Compensation

In fact, previous measurements showed that the various pixels within a DEPFET matrix show different pedestal currents. The variation of pedestal currents among the pixels of a single matrix must be expected to be in the order of $12\mu A$ and can even increase by irradiation. For a single channel of the DCDB, which is supposed to successively read a certain amount of pixels, this translates into a dynamic variation of input pedestal currents. In order to cope with that, a dynamically controllable compensation mechanism is required in addition. This job is done by the channel's sub-circuit called *DAC*, which is shown in figure 3-4.

The DAC circuit consists of three identical globally configurable current sources *PDAC*. The respective switches that connect the sources to the channel's input node are controlled by the two signals *DAC0* and *DAC1* in the following way. *DAC0* is directly assigned to the activation switch of only one of the sources, while *DAC1* is assigned to the remaining two. That means, *DAC0* and *DAC1* can be regarded as the two bits within a two-bit wide bus, where the assigned binary number directly translates into the

multiplication factor of the unit current that is added to the channel's input node. The unit current in this case is defined by PDAC. The DAC0 and DAC1 signals are received dynamically, that means synchronously to the analog-to-digital conversion cycle, from off-chip via the DCDB's digital block (refer to section 4.2.3).

If the pedestal current for each of the pixels is known a priori, the DAC can be used to compensate for the dynamic fraction of the offset¹. Fortunately, for the application of reading DEPFET pixel matrices, this is indeed the case. As discussed in section 2.1.3, dedicated pedestal measurements for every pixel of the DEPFET matrix are necessary in order to run in single sampling mode, anyway. This pedestal information can be used to determine the correct DAC setting for every pixel individually.

An important implementation detail in the context of dynamic offset current compensation is that the PDAC current sources add a certain current to the channel's input node rather than subtracting it. In general, this is not affecting the quality of the compensation, at least as long as the increased mean offset can be subtracted again. This means that the NSubIn current source must be strong enough to subtract not only the constant fraction of the input offset current but also the dynamic compensation current.

3.5 Configuration Summary

The DCDB's analog-to-digital conversion channel contains a whole bunch of configurable elements. These are screws to adjust the system. In general, they can be separated by the following aspect. On the one hand, there are those configurations that define and adjust the working points for the various sub-blocks of the channel. On the other hand, some are used to select the channel's mode of operation. Table 3-1 summarizes the most important configurable elements and gives a brief description. A complete list of all DCDB configuration registers is provided in [46].

Element	Brief Description	Config. Name	Config. Type
PInjSig	Test signal injection current source.	VPInjSig	7 bit bias DAC
InjectLoc	Global activation switch for the PInjSig current source.	InjectLoc	Register
InjectStrobe	Global dynamic activation switch for the PInjSig current source.	-	External pin: TDI (JTAG data input)
EnInjLoc	Local activation switch for the PInjSig current source, set individually for every channel.	EnInjLoc	Registers. One per channel.

Table 3-1 Summary of the most important configurable elements of the DCDB's analog-to-digital conversion channel.

1. One could think of the DAC circuit as extending the effective dynamic range of the DCDB conversion channel by pushing a larger range of input signal into the real (smaller) input range of the ADC. But it has to be taken into account that some knowledge about the input signal is required in advance.

Element	Brief Description	Config. Name	Config. Type
VDC	The gate of the cascode transistor in a channel's connection to the monitor is either shorted to ground (set) or biased (reset).	VDC	Register
EnDC	Local switch to connect the channel to the common monitor bus, set individually for every channel.	EnDC	Registers. One per channel.
AmpOrADC	Local switch to connect the calibration circuitry either to the input or the output node of the TIA, set individually for every channel.	AmpOrADC	Registers. One per channel
PDAC	Current source that defines the unit current for the dynamic offset current compensation.	VPDAC	7 bit bias DAC
NSubIn	Current source for coarse-grain static offset current compensation.	VNSubIn	7 bit bias DAC
NSubOut	Current source for fine-grain static offset current compensation.	VNSubOut	7 bit bias DAC
EnDKS	Switch for enabling the double correlated sampling mode.	EnDKS, EnDKSB	2 registers, set complementary
TCP	Current source at the TIA's input stage.	VTCP	7 bit bias DAC
TCPL	Load current source at the TIA's input stage.	VTCP	7 bit bias DAC
TCCasc	Cascode transistor at the TIA's input stage.	VTCCasc	7 bit bias DAC
CapL	Four switches. Connect low-pass filtering capacitors to the output node of the TIA's input stage.	CapL[3:0]	Registers. One per switch
AmpSFON	Switch. Activates the TIA's output stage.	AmpSFON	Register
TCSFN	Sinking current source at the TIA's output stage.	VTCSFN	7 bit bias DAC
Cap	Four switches. Add up to four extra capacitors to the TIA's feedback path.	Cap[3:0]	Registers. One per switch
En30, En60, En90, En120	Four switches. Define the TIA's resistive feedback path.	En30, En60, En90, En120	Registers. One per switch
AmpPBias	Load source of the amplifier that is used for current memory cell and comparator.	VAmpPBias	7 bit bias DAC

Table 3-1 Summary of the most important configurable elements of the DCDB's analog-to-digital conversion channel.

Element	Brief Description	Config. Name	Config. Type
FBPBias	Head current source of the transconductor's differential pair. (Used for CMC and comparator.)	VFBPBias	7 bit bias DAC
PSource	Output manipulation source of the CMC. Also used for the threshold of the comparator.	VPSource	7 bit bias DAC
PSource2 (a/b)	Tail current sources of the transconductor's differential pair. (Used for CMC and comparator.)	VPSource2 (Same config. for both sources)	7 bit bias DAC
PDel	Delay element of the CMC's control signals.	VPDel	7 bit bias DAC
PSourceCasc	Cascodes within the PSource output manipulation sources of CMC and transconductor.	VPSourceCasc	7 bit bias DAC
FBNCasc	Cascode transistors of the transconductors.	VFBNCasc	7 bit bias DAC
RefFB	Second (fixed) input of the transconductor's differential pair.	VRefFB	7 bit bias DAC
NMOS	Potential protecting circuitry within the comparator.	VNMOS	7 bit bias DAC
VDDA	Main supply of the DCDB's analog domain.	-	External supply voltage.
AmpLow	Ground node for the TIA's input stage and the amplifiers in CMC and comparator.	-	External supply voltage
RefIn	Reference Potential.	-	External supply voltage

Table 3-1 Summary of the most important configurable elements of the DCDB's analog-to-digital conversion channel.

The Digital Domain of the DCDB

Abstract:

Together with the previous one, this chapter completes the detailed report on the DCDB design, as it is focused on its digital domain. The chapter starts with a general discussion about what and how it is implemented. After that the entire digital development process from the logic description and simulation to the physical implementation is explained.

4.1 General Considerations

As the DCDB's main business is the digitization of analog input signals, the analog-to-digital conversion channel is certainly the most important building block. However, although the ADC is already producing digital output, there is still some work to do before the analog channels can be operated properly and the data can be sent off the chip.

4.1.1 Digital Tasks

First of all, the fact that the ADC is producing digital data in a redundant representation needs to be addressed. Simply sending this data off the chip as it is produced by the ADCs would waste transmission bandwidth. So it is a quite obvious idea to convert the data into a non-redundant data format first. The straight-forward way is to implement some conversion logic that produces standard binary code.

Secondly, the available chip area and also the available routing resources on the half-ladder module limit the amount of digital output pads. An early floorplan of the DCDB showed that beside the 256 analog inputs, power supplies and debugging connections, 64 digital data output pads are feasible. So a serialization of the produced data is necessary that maps the analog channels to the available outputs. This mapping strategy, however, in any case holds a trade-off that needs to be faced. As described in section 3.2, the analog channels produce n pairs of bits within n operation cycles. After the conversion to standard binary format, this is equivalent to $n + 1$ bits of data, still produced within n cycles, which is a quite odd relation. There are two main options of coping with this situation.

1. The easiest solution is to discard the least significant bit. This results in a more natural relation of one produced bit in standard binary format per cycle. But obviously, there is the drawback that the analog to digital conversion accuracy is reduced by $1/n$.
2. Increasing the data processing speed by $1/n$ is a way to overcome this odd relation of data production per cycle. This, however, results in a much more difficult digital design due to the introduced clock domain crossing between clocks with odd relation, at least for the relevant values of n . Furthermore, the mapping of n bit data words to 64 output pads might be odd as well, which results inevitably in unused transmission bandwidth. Nevertheless, it would also be a feasible method.

The DCDB was specified to produce conversion results represented by eight bits of data each. So either option one with eight analog to digital conversion cycles, or option two with only seven conversion cycles could be used. It is decided to favour option one, coping with the higher number of analog to digital conversion cycles, but for the sake of an easier digital design and serialization strategy. The latter is determined as follows: The 64 output pads are split into eight groups of eight pads each. The 256 data producers are split accordingly into eight groups. The eight bit data words of the 32 producers within a group are transmitted in parallel via the associated eight pads. So the frequency on these busses is determined to be 32 times the sample rate of the analog channels, which is $320MHz$.

The third purpose for digital logic within the DCDB design is managing the data input. In fact, the analog channels need both, dynamic data input for controlling the dynamic offset compensation circuitry as well as static configuration data. The same problem of limited interconnection as for the data output lines holds here, too. Especially for the dynamic data input, as only 16 pad were reserved for this purpose during the floorplanning. Digital deserialization is used here in the same arrangement as the serialization of the data outputs. For configuration purpose, internal shift register chains encapsulated within a JTAG compatible interface are implemented. JTAG offers a clever way to combine multiple configurable devices into a single chain and is therefore the favoured configuration protocol for all the chips on a half-ladder module. Although there are no really strong arguments, JTAG was favoured over other protocols like I²C or SPI.

4.1.2 From Full Custom to Synthesized Digital Logic

Historically, all the DCD versions prior to the DCDB (especially the DCD2 [42]) and also the predecessor ASIC called CURO [21] were entirely full custom designs. Large fractions of the handcrafted digital logic, particularly for data format conversion, were placed very close to the analog channels. Later on, these islands of digital logic, spread all over the design, were believed to induce noise and therefore degrade the system's performance. This led to the idea of separating analog electronics from digital logic completely, forming a single large digital logic block on the one side of the chip, while placing the analog channels on the other side. For the DCDB, however, with its 256 analog channels, developing such a large digital logic block with the methods of full custom design would be a very cumbersome task. A considerable alternative is to make use of digital implementation methodologies based on synthesized designs. This is indeed a very attractive option, due to the advantageous side effects that come along with it. Apart from the extremely enhanced development speed, there is first of all the easier way of committing changes of the implemented logic which makes the design much more flexible. Secondly, the simulation of the design, not only the digital logic but the DCDB as a whole as well as the entire DEPFET readout electronics chain is the key to success for the DEPFET vertex detector project. With a full custom design, analog simulation methodologies would have to be applied for its verification. For simulating digital designs, however, this is like taking a sledgehammer to crack a nut. Using appropriate digital simulation techniques instead, brings a much higher level of abstraction. This results in much fast runtimes and therefore offering the possibility of a much more comprehensive verification coverage of the design. Finally, these considerations lead to the decision to synthesize the DCDB's digital logic block.

4.1.3 Revision History

The first version of the BELLE-II-type DCD chip, the DCDBv1, was developed in autumn 2009. In the mean time up to summer 2011 new improvement ideas mostly concerning the design's analog performance lead to the decision to produce a small-size test chip (DCDB-TC) with a reduced number of channels and later on to resubmit the full-size design, which is named DCDBv2. Each of these chips is of course equipped with a digital logic block, individually adopted in terms of size and channel count. Due to the development time of about one and a half years with three chip submissions,

however, there were also a lot of opportunities to enhance the digital design parts. This is mainly because of improvements of the used software tools as well as the skills in working with these. So throughout this chapter, differences between the various versions are denoted.

4.2 Logic Development

4.2.1 Data Format Conversion

Due to the specific cyclic implementation of the DCDB's analog-to-digital converters, its output code contains redundant information. Converting the data to a non-redundant format, such as standard binary, is therefore highly desirable in order not to waste the limited output bandwidth. The conversion is implemented using a very simple finite state machine with only two states as it is proposed in [45]. This algorithm processes the digits of the ADC output code, represented with two bits each, in a sequential manner, which is a very natural way since the digits are produced sequentially by the cyclic ADC. Hence, the conversion can simply be regarded as a single pipeline stage. However, there is a drawback due to the fact that the data must be provided to the state machine with the least significant digit first, while the data stream coming from the ADC is just the other way round, that is most significant digit first. To this end, the 16 bits of ADC output code need to be registered entirely, before the conversion can even start.

The corresponding logic is illustrated in figure 4-1. The data arriving from the ADC is stored sequentially into a bi-directional shift register, either from the left or from the right side. After eight cycles, when the entire analog to digital conversion result is stored, the shift direction flips. That means the following data from the ADC is shifted in from the other side, while simultaneously the previously stored data is shifted out in the reversed direction, namely with the least significant digit first. In that way, it can easily be processed by the subsequent state machine.

The state machine operates as indicated by the graph in figure 4-1. When beginning a new conversion, the state machine must be in state A. The given logic expressions determine the valid transition for a respective input digit, as well as they provide the corresponding bit of the conversion result. They are interpreted in the following way:

$$(in, out) \quad in \in \{-1, 0, 1\} \quad out \in \{1, 0\}$$

in refers to the input digit, while *out* indicates the conversion result bit. Since the data is processed starting with the least significant bit, the output is generated in the same way. After having processed all such redundant digits belonging to the same value, the final state of the state machine can be interpreted as the result's most significant bit (MSB). If it stops in state A, the MSB is assigned a 0, otherwise it is assigned a 1. That is equivalent to the sign bit of the conversion result, as it is represented in two's complement.

The clock frequency for the conversion logic is determined by the ADC operation speed in order to have both blocks working synchronously. The analog sampling rate is designed to be $100ns$. Since two ADCs are working in parallel within each analog channel, the effective sampling rate per ADC is $200ns$. The eight output digits per ADC

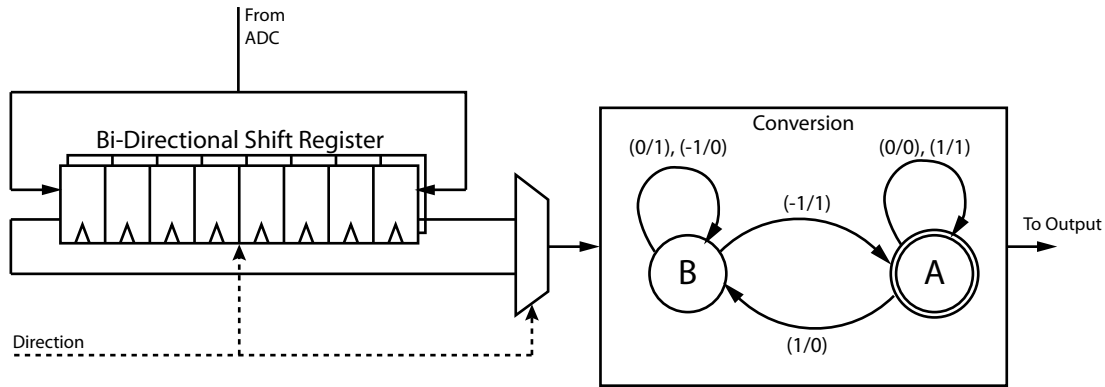


Figure 4-1 Schematic of the data conversion logic. It is used for translating the ADC output into a non-redundant format.

must be processed within this time, resulting in $25ns$ per digit and accordingly a clock frequency of $40MHz$.

4.2.2 Output Serialization

Right after the conversion into two's complement format, the data needs to be sent off the chip to the DHP for first order data processing. Since, however, the number of off-chip connections is limited, a serialization needs to take place. It is decided to combine the outputs of 32 analog channels on a single eight bit wide output link of the DCDB. This results in a two-step serialization. The first step derives from the fact that the two ADCs within an analog channel are working alternately. As a consequence, the 64 conversion logic units assigned to these ADCs are divided into two groups that finish their conversion cycles alternately as well. Hence, with a period of $100ns$ either of the two groups forward their results to the serialization unit. The second serialization step is then multiplexing the 32 bytes of data to the output bus.

The clock frequency of the serialization logic is derived from the need to transfer 32 bytes of data on a one byte wide link within $100ns$. This is $3,125ns$ per data byte and $320MHz$ clock frequency accordingly. With the chosen microelectronics technology, this output frequency is manageable even with standard logic cells. There is no need for special high speed transceiver elements. However, designing the serialization architecture that is working at that speed is not really a trivial task. In a first order approach, one would try to implement a large multiplexer structure as illustrated in figure 4-2 a). This would end up in a huge net of combinatorial logic that is not able to operate at the target frequency. A more sophisticated architecture is presented in figure 4-2 b). By using a fast shift register chain instead of a single register, the large multiplexer structure can be broken down to a set of fairly small multiplexers with up to three inputs. Since this approach meets the requirements, it is the favoured implementation for the DCDB.

Still a potentially critical task is the clock domain crossing that cannot be avoided by any serialization architecture. While the data is produced in the $40MHz$ clock domain, the serialization must obviously be implemented in the $320MHz$ domain. In this special case, however, the involved clock signals are related, since they are generated on the

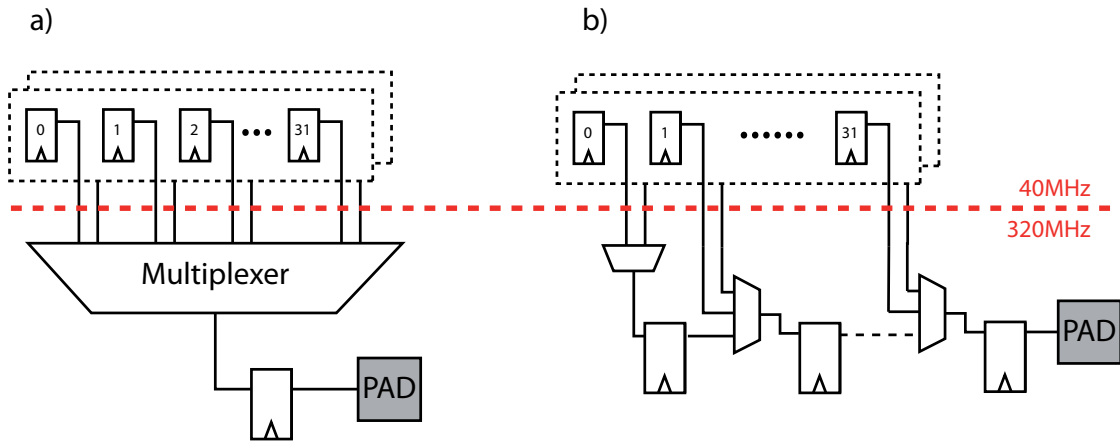


Figure 4-2 Illustration of two different serialization architectures. a) Single multiplexer structure that is serializing all the input signal onto a single output register. b) Fast shift register that can be loaded in parallel.

chip from a common clock source. Consequently, the issue of synchronizing these signals is postponed to the physical implementation step, where it needs to be addressed again (refer to section 4.5.6). During the design phase, in turn, they can be considered as synchronous.

4.2.3 Input Value Distribution for the Dynamic Offset Compensation

The current sources for dynamic offset compensation residing in each analog channel need to be provided with two bits of data for each conversion. That data is received from off-chip via a two-bit wide bus per group of 32 analog channels. So a deserialization inside the digital domain is necessary. For its implementation the same strategy as for the output data serialization is applied, just with the data flowing in the opposite direction. Each of the two input bits are fed into a 32 bit long shift register. Once a set of 32 bits is sampled, the contained data is copied into a mirror register from where it is distributed to the associated analog channels.

4.2.4 ADC Control Sequence Generation

The algorithmic implementation of the ADCs used in the DCDB's analog channels requires some steering activity from external in order to operate properly. A detailed description of the steering signals, as well as a timing diagram showing the sequence to be driven on them is given in [46]. In principle, these steering signals could be provided from anywhere, even from off-chip or individually generated inside every pixel. Certainly, the most advantageous location is inside the common digital block of the DCDB. In that case, additional off-chip signals are avoided and the electrically disturbing digital logic inside the analog channels is reduced to the minimum. Furthermore, if the ADC steering logic is a part of the DCDB's digital logic block, it is by default a part of its simulation environment and therefore simplifies the verification of the synchronization between the ADCs and their associated data conversion units.

In order to meet the timing requirement of the ADC's steering sequence, a third clock domain running at a frequency of 80MHz is necessary. Again, this clock is generated on-chip, assuring a certain relationship to the other clocks on the chip.

4.2.5 Clocking and Resetting Scheme

The DCDB's entire operation is synchronized by means of only two signals, the `BITCLK` and the `SYNC_RESET` signal. Via `BITCLK`, a 320MHz clock signal must be provided to the DCDB. While the serialization logic uses this clock directly, the two derived clocks of 80MHz and 40MHz for the ADC steering sequence generation and the conversion logic are generated internally. Simple counter circuits are used as clock dividers. A proper phase relation between all these clocks is assured by the physical implementation tool as described in section 4.5.6.

The `SYNC_RESET` signal is used as synchronous reset for the entire digital logic of the DCDB. It affects all controlling registers in every clock domain. In particular, the `SYNC_RESET` signal is also used for resetting the internal clock divider circuits, which is very important for keeping all clock phases synchronized. Additionally, due to the fact that the `SYNC_RESET` signal is synchronous to the `BITCLK` input clock, it must be synchronized into the 80MHz and 40MHz clock domains. Here, it is a very critical task to really make sure that all the local (that means synchronized) reset signals of the clock domains are released at the same time. All this is again very challenging for the physical implementation tool that needs to keep track of all the relative signal runtimes to the various end nodes all over the design.

The DCDB's entire digital logic, including the data conversion, ADC steering and serialization units turned out to have a period of 128 `BITCLK` cycles. That means, the logic state of the entire DCDB digital block is the same, namely the reset state, every 128 cycles. This fact can be utilised to implement a very simple mechanism to recover from temporary radiation effects like single event upsets (SEU). In general, their occurrence during the operation of the DCDB is very likely, since its environment, which is the BELLE-II detector, is a very harsh one for electronic devices. The strategy is to pull the synchronous reset signal every 128 cycles. In normal operation, this would not affect anything at all. However, if for example a bit flip occurs in any of the chip's control registers, the logic state recovers in not more than 128 clock cycles.

4.2.6 JTAG Configuration and Debugging Interface

The DCDB provides a JTAG compatible¹ interface for configuration and debugging purpose. There are two separate register chains for configuring the analog channels. The first one contains registers for global settings, influencing all analog channels at once. The second one is used for those configurations that can be set individually for each channel. For both chains, the digital block only provides the interface to the register chains. That means it provides input to the first register as well as it receives the output of the last one. The registers themselves are full custom implementations in the analog domain for simpler reuse of former designs during the implementation phase of the chip.

1. The JTAG standard defines the JTAG reset signal (TRST) to be low-active. There is a bug in the DCDBv1, as this signal is designed as high-active. The bug is repaired for the versions DCDB-TC and DCDBv2.

Additionally, there is a boundary scan chain along all digital off-chip connections (except the JTAG signals) plus a single internal one. The first version of the DCDB, that is DCDBv1, provides only the *EXTEST* functionality of the boundary scan chain as defined in the JTAG standard [47]. That is necessary for electrically checking the off-chip connections. The *SAMPLE/PRELOAD* functionality for the validation of the chip's internal logic via JTAG is available for DCDB-TC and DCDBv2. Further technical information about the DCDB's JTAG interface can be found in [46].

4.2.7 Digital Test Signal Injection

In order to simplify the testing of the chip, a digital test signal injection is implemented in addition to the DCDB's normal operation logic. Extending the testability provided by the JTAG interface, the chip is able to provide a well-defined pattern at the digital data outputs. It is implemented by simply having a multiplexer at every data input from the analog channels, that selects either the real data from the analog-to-digital conversion or a certain constant value. That constant value is for simplicity fixed during the physical implementation and therefore unchangeable.

By injecting the test data right at the border between the analog and the digital domains of the chip, large fractions of the digital logic are involved in producing the output pattern and therefore covered by this test. The constant input values are selected on the premise that after conversion and serialization all the data output buses are distinguishable and perform at least partially a toggling signal with the period of only one clock cycle¹. The latter can then be used for the determination of the chips's maximum operation frequency.

4.3 Verification

A very important and often even very time-consuming part of ASIC development, including both analog and digital electronics of course, is the functional verification by means of simulation. This holds in particular for the DCDB. In fact, its digital logic block is of rather low conceptual complexity. Nevertheless, the simulation effort is considerable, not least due to the fact that the DCDB is a mixed signal design, having analog and digital parts on the same chip. This requires special and rather new simulation techniques that are sometimes quite hard to handle.

The availability of adequate software tools for performing these simulation tasks at the Chair of Circuit Design, Heidelberg University, is indeed very acceptable. Due to a participation in a partnership program with *Cadence Design Systems Inc.*, the students and employees can have direct access to the newest design and verification products of that company. Hence, the *Incisive Unified Simulator* verification suite in the versions 8.2 and 9.2 is used.

1. Both validation capabilities of the output pattern hold for DCDB-TC and DCDBv2, but only for a subset of output buses of the DCDBv1.

4.3.1 Digital-Only Functional Verification

Simulating sub-blocks concurrently with the development of the digital logic is simply a very common technique of designing. Small stimuli generators are used to produce input to a certain piece of logic, while the output verification is done manually by investigating the output waveforms. In that way, designers very quickly get an idea about whether the logic is behaving as expected or not. However, the functional coverage of such kind of tests is rather low, since typically a very special type of input stimuli is generated. Non-automated waveform checking is also a quite error-prone method. Furthermore, the flexibility of such generators is rather low. That means for every new type of input sequence the generator must be created almost from scratch again.

The DCDB's digital logic is developed in the same way. But in order to make absolutely sure that the design is really doing its job correctly, much more sophisticated verification techniques must be applied once the design phase is completed. To this end, a full-blown verification environment based on the SystemVerilog hardware verification language in combination with the *Open Verification Methodology* (OVM) is developed.

The Open Verification Methodology

OVM [48] is a methodology-based class library for SystemVerilog that aims for enhancing both, the effectivity and the productivity of hardware verification environments. Historically, the first version of OVM class library was released in 2008 by the two EDA¹ tool vendors *Cadence Design Systems* and *Mentor Graphics*. Before that time, the tools of each vendor had their own verification strategy, preferred language and class libraries, without being compatible to each other. OVM was then the result of the first efforts of those two companies to unify their systems. While in the early days working with that class library was rather hard [49] due to very poor tool support and availability of documentation, OVM has evolved over the past years. Nowadays, the library is released in its second version where bugs are fixed, documentations are available from many sources and the tool support has improved as it is integral part of the latest revisions of the simulation software suites. However, since summer 2010, the story of OVM is continued under the name *Universal Verification Methodology* (UVM) [50] for which even the third big EDA tool vendor, *Synopsys*, has already announced support.

The DCDB's OVM Verification Environment

The basic idea of an advanced verification environment is not to let the designer explicitly define the input stimuli for the Design Under Test (DUT), but to generate them autonomously. In that way the verification is much more flexible, while in general improving the functional coverage at the same time. In the first order, the input stimuli are generated randomly, producing any possible input sequence. Depending on the DUT, this can be useful, since either any input sequence is a valid one or the DUT is supposed to handle even invalid input sequences on its own. In most cases, however, having a really random input to the DUT is not leading to the desired results, at least not in a reasonable time. Therefore, it is possible to specify constraints that must be met by

1. Electronic Design Automation

randomly generated stimuli in order to be driven on the DUT's interfaces. This can be as simple as defining a range of valid values for a certain input bus, but complex sequences and protocols can be defined as well. This methodology is known as *Constraint-Random Testing*.

The DCDB verification environment implements Constraint-Random Testing using OVM in the following way. Each of the DCDB's interfaces, such as the JTAG interface or the ADC raw data input interface, is assigned to a pair of so-called *driver* and *monitor* units (sometimes having only a monitor is sufficient). While the driver unit drives the control signals of a certain interface, the monitor records and interprets all actions on that interface. Each driver is assigned to its own sequence generator, having the random data generator implemented and the corresponding constraints defined inside. The monitor, in turn, forwards the collected interface actions to a common scoreboard. The scoreboard keeps track of all interactions and implements the algorithms to decide whether the DUT is behaving correct or not.

Due to the existence of the boundary scan chain along the DCDB's digital I/Os, those signals must be verified in two separate modes, resulting in two separate configurations of the verification environment. First of all, there is the regular functional mode and its appropriate verification environment as illustrated in figure 4-3. It verifies the data flow in both directions. That is on the one hand the flow from the ADC raw data input through the conversion logic and the serializer to the high speed output link. On the other hand there is the high speed serial input and deserialized output for the data transfer to the pedestal compensation circuit (denoted *Offset DAC*). The JTAG interface is connected here as well, but only in order to ensure that the boundary scan cells are transparent and do not disturb the simulation.

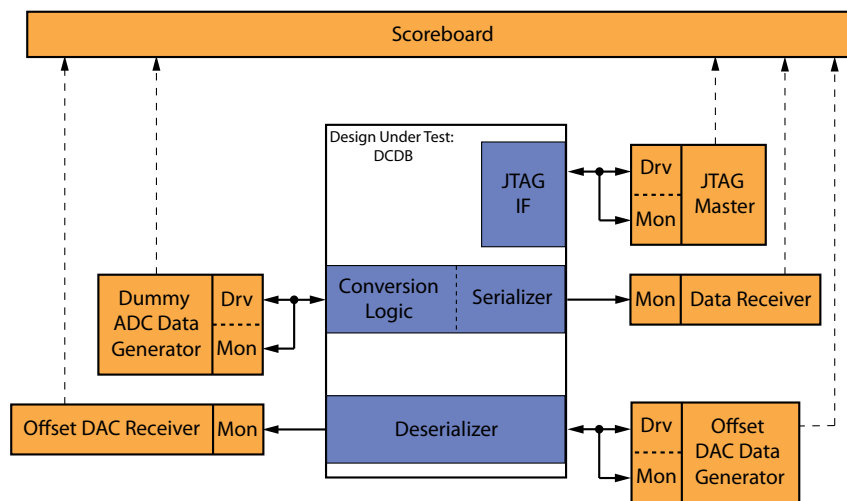


Figure 4-3 Block diagram of the OVM-based verification environment for the DCDB in functional mode.

Secondly, the DCDB can be operated in test mode using the JTAG interface only. The verification environment for that case is presented in figure 4-4. The JTAG master issues commands to the DUT in order to read and write the various register chains. First, there are the two configuration chains for the ADCs which, however, are actually part of the analog domain and must therefore be replaced each by a dummy register chain within the verification environment. In addition, there is a boundary scan unit connected to all the inputs and outputs included in that chain. It is used to verify the EXTEST functionality of the boundary scan chain. Unfortunately, it is very hard and cumbersome to check also the SAMPLE/PRELOAD functionality in this configuration. The easiest but rather unattractive way to do so is to exclude the JTAG interface together with the boundary scan chain from the rest of the DCDB design, gaining access to the internal signals that connect the boundary scan cells to the core logic. This, in turn, requires again a separate version of the verification environment that is not shown here.

In both verification modes the scoreboard is used to check whether the data is flowing correctly through the design. The results are printed onto the simulator's console window as presented in figure 4-5 and figure 4-6. In fact, this is not a trivial task, since it inevitably results in a kind of re-implementation of the DCDB's logic inside the verification environment. From the technical point of view, this is not as bad as it seems, since the SystemVerilog language specifies much more powerful, software-like expressions that make the designer's life a little easier. These language elements are mostly not synthesizable, but that is not important at all for the description of the verification interface. In contrast, for the verification environment's quality this fact might be judged as a drawback, because the designer could tent to reuse parts of the DUT design for the verification environment, leading to the situation where the same mistakes are done on both sides and so cannot be detected. The best practice approach to overcome this drawback is to have the DUT and the verification environment created by different designers. It is, however, a matter of fact that this is not always possible, in particular for the DCDB design. In that case, the designer must be fully aware of that risk and avoid any mixture of the two parts.

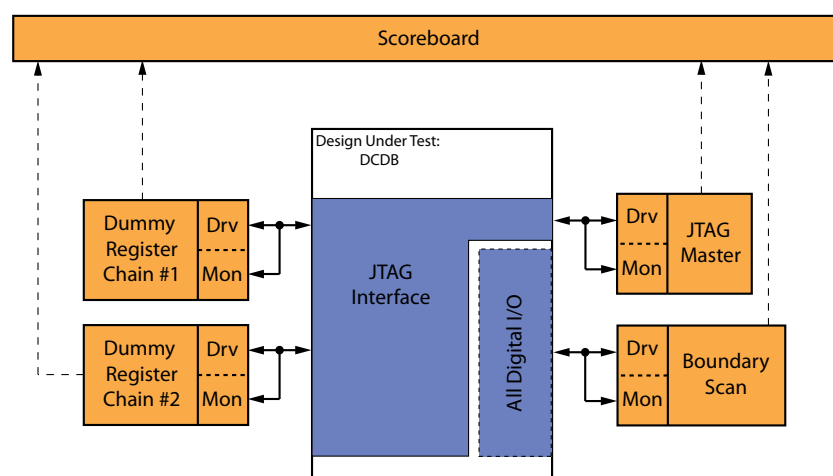


Figure 4-4 Block diagram of the OVM based verification environment for the DCDB in test mode.

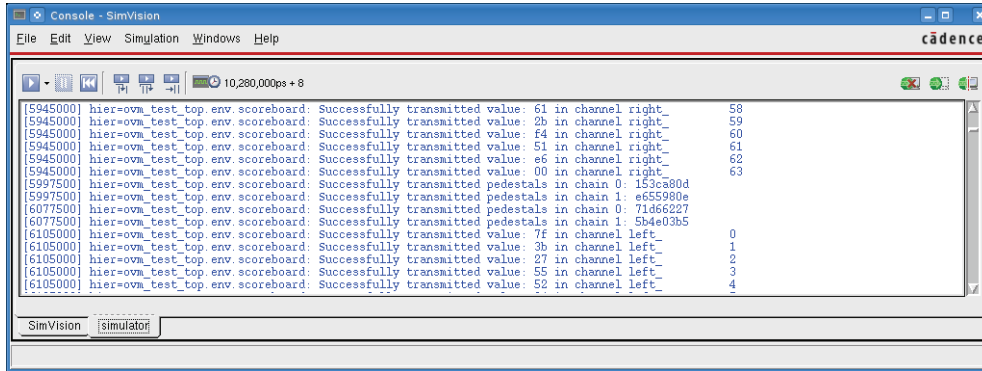


Figure 4-5 Text output on the simulation tool's console window that is produced by the scoreboard during the verification of the DCDB's functional mode.

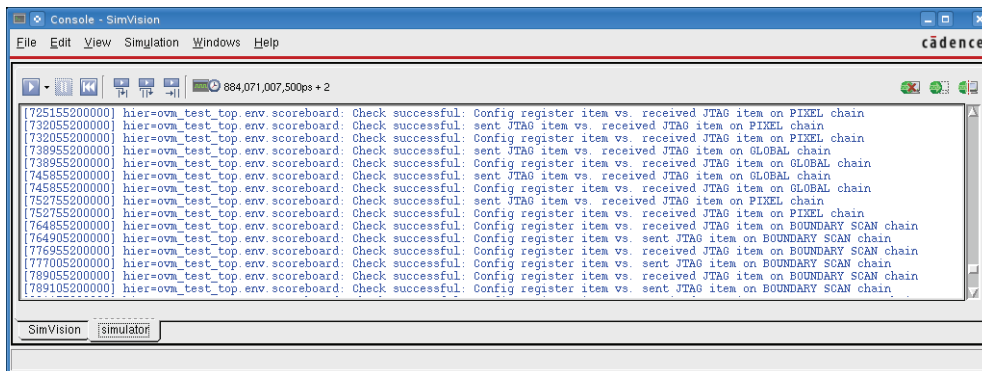


Figure 4-6 In contrast to figure 4-5, this picture shows the text output generated during the verification of the JTAG interface and the associated register chains.

4.3.2 Mixed-Mode Simulation

Once the digital logic is proven to work as expected, at least all parts except from the ADC control sequence generator, which has not been focused so far, the next step of the verification process can be taken. That is a combined verification of the digital logic together with the analog circuits of the ADC channel. In fact, state-of-the-art hardware verification tools are capable of combining the digital and the analog world within the same so-called *mixed-mode simulation*, allowing both parts to interact with each other. Nevertheless, really setting up such kind of simulation environment can be a cumbersome and time-consuming task.

In general, there are two different ways of setting up a mixed-mode simulation environment, the analog-on-top flow and the digital-on-top flow¹. The two flows are distinguished simply by the tool that is used to create them. For the analog-on-top flow digital design parts are imported into the analog design environment, while for the

1. This holds at least for the Cadence tool flow.

digital-on-top flow the situation is vice-versa as the analog design parts are imported into the digital design environment. On first sight, this distinction might be rather nebulous. But indeed, the design techniques for digital and analog electronics are completely different, and so are the use models of the corresponding tools. Mainly, the two flows exist only for the comfort of the designers, as they can work with the simulation tool they are familiar with. Anyway, under the hood both flows result in the same execution. The analog simulator is used for the analog components, while the digital simulator processes the digital parts.

A very important technical detail of mixed-mode simulations is the modelling of the interfaces between the two domains. The need for some kind of conversion is pretty obvious, since digital simulation typically operates on logic states like „high“, „low“ and sometimes also „tri-state“, while in the analog world there are continuous values of currents and voltages. In fact, currents are not directly translatable, but the conversion of voltages to logic states and vice versa is straightforward. In the easiest case, there is simply a single threshold value deciding whether a certain voltage value is translated into a „high“ or a „low“ state. Technically, the conversion is implemented inside the so-called *connect modules*, that can be instantiated either manually in the code or automatically by the tool. The former method leaves the decision which connect module to use for the designer, while the automatically instantiated modules can only be selected from a set of predefined ones. In any case, at least for the Cadence flow, the connect modules are described using a special mixed-mode programming language called *VerilogAMS*. The designer is free to create custom-made connect modules by means of VerilogAMS, having control over even more complex signal translation properties like transition time and so on.

The concrete DCDB mixed-mode simulation environment is kept rather simple, as illustrated in figure 4-7. It is set up using the digital-on-top flow. An ideal current source, which is modelled in VerilogAMS, is used for generating an analog input signal to the ADC channel. The ADC channel is represented by an analog netlist file, which is generated from the real analog ADC design. That means the ADC used for simulation is not just any ADC model, but it is the real ADC that is implemented inside the DCDB. However, only a single ADC channel is used, in order to keep the simulation time reasonable. The digital logic block is attached to the ADC channel for steering and data conversion purpose using the simulation tool's default connect modules. A simulation test bench surrounding that setup provides appropriate infrastructure signals for the digital logic block, like clock and reset, as well as steering input to the signal generating current source. The resulting waveform output is illustrated exemplarily in figure 4-8.

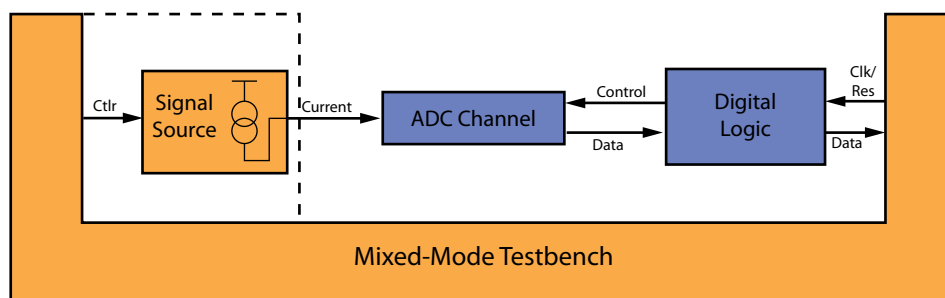


Figure 4-7 Block diagram of the DCDB's mixed-mode simulation environment.

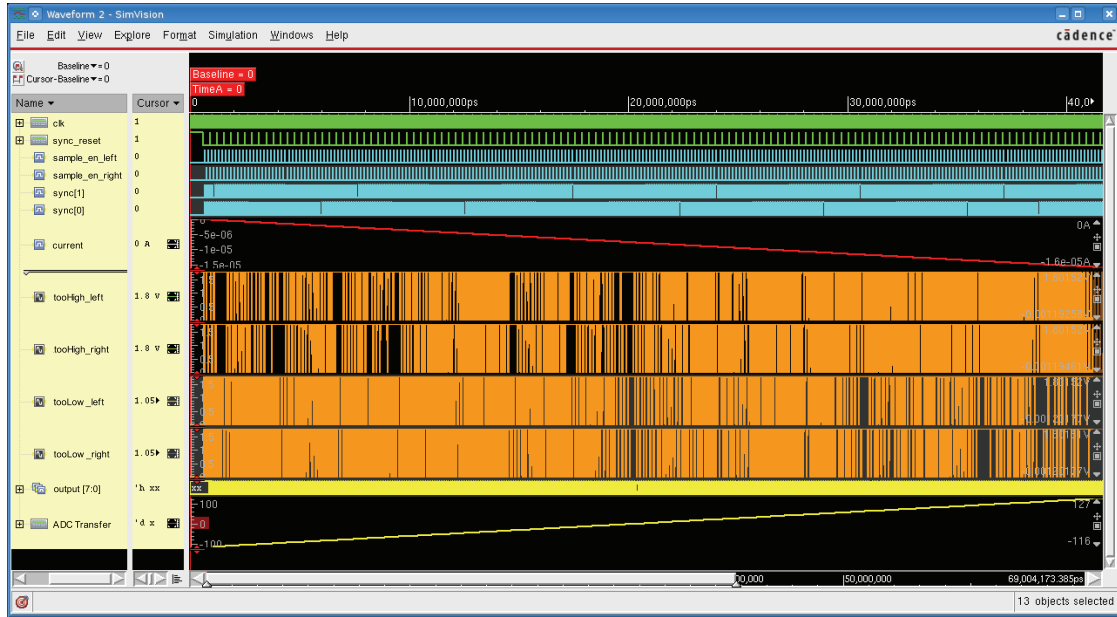


Figure 4-8 Waveform snapshot showing the DCDB mixed-mode simulation. Beside the infrastructure and ADC steering signals, the snapshot shows also the input signal current, the corresponding ADC outputs and the resulting ADC transfer curve after the conversion by the digital logic.

4.3.3 Concluding Remark

The two simulation environments described so far provide an exhaustive functional verification of the DCDB's digital logic block. After having performed these simulations successfully, one can consider it as functionally correct and continue with the next step in the design process, which is the physical implementation. However, once finished even with that, it is best practice to return to the design simulation again for the so-called *Post-Place-and-Route simulation*. Up to now, only the design's logic functionality is proven to be correct. But in order to make sure that its translation into physical hardware is working as well, in particular at the target operation speed in terms of clock frequency, the design description must be extended by the delay information that are obtained not before the physical implementation is done.

4.4 Standard Cell Library Development

As the first step on the way to the DCDB's physical implementation, two decisions must be made. The first one is about the microelectronics technology to be used for implementing the chip. It is obvious that for a chip like the DCDB, which combines analog and digital components on a single substrate, both domains use the same technology. So it must fit to both demands. Secondly, in conjunction with the chosen

technology, a standard cell library for implementing the digital logic in semi-custom design flow has to be selected. Both decisions are very critical, since they are made very early in the design process, in particular for the analog domain, and have enormous impact on the final performance of the chip.

Indeed, the technology selection for the DCDB is rather pragmatic. Over several years the analog design experts at the Circuit Design Group of Prof. Dr. Peter Fischer, Heidelberg University, gained expertise in using the 180nm Mixed-Mode and RFCMOS 1P6M technology from United Microelectronics Corporation (UMC). In particular, the DCD2 was designed with it, which can fairly be considered as a proof of principle for the DCDB. The measurement results published in [42] show that the requirements can be met using that technology. Furthermore, large fractions of the design, at least for the analog domain, can be reused, saving both development time and costs.

Once the technology is fixed, the standard cell library has to be selected. There are commercially available libraries for most of the technologies, sometimes even from various vendors. The most common way is to choose and buy one of these. In the special case of the DCDB, however, the situation is a little different. From the electrical point of view, the requirements for the standard cells are rather tough. Firstly, due to the fact that the DCDB houses analog and digital circuits, special care is necessary to make sure that the digital logic is not disturbing the analog part by emitting noise through the silicon substrate. Usually, guard-rings surrounding the digital elements on a mixed-mode ASIC are used to handle that issue. But for the DCDB as a very low noise device, this is believed to be not enough. A much safer way to prevent digital logic from harming the analog block is to really separate the substrates for the digital and the analog parts of the design. This can be realized by using a technology that provides the so-called *Tri-Well*, which allow not only to place PMOS but also NMOS transistors inside a well structure, separating all active devices from the bulk substrate. In addition, noise distribution through the substrate can be avoided by keeping their bias independent from the supply of the active devices. To this end, it is desirable to have substrate contacts connected with a dedicated supply rail, separately for PMOS and NMOS transistors, in each and every cell of the digital design. The second requirement to the standard cells concerns their tolerance to irradiation. In general, microelectronic transistors suffer from permanent damage due to irradiated oxides. Irradiating the oxide causes positive charge to accumulate inside, resulting in parasitic n-channels in the p-substrate of NMOS transistors. By means of these parasitic channels, leakage currents arise which can cause permanent malfunctions of the electronics. There are special design techniques, like circular NMOS transistor layouts (also known as *enclosed layout transistor* - ELT), that help to overcome this issue. Consequently, using circular transistors for the standard cells of the DCDB, would give an additional safety margin in terms of radiation hardness.

In fact, since the system-on-a-chip (SoC) approach is getting more and more common in the digital microelectronics industry, standard cell libraries are available that address the needs of mixed-mode designs. In other words, there are libraries available that provide separate wells with individual supplies for the active devices and hence fulfil the first requirement for the DCDB design. But there are only very few libraries that use circular NMOS transistors for improved radiation tolerance, like the so-called *DARE* library [51] for example. It seems, however, that there is no library that combines both features. To

this end, a custom-made standard cell library must be developed that has both, substrate contacts in each cell with dedicated supplies and circular NMOS transistor layouts¹.

4.4.1 Radiation Hard Standard Cell Library: First Approach

A qualified starting point for the development of a radiation tolerant standard cell library is the diploma thesis of M. Bruder at the Circuit Design Group [52]. He created the schematic and the layout for 17 cells in total, using radiation hardening design techniques. Among others, there are ten combinatorial cells as well as three types of flip-flops. In order to make these cells usable for the digital physical implementation flow, he additionally created scripts that extract the cell's geometry information as well as initiate simulations for determining their timing behaviour.

Using this preparatory work, a test chip (DTC1) was submitted on a multi-project wafer run via EuroPractice in order to verify the library. First, the usability of the generated characterization files has to be proven. Furthermore, measurements with the produced ASIC can be compared to the expected performance based on simulations, which evaluates the quality of cell characterization.

The DTC1 contains 64 channels of data conversion and serialization logic, which is very similar to that used for the DCDB, and is intended to run with clock frequency of up to 400MHz. The final layout of the chip is presented in figure 4-9. The size of the core including the supply rings around it is roughly $1200 \times 1200 \mu m^2$.

However, after production the measurements revealed mainly two aspects. On the one hand, the logic is working in the expected way. That means for a given sequence of input values the chip produces the expected output. In other words, the logic can be considered

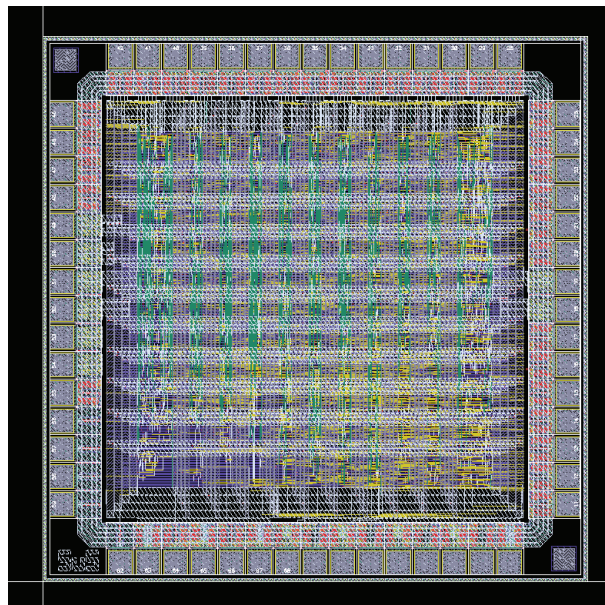


Figure 4-9 Layout of the DTC1.

1. The description given here follows the chronological development process. Unfortunately, this decision had to be revised later on (refer section 4.4.3).

functionally correct. The maximum operation speed, in turn, is limited to about 200MHz , which is only half of what the logic is designed for. In order to analyse the problem, a capacitance-extracted full analog simulation was performed. In accordance with the previous results, this simulation also claims that the design should run at the target clock frequency, which points to bad models of the circular transistors or even a bad production.

4.4.2 Standard Cell Library: Second Approach

The lesson that is learned from the DTC1 measurements is that the characterization of the radiation hard standard cell library in the state as described in [52] is not resulting in a logic with predictable timing behaviour and therefore needs to be improved. However, a much more critical fact regarding the DCDB development is that the logic for the 64 channels of the DTC1 occupy about half of the area that is estimated for the 256 channels of the final DCDB. The layouts of the cells in the radiation hard standard cell library are simply too large. This is not because of a bad design, but it results from the fact that circular transistors are much larger than the normal linear ones.

Various ways are possible to overcome this issue, but unfortunately, all of them are rather dramatic and result in the abandonment of a major design goal. First of all, it is possible to return, at least partially, to a full custom ASIC design. That means to give up either parts of or even the entire synthesized digital logic and replace it by hand-crafted electronics, which is usually more dense. A second solution would be to reduce the digital functionality of the DCDB by outsourcing parts of the logic, for instance to the DHP. The third way could be to trust in the intrinsic radiation tolerance of the selected microelectronics technology and use standard cells without circular transistors.

Although the operation environment of the DCDB is very harsh, skipping the use of circular transistors is the preferred way to go. It is believed that designing with the safety margin in terms of radiation tolerance provided by the circular transistors can be considered as very conservative. Irradiation tests with the former DCD2 chip, whose digital logic cells are not designed with circular transistor layouts as well, encourage this assumption [36].

Triggered by the decision to dismiss the circular transistors, even the decision to use a self-developed library rather than a commercial one must be reviewed, since one of two arguments for it is gone. The remaining argument, the need for separate substrates with bias contacts in every cell, might not be a strong one since there are mixed-mode ASICs on the market that face the same problem of digital logic interfering the analog circuits. So there must also be a commercial solution for that. However, the expertise gained by the development of the radiation hard standard cell library might also be good enough to produce a library that has both, smaller cells and a characterization that leads to a predictable timing. Consequently, a second attempt was made that resulted in the production of a second test chip, the so-called *DTC2*.

The DTC2, whose layout is shown in figure 4-10, implements the same digital logic as its predecessor. But in contrast to the DTC1, there are three separate blocks with only eight data processing channels each, that are built using three different standard cell libraries. The first block is implemented using the identical library as it is used for the DTC1. It serves as a reference and is expected to behave exactly like the logic on the DTC1. The second block uses the same library as the first one, except from a bug-fix in

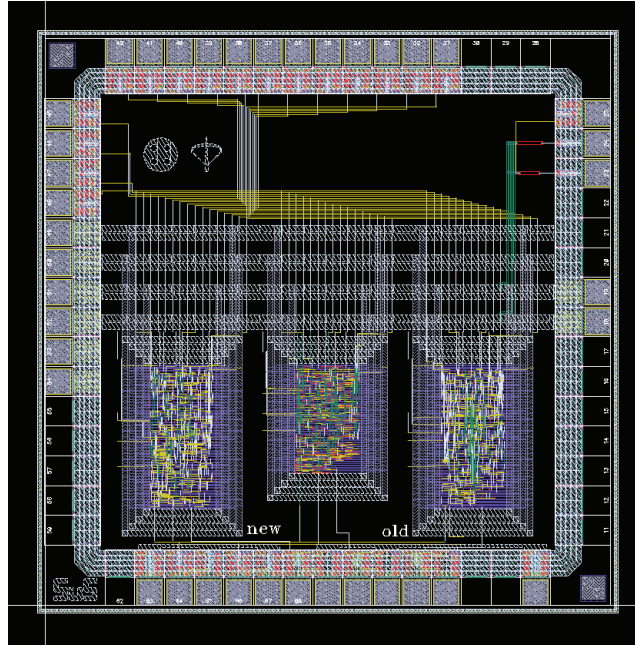


Figure 4-10 Layout of the DTC2.

the layout of the flip-flop, which might possibly be responsible for the bad timing behaviour. The third block, finally, is built from a newly developed standard cell library, which provides - as discussed above - no circular transistors but still uses the Tri-Well approach, has got substrate contacts and separate substrate supply rails in every cell. Common to all of these libraries is that they are characterized using a proprietary tool called *Encounter Library Characterizer (ELC)* provided by Cadence. This tool is favoured over the set of scripts used before for several practical reasons. The first one is simply because the tool is available in the Encounter Toolkit, which is necessary for digital physical implementation anyway, and there is the opportunity to have vendor support for it. But it is also a good way to crosscheck the results with those obtained by the scripts. Furthermore, the ELC is able to produce cell characterizations in ECSM (Effective Current Source Model) representation for more accurate timing analysis rather than the old-fashioned NLDM (Non-Linear Delay Model)¹.

Measurements with this chip reveal that the first two blocks behave nearly identically and still far below the simulated performance. First of all, that means the possibility of a production issue for the DTC1 is very unlikely. Secondly, the bug in the flip-flop's layout seems not to be the cause of the timing issue, since its repair is not improving the behaviour. The third block, however, is doing well as its maximum operation speed even exceeds the expectations. In turn, this fact proves that the ELC characterization tool is working properly and beyond that, it shows that the characterization scripts from [52] do not necessarily need to be blamed for the issues of DTC1.

1. In contrast to the NLDM, the ECSM models the standard cells as voltage-controlled current sources rather than a black box with input capacitance and output timing behaviour.

4.4.3 The Standard Cell Library for the DCDB

The standard cell library tests with the two chips DTC1 and DTC2 lead to two major results. The first and rather depressing one is that a library providing extended radiation tolerance by means of circular NMOS transistor layouts is not applicable for the DCDB. Not only because the cells are too large, but also due to the bad characterization even with the new ELC tool. The second result however, which is indeed fairly encouraging, is the availability of a well characterized standard cell library providing sufficiently small cell sizes and individual substrates with dedicated contacts and supply rails in every cell. Although containing a lot less cells than commercial libraries usually do, estimations on size and timing based on the DTC2 justify the decision to go for the self-developed library instead of changing to a commercial one. The approach of a library containing cells with circular transistors is not followed up any further, or at least postponed until radiation tests show the need for a more radiation tolerant design. Summarizing the discussion about the standard cell library for the DCDB development, table 4-1 lists the available cells and gives a short description of their functionality.

#	Name	Description
1	UCL_AND2	AND gate with two inputs.
2	UCL_BUF	Buffer with driving strength of one unit.
3	UCL_BUF4	Buffer with driving strength of four units.
4	UCL_BUF16	Buffer with driving strength of 16 units.
5-9	UCL_CAP5 - UCL_CAP9	Decoupling capacitor cells with widths of five to nine times the minimal cell width.
10	UCL_DFF	D-Flip-Flop with UCL_INV as output driver.
11	UCL_DFF_LP	D-Flip-Flop with UCL_INV_LP as output driver.
12	UCL_DFF_LP2	D-Flip-Flop with UCL_INV_LP2 as output driver.
13	UCL_DFF_RES	D-Flip-Flop with asynchronous reset.
14	UCL_DFF_SET	D-Flip-Flop with asynchronous set.
15	UCL_FILL	Filler cell of minimal width.
16	UCL_GTINVS	Gated inverter.
17	UCL_INV	Inverter with driving strength of one unit.
18	UCL_INV4	Inverter with driving strength of four units.
19	UCL_INV_LP	Inverter based on UCL_INV with aggressive reduction of transistor width for lower power consumption.
20	UCL_INV_LP2	Inverter based on UCL_INV with intermediate reduction of transistor width for lower power consumption.
21	UCL_MUX2	Multiplexer with two inputs and inverted output.
22	UCL_NAND2	NAND gate with two inputs.

Table 4-1 List of the gates available in the standard cell library for implementing the DCDB. The gates #29 to #31 are available since the DCDB-TC development, the gates #32 to #35 were added for the DCDBv2.

#	Name	Description
23	UCL_NOR2	NOR gate with two inputs.
24	UCL_NOR2_2	NOR gate with two inputs and increased driving strength.
25	UCL_OR2	OR gate with two inputs.
26	UCL_TIEHI	Tie-High gate, statically providing a logical „high“ at its output.
27	UCL_TIELOW	Tie-Low gate, statically providing a logical „low“ at its output.
28	UCL_XOR	XOR gate with two inputs.
29	UCL_AOI21	AND-OR-INVERTED mixed gate with three inputs. $f = \neg((a \wedge b) \vee c)$
30	UCL_NAND2A	NAND gate with two inputs. One of the two inputs is inverted.
31	UCL_OAI21	OR-AND-INVERTED mixed gate with three inputs. $f = \neg(a \wedge (b \vee c))$
32	UCL_BUF8	Buffer with driving strength of eight units.
33	UCL_DFF_LP4	D-Flip-Flop with a doubled UCL_INV_LP2 as output driver.
34	UCL_CGI2	CARRY-GENERATOR-INVERTED mixed gate with three inputs. $f = \neg((a \wedge b) \vee (a \wedge c) \vee (b \wedge c))$
35	UCL_MUX2A	Multiplexer with two inputs. One of the two inputs as well as the output are inverted.

Table 4-1 List of the gates available in the standard cell library for implementing the DCDB. The gates #29 to #31 are available since the DCDB-TC development, the gates #32 to #35 were added for the DCDBv2.

4.5 Physical Implementation

With the availability of the standard cell library the final precondition for starting the DCDB's physical implementation is fulfilled. This section gives a detailed description of all the processing steps that have to be performed in order to produce a microelectronic layout which is ready for production.

As already mentioned, the Chair of Circuit Design, Heidelberg University, has access to the latest design and verification tools from *Cadence Design Systems Inc.* It is therefore very convenient to use the tools of this vendor for the physical implementation of the DCDB, not only of the digital logic, but of course also regarding the analog design parts. For the digital domain, the tool-kit of choice is the *Encounter Digital Implementation System*. While the DCDBv1 and the DCDB-TC are both implemented using the tool version 8.1, for the latest chip revision, the DCDBv2, the tool version 10.1 is employed. The remaining part of this chapter is mainly focused on processing the DCDB's physical implementation by means of this tool. Additional information about it is available in

[53], [54], [55], [56] and [57]. Cadence's analog design environment is called *Virtuoso*. The DCDBv1 development started with tool version 6.1.3. Later, the versions 6.1.4 and 6.1.5 were used for DCDB-TC and DCDBv2. As mixed-mode chip designs become more and more important even for commercial applications, Cadence has been putting effort on improving the co-operation of their digital and analog design tools. This leads to a very comfortable situation for the DCDB development, as analog and digital parts can be developed separately with the respectively specialized tools. In a later integration phase, either of the tools can be used to combine the blocks, since each of them is able to read the other's design files. In the case of the DCDB, the so-called *Analog-On-Top* flow is employed, meaning that Virtuoso is used for combining analog and digital sub-designs and for performing final design rule (DRC) and layout-vs-schematic checks (LVS).

4.5.1 Constraining the Design

The first step of digital physical implementation is to translate the behavioural-based functional description of a design into a logically equivalent netlist of standard cells. This is far from being a trivial task, because in general, the translation is not unique. Indeed, depending on the respective design, the solution space might be even too large for the used tool to find the best realization. And beyond that, it is often not even clear to the tool how to judge the solutions in order to find the best. It simply cannot know what to optimise for.

As a consequence it is necessary to help the tool by providing additional meta information about the design. The most important information is certainly about the desired speed of the chip. That means, the design's clock signals together with their target toggling frequencies need to be identified. Knowing that, the tool is able to analyse the combinatorial logic paths between the flip-flops in the design and prune those realizations which would possibly violate the given timing constraints. Another important information is about the design's connection to the outside world. That is mainly concerning the timing behaviour of the off-chip signals, both, inputs and outputs, together with related information like capacitances that have to be driven by output signals and so on. In addition, also defining the design's maximum power consumption is possible.

In the case of the DCDB, the timing constraints are rather straightforward. There are two global clock inputs, the main clock and the TCK signal of the JTAG interface. The former is constrained to a frequency of 320MHz ¹, while the JTAG interface has very relaxed timing requirements and is therefore set to 20MHz . The internally generated clock signals for the data conversion logic and the ADC steering sequence generation must be mentioned in the constraint file (.sdc file) as well, since the tool needs to recognize them as clocks having a fixed relation to the main clock in order to allow regular data signals to cross the clock domains properly. In addition, all input and output data signals must be constrained to a certain delay relative to the capturing or generating clock edge. Outputs, especially those driving long wires across the chip to the ADCs in the analog domain have a load capacitance of 300fF (10fF for others) assigned to them, which ensures the use of adequate buffer cells.

1. For the DCDBv1, the main clock is constraint to 363MHz , for the DCDB-TC it is 400MHz . Over-constraining provides safety margin for coping with uncertainties of, for example, cell characterization or process variation.

Since keeping the power consumption of the chip as low as possible is also one of the major design goals, a corresponding constraint is formulated that instructs the tool not to let the DCDB's digital block consume more than $700mW$. However, determining the power consumption of an ASIC during design phase is a rather difficult task, as it is in general strongly related to the chip's operating environment in terms of the received input stimuli signals and the resulting output signal sequence. In particular, this is true for the digital logic block of the DCDB, because its main functionality is actually data conversion. The toggling rate of the generated output values, for example, is strongly depending on the input values coming from the ADCs, and so is the power consumption of the design. Therefore, in order to let the tool calculate a good estimation of the design's power consumption, a simulation snapshot is necessary providing the information about how the design is going to work. It is obvious that the quality of the power estimation and so the ability of the tool to meet the power constraint is to a large extend depending on a realistic simulation snapshot. This snapshot can easily be generated using the simulation environment for the DCDB as described in section 4.3.

4.5.2 Encounter: Standard vs. MMMC Flow

Before starting the physical implementation, it must be clear whether to use Encounter with the standard or the multi-mode-multi-corner (MMMC) flow. For the standard flow, the tool needs beside the logic description, of course, at least one set of standard cell characterization information (.lib file) together with one set of design constraints. Equipped with these inputs, Encounter is enabled to go through all the steps of physical implementation, trying to meet all given constraints based on the timing (and eventually power) information from the standard cell characterization.

Although being comparatively easy to set up, this flow has, however, two major disadvantages. The first one is related to the fact that only a single.lib file is loaded. Usually, standard cell libraries are characterized using a variety of combinations of values for supply voltage, environmental temperature and manufacturing process variations. Each of these combinations is stored into a separate .lib file. Hence, if only one of these files is loaded, the tool has obviously no chance to check whether the design is working under conditions other than those the single characterization file is generated for. This might cause conflicts in choosing the right .lib file to load. On the one hand, for making sure that the design meets operation speed constraints, one would tend to use the worst case characterization. On the other hand, the best case set is necessary to make sure that there are no hold time violations in the design.

The second disadvantage of the standard flow is that the tool is not aware of designs that might have several operation modes. It simply takes the design as a whole and optimises what ever the given set of constraints tell that is worth to optimise. But there might be separate parts within the same design that are actually barely related to each other, although interconnections exist between them. In that case the tool will optimise them, even if it is not necessary.

The multi-mode-multi-corner flow is an adequate way to overcome these issues. It allows both, specifying multiple characterization files that cover multiple process and environmental corners as well as defining several operation modes within the same design. Each of the modes is assigned to its own set of design constraints, covering only

those aspects that are related to that certain mode. The tool then tries to optimise every mode in every corner concurrently.

While the DCDBv1 and the DCDB-TC are implemented using the standard methodology, the development of the DCDBv2 benefits from the MMMC flow. For the DCDBv1 and the DCDB-TC over-constraining on a typical-condition characterization is the best practice for coping with process and environmental variations. This is not necessary for DCDBv2. Here, the clock constraint is set to the real target operation speed of $320MHz$, because the physical implementation tool is actually working on the worst case library. But since the best case library is provided as well, the tool is also able to eliminate hold time violations. In order to benefit from the multi-mode capabilities of that flow as well, two modes can be identified for the DCDB design. The first one is obviously the normal operation mode, but the DCDB's JTAG interface can fairly be regarded as a second mode, since it operates completely independent from the rest of the design. Indeed, when working with the standard flow, the fact that these parts of the design are not handled separately introduces difficulties. The JTAG block is connected with the logic part for the normal operation via the boundary scan chain. Although these connections are never used dynamically, Encounter is not aware of this and finally fails to optimise them. Special constraints are necessary work around this issue. When working with the MMMC flow, in turn, this problem simply does not occur.

4.5.3 Synthesis

Generally, in the synthesis step, a behavioural-based design description together with an appropriate set of constraints is translated into a netlist of available standard cells. The DCDB's digital block design is written in Verilog hardware description language, which supports both types of representation. So basically, the synthesis tool, which is a part of the Encounter tool package and is called *RTL Compiler*, takes the Verilog design files, the constraints as well as timing and eventually physical information about the standard cells in the library and produces a single Verilog file containing only gate instances and their connections.

Using the provided constraints, the synthesis tool tries to optimise the combinatorial logic paths between the flip-flops in order to make them meeting the timing requirements. However, at this early stage of the physical implementation process, the path delays can only be estimated on base of the delays introduced by the used cells. The delay introduced by the interconnection between them is totally unknown or can only be estimated very coarsely using so-called *wireload models* as long as there is no information available about the actual distance between connected cells on the chip. In principle, there are two ways to deal with that issue. The easiest but rather inaccurate one is overconstraining. That means the constraints provided to the synthesis tool must be tougher than the real target constraints, making the tool putting higher effort on optimization. If possible, the tool produces a netlist where these virtual constraints are met and the gained margin can be used to compensate interconnection delays once they are known. The more accurate but also more complex way to cope with unknown interconnection delays is a placement-aware synthesis run. That means the synthesis tool performs a more or less precise placement estimation or even calls Encounter's placement engine for producing a real placement of the generated netlist in order to improve the estimation of the interconnection distances. However, the most realistic and

useful placement result is obtainable only if additional information about relevant boundary conditions like pin position, for example, is available, which is defined later in the floorplanning phase. At this point, the synthesis step becomes an iterative procedure, since floorplanning (refer to section 4.5.4) can only be done with a netlist being available from a previous first-order synthesis run.

Cell	DCDBv1	DCDBv2
UCL_AND2	4250	4885
UCL_AOI21	-	3984
UCL_BUF	527	352
UCL_BUF4	-	37
UCL_BUF8	-	5
UCL_BUF16	-	38
UCL_CGI2	-	1
UCL_DFF	17	43
UCL_DFF_LP	19819	19551
UCL_DFF_LP2	264	15
UCL_DFF_LP4	-	10
UCL_DFF_RES	23	23
UCL_DFF_SET	2	2
UCL_INV	593	14
UCL_INV4	5	867
UCL_INV_LP	12764	8445
UCL_INV_LP2	338	59
UCL_MUX2	10746	7491
UCL_MUX2A	-	3738
UCL_NAND2	20285	9198
UCL_NAND2A	-	2415
UCL_NOR2	2516	822
UCL_NOR2_2	512	2
UCL_OAI21	-	646
UCL_OR2	19	543
UCL_TIELOW	12	60
UCL_XOR	512	58
Total Cell Count	73204	63304
Total Area (μm^2)	2465974.9	2398740.48

Table 4-2 Standard cell usage statistics for DCDBv1 and DCDBv2.

For DCDBv1 and DCDB-TC, the timing closure is achieved using the over-constraining method. Due to the fact that there is no wireload model available in the standard cell library, both designs are temporarily over-constrained to an operation frequency of 588MHz , which corresponds to $\sim 162\%$. Each of the two designs meets this constraint by a few picoseconds. The DCDBv2 development, in turn, benefits from a placement-aware synthesis run. In that case, not only an estimation but the real placement is produced by calling the Encounter placement engine during the synthesis step. As a result, the synthesis step ends up with a netlist that is already placed, meeting the regular timing constraint with 20ps of positive slack left in the worst case path with the worst case cell characterization.

Table 4-2 summarizes the synthesis results of DCDBv1 and DCDBv2 in terms of cell usage and corresponding total area. The DCDB-TC design is excluded here, since it is only a test chip with a lot less channels. So the synthesis result would hardly be comparable to the others. The DCDB versions one and two, however, can be considered as identical, since the minor changes in version two can be neglected. Comparing the cell usage statistics for these two designs, one recognizes that adding gates with combined logic functionality, such as UCL_AOI21, UCL_MUX2A, UCL_NAND2A and UCL_OAI21, helps the tool a lot to decrease the overall amount of cells by about 14%. As an effect of the placement-aware synthesis, the amount of used buffers and inverters with high driving strength like UCL_BUF4, UCL_BUF8, UCL_BUF16 and UCL_INV4 in this early implementation phase is much higher for the DCDBv2 design.

4.5.4 Floorplanning and Placement

Once a design is translated into a gate-level netlist, its cells can be placed onto the silicon area. The placement can be run as individual step, or, in case of a placement-aware synthesis, it can be called automatically by the synthesis tool. However, in both cases a floorplan is required prior to that. During the floorplanning step, guidelines for the placement are created. That is, first of all, the definition of the available chip area in terms of edge length, aspect ratio or even more complex shapes. Second, the various input and output pins must be fixed to their target positions. In most cases it is a good idea to reserve a certain fraction of the chip area for power routing, usually as a ring-shape surrounding the core area. Depending on the complexity of the design, it might be necessary to define regions within the chip area for several or even each of the design's subdivisions on a certain hierarchy level, in order to make the tool being able to solve the problem at all or at least within reasonable time. After having created such a floorplan, the actual placement procedure can be started. The placement engine is able to optimise for a number of different design goals, such as delay or area reduction, for example. Furthermore, if adequate simulation results are provided, it can run in power-aware mode, meaning that highly switching nets are identified and automatically kept short.

The floorplanning for the DCDB is rather simple. (While the floorplans for the DCDBv1 and DCDBv2 are identical, the DCDB-TC is not considered here.) The digital block area has a rectangular shape with edge lengths of $3000\mu\text{m} \times 1501\mu\text{m}$. Subtracting the reserved area for the power rings leaves $2823\mu\text{m} \times 1325\mu\text{m}$ for the core logic. This leads to a placement density of about 66% for DCDBv1 and about 64% for the DCDBv2, which is perfectly relaxed with regard to later optimization steps and insertions of decoupling capacitances. The inputs and outputs connecting the digital block to the

DCDB's analog part are located on the top edge, while all digital off-chip connections are placed at the bottom edge. Due to the fact that the overall combination of analog and digital parts of the DCDB is done in the analog design environment (Analog-On-Top flow), I/O cells and their exact location are not considered during the digital physical implementation. Instead, these cells are added and connected afterwards manually. Therefore, a „pin“ in this context is only a piece of metal regarded as a wire that begins (or ends) at the edge of the digital block area.

Although the DCDB's digital block design has a relatively clear intrinsic segmentation due to the large number of individual channels, no corresponding segmentation of the available silicon area is performed. That means, in principle, every cell could be placed everywhere. Obviously, on the one hand, this makes the design difficult for the tool to handle. But on the other hand, as long as the tool is able to deal with the complexity, which is indeed the case for the DCDB, it is best practice to do so, since unnecessary constraints mostly degrade the final performance.

With the created floorplan, placing the design is then straightforward. As mentioned above, the placement engine is either called directly within Encounter, like for the DCDBv1, or the placement is run under the hood of the (second-order) synthesis step as for the DCDBv2. In both cases incremental placement optimizations are possible, and indeed necessary in order to keep meeting the timing requirements.

4.5.5 Power Planning

Following to the cell placement, power planning is the next task. Usually the power is distributed over a digital design by means of three different structures. First of all, each standard cell is typically equipped with a strip of metal on the top and the bottom edge, mostly on the technology's first metal layer, regarded as power and ground supply for the transistors. Once several of these cells are placed back-to-back in a row, which is the standard placement strategy in digital physical implementations, these strips connect

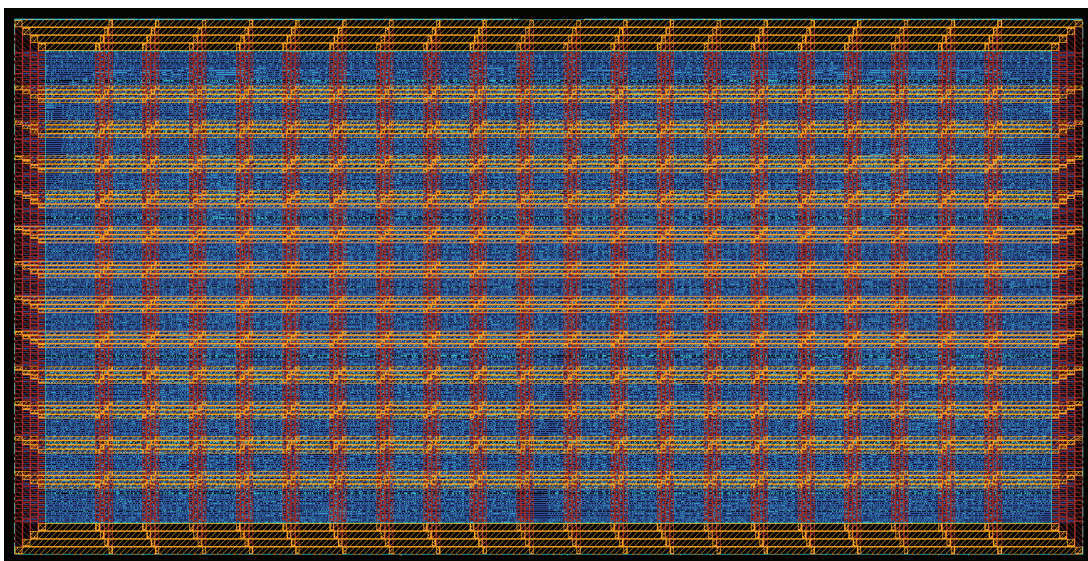


Figure 4-11 Picture of the DCDBv1 layout after floorplanning, placement and power planning.

each other building macroscopic power rails¹. The various power rails of the same polarity are then connected to each other by means of power rings surrounding the entire design. In principle, these rings can be implemented using any of the available metal layers, but practically, the decision about which layer to use for the rings is strongly depending on technology parameters. Usually the upper metal layers of a microelectronics technology are thicker and have therefore rather low electrical resistance compared to lower ones. However, considering the electrical resistance of the via stack, that is necessary to connect the power rails on the lower level to an upper metal layer with advantageous electrical properties, one might end up with the conclusion that implementing the rings on a lower layer is the better choice. The third power distribution structure are sets of wide stripes crossing the chip area in horizontal and vertical direction. They connect the opposite sides of the power rings and all crossed standard cell level power rails. In large designs, these stripes are very useful to minimize the effective resistance of the power distribution network. However, since they are crossing the entire placement area, the stripes and especially the via stacks for connecting the rails compete with all the other signals for the available routing space. Consequently, there is a trade-off between the quality of the power network and the resource reservation for the signal routing.

The power distribution network for the DCDB is illustrated in figure 4-11. It is identical for the chip versions one and two. The cell level power rails are implemented on the first metal layer. The power rings all around the core area use the two uppermost layers, which are metal five and six. The stripes are implemented using the same layers as the rings, reserving the lower layers for signal routing. The number of sets of stripes crossing the core area is 20 in vertical and 12 in horizontal direction. Due to the fact that the standard cell library provides extra power supply rails for n-wells and p-wells, there are four rings and four stripes per set in total.

Indeed, this power network tends to be over-designed. Nevertheless, due to the rather low density of the design no unsolvable routing congestions occur. It is therefore decided to go for a very conservative power network, which relaxes the effort that needs to be spent on its analysis.

4.5.6 Clock Tree Synthesis

After the placement phase the location of the design's cells on the chip is considered fixed, at least to a large extend. In turn, the routing has not been done yet, so the routing resources, at least on the lower layers, are barely touched. It is therefore the right time to distribute the most important signals of the design, which are the clocks of course.

Since it is the basic idea of a synchronous digital design to have all flip-flops sampling their input values at the same time, it is very important to make sure that the clock signal distribution over the design is balanced. That means the clock edges reach all the flip-flops simultaneously. In general, there are two common ways to physically implement the clock signal distribution, the *mesh* and the *tree* structure. With the mesh structure approach, the clock is distributed via a grid of horizontal and vertical lines. Due

1. Obviously, for designs having a density less than 100%, there are gaps between the cells which interrupt these rails. In that case filler cells providing the necessary rail connections must be used to fill these gaps. Alternatively, the physical implementation tool might also automatically recognize and fix them.

to the combination of extensive buffer usage and the existence of intersection points connecting the lines of the grid, a clock mesh provides a comparatively low clock insertion delay but has a rather high power consumption at the same time. The clock tree approach uses a completely different routing strategy. Here, as the name implies, the clock signal is routed in a tree shape with the clock source being the root and the flip-flops being the leave nodes of the tree. Hence, in contrast to the mesh approach, for every flip-flop there is exactly one path to the clock source. Compared to clock mesh structures, clock trees tend to require less of both, routing resources and power, while introducing longer insertion delay. Nowadays, the clock tree approach is most common for digital designs, while the clock mesh structure is only barely used and is not discussed here any further.

Implementing the clock tree is done by the clock tree synthesis engine. That means, in first order, calculating the required depth of the tree, defining and placing the necessary buffers and finally implementing the routing paths from the clock source via the buffers to the flip-flops. Simultaneously, the clock synthesis engine needs to honour and balance several optimization goals like the minimization of the power consumption for the clock buffers, the total clock insertion delay, the quality of the tree balance and so on. In addition, the clock tree synthesis offers also a great potential for optimizations of the design's timing behaviour. For example, selectively unbalanced leave nodes of the clock tree can help to swap unused timing budget of a combinatorial path, the so-called *positive slack*, to neighbouring paths by adjusting the sampling point of the connecting flip-flop cell. Performing all these optimizations makes the clock tree synthesis a very complex and time consuming task.

All versions of the DCDB digital designs use clock trees for distributing their clock signals. For setting up a clock tree synthesis with Encounter, a lot of configuration work has to be done. In particular for the DCDB design, there are a few issues to tackle in this context. First of all, beside the TCK signal of the JTAG interface, the DCDB digital block design has only a single main clock input, which is supposed to toggle at a rate of 320MHz , while internally two more clock signals must be derived by division, as described in section 4.2.5. These clock signals must be defined as such in order to make the tool synthesize clock tree structures for them. Additionally, their frequency relations among each other needs to be defined, although this information would actually also be extractable from the design. This enables the tool to do proper timing analysis and optimization on paths crossing the clock domains. Consequently, as long as the tool is able to meet the timing at all, no dedicated synchronization techniques must be applied to those paths, since the clock tree synthesis engine is aware of the relation of the involved clocks.

As already mentioned, implementing a tree structure for distributing the clocks over the chip introduces signal delay, the so-called *clock insertion delay*. For the DCDB designs, at least for the full size chips DCDBv1 and DCDBv2, the insertion delays for the 320MHz and the 40MHz clocks are in the order of 2ns . While for the latter this is not affecting the timing at all, the insertion delay does cause problems for the former, since it is in the order of the clock's period. In principle, as ensured by the clock tree synthesis engine, all flip-flops of that clock domain face about the same delay. So it is not influencing the timing between flip-flops. However, regarding the input and output signals of the design, it is highly desirable to define and constrain the validity of these signals relative to the input clock. In the context of an output signal, for example, this translates to a statement saying that a clock edge arriving at the chips clock input

influences the output signal and makes it being valid after a certain amount of time. Extending the idea of a synchronous digital design beyond the chip's borders to inter-chip connections, one would demand that this certain amount of time is fairly less than a period of the related clock. In that case, the connected chip that is supposed to sample the output signal could simply use the next edge of the same clock to sample the data signal, just as it works inside a single chip. Obviously, this is not possible with a clock insertion delay being in the order of the clock's period. The most comfortable way to solve this issue would be to introduce a delay element, like a delay-locked loop (DLL) for example, into the clock tree, that extends its delay to (exactly) one clock period. In that case, whenever a clock edge arrives at the root of the clock tree, there is a clock edge at the tree's leaves simultaneously. That is, of course, not the identical clock edge but the edge that occurred one period before. However, this fact can be neglected, so that the virtual clock insertion delay is (almost) zero. As an alternative solution, it is also possible to live with the clock insertion delay by providing an additional output clock signal, the so-called *return clock*, that is actually connected to a leave node of the clock tree. Hence, that signal undergoes the same delay as the clock signals that arrive at the various flip-flops. This return clock can then be used within a connecting chip to sample its inputs. In the context of the DCDB, the latter approach has to be used, since there is simply no such DLL element available¹.

4.5.7 Signal Routing

Once the clock routing is done, the rest of the signals can be routed, too. This step is performed by the routing engine of Encounter. Like all the development steps before, the

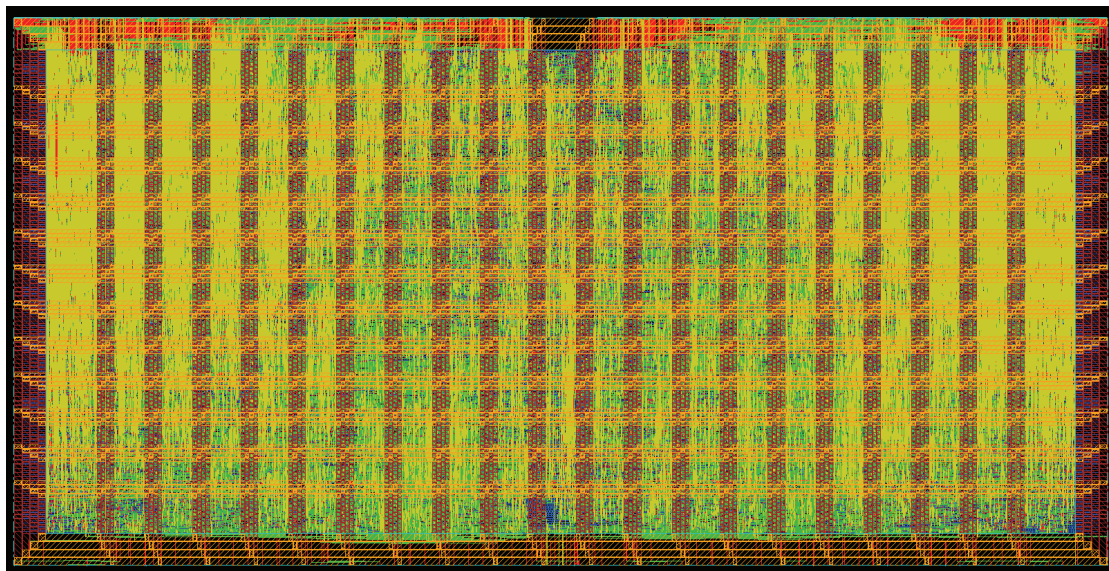


Figure 4-12 Picture of the fully routed DCDBv1 layout.

1. Nevertheless, the output signals are constrained relative to the clock tree's source node for all versions of the DCDB. Due to the explained reasons, these constraints cannot be met by the tool and the resulting warnings are ignored during the development of the DCDBs. Alternatively, it would be possible to constrain the outputs relative to the return clock, enabling the tool to achieve timing closure for the entire design.

routing engine is able to honour some optimization goals on-the-fly. The most important optimization goal is certainly the timing, which is activated by default. If the tool is instructed to perform timing-driven routing, the critical nets are processed with higher priority, leading to the shortest possible connection. Additionally, the tool may also add buffers to actively improve the timing. Another important optimization goal is the signal integrity. If instructed, the tool automatically minimizes crosstalk between nets by increasing distances between potential aggressor and victim nets and avoiding long parallel lines. In order to improve the production yield, the routing engine is also able to optimise the use of vias in two ways. In the first step, the number of vias in the signal routing is reduced by avoiding unnecessary layer changes. Afterwards, remaining vias on paths with uncritical timing are doubled.

Timing and signal integrity optimizations are activated for all three versions of the DCDB design. The technology layers to be used for signal routing is set to two to four for DCDBv1 and DCDB-TC, while additionally using the fifth layer is allowed for the DCDBv2. Via optimization is used since DCDB-TC. Figure 4-12 shows a picture of the fully routed DCDBv1 layout.

4.5.8 Timing Analysis

After the routing step is completed, the design's timing is fixed. If instructed, the tool can now produce a very precise timing report containing the final results for all the paths in the design. Figure 4-13 provides an excerpt of the timing report for the DCDBv1. It shows the detailed information about the path having the worst timing behaviour within the register-to-register group. This is the group of paths that start and end at registers of the design. In other words, paths that include inputs or outputs of the design are not considered here. The most important information is provided right in the first line, that is the timing defined by the given constraints is met. Consequently, since it is the worst path, all other paths in the design meet the timing as well.

Having a closer look to that report, one finds a comparison of the path delay versus the available time. The so-called *Timing Path* lists all the instances contributing to the signal delay from the issuing flip-flop (referred as „Beginpoint“) through the combinatorial logic elements to the sampling flip-flop (referred as „Endpoint“) including the clock insertion delay from the source of the relevant clock through the clock tree to the issuing flip-flop. This Timing Path adds up to a total delay of $4.137ns$. In order to verify whether the sampling flip-flop is actually able to sample that signal correctly, the delay of the Timing Path must be compared to the delay related to that sampling flip-flop. This is the insertion delay from the clock source to the sampling flip-flop (referred as „Other End Time“), its input signal setup time with a negative sign and, of course, the period of one clock cycle. This results in $4.249ns$, which is longer than the delay of the Timing Path. Hence, the timing is met, reserving a margin, the so-called *Slack Time*, that is equal to the difference of these two numbers.

The corresponding report for the DCDB version two is presented in figure 4-14. Obviously, the worst path in this design is not the same as for the DCDBv1, but nevertheless, it is meeting the timing constraints, and so do all the other paths. In


```

Path 1: MET Setup Check with Pin column_pair_I7/pedestals_chain_I/\chain0_out_reg[15] /CLK
Endpoint: column_pair_I7/pedestals_chain_I/\chain0_out_reg[15] /D (v) checked with leading edge of 'clk'
Beginpoint: column_pair_I7/pedestals_chain_I/store_reg/NQ (^) triggered by leading edge of 'clk'
Path Groups: {reg2reg}
Other End Arrival Time 1.704
- Setup 0.205
+ Phase Shift 2.750
= Required Time 4.249
- Arrival Time 4.137
= Slack Time 0.112
Clock Rise Edge 0.000
= Beginpoint Arrival Time 0.000
Timing Path:

```

Instance	Arc	Cell	Slew	Delay	Arrival Time	Required Time
clk ^			0.000		0.000	0.112
clk_L1_I0	EIN ^ -> AUS ^	UCL_BUF16	0.133	0.240	0.239	0.351
clk_L2_I1	EIN ^ -> AUS ^	UCL_BUF4	0.085	0.139	0.379	0.490
clk_L3_I1	EIN ^ -> AUS ^	UCL_BUF	0.141	0.132	0.511	0.622
clk_L4_I0	EIN ^ -> AUS v	UCL_INV4	0.138	0.099	0.609	0.721
clk_L5_I1	EIN v -> AUS v	UCL_BUF16	0.184	0.291	0.900	1.012
clk_L6_I6	EIN v -> AUS v	UCL_BUF16	0.635	0.389	1.289	1.401
clk_L7_I113	EIN v -> AUS ^	UCL_INV4	0.522	0.447	1.736	1.848
column_pair_I7/pedestals_chain_I/store_reg	CLK ^ -> NQ ^	UCL_DFF_LP	0.213	0.389	2.125	2.237
column_pair_I7/pedestals_chain_I/FE_OFC524_store	EIN ^ -> AUS v	UCL_INV_LP	0.128	0.079	2.204	2.315
column_pair_I7/pedestals_chain_I/FE_OFC525_store	EIN v -> AUS ^	UCL_INV	0.530	0.272	2.475	2.587
column_pair_I7/pedestals_chain_I/g1441	EIN ^ -> AUS v	UCL_INV	2.600	1.262	3.738	3.849
column_pair_I7/pedestals_chain_I/g1302	SEL v -> AUS v	UCL_MUX2	0.667	0.400	4.137	4.249
column_pair_I7/pedestals_chain_I/\chain0_out_reg[15]	D v	UCL_DFF_LP	0.667	0.000	4.137	4.249

```

Clock Rise Edge 0.000
= Beginpoint Arrival Time 0.000
Other End Path:

```

Instance	Arc	Cell	Slew	Delay	Arrival Time	Required Time
clk ^			0.000		0.000	-0.112
clk_L1_I0	EIN ^ -> AUS ^	UCL_BUF16	0.133	0.240	0.240	0.128
clk_L2_I1	EIN ^ -> AUS ^	UCL_BUF4	0.085	0.139	0.379	0.267
clk_L3_I1	EIN ^ -> AUS ^	UCL_BUF	0.141	0.132	0.511	0.399
clk_L4_I0	EIN ^ -> AUS v	UCL_INV4	0.138	0.099	0.609	0.498
clk_L5_I1	EIN v -> AUS v	UCL_BUF16	0.184	0.291	0.900	0.789
clk_L6_I6	EIN v -> AUS v	UCL_BUF16	0.635	0.389	1.289	1.177
clk_L7_I122	EIN v -> AUS ^	UCL_INV4	0.364	0.413	1.702	1.590
column_pair_I7/pedestals_chain_I/\chain0_out_reg[15]	CLK ^	UCL_DFF_LP	0.364	0.002	1.704	1.592

Figure 4-13 Excerpt from DCDBv1 timing analysis report, showing the timing of the worst path in the reg2reg group.

principle, the report is structured in the same way as the one in figure 4-13. The main difference that one might recognize is the value of the phase shift, which is actually the duration of the clock period. This is because the two designs are constrained differently, as described in section 4.5.2. $3.125ns$ is actually the target clock period for both designs. However, the DCDBv1 uses the typical case standard cell library characterization and must therefore be over-constrained. In contrast, the timing analysis for the DCDBv2 in figure 4-14 refers to the worst case standard cell library characterization.

4.5.9 Finalizing the Design

The design is now fully placed, routed and the power supply is arranged as well. It is ready for being finalized. That is mainly all about filling the still remaining gaps between the standard cells. In order to do so, dummy cells are placed in these gaps, filling up the available chip area to 100%. The selection of cells to be used as filler cells is user defined. Typically, there are three facts to consider while making this selection. First of all, the placement engine very often places cells with a minimal gap in-between as defined in the technology description. Therefore, standard cell libraries usually provide a cell with minimal width, having no purpose other than being used to fill these gaps. Secondly, if the library comprises capacitor cells, those can be used to place decoupling capacitors into the empty spaces right next to the functional cells of the design. Finally,

in large commercial designs it is a very common technique to spread unconnected functional cells, the so-called *spare cells*, over the unused chip area. In case of a design failure that is discovered after or during production, it might be possible to solve the issue by using spare cells and a changed routing mask. If so, fixing the chip in that way is much less expensive than starting the production anew, since the silicon implantation process typically causes a large fraction of the total production costs.

The standard cell library used for the DCDB development contains a minimal width filler cell and several capacitor cells (refer to table 4-1). These cells are used for filling up the remaining empty spaces of the design. There are no functional cells used as fillers for the DCDB's digital logic block, since the chip production is done via a multi-project wafer run. So in any case, there is no chance to do a wafer-level design correction at all.

```

Path 1: MET Setup Check with Pin column_pair_I2/left_column_I/shift_regs_reg[2][7]/CLK
Endpoint: column_pair_I2/left_column_I/shift_regs_reg[2][7]/D (^) checked with leading edge of 'clk'
Beginpoint: column_pair_I2/counter_val_20_reg/Q (v) triggered by leading edge of 'clk'
Path Groups: {reg2reg}
Analysis View: view_PM worst_functional
Other End Arrival Time 3.115
- Setup 0.143
+ Phase Shift 3.125
= Required Time 6.097
- Arrival Time 6.067
= Slack Time 0.031

Clock Rise Edge 0.000
= Beginpoint Arrival Time 0.000
Timing Path:
+-----+-----+-----+-----+-----+-----+
| Instance | Arc | Cell | Delay | Arrival | Required |
| Time | Time |
+-----+-----+-----+-----+-----+-----+
| clk_L1_I0 | clk ^ | UCL_BUF16 | 0.426 | 0.000 | 0.031 |
| jtag_I/BS_input_cell_clk_I/g46 | EIN ^ -> AUS ^ | UCL_INV4 | 0.074 | 0.500 | 0.530 |
| jtag_I/BS_input_cell_clk_I/g45 | EIN0 v -> AUS ^ | UCL_MUX2A | 0.165 | 0.665 | 0.696 |
| clk_bsc2core_L1_I0 | EIN ^ -> AUS ^ | UCL_BUF4 | 0.294 | 0.959 | 0.990 |
| clk_bsc2core_L2_I0 | EIN ^ -> AUS ^ | UCL_BUF16 | 0.388 | 1.347 | 1.378 |
| clk_bsc2core_L3_I0 | EIN ^ -> AUS ^ | UCL_BUF16 | 0.370 | 1.717 | 1.748 |
| clk_bsc2core_L4_I1 | EIN ^ -> AUS ^ | UCL_BUF16 | 0.464 | 2.181 | 2.212 |
| clk_bsc2core_L5_I19 | EIN ^ -> AUS ^ | UCL_BUF16 | 0.467 | 2.648 | 2.679 |
| clk_bsc2core_L6_I199 | EIN ^ -> AUS ^ | UCL_BUF16 | 0.444 | 3.092 | 3.122 |
| column_pair_I2/counter_val_20_reg | CLK ^ -> Q v | UCL_DFF_LP4 | 0.635 | 3.727 | 3.758 |
| column_pair_I2/FE_OCPC8601_n_244 | EIN v -> AUS v | UCL_BUF | 0.207 | 3.934 | 3.965 |
| column_pair_I2/left_column_I/FE_RC_590_0 | EIN1 v -> AUS ^ | UCL_NOR2 | 0.112 | 4.046 | 4.077 |
| column_pair_I2/left_column_I/FE_RC_589_0 | EIN ^ -> AUS v | UCL_INV | 0.146 | 4.192 | 4.222 |
| column_pair_I2/left_column_I/FE_OCPC6610_FE_OFN3615_n_8 | EIN v -> AUS v | UCL_BUF | 0.201 | 4.393 | 4.424 |
| column_pair_I2/left_column_I/g12640 | EIN v -> AUS ^ | UCL_INV4 | 0.129 | 4.522 | 4.552 |
| column_pair_I2/left_column_I/FE_OFCC9928_n_16 | EIN ^ -> AUS ^ | UCL_BUF4 | 0.286 | 4.808 | 4.838 |
| column_pair_I2/left_column_I/FE_OFC4816_n_16 | EIN ^ -> AUS v | UCL_INV | 0.204 | 5.011 | 5.042 |
| column_pair_I2/left_column_I/FE_OFC4817_n_16 | EIN v -> AUS ^ | UCL_INV4 | 0.229 | 5.240 | 5.271 |
| column_pair_I2/left_column_I/FE_OFC4819_n_16 | EIN ^ -> AUS v | UCL_INV | 0.213 | 5.453 | 5.484 |
| column_pair_I2/left_column_I/FE_OFC4820_n_16 | EIN v -> AUS ^ | UCL_INV4 | 0.156 | 5.610 | 5.641 |
| column_pair_I2/left_column_I/FE_RC_169_0 | EIN0 ^ -> AUS ^ | UCL_AND2 | 0.215 | 5.825 | 5.855 |
| column_pair_I2/left_column_I/FE_RC_170_0 | EIN0 ^ -> AUS v | UCL_NOR2 | 0.135 | 5.960 | 5.990 |
| column_pair_I2/left_column_I/g11684 | EIN0 v -> AUS ^ | UCL_NAND2 | 0.107 | 6.067 | 6.097 |
| column_pair_I2/left_column_I/shift_regs_reg[2][7] | D ^ | UCL_DFF | 0.000 | 6.067 | 6.097 |
+-----+-----+-----+-----+-----+-----+

Clock Rise Edge 0.000
= Beginpoint Arrival Time 0.000
Other End Path:
+-----+-----+-----+-----+-----+-----+
| Instance | Arc | Cell | Delay | Arrival | Required |
| Time | Time |
+-----+-----+-----+-----+-----+-----+
| clk_L1_I0 | clk ^ | UCL_BUF16 | 0.426 | 0.000 | -0.031 |
| jtag_I/BS_input_cell_clk_I/g46 | EIN ^ -> AUS ^ | UCL_BUF16 | 0.426 | 0.426 | 0.395 |
| jtag_I/BS_input_cell_clk_I/g45 | EIN ^ -> AUS v | UCL_INV4 | 0.074 | 0.500 | 0.469 |
| jtag_I/BS_input_cell_clk_I/g45 | EIN0 v -> AUS ^ | UCL_MUX2A | 0.165 | 0.665 | 0.634 |
| clk_bsc2core_L1_I0 | EIN ^ -> AUS ^ | UCL_BUF4 | 0.294 | 0.959 | 0.929 |
| clk_bsc2core_L2_I0 | EIN ^ -> AUS ^ | UCL_BUF16 | 0.388 | 1.347 | 1.317 |
| clk_bsc2core_L3_I0 | EIN ^ -> AUS ^ | UCL_BUF16 | 0.370 | 1.717 | 1.687 |
| clk_bsc2core_L4_I1 | EIN ^ -> AUS ^ | UCL_BUF16 | 0.464 | 2.181 | 2.150 |
| clk_bsc2core_L5_I18 | EIN ^ -> AUS ^ | UCL_BUF16 | 0.470 | 2.651 | 2.621 |
| clk_bsc2core_L6_I178 | EIN ^ -> AUS ^ | UCL_BUF16 | 0.462 | 3.113 | 3.082 |
| column_pair_I2/left_column_I/shift_regs_reg[2][7] | CLK ^ | UCL_DFF | 0.002 | 3.115 | 3.085 |
+-----+-----+-----+-----+-----+-----+

```

Figure 4-14 Excerpt from DCDBv2 timing analysis report, showing the timing of the worst path in the reg2reg group.

4.5.10 Tape-Out and Design Transfer to the Virtuoso ADE

With the placement gaps being filled, the design is now ready for production, or it can be considered as building block within a higher level integration phase. In both cases, the design together with some meta data must be streamed out of the Encounter environment. This is in first order the entire geometry information, represented in a .gds file. Beside that, a Verilog file containing the final netlist with all buffers and fillers as well as the according delay information file (.sdf file) can be generated. By means of the latter two files, the final design can be verified by a Post-Place-and-Route simulation.

In the case of the DCDB development, this digital block is only considered to be a building block. The design's geometry information is therefore imported into Virtuoso again¹, where the top level assembly is done. That is putting the digital block and the analog design part together, as well as adding and connecting I/O-cells and power pads. In order to enable Virtuoso to perform a layout-versus-schematic check for the entire DCDB design, the Verilog netlist must be imported as well. The resulting final layout is shown in figure 2-15 on page 31.

1. Recently, Cadence enabled the digital and analog design tools, Encounter and Virtuoso, to use a common design database called *Open Access*. By using Open Access, interchanging designs is supposed to be more straightforward. However, it is not used for the DCDB development and is therefore not considered here.

The DCDB Test Environment

Abstract:

This chapter covers the development of the test environment for the DCDB. It consists of three sections discussing the hardware as well as the software aspects. Of special interest is certainly the firmware development for the FPGA that is used to interact with the DCDB. The problems arising in that context together with the appropriate solutions are presented.

The quality of the PXD detector system for BELLE-II is significantly depending on the quality of the DCDB, since it is the DCDB that actually measures and samples the signals from the DEPFET detector. It is therefore of high importance to characterize the DCDB chip accurately. In order to do so, the DCDB test environment is developed, which is described in the following.

5.1 The Hardware Components

5.1.1 Electrically Interfacing the DCDB

As explained in chapter 2, it is very advantageous for the half-ladder module assembly to design the DCDB with bump bonds rather than wire bonds and to reduce the number of interconnections as far as possible. A big contribution to the latter design goal is achieved by using low swing single-ended signals for the high speed digital inputs and outputs. This is fairly feasible, since the receiver chip is placed right next to the DCDB on the half-ladder module. For chip testing, however, both aspects cause serious difficulties, as the following two questions must be solved. The first is about where to flip the DCDB onto, establishing an electrical connection to as many pads as possible. Because of the narrow pitch between the bond pads, an adequate counter part is hardly realizable using standard printed circuit board (PCB) technologies. Putting it into a standard chip housing is not feasible either, since the electrically weak output signals of the DCDB most probably would not be able to drive the traces all the way through the housing and via a PCB to a receiving chip. This directly leads to the second problem, that is about which chip to use for receiving these signals at all.

The solution to these problems is shown in figure 5-1. The microchip technology of the Max-Planck-Institut (Munich, Germany) is used to build the so-called *wire-bond adapter*, that allows to connect the DCDB by means of wire-bonds for testing. The DCDB can be flipped onto the wire-bond adapter, which in turn, is glued onto a PCB and electrically connected using wire-bonds. The sensitive high speed digital signals, however, are not directly connected to wire-bond pads. Just like in the final half-ladder design, the communication partner is located right next to the DCDB, that is also on the wire-bond adapter. For the sake of simplicity, this is certainly not the DHP, which is of course a very complex design by itself, but it is another ASIC called DCDRO. („RO“ stands for „read out“.) The DCDRO is a chip solely developed for electrically translating the low swing single-ended signals of DCDB to standard LVDS and vice versa. It is produced using the same technology as the DCDB, which is the UMC 180nm Mixed-Mode and RFCMOS 1P6M via a EuroPractice multi-project wafer run.

In order to keep the test environment development simple by reducing the number of signals to interconnect the DCDRO, the chip provides a static 2:1 multiplexing functionality. That means the DCDRO is able to multiplex the DCDB's eight independent sets of input and output buses to only four sets of those buses. Consequently, the number of wire bond pads is reduced as well. During the development of the wire bond adapter itself, however, it turned out that the single available metal layer is not sufficient to route all the digital signals between the DCDB and the DCDRO. In the end, the available space allows to route only half of the input and output signals,

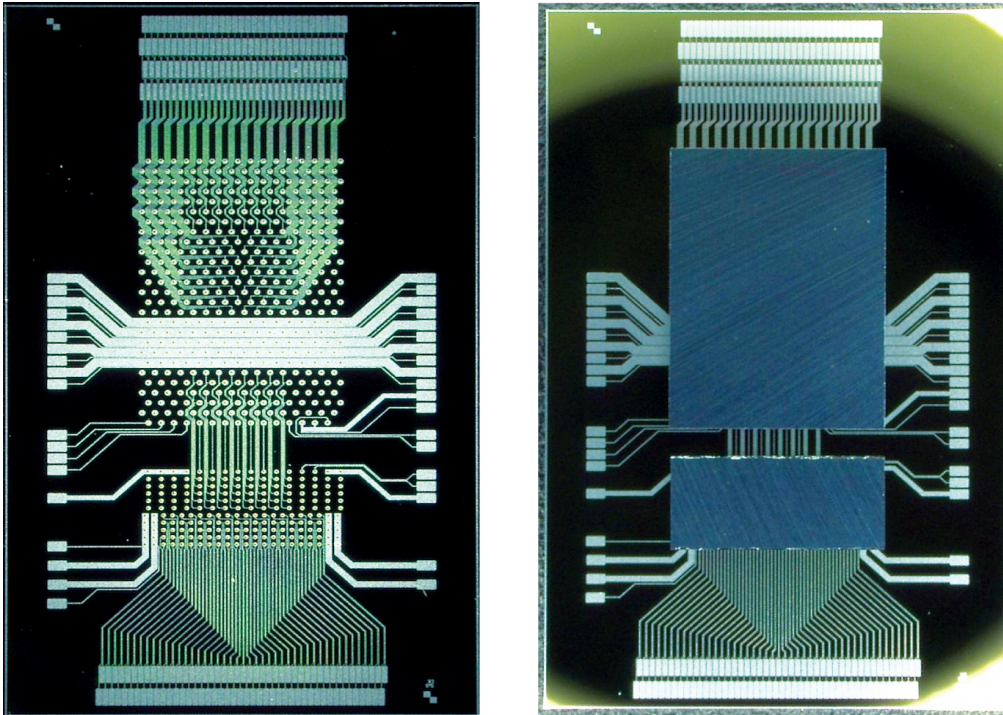


Figure 5-1 Picture of the wire-bond adapter with (right) and without (left) assembled chips. The wire-bond pads on the top are connected to the DCDB inputs (upper footprint). The pads for power supplies and control signals are placed to the left and to the right. The DCDB's digital signals are routed to the DCDRO (lower footprint), whose LVDS signals are connected to the bond pads at the bottom.

which means that the DCDRO's multiplexing functionality is not useful in this setup. Nevertheless, this test environment allows to electrically connect inputs and outputs of half of the DCDB's channels as well as all control signals and power supplies. This is fairly sufficient to make an accurate statement about the design's performance and to use it within a prototype system of reasonable size.

5.1.2 The DCDB Test Board

The wire-bond adapter with the DCDB and the DCDRO on top is then assembled onto the DCDB test board as presented in figure 5-2. This board provides all the infrastructure that is necessary to run the DCDB. That is, first of all, the power supply, separate for analog and digital parts of the DCDB as well as for the DCDRO. The respective decoupling capacitors are arranged on the bottom side of the PCB. A large system connector is used to connect the general purpose FPGA board (refer to section 5.1.3) that is used for steering and reading out the DCDB. Test input signals can be sent to the DCDB via its monitor bus or one of the few wire-bonded input channels. They are accessible by means of SMA connectors on the test board. Additionally, the PCB provides several test points for probing the digital signals from and to the DCDRO/DCDB.

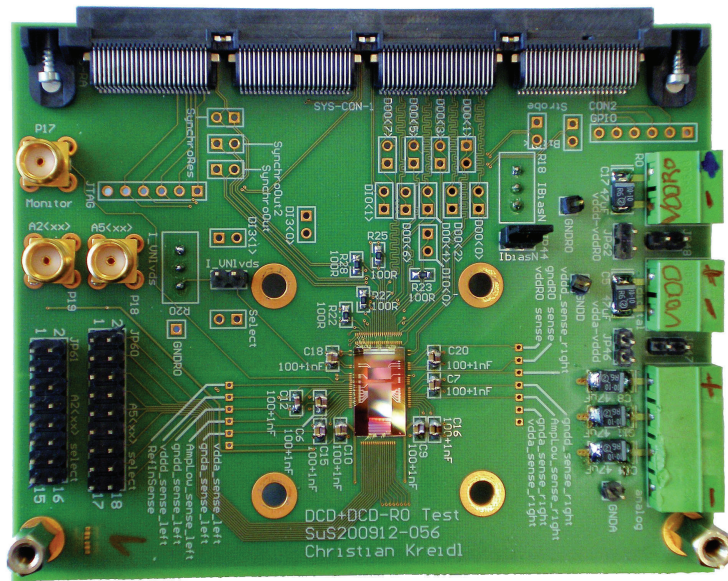


Figure 5-2 Picture of the DCDB test board (top side).

Unfortunately, the geometrical fan-out of the digital signal pads on the wire-bond adapter, which are connected to the DCDRO, turned out to be too narrow for being bonded entirely¹. To this end, only two of the four DCDRO I/O channels are actually connected to the PCB. Finally, the number of DCDB/DCDRO signals to be connected is ten differential lines per pair of input and output data buses, five differential control signals and another five single-ended signals for connecting the DCDB's JTAG interface. That is 25 differential signals and 5 single-ended ones in total.

5.1.3 General Purpose FPGA Board

In general, the most convenient way to rapidly build a flexible environment for testing a new ASIC is to use a programmable logic device, like a FPGA for example, that is interacting with the chip. Within certain limits, a FPGA can be programmed freely and is able to implement a large variety of custom logic for communicating with the chip. Consequently, this approach is also taken for testing the DCDB.

The general purpose FPGA board that is used for this task was developed by the group of Prof. Dr. Norbert Wermes at the University of Bonn (Germany), originally for the test environment of the DCD2 chip, the predecessor of the DCDB. Using this already available board is obviously a very straightforward idea, since the requirements concerning the control and readout logic as well as number of connections remained similar. A picture of this development, which is lovingly called *V4Board*, is shown in figure 5-3. More technical information about it, beyond the brief presentation here, is provided in [58] and [59].

1. In a second version of the wire-bond adapter this fan-out is widened. However, since the second version is again not fully compliant with the DCDB test board, it is only used in conjunction with the DEPFET readout prototype hybrid board as presented in section 6.1.3.

The heart of the V4Board is a Xilinx Inc. FPGA of type Virtex4 LX40 FG1148 (Speed Grade -10), offering 41472 logic cells, 216kB of usable on-chip block memory as well as 640 general purpose inputs and outputs [60]. In addition, there are several peripheral devices available on the V4Board, such as two memories, 288Mb RLDRAM and 16Mb asynchronous SRAM, power regulators and a dedicated system monitoring device. A JTAG header is available allowing to access the FPGA's JTAG interface from external. This is very attractive, since it permits firmware debugging directly inside the FPGA via internal logic analyser cores like Xilinx's ChipScope toolkit. The RJ45 connector is not an ethernet connector but it can be used for general purpose, since it is directly connected to some of the FPGA's user I/Os. Its dedicated use, however, is the connection of an external trigger logic unit [61], which is required for triggered operation of a DEPFET readout prototype. The most important features of this PCB with regard to the DCDB test operation is certainly the very wide system connector and the USB 2.0 add-on card, that allows for establishing a PC communication to the FPGA. (The latter is not shown in figure 5-3.)

There are 32 differential and 26 single-ended signal lines routed directly from the FPGA to the system connector. By using another add-on board, that is bridging the spare connectors on the top side of the V4Board, additional five differential and 32 single-ended signals between system connector and FPGAG are available. That is 37 differential and 58 single-ended connections in total, which can be used by test boards like the one for the DCDB in order to connect the chips directly to the FPGA. Referring to the numbers of required connections given in section 5.1.2, the V4Board is perfectly suitable for this purpose.

The USB 2.0 add-on card mainly consists of a microcontroller and a storage device that holds its program code. The microcontroller provides a USB 2.0 compliant interface and several others for connecting peripheral devices. Two of these, a memory interface and a software-configurable high speed FIFO interface, are connected to the FPGA of the

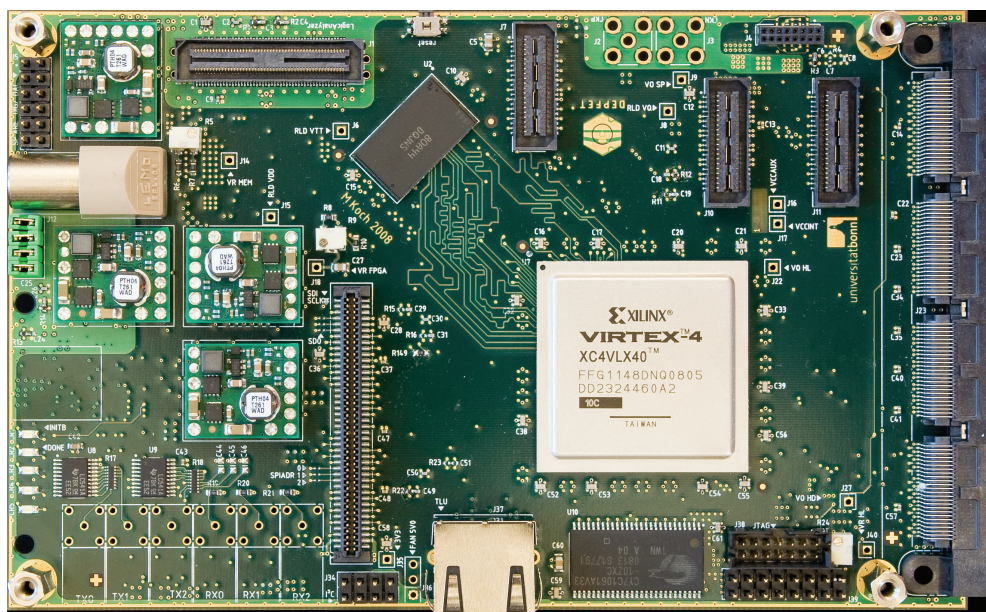


Figure 5-3 Picture of the V4Board, developed at the University of Bonn [59].

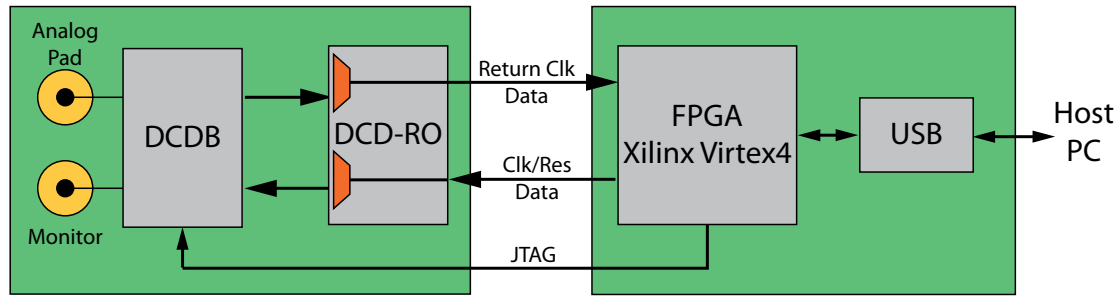


Figure 5-4 Schematic drawing of the DCDB test hardware setup.

V4Board. Indeed, measurements revealed that the data throughput between FPGA and a connected PC using the high speed FIFO interface is only about $12MB/s$, while the theoretical maximum of USB 2.0 is $60MB/s$. Nevertheless, this transfer rate is believed to be sufficient for all relevant tests of the DCDB.

5.1.4 The DCDB Test Environment Hardware Setup

The schematic drawing in figure 5-4 summarizes the test environment hardware for the DCDB. The DCDB's analog test inputs are connected to SMA connectors on the test board. While its JTAG interface is directly connected to the Virtex4 FPGA, the majority of the DCDB's digital communication with the FPGA is bidirectionally converted by the DCDRO repeater chip. The FPGA is connected to a host PC via a USB 2.0 interface for configuration and steering purpose.

5.2 The DCDB Test Firmware

5.2.1 Complexity Distribution: Software vs. Hardware

Within the DCDB test environment the FPGA plays a major role acting as a mediator between the DCDB on the one side and the host PC on the other side, and it is the designers choice which degree of complexity to implement in here. In order to run the DCDB, it is obviously mandatory to have the clock and the reset, the two primary steering signals, generated at some place. Synchronously, the data transfer to and from the DCDB must be managed. In addition, there is a JTAG interface that has to be used to configure the chip. All this has to be implemented somewhere. Defining this place is a fundamental design decision at this point.

In principle, there are two ways to go. The first one is to implement a direct mapping of software-addressable registers inside the FPGA to the various signals of the DCDB. The chip can then be steered directly from the software running on the host PC by accessing these registers. Beside the communication to the external USB microcontroller, there would be no complexity at all inside the FPGA, since the entire intelligence for operating the DCDB would have to be implemented in software. The second way is the complete

opposite. Here, the entire complexity of steering the chip and synchronizing the data streams is implemented inside the FPGA, while only a very light-weight software on the host PC is used for configuring the FPGA and sending/receiving chunks of data on a higher level of abstraction.

In first order, pushing the complexity upstream to the software is a very attractive approach. This is because software is much more flexible to handle and easier to create than a firmware for a FPGA, which is actually programmed using hardware development techniques similar to those applied for building the DCDB's digital block. In particular, it is the much simpler verification that makes the software-oriented solution being desirable. However, this gain in comfort for the developer is bought dearly, since there is a considerable and in fact show-stopping disadvantage coming along with it. That is, the designer has got only rudimental control of the speed of the interface signals. By steering the chip from software, the control of the signal timing is completely left to the lower software and hardware levels of the host PC and the communication channel to the FPGA. This is the PC's operating system and, in the case of the DCDB test environment hardware, the USB components. A very optimistic estimation for the upper bound of the achievable toggling rate at the DCDB's interface signals in this case is given by halving the clock frequency of the interface between the FPGA and USB microcontroller, which is $48MHz$. It is now obvious that designs, for which an operation speed higher than $24MHz$ is a major design goal, cannot be operated using this approach on the present hardware. As a consequence, the intelligence for steering and reading the DCDB must be placed inside the FPGA, in order to be able to run the chip with the target frequency of $320MHz$.

5.2.2 The Conceptual Structure

The structure of the firmware for the FPGA in the DCDB's test environment is dominated by the need for three different clock domains, resulting in three separate blocks. This is the USB block, the DCDB operation related logic and a third unit consisting of all the things that are not related to any of the former two. A schematic drawing of the firmware structure is presented in figure 5-5.

The USB block is, as the name implies, related to the data transfer between the external USB microcontroller and the FPGA. It consists of a communication interface providing this service to other sub-parts of the firmware as well as a register file for keeping system configuration and status information. As mentioned in section 5.1.3, there are two electrical interfaces available to establish a data transfer between these two devices, the high speed FIFO interface and a memory interface. Although using the memory interface seems to be a quite natural way of implementing the register file, it is not used as such for scalability reasons. Instead, the entire communication is done via the high speed FIFO interface. In order to do so, the USB interface block inside the FPGA is a dual-layer communication engine, implementing the low-level protocol for interacting with the microcontroller as well as a simple packet-based data transfer protocol on top of it. This data transfer protocol is very lightweight for not to deteriorate the effective data transfer rate too much. It simply adds a header containing command, address and data length information ahead of a chunk of data. In this way, there is full freedom of accessing the various parts of the design, while having only a plain FIFO interface underlying.

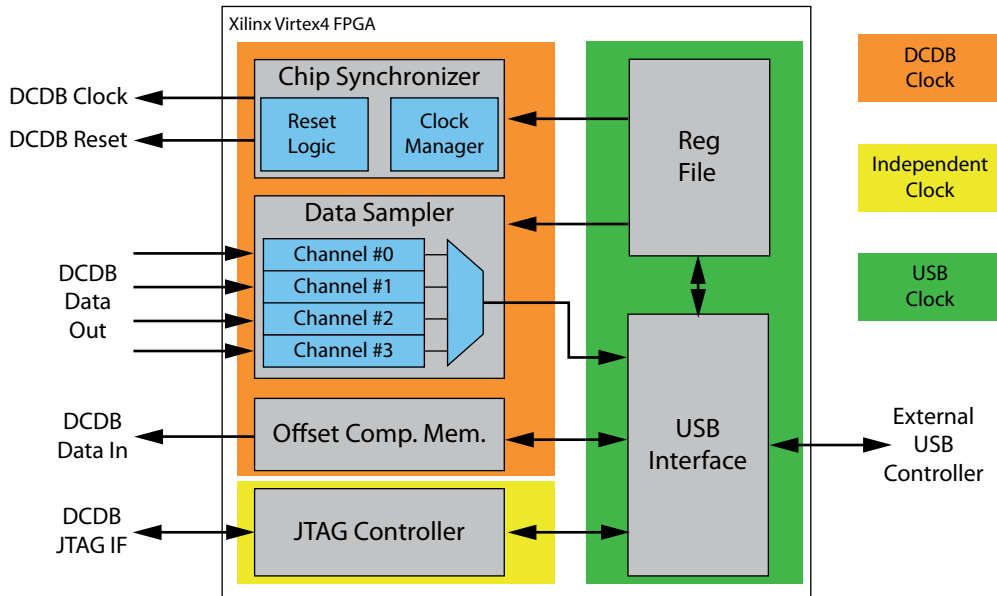


Figure 5-5 Internal structure of the FPGA firmware for the DCDB test environment.

The second block is dedicated to the DCDB operation. This encapsulates mainly three tasks. That is the clock and reset signal generation, providing input data to the DCDB as well as receiving data from the chip. A primary feature of the DCDB test environment is to allow for scenarios with the DCDB running at a variety of clock frequencies. In order to do so, it is necessary for the clock generating circuit inside the FPGA, the so-called *Digital Clock Manager (DCM)*, to be configurable by software during runtime. Obviously, the data transfer logic to and from the DCDB must be aware of changing clock frequencies, which requires for flexible synchronization techniques at the borders to other clock domains. Beyond that, there are several serious issues to tackle in this context concerning the details of implementation. The most important of them are discussed in the following section 5.2.3.

The third block of the firmware contains the logic that is neither related to the USB communication nor to the DCDB operation. Beside several small and very specialized logic units, the JTAG master for configuring the DCDB is the most important one in this group. On first sight, it might look rather curious to implement the JTAG master apart from all the other DCDB related elements. With a look into details, however, this turns out to be a very clever approach. First of all, there is no need for the JTAG interface to run at variable clock speeds. This would rather make the implementation needlessly complicated. Secondly, there might be scenarios, like testing the boundary scan chain for example, where the chip must be accessed via JTAG while the rest of the signals, including the clock, is completely quiet.

5.2.3 DCDB Communication and Data Processing Issues

Due to the DCDB's target interface speed of 320MHz , an adequate test environment for that chip is required to operate at least that fast, rather even faster. It is, however, far from being a trivial task to make core logic of reasonable complexity inside a Virtex4

FPGA working with a clock in that frequency region. The situation is additionally complicated by the fact that the specification of the DCM, which is used for synthesizing the DCDB clock, limits the maximum frequency for the dynamically configurable clock output to only 300MHz [62].

There are two ways to work around the latter issue of too low maximum clock speed. The first one is to use a non-configurable but fixed-frequency clock output of the DCM instead, at least for test scenarios with frequencies higher than 300MHz , since in this case the maximum specified clock frequency is 400MHz . The second and rather crude way is simply to do not care about the specification. In fact, measurements with the available FPGA showed that its DCMs seem to produce stable clock signals with frequencies even beyond 500MHz . To this end, the primary solution is to run the DCM out of specification, while there is still a valid but less comfortable alternative.

In order to handle the potentially too slow core logic of the FPGA, the fraction that really must run at full DCDB speed is reduced to the absolute minimum. Finally, it turns out that only the reset generation part, which is just a seven-bit counter and a little related logic, actually has to run that fast. This is feasible. The entire rest of the DCDB operation related design parts can be slowed down by applying serializer and de-serializer to the data streams directly inside the FPGA's I/O cells and parallelising the data processing. Indeed, this approach results in more used logic elements roughly by the factor of (de-)serialization, but this exactly fits the excellence of a FPGA.

Another issue in this context is the synchronization of data streams from the DCDB to the FPGA. As already discussed in section 4.5.6, there is a considerable delay on that streams due to the clock insertion delay inside the DCDB. In principle, there are again two ways to manage this situation. The first one is to use the return clock signal, which is provided by the DCDB, inside the FPGA to sample the data, resulting in separate clock domains for the two data stream directions. The second solution is simply to keep using only a single clock domain for sending data to and receiving data from the DCDB by adjusting the relevant fine-grain delay elements of the FPGA's input cells in order to find the valid sampling points. However, it is common to both approaches that in addition there is a coarse-grain synchronization necessary in terms of clock cycles. This situation is even further complicated due to the fact that the delay to be compensated is of course not related to the clock frequency but fixed by the DCDB design and the physical interconnection between the FPGA and the DCDB. Hence, even the coarse-grain synchronization must be adjusted to the actual clock frequency of the test scenario. Finally, the implementation complexity of the two approaches is rather similar, thus there is no clear preference for one of them. So for the realization of the firmware, the two data stream direction are implemented in the same clock domain. The return clock signal remains unused.

5.3 The DCDB Test Software

The last element in the readout chain of the DCDB test environment is the software tool running on the host PC. It is created for the Linux operation system using the C++ programming language. The internal structure of the DCDB test software is divided in

three separate parts. This is a low-level part for interacting with the FPGA, a high-level part where the measurement algorithms are implemented and a graphical user interface.

As a consequence of the decision to develop a rather complex FPGA firmware, the low-level part of the software is not involved in DCDB communication directly. Here, the firmware already introduces a certain level of abstraction. A typical task for such a low-level software function in this context is for example to provide reading and/or writing access to a register inside the FPGA by means of the custom-made data transfer protocol as described in section 5.2.2. By using these rudimental access functions, more complex ones can be built like for instance reprogramming the clock generating DCM or configuring the DCDB via JTAG.

The measurement algorithms of the high-level software part use these access functions to drive test sequences on the chip. In principle, this is all about varying the DCDB's input signal in a certain way while capturing and analysing its output. Input signal manipulation can be done either directly inside the DCDB by configuration via JTAG, or from outside using an external signal generator that is connected to one of the DCDB's analog inputs. To this end, the software also includes a GPIB (General Purpose Interface Bus, IEEE-488) communication engine that allows to access and automatically steer compatible laboratory devices like a Sourcing-Measurement Unit (SMU) or Pulse Generator for example. For enhanced user-friendliness the DCDB test software provides a graphical user interface that covers all of the software tool's functionality. It is created by means of the Qt framework [63], which is currently available in version 4.7. The obtained measurement results can be displayed via the integrated plot environment called *KUPE*, which is custom-made and available in the intellectual property pool of the Circuit Design Group. A picture of the graphical user interface is shown in figure 5-6.

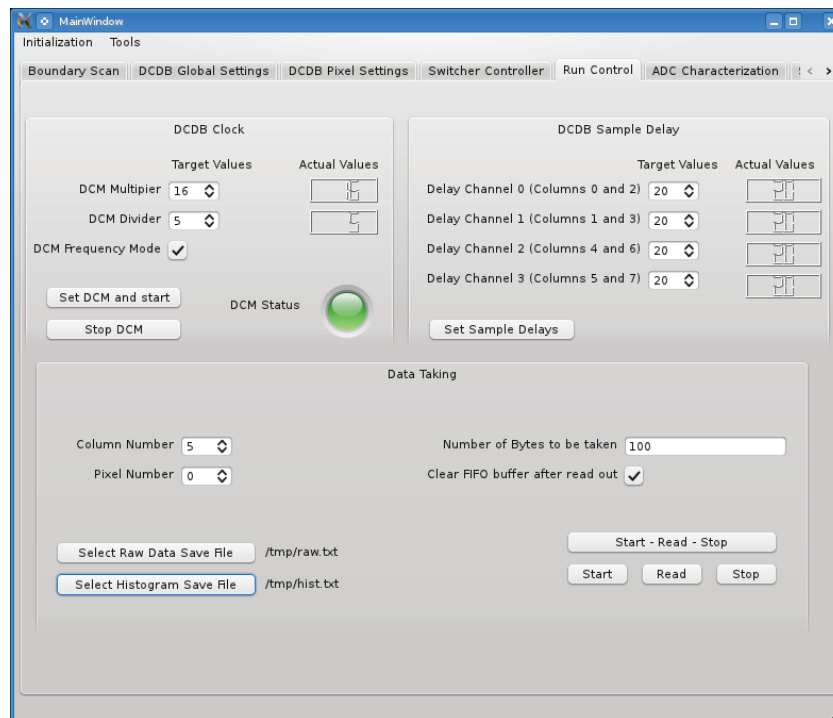


Figure 5-6 Screenshot of the DCDB test software's graphical user interface.

The DCDB-based Detector Prototype

Abstract:

In the following, the hardware and software aspects of a DCDB-based DEPFET prototype system are discussed. The main focus is placed on the features and the implementation details of the firmware for the FPGA, which is responsible for controlling and readout tasks.

The DCDB test environment, which was introduced in the previous chapter, is mainly focused on the characterization of the DCDB chip itself. Once the details of the DCDB's performance are known, it is self-evident to aim for a prototype system consisting of a DEPFET detector matrix, a Switcher steering chip and a DCDB in order to show that the combination of these devices is really suitable for physics applications. In that context, this is a preparatory chapter, describing the hardware and software elements of that system.

6.1 The Hardware Platform

6.1.1 DEPFET Detector and Switcher Chip Selection

The general setup of the DCDB-based DEPFET readout prototype system is rather clear and straightforward: there must be a DEPFET matrix that is steered by a Switcher chip and read out by a DCDB. The DCDB, in turn, has to be connected to a controlling device. Here, the most convenient solution is to use the same chain of DCDRO and FPGA as already implemented for the chip test setup. However, the selection of the DEPFET detector and Switcher type are not that clear in advance. In principle, older versions of both devices that have been used within such prototype systems before could be used as well, depending on the focus of the measurements that are done with it. Having a setup with only the DCDB as new device allows to compare the measurement results to older ones and to extract the influence of the DCDB. Using the BELLE-II versions of the chips instead, gives the most accurate information on the target final design. Actually, up to summer 2011 three combinations of devices have been built. First tests focused on a system with predecessor versions of both, Switcher and DEPFET detector, because the respective BELLE-II versions were not available at that time. For the second system, the older Switcher is replaced by the SwitcherB, while the detector version is kept. Finally, the third system is assembled exclusively with BELLE-II versions of the devices. For the focus of the present work, however, the intermediate version comprising SwitcherB chips, the DCDB and the predecessor DEPFET detector PXD5 (ILC-type) is of most importance, because most measurements have been done with it.

The technical details of the used PXD5 DEPFET detector matrix differ significantly from those given in section 2.2 for the BELLE-II version and even from the corresponding prototype PXD6, so the relevant values are given here. The PXD5 matrix is built on an unthinned wafer of $450\mu m$ thickness and has 16384 pixels in total, organized in a 64×256 pixel matrix structure. Each pixel covers an area of $32\mu m \times 24\mu m$ ¹. The steering and the readout of the matrix is organized in double-row granularity. That means there are 128 pairs of gate and clear lines as well as 128 drain lines. So each logically addressable gate/clear pair actually connects to two physical pixel rows of the matrix that are read in parallel. While the drain lines are all routed to wirebond pads on the bottom side of the matrix, the pads of the gate and clear pairs are

1. The PXD5 matrices are designed with a variety of pixel sizes. The numbers given here correspond to the type of matrix that is used for the setup which is focused here.

split. Gates are connected at the right side of the matrix, clear pads are located on the left side.

When regarding the details of the SwitcherB implementation as described in section 2.3, however, one finds that the interconnection of SwitcherB and PXD5 matrix is not as trivial as it might seem. First of all, the SwitcherB offers only 32 output channel pairs. This is only a fourth of the matrix connections. The second issue is similar to what has already been seen for the DCDB connection, as the SwitcherB does not have any wire bond pads but fully relies on bump bonds. It is of course a straightforward solution to the latter problem to go the same way as for the DCDB. That is the development of a suitable wirebond adapter for the SwitcherB. But right as experienced before, the existence of only a single metal layer for the chosen production technology leads to the fact that only 16 SwitcherB output channels can be routed to wirebond pads. So considering the mismatch of steering lines of matrix and SwitcherB, the situation can be handled in two ways. Either having only 16 lines (32 physical rows) connected or using more SwitcherB chips. From physics point of view it is of course very desirable to read as much matrix pixels as possible, but due to the physical dimensions of the involved chips, however, the latter is not really an option. So the system is built with only 16 pairs of steering lines being connected. Finally, the last cumbersome detail concerns the fact that the PXD5 matrix is designed such that the gate and clear contacts are located at opposite sides. To this end, two SwitcherB chips are actually necessary, one at the right and one at the left side of the matrix, in order to avoid awkward bonding schemes and bond wires crossing the volume over the detector's active area.

6.1.2 FPGA-based Controlling and Readout System

Technically, the most challenging part of the readout sub-system for the DCDB-based DEPFET prototype is the interaction with the DCDB at full speed. Here, it has been shown that the V4Board as used for the DCDB tests is a perfectly suitable candidate. The only drawback, however, is its USB 2.0 interface for PC communication. Its measured data throughput of 12MB/s is more than two orders of magnitude lower than the data production rate of the DCDB at target speed. Fortunately, this is not a serious issue, since a triggered readout scheme is used with the trigger processing taking place at FPGA level, which means that the DCDB can be operated at target speed anyway, independent from any data transmission restrictions. In addition, the board is able to provide all the required interconnections, in particular a sufficient number of lines at its system connector in order to interface also the SwitcherB ASICs beside the DCDB. It is therefore a very convenient decision to use the V4Board as readout sub-system for the DCDB-based DEPFET prototype.

6.1.3 The Hybrid Board

As a result of the previous considerations about which devices to use and what readout sub-system to connect, the boundary conditions to realize the setup onto a printed circuit board are defined. Based on that the hybrid board (Hyb 4.1) has been developed at the Max-Planck-Institut (Munich, Germany) [64]. A picture of it together with a simplified schematic of the chip interconnections is presented in figure 6-1. An important but not shown detail of the hybrid board is the hole in the PCB underneath the detector. It is

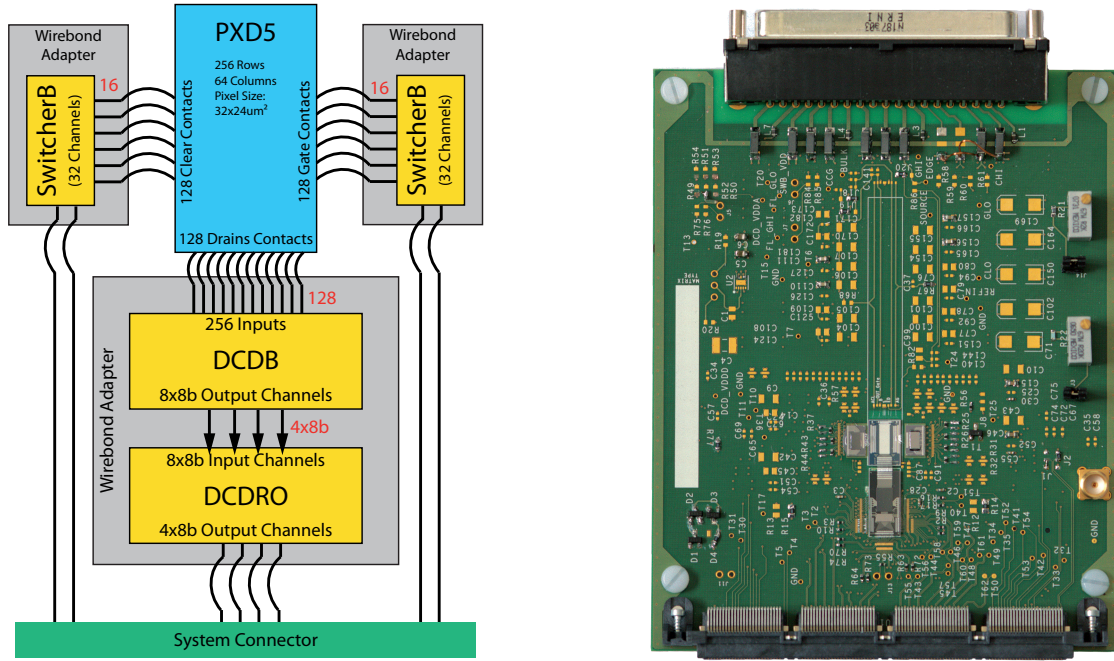


Figure 6-1 Interconnection scheme of the DEPFIET detector, the SwitcherB chips and the DCDB on the hybrid board (left). Picture of the Hybrid 4.1 (right).

large enough to completely uncover the back side of the detector's active area. This allows to expose both sides of it to incident particles. Furthermore, in case of a telescope setup in a beam test, for example, unnecessary material is excluded from the beam, which reduces multiple scattering effects.

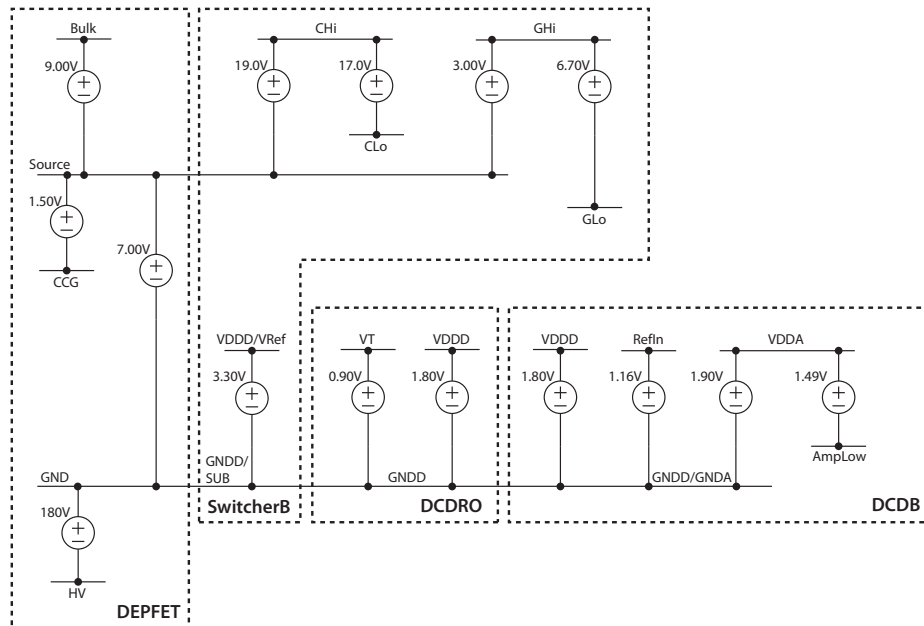


Figure 6-2 Powering scheme for the DCDB-based DEPFIET prototype system.

6.1.4 Powering Scheme

The powering scheme as it is used with the hybrid board is illustrated in figure 6-2. In anticipation of the results presented in chapter 7, the voltages for the DCDB are the optimal rather than the nominal ones. The V_T potential is not originally necessary for the DCDRO, however, due to an implementation detail of the hybrid board, it is required for signal termination. The voltages for steering the detector (CH_i , CL_o , GH_i , GL_o) are considered exemplary and depend on its target operation point. The bias potentials for the DEPFET correspond to the used PXD5 version of $450\mu m$ thickness and may change for future PXD6 setups.

6.2 The Readout Firmware

6.2.1 Overview

Just like for the DCDB test environment, the FPGA is the heart of the readout subsystem. Figure 6-3 provides a schematic of its firmware. In general, most of the firmware for the DCDB test environment is reused here, which is quite convenient, since the issues and solutions for the DCDB configuration and operation as well as the host communication remain the same. Only the data sampler unit has to be rebuilt largely. The reason is that for the DCDB test system the incoming data streams are analysed in the way that only output data of a single ADC is extracted. Here, however, the main goal is to read and capture large continuous data streams, that are regarded as frames of the DEPFET matrix.

In addition, new functional units are required. First of all, this is the SwitcherB controller, which is, as its name implies, responsible for steering the two SwitcherB chips synchronously to the DCDB. The second is the Trigger Logic Unit (TLU) Controller, that processes incoming trigger signals. Both functional blocks are studied in more details during the following sub-sections. Thirdly, since the JTAG interfaces of the DCDB and the SwitcherBs are not chain-connected on the hybrid board, a second instance of the JTAG Controller block is necessary to configure the SwitcherBs. Finally, the fourth new element of the firmware is the protocol engine. This block is used to structure the outgoing data stream towards the host PC by encapsulating raw frame data together with the associated meta information. As optional features, first order data processing like common-mode and pedestal subtraction as well as zero suppression for data reduction are developed [65] (not shown in the schematic). Undoubtedly, these are extremely useful features when targeting the full-blown readout system. At this prototype state, however, the raw analog quality and the pure performance as particle physics instrument are of major interest. So it is necessary to receive unfiltered data from the device, which means that these techniques cannot be applied here.

6.2.2 Trigger Processing

The firmware in its current state implements a triggered readout system. This means that the decision whether to dismiss or keep a captured frame is taken at such an early stage

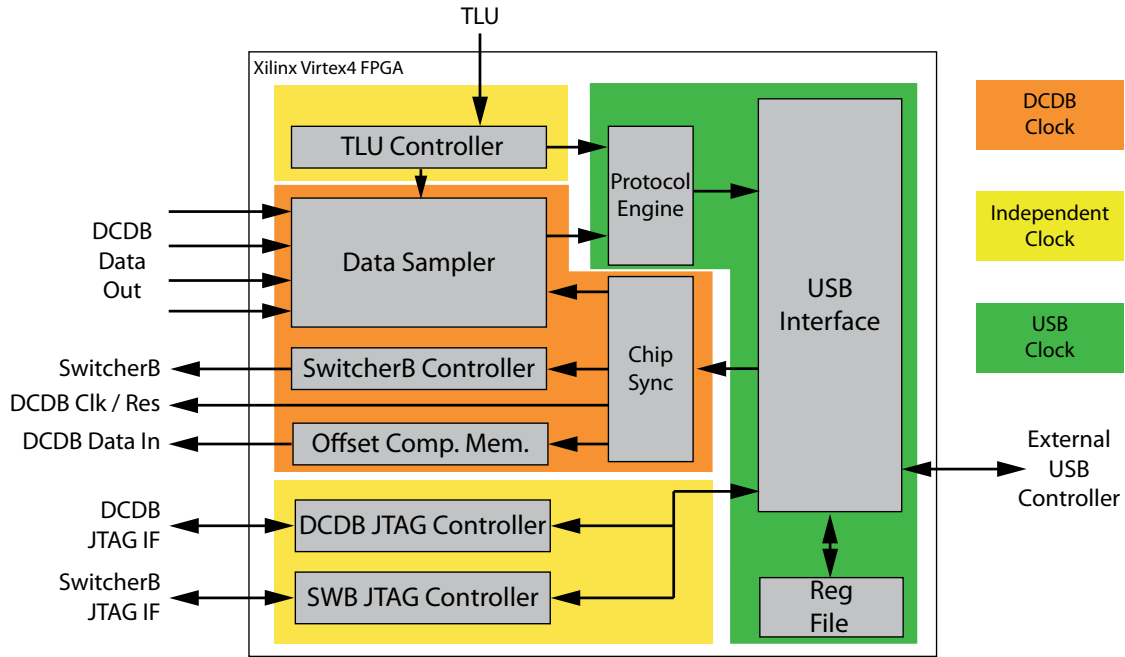


Figure 6-3 Schematic of the FPGA firmware for the DCDB-based DEPFET prototype system.

in the data processing chain that the potential rejection can take place already at FPGA level. So the data does not need to be read and stored entirely at the host PC in order to enable some kind of computer program to scan it afterwards. As already mentioned, this has the very advantageous effect that the matrix readout rate can be decoupled from the available data transmission bandwidth, although nowadays communication and computation systems exist that could stand data rates as produced by the DCDB. The actual reason for applying trigger processing at FPGA level, however, is backward compatibility to existing prototype systems and measurement setups that the DEPFET collaboration has developed over the past years [66].

The standard trigger producer used for DEPFET prototype systems in the recent years is the Trigger Logic Unit (TLU) that has been developed under the EUDET project [61]. It allows the connection of up to six detector readout devices in parallel. The communication between the TLU and the device is established by means of a very simple protocol. It comprises the broadcast of a trigger signal and the transmission of a corresponding serial number. Connected devices can indicate their busy state via a dedicated wire. Triggers are broadcasted only if all connected devices are ready. Within the firmware for the DEPFET readout prototype the TLU Controller block implements the communication protocol to the TLU and forwards incoming trigger events to the data capturing logic. An important detail in this context is the synchronisation of trigger and data stream. First of all, the trigger signal production delay must be considered. Second, once a trigger occurs it must be made sure that the data taking starts immediately regardless of frame boundaries in the data stream. Otherwise, physics events could be lost. In order to provide these features, a sufficiently large ring buffer is used to temporarily store incoming data until it is clear whether it belongs to a triggered frame or not. In addition, the index of the matrix row, that is addressed by the time the trigger

occurs, is stored and sent along with the data as meta information, allowing for the frame boundary reconstruction afterwards.

6.2.3 Trailing Frames

Another data capturing feature that is very interesting from the physics point of view is the device's ability to store trailing frames. That means multiple subsequent frames are captured for a single trigger event. Assuming a reasonably low occupancy of the detector, this feature allows for detailed efficiency studies of the detector's clearing mechanism. Moreover, by adjusting the trigger synchronization delay accordingly, it is even possible to capture frames ahead of a trigger event.

6.2.4 SwitcherB Controller

The SwitcherB Controller is responsible for steering the two SwitcherB chips. Although this seems to be a rather simple task when regarding the SwitcherB's functionality, the timing of the steering signals relative to each other and relative to the DCDB must be considered critical for the system's performance as an instrument for particle physics applications. On the one hand, the period of activation and clearing of the matrix rows must be equal to the DCDB sampling period. On the other hand, the phase shift between matrix steering and the DCDB's sampling point must be configurable at a very fine grain in order to allow for an adjustment of the involved signals in terms of their transition times.

The required flexibility is provided by implementing the switcher controller as a microcode engine rather than as a fixed finite state machine. That means the steering sequence for the SwitcherB is not generated by logic but read from a memory where the plain sequence is stored. This memory is writable from external at any time. Figure 6-4 shows a schematic of this functional unit. The synchronism to the DCDB operation is established by running the memory readout logic with the same clock as used for controlling the DCDB. As a consequence, the granularity of the SwitcherB steering sequence adjustment is derived from the number of DCDB clock cycles per sampling period. This is 32 steps per matrix row, which corresponds to a step width of 3.125ns at target speed.

Regarding the details of implementation, there is a difficulty arising from the fact that the DCDB is starting its digitization job immediately once its reset is released, while the SwitcherB needs some startup phase. This is caused by the SwitcherB's shift register, which cannot be reset but has to be filled entirely with meaningful data prior to starting its actual operation. Implementing this in the context of a microcode engine requires a special way to read the memory. Actually, the available memory space is divided into two sections, denoted *startup sequence* and *run sequence*. The startup sequence is stored into the first section of the memory and its end address is marked as *startup barrier*. The run sequence is stored into the remaining memory space. The strategy is that the startup sequence is meant to bring the SwitcherB into a defined state from which it can start its real operation instantaneously. Therefore, it is executed as soon as the memory is programmed. Starting the DCDB for readout is prohibited during that time. Once the startup sequence is done, the system is ready to run. Starting the run leads to an endless looping through the run sequence. After having stopped the run again, the startup

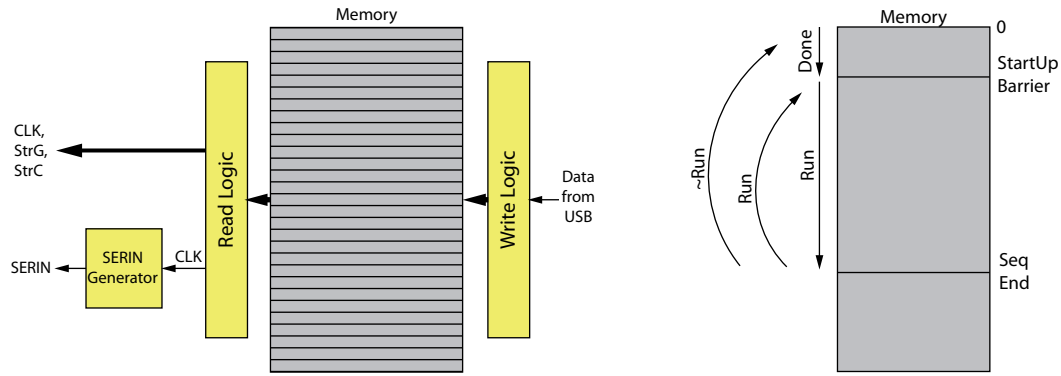


Figure 6-4 Schematic drawing of the SwitcherB Controller (left). The way the memory is read by the Read Logic is illustrated on the right hand side.

sequence is performed once more in order to make the SwitcherB being prepared for a restart.

6.3 The Data Acquisition Software

The backward compatibility of the DCDB-based DEPFET prototype system in terms of trigger processing and data preparation simplifies its integration into the DEPFET collaboration's existing data acquisition (DAQ) environment [67]. That is a Linux-based software system, implemented using a client/server architecture. The latter allows for distributing the various tasks of the system, like data capturing, event building, monitoring and data analysis, among a multiplicity of computation units within a network. This is a very advantageous approach. First of all, some of the tasks are fairly resource consuming. So within a distributed system the tasks can be prevented from interfering with each other in terms of resource allocation. Second, the software structure can be easily adopted to the environmental conditions and restrictions of a beam test campaign, where the experiment must be run and maintained from a considerable spatial distance.

An additional feature of the DAQ system is a ROOT based monitoring tool. It can be used as online monitor obtaining its input data directly from the event builder, or alternatively for offline data analysis of stored data. The tool provides basic data processing features, such as pedestal and common-mode correction, cluster reconstruction and simple track finding.

The DEPFET DAQ system has also been fully integrated into the DAQ framework of the EUDET project [68]. This allows the compatible devices, in particular the DCDB-based DEPFET prototype system, to be run as design under test (DUT) within the EUDET telescope.

DCDB

Characterization

Abstract:

This chapter provides a comprehensive and detailed characterization of the DCDB. It comprises digital functionality checks and an extensive series of analog measurements. The latter focuses on both, very fine-grain performance analyses on selected channels as well as the determination of major quality parameters for all channels on a single chip.

In the following, the DCDB characterization in terms of meaningful performance measurements is presented. All of them correspond to the first full-size version of the DCDB, which is the DCDBv1. So the design version is not denoted anymore throughout this chapter. The organisation of this chapter follows very naturally the bring-up order of the chip. The first section covers the measurements of the DCDB's digital domain, since its proper functionality is a pre-requirement for all other measurements. The second section then provides a detailed examination of the various sub-blocks of the analog-to-digital conversion channel. After that, a full chip analysis provides information about the homogeneity of the performance over the various channels of the chip.

A general problem in the context of DCDB testing is that only very few fully assembled modules are actually available. This includes both, DCDB test boards as well as the hybrid boards for DEPFET readout prototyping. To be precise, until April 2011 there is only a single one of each of the two boards available at the Circuit Design Group, which is fully assembled and sufficiently functional. Most of the measurements presented in this chapter are obtained from the DCDB on the hybrid board, referred as *Golden Module*. There is another fully assembled but partially broken hybrid board, which is only of limited usefulness. The reason for the low availability of chip samples is mainly related to yield issues concerning the chip production as well as the assembly of the enormously complex system.

7.1 Digital Functionality Checks

The quality of the DCDB and its worthiness for the BELLE-II PXD detector project is determined only by its analog performance. But in order to achieve any analog performance at all, the digital part of the design simply has to work. Proving that is the purpose of the following measurements.

7.1.1 Power Consumption of the Digital Block

The first thing to do when initially starting up a new ASIC is to check whether the power consumption is within a reasonable range in order to detect fatal production or even design mistakes. To this end, figure 7-1 provides the measured power consumption of the digital logic block in relation to the frequency of the clock.

The diagram shows two curves. The blue one gives the power consumption of the digital block during normal operation¹. In contrast to that, the red curve illustrates the consumption for the case that the logic is held in reset state. There are two separately labelled y-axes indicating the current measured in ampere and the power in watt. The constant translation factor is the supply voltage of 1.80 V . For each of the measurements, the supply voltage has been regulated to this value by using the sense lines provided by the wirebond adapter.

1. The digital test pattern injection is activated. In fact, this is not generating a typical input pattern for digital logic, but it is at least a reproducible one.

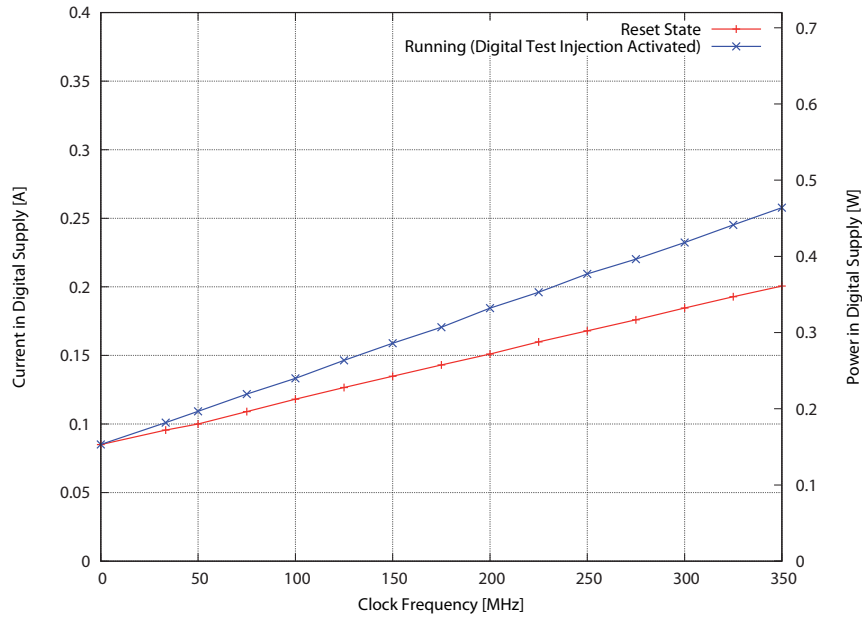


Figure 7-1 Measured current and power consumption of the DCDB's digital logic.

If the clock is switched off (refer to the measurements at 0MHz) both curves show the same power consumption, which is fairly trivial. However, one would expect that the static power consumption of digital logic is negligible, so that would be much less than the measured 90mA . The reason is that the digital I/O cells are powered by the digital supply as well, and they do have a static power consumption. There is no separate I/O supply.

If the clock is switched on, the power consumption rises linearly with the frequency, just as expected. However, the slope of the two curves is different. This is because only the clock distribution network contributes to the power consumption if the system is clocked while being statically held in the reset state. Whereas during normal operation, there is in addition the power consumption caused by charging/discharging the internal control and data lines.

For nominal operating conditions with 1.80V supply voltage and a 320MHz clock frequency, the measured power consumption of the digital logic block is 0.43W .

7.1.2 Clock Insertion Delay

The simplest of all digital tests is to prove whether the injected clock signal is returned via the DCDB's return clock output. Since the clock input and the return clock output are root node and leaf node to the same clock tree, there is only a chain of buffers and thus no sequential logic between them. The only fact that makes this measurement a little difficult is that both pins are part of the DCDB's boundary scan chain. Indeed, the input cells of the boundary scan chain do not implement the SAMPLE/PRELOAD functionality as required by the JTAG standard. That means, from the core logic's point of view these input cells are always transparent. The output cells of the boundary scan chain, in turn, are not necessarily transparent on startup. So in order to make sure to see

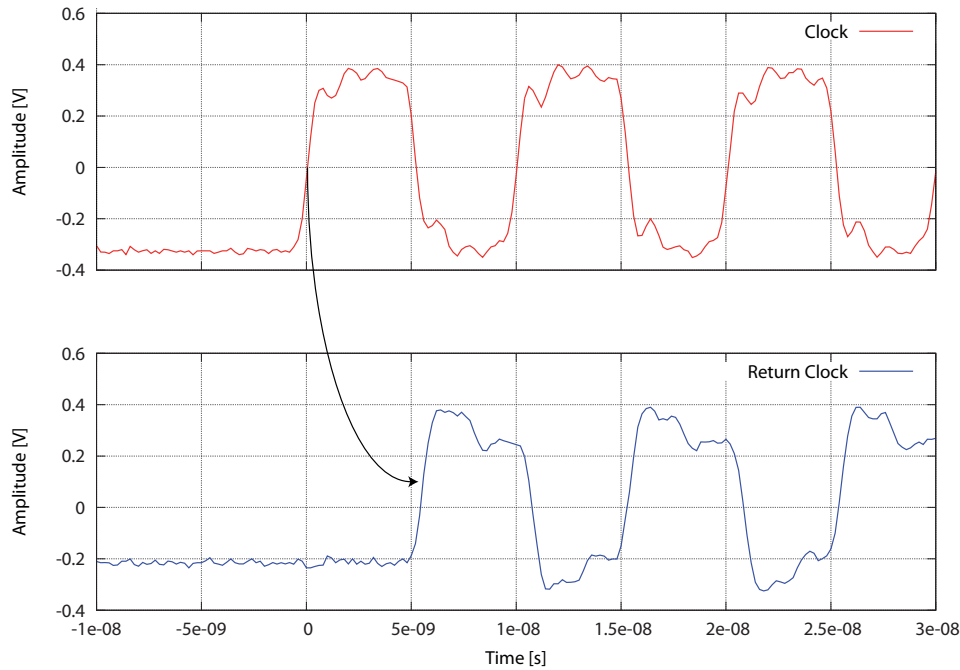


Figure 7-2 Measurement showing the DCDB clock signal (upper curve) together with the return clock.

the return clock at the output, the TAP controller of the JTAG block must be reset. Since the TRSTB reset mechanism is implemented here, this is done by simply pulling that signal low.

Figure 7-2 shows an oscilloscope measurement capturing the clock input as well as the return clock output. The two signals are probed via test points on the DCDB test board. The obvious result is that there is a properly shaped clock signal at the return clock output. The absolute delay between corresponding edges of these signals is 5.48 ns . The digital block is alive, even though up to now only in a very rudimentary way!

7.1.3 Digital Test Signal Injection

As described in section 4.2.7, the DCDB's digital block can be fed with a certain pattern of constant input values instead of real analog-to-digital conversion results from the ADCs. The related multiplexers, that internally select either of the two signal sources, are controlled by a configuration register which is accessible via the JTAG interface. Thus, generating the digital test output pattern is not only verifying the data processing logic but also the JTAG configuration interface.

Figure 7-3 provides a simulation snapshot that shows exemplarily what pattern to expect from the two most significant bits of the DCDB's output bus #5 on a very short time scale. The corresponding oscilloscope measurement is presented in figure 7-4. One finds that the measurement fits the simulation.

Actually, the generated output pattern does not only affect the two most significant bits but the entire output bus. In addition, the pattern's period is as long as 128 clock cycles. It is therefore hardly possible to capture and present meaningful oscilloscope plots

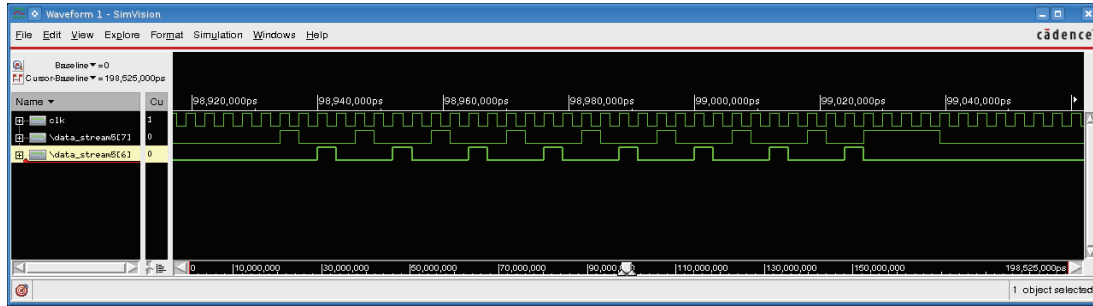


Figure 7-3 Simulation snapshot of a part of the signal pattern on DCDB's output bus #5 generated by the digital test signal injection.

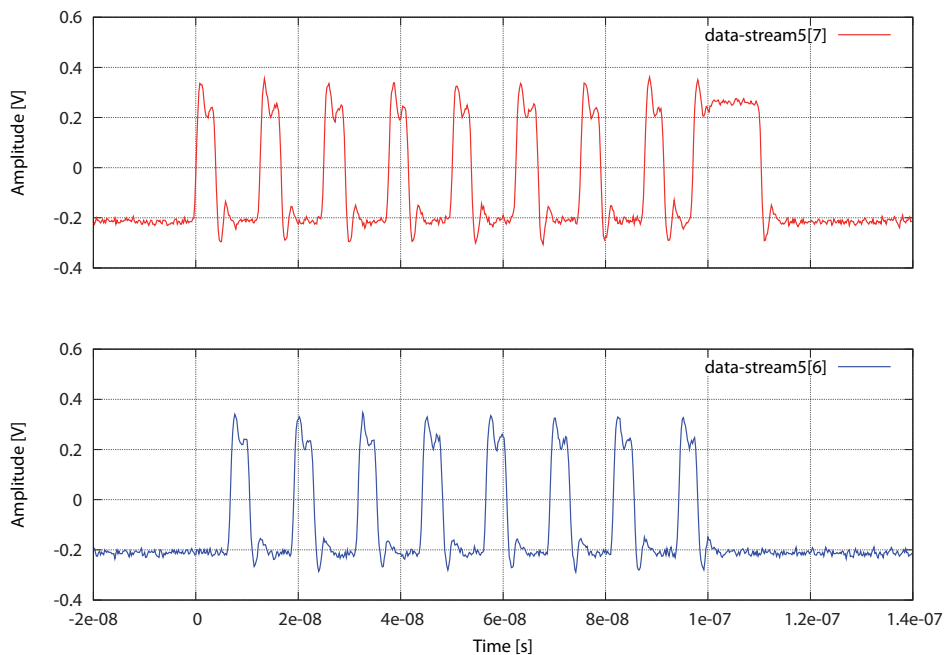


Figure 7-4 Oscilloscope measurement of the two most significant bits of the DCDB's output data bus #5 with digital test signal injection enabled.

showing the entire pattern. To this end, a ChipScope logic state analyser core is embedded into the firmware of the FPGA on the V4Board. By using this tool, much more data can be sampled for longer time. Figure 7-5 shows a screenshot of the ChipScope window after capturing the test pattern of output bus #5 for a certain amount of clock cycles. As explained in section 5.2.3, the data is deserialized immediately after entering the FPGA. This is why the test pattern is spread over four buses of eight bits each. A simulation snapshot showing the same data stream is presented in figure 7-6 for reference. In order to simplify the comparison of simulation and measurement, the simulation snapshot displays the generated data in the same deserialized way. Again one recognizes that simulation and measurement match even on a longer time scale.

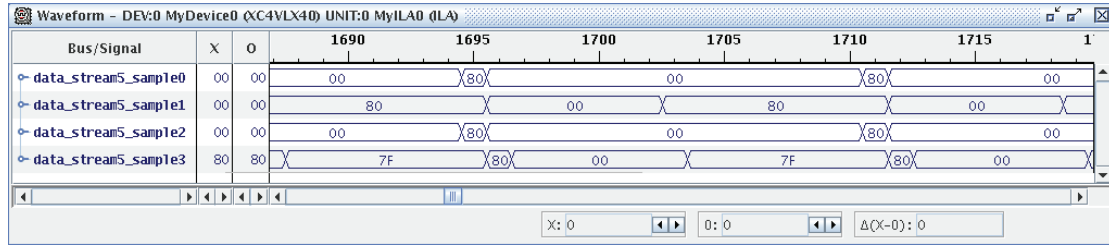


Figure 7-5 Screenshot of the ChipScope Logic Analyser software tool. It shows the deserialized data stream on the DCDB's data output bus #5 with test injection being enabled.

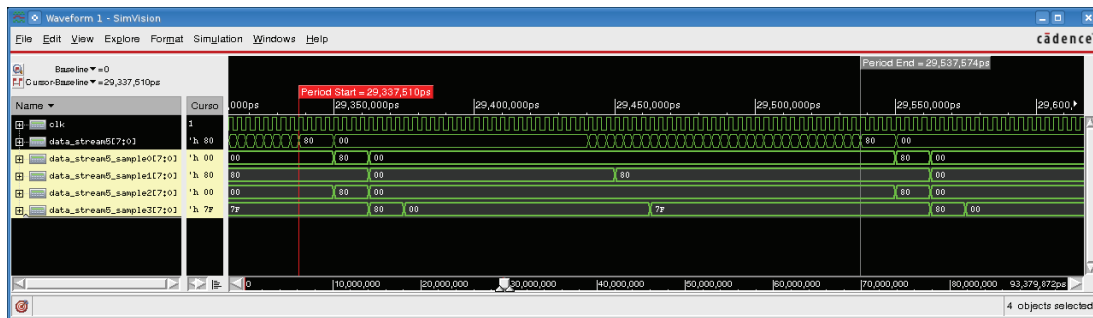


Figure 7-6 Simulation snapshot showing an entire period of the generated test pattern at the DCDB's data bus #5. For convenience, the pattern is deserialized in order to compare it to the measurement in figure 7-5.

7.1.4 Maximum Operation Speed

The last one in the series of purely digital checks focuses on the operation speed of the DCDB's digital logic block. The question is as follows: Assuming a given supply voltage of the digital domain, what is the maximum clock frequency the logic is able to work with? In this context, „to work“ means that there are no setup time violation on any of the internal logic paths.

Although it does not completely cover all the paths of the design, the best effort approach to realise such a test for the DCDB is to use its digital test signal injection. The simplifying assumption here is that the chip is considered working for a certain combination of supply voltage and clock frequency, if it produces the expected output pattern. The result of this experiment is illustrated in figure 7-7.

On the lower end of the x-axis, there is a maximum clock frequency of 100MHz measured for a supply voltage of 1.08V. The nominal operation speed of 320MHz is reached at 1.64V. Even a clock frequency of 350MHz is possible, which is more than the system is actually designed for.

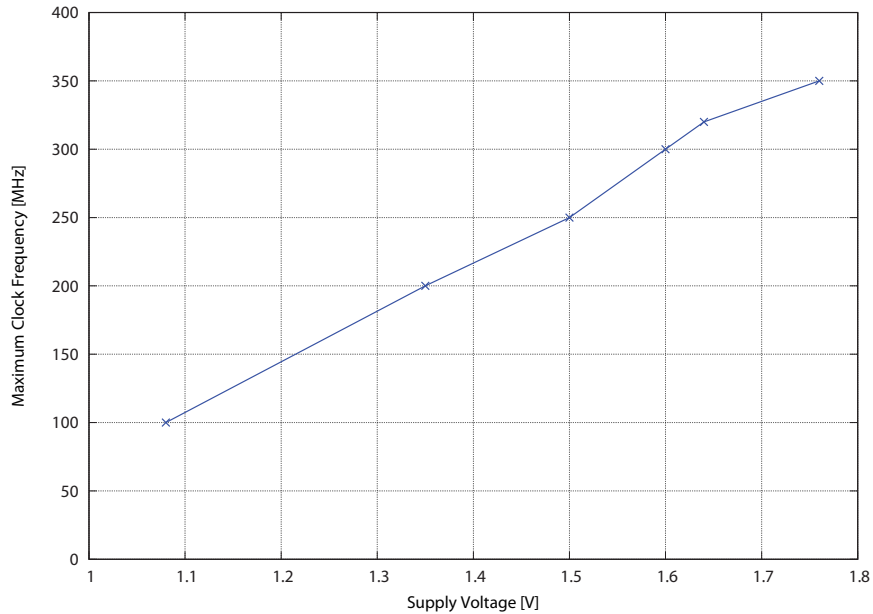


Figure 7-7 Relation of supply voltage and maximum operation speed of the DCDB's digital logic block.

7.2 Detailed Analog Channel Measurements

In this section of detailed analog measurements, the various building blocks of a channel are analysed separately, providing a deeper understanding of their individual functionality as well as their contribution to the overall analog performance. The series of measurements starts with an examination of the most basic block, the current memory cell (CMC). Secondly, the operation performance of a single ADC is shown. After that, the transimpedance amplifier (TIA) is focused. Finally, the combined performance of TIA and ADC is investigated.

An important fact about the DCDB design in the context of analog measurements is that it is not possible to (de-) activate a single one or a fraction of channels. That means, although only a single channel of the DCDB might be focused for a certain test, all the others are operating in the same way. Thus, all the effects that may arise from the influence of the large number of channels, such as voltage drops for example, are implicitly contained in all the following results.

7.2.1 The Current Memory Cell

The main purpose of the current memory cell is, of course, the storing of a current. Beside that, the current memory cell can be regarded as a regulator circuit, that aims for keeping the potential of its input node constant. So for every input current within the CMC's dynamic range, the input node's potential is expected to show almost no change,

while it should rise/fall instantaneously and rapidly once the input current exceeds the dynamic range. This behaviour can be measured.

Establishing the stimulating current and measuring the resulting potential is done from external via a Source Meter Unit (Keithley Instruments Inc., Model 2400 SourceMeter), that is connected to the monitor bus of the DCDB. The EnDC switch is closed only for the single channel, which is addressed for this measurement. The cascode transistor within the monitor path is excluded via the VDC configuration register. The transimpedance amplifier is bypassed by setting the AmpOrADC switch accordingly. The unavoidable resistive influence of these three transistors to the measurement result is compensated by calibration. Inside the channel, the connection to one of the ADC circuits is arranged by activating the test mode (Enable Test) and statically closing one of the two sample switches (SmpL or SmpR). The two Sync configuration registers are set to zero in order to bring the CMC into write state. Here, the current is not additionally manipulated inside the CMC, resulting in a dynamic range that is expected to be symmetric in positive and negative input currents¹.

Config / Supply	Value
VAmpPBias	120
VFBPBias	120 (high) / 60 (low)
VPSource	120 (high) / 60 (low)
VPSource2	120 (high) / 60 (low)
VPSourceCasc	40
VFBNCasc	0
VRefFB	60
VDDA	1.8V (sensed)
RefIn	1.16V (sensed)
AmpLow	360mV (sensed)

Table 7-1 Configuration and supply values for the CMC measurements.

Figure 7-8 presents the result of this measurement for two different CMCs and two different bias settings. Diagram a) is obtained from a CMC that is located in the vicinity of the DCDB's power supply pins. Its coordinates within the 16×16 matrix in which the channels are arranged on the DCDB is C10-R0-L. That means column #10, row #0, left ADC. In contrast, diagram b) shows the behaviour of a CMC that is as far away as possible from the root of the power supply. The corresponding coordinates are C10-R15-L. Each of the two cells is measured at two different operation points in terms of the bias currents of the three sources FBPBias, PSource and PSource2. The low bias setting corresponds to the nominal operation condition as defined by simulation. The nominal settings for the other relevant configurations are listed in table 7-1.

For the nominal low bias setting, the CMC of diagram a) shows a dynamic range of $16\mu A$ in the boundaries $[-8.0\mu A, 8.0\mu A]$. This fits perfectly to the expectations! At the same conditions, CMC b) has a dynamic range with only very slightly reduced width

1. Currents flowing into the CMC are considered positive, those flowing out of it are considered negative.

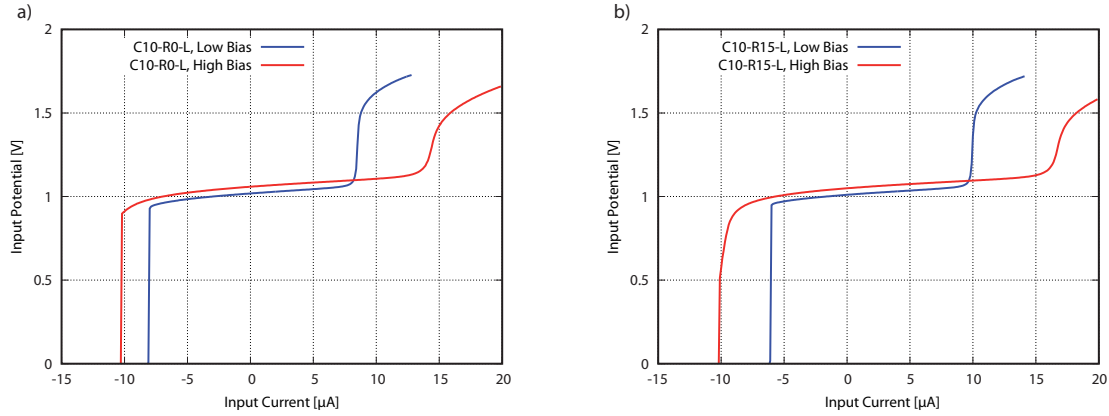


Figure 7-8 Input characteristic of a current memory cell (CMC) for two different bias settings. Diagram a) shows the characteristic of a CMC that is located at the lower end of the analog domain, right next to the DCDB's power supply pins. Diagram b) provides the results of the same measurement targeting a CMC at the upper end of the analog domain.

of $15.9\mu A$. But here the boundaries are $[-6.0\mu A, 9.9\mu A]$, so there is a significant shift towards the positive currents by $2\mu A$. Most probably this shift is caused by on-chip voltage drops. When changing from low to high bias settings, two major observations can be made. The first one is that the increase in dynamic range is less than the expected factor of two, which can be explained by current sources that operate near their maximum, even for nominal settings. This seems to be true in particular for the FBPBias source that is responsible for the rather smooth end of the dynamic range of CMC b) for negative currents. The second observation is an additional shift of the dynamic range towards the positive currents, that is nearly equal for both CMCs. This cannot be explained by voltage drops, but rather points to unbalanced strengths of the involved current sources. It seems that PSource2, the sinking source within the transconductor, tends to be too strong at the high bias settings.

The quality of the CMC characteristic has direct influence on the ADC operation performance. Due to the ADC's algorithmic nature it is essential for the CMC to have a dynamic input range that is symmetric in positive and negative currents. In other words, the width of the symmetric fraction of the CMC's dynamic range is an upper bound for the resulting dynamic range of the ADC. The eventually remaining part, which is out of symmetry, is of no use for this application. Indeed, by adjusting the three relevant current sources relative to each other, at least that contribution to the asymmetry which is caused by higher bias settings can be compensated. In this case, however, it turns out that the CMC's stability within the ADC operation is significantly degraded.

In order to further evaluate the quality of the CMC design, more measurements determining the performance stability are required. Figure 7-9 presents the results of a detailed investigation of influences on the CMC's input characteristic caused by variations of the supply voltages VDDA, AmpLow and RefIn as well as the RefFB bias configuration. All these measurements are performed with the same cell and identical settings: It is again the cell C10-R15-L, which is located far away from the root of the chip's power supply, operated at high bias settings.

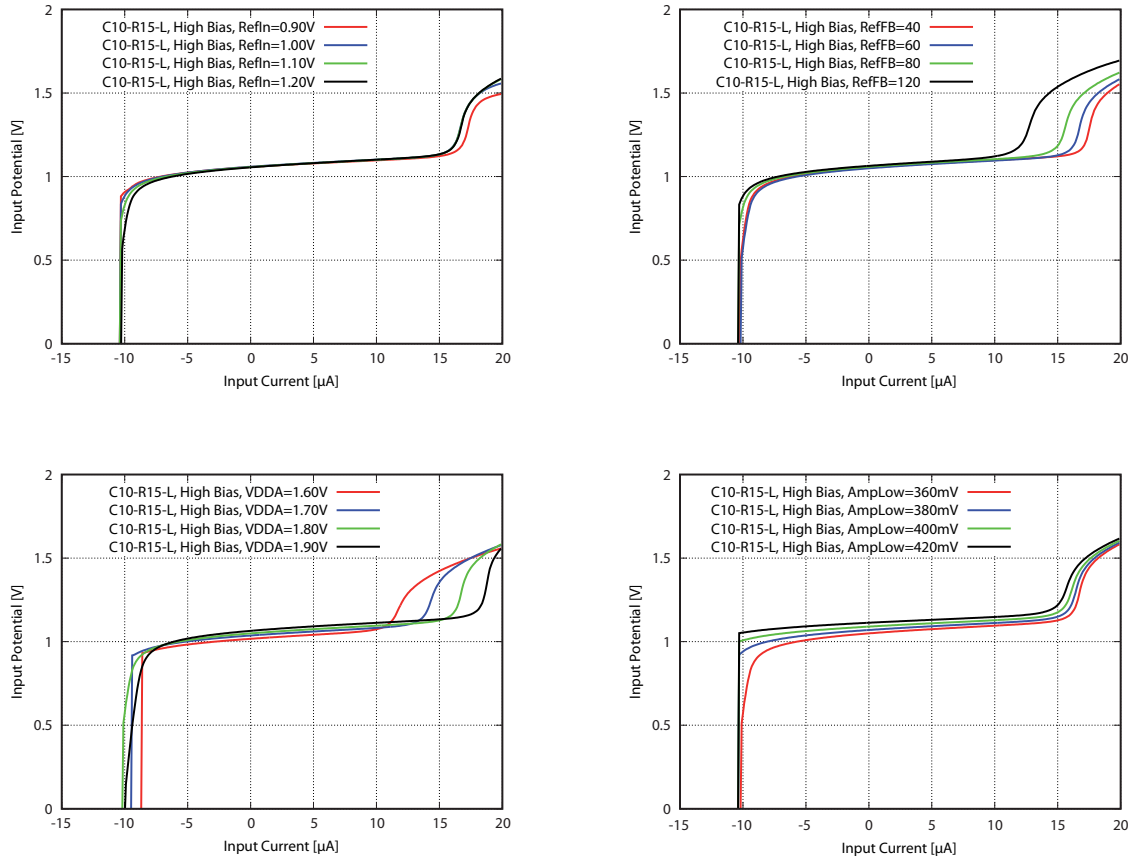


Figure 7-9 Influence of variations of the supply voltages VDDA, AmpLow and RefIn as well as the RefFB bias configuration on the current memory cell's input characteristic.

First of all, a variation on the RefIn potential has almost no effect on the CMC input characteristic. This is expected, since it is not directly in touch with the input node when the cell is in write state. Changing the RefFB setting is effecting the end of the dynamic range for positive currents. This behaviour can be explained in the following way. A high configuration value for RefFB leads to comparatively low gate potential of the transistor at the output side of the transconductor's differential pair. That is why the current from the FBPBias source can hardly be switched off completely at this side, which reduces the maximum amount of current the transconductor is able to sink. Varying the VDDA supply voltage leads to an interference of several effects, so it is hard to really name them separately. Nevertheless, it is believed that the major contributions rather come from the bias generation circuits than from the CMC itself. For example, lowering the potential leads to similar effects on the positive end on the dynamic range as already observed in the RefFB configuration value sweep. Since the RefFB bias generation is mainly based on a voltage divider circuit, a relation is likely here. In the same way, the effects on the other end of the dynamic range can be linked to bias changes of FBPBias. Finally, the vertical (potential) shift effect of AmpLow variations is rather obvious as this potential is used as virtual ground for the CMC's amplifier and thus defines the potential of its input node.

7.2.2 The Analog-to-Digital Converter

The next step in the series of detailed analog measurements is the examination of the ADC's performance. Here, the first task is simply the determination of the ADC's characteristic in terms on dynamic range, noise and linearity.

Config / Supply	Value
VAmpPBias	120
VFBPBias	120 (high) / 60 (low)
VPSource	120 (high) / 60 (low)
VPSource2	120 (high) / 60 (low)
VPSourceCasc	40
VFBNCasc	0
VRefFB	60
VPDel	127
VNMOS	60
VDDA	1.85V (sensed)
RefIn	1.16V (sensed)
AmpLow	360mV (sensed)

Table 7-2 Configuration and supply values for the ADC measurements.

The basic idea of the measurement setup is to let the ADC operate in its normal mode while sweeping the input signal very slowly. Normal operation mode means that the switches inside the ADC are steered in accordance to the algorithmic procedure at target speed. So a new input signal is sampled every 200ns , while the meantime is used for conversion. The input signal to the ADC is changed much slower by several orders of magnitude, so it can be considered constant for the scope of a single conversion. Hence, the following measurements are of semi-dynamic nature. The input signal is generated by the PInjSig current source. Due to the fact that the transimpedance amplifier is not used and thus is switched off here, the signal source is directly connected to the common input node of the channel's two ADCs. The NSubOut current source is used to subtract a constant current from this node in order to provide both, positive and negative currents to the ADC. The two current sources are externally calibrated. The baseline configuration settings and supply potentials are listed in table 7-2.

Figure 7-10 presents the results of such measurements for the two ADCs C10-R0-L and C10-R15-L, both with low and high CMC bias settings (VFBPBias, VPSource and VPSource2). The transfer curves in the diagrams a) and b) show the relation of the input signal current to the generated digital output code. For an ideal ADC, a sweep of the input signal should generate a straight line from the one end of its codomain to the other, in this case from the value 127 for large negative currents to -128 for large positive currents¹. Here, each point of the transfer curves is the mean value of 100 samples taken for the respective input signal. The analysis of the curves covers the following aspects:

1. Here, a negative current flows out of the ADC, a positive current flows into the ADC, respectively.

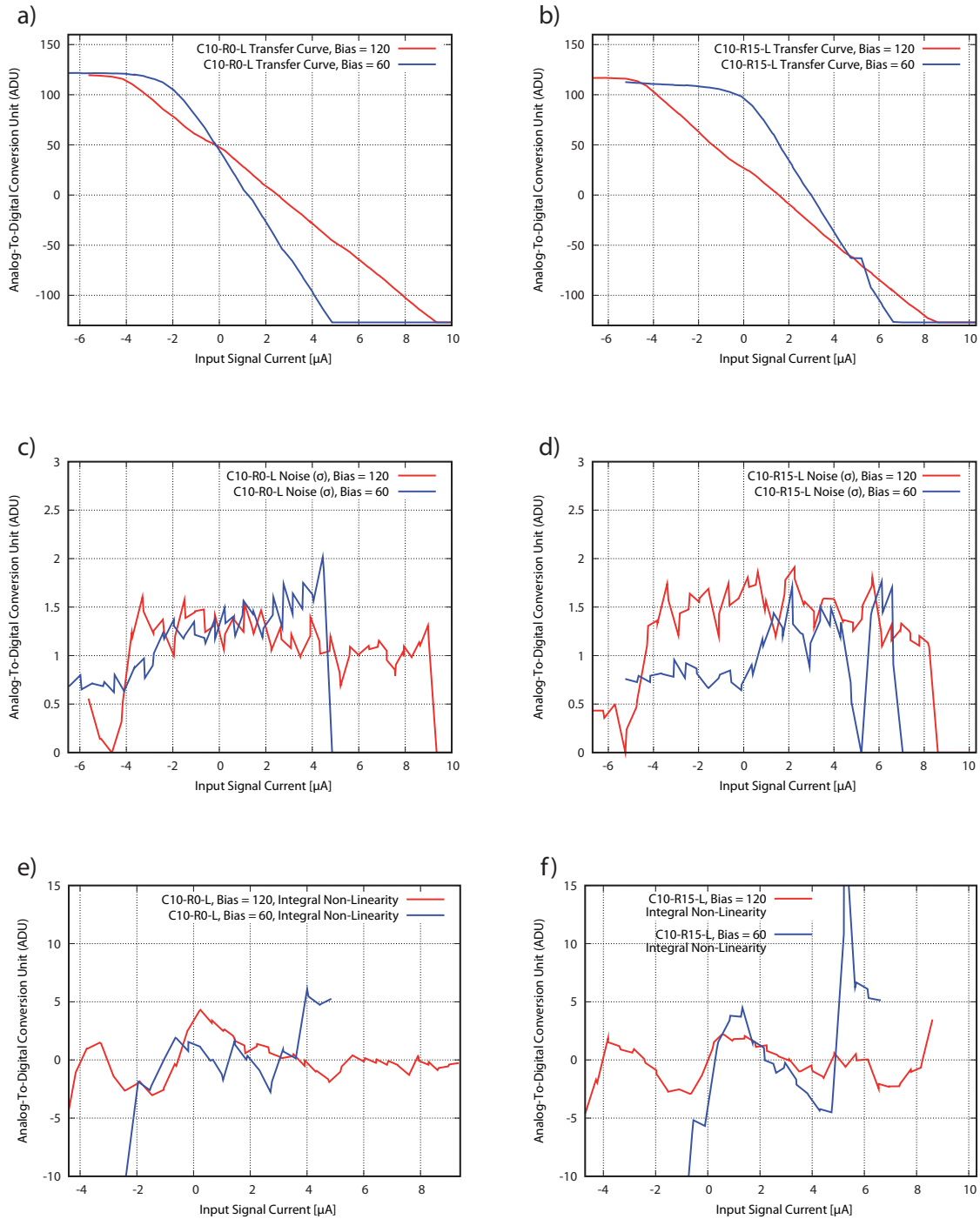


Figure 7-10 Performance analyses of two ADCs, located at C10-R0-L (near power supply pins) and C10-R15-L (far away from power supply pins). The transfer curves as well as noise and linearity analyses are given for different CMC bias settings.

noise, linearity and dynamic range in conjunction with the resulting gain. The noise analyses are provided in diagrams c) and d). The noise is calculated separately for every input signal of the ADC by the standard deviation (σ) of all the samples for that particular signal. The linearity of the transfer curve is evaluated by the so-called *Integral*

Non-Linearity (INL), which is its deviation from a (best) straight line fit, shown in diagrams e) and f). The gain of the ADC is given by the ratio of input signal and resulting output code. It is obtained from calculating the slope of the straight line fit. Finally, the input signal range for which the ADC's output code is not equal to either the minimum or the maximum value and its transfer curve shows a reasonably low non-linearity is considered being the ADC's dynamic range.

With high bias settings the ADC C10-R0-L shows a transfer curve with acceptable linearity for the input signal range of $[-4.4\mu A, 9.4\mu A]$. Obviously, it is not centred around $0\mu A$ input current but shifted towards the positive currents. This is believed to be caused by charge injection effects at the switches used within the CMCs and can be compensated by adopting the RefIn potential. Lowering the bias settings to the nominal (low) values results in two major differences. The first one is the smaller dynamic range. This effect is expected, because the PSource current source, which is used to manipulate the currents during the conversion procedure, is weaker in this case. So the range of currents that can be kept within the CMC's operation window is smaller. The second observation is that the transfer curve shows a noticeable non-linearity for input signals at the negative end of the dynamic range. Such signals are therefore not considered valid. Thus, the effective dynamic range for that bias setting is $6.8\mu A$ with the boundaries $[-2\mu A, 4.8\mu A]$. The noise performances of the two alternatives do not differ very much. $1.4ADU$ is measured for the nominal setting, while it drops to $1.2ADU$ for the higher bias.

The situation is similar for the ADC C10-R15-L. For the high bias setting, the ADC's dynamic range reaches from $-4.7\mu A$ to $8.6\mu A$, so it is $13.3\mu A$ in total. The nominal (low) bias setting leads again to a smaller dynamic range that is additionally cut for non-linearity reasons. Furthermore, a step appears in the transfer curve at an input signal of about $5\mu A$, indicating missing output codes. The noise performance, however, is again not varying too much between the two options. There are about $1.2ADU$ for the nominal setting versus $1.5ADU$ for the higher bias.

There is another effect that both investigated ADCs seem to suffer from, independently from the bias setting. The maximum digital code that is reported by the two ADCs is not the maximum number of the codomain, 127, but 119 for C10-R0-L and 117 for C10-R15-L. This is dominantly caused by non-linearities of the NSubOut current source, which is operated at its maximum in order to make the ADCs showing their full range. For the final performance this effect can be neglected, since the NSubOut current sources do not have to be used when the transimpedance amplifier is activated.

As a conclusion, for both examined ADCs the high bias setting leads to the more useful results. These are a larger dynamic range and a better linearity. So the major result from this analysis is to select the value of 120 as new nominal bias configuration for the three current sources FBPBias, PSource and PSource2 of the current memory cell and the comparator circuit.

7.2.3 The Analog-to-Digital Converter: Stability Measurements

Beside its pure functionality, the stability of an ADC's performance for varying operating conditions is a further very important quality parameter. Figure 7-11 shows the results of adequate measurements to determine the robustness of the two ADCs C10-R0-L and C10-R15-L. The measurement setup is the same as used above. Both

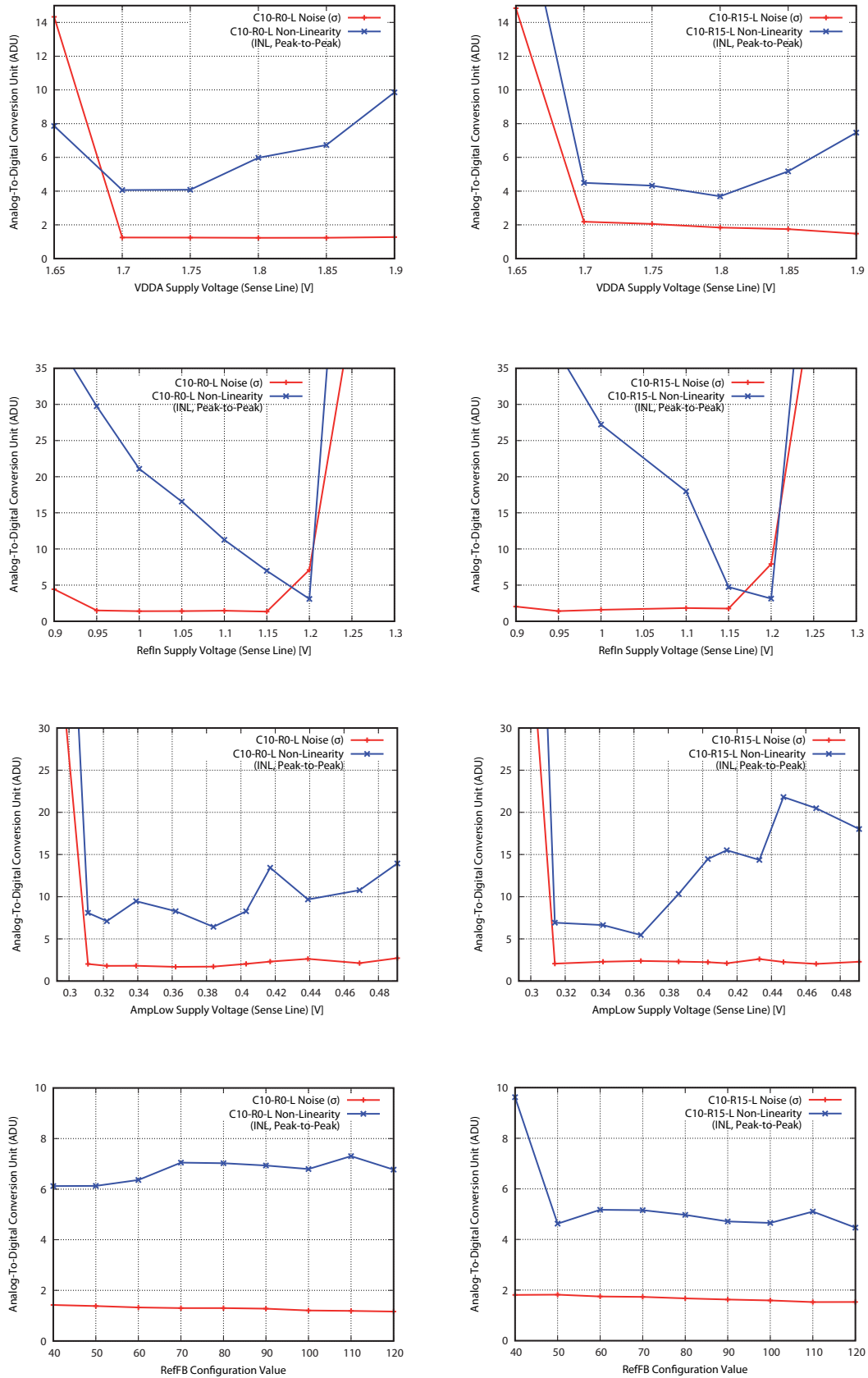


Figure 7-11 Stability analyses for the ADCs C10-R0-L and C10-R15-L: noise and linearity versus supply voltages and RefFB configuration.

ADCs are operated with the new nominal bias settings of 120 and nominal speed. For the other configurations the values given in table 7-2 are kept. The measure of stability is chosen to be the noise and the linearity. The noise is calculated as standard deviation (σ) of all sampled values over all input signals. For the linearity statement, the peak-to-peak distance of the Integral Non-Linearity (INL) is calculated.

A sweep of the VDDA supply potential around the chip technology's nominal operation voltage shows that a minimum of $1.70V$ is necessary to bring the noise behaviour of both ADCs into a reasonable range. For higher voltages it remains stable on a low level. The linearity, however, seems to show optima: at the nominal supply of $1.80V$ for C10-R15-L and even below that for C10-R0-L. The nominal supply voltage seems to be a good compromise.

When searching for a valid operation point in terms of RefIn potential, the first fact to recognize is that the simulated nominal value of $0.9V$ is not a good option to start from because of enormous linearity issues. In fact, looking into the raw data of these measurements reveals that the corresponding transfer curves show steps in the order of 20% of the output code range. The linearity is improving rapidly, however, for higher potentials of RefIn, at least up to a value of $1.20V$. This point seems to be a hard limit for a proper functionality. The reason why the system is so sensible on the RefIn potential is most probably the fact that this potential is actually used for several more or less unrelated purposes. Beside its use cases within the contexts of current memory cell and comparator circuits, it is used beyond that, for instance, as bulk potential for several switch transistors like SmpR and SmpL. As a consequence, the RefIn supply potential must be considered critical for the ADC performance and the optimal operation point lies within a very narrow range of a few $10mV$ around $1.20V$.

Analysing the influence of AmpLow on the ADC performance leads in first order to the same fact as already seen for the RefIn potential: the nominal value from simulation, here $0.3V$, is too low. In order to make both examined ADCs operate in a reasonable way, a minimal potential of about $0.32V$ is required. When further increasing the potential, the influence of AmpLow can be considered less aggressive compared to RefIn. The noise is stable on a low level, while the linearity is not degrading significantly in either of the two ADCs before $0.38V$. The best compromise between the two ADCs is a value of $0.37V$, while $0.35V$ would be a rather conservative decision, since it provides a 10% safety margin in both directions.

The last one in this series of stability analysis measurements is focused on the influence of the RefFB configuration, which is used to adjust the operation point of the transconductor's differential pair. The only serious restriction resulting from these measurements is that the setting must not be too low. That means at least 50. For even higher values, only very slight effects on noise and linearity can be observed. That is why this configuration can fairly be considered uncritical.

7.2.4 The Analog-to-Digital Converter: Dynamic Behaviour

The ADC measurements and analyses presented so far are considered semi-dynamic, since with respect to the ADC operation the input signal is kept constant. The next step towards an investigation of the ADCs behaviour in a real operation scenario is to measure its output for a changing input signal.

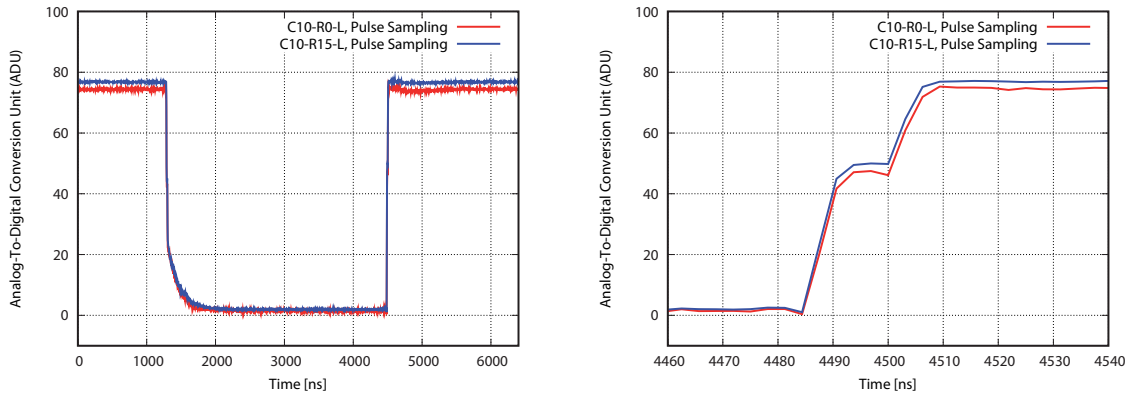


Figure 7-12 Dynamic ADC measurement: ADC output code for a pulsed input signal, full period (left) and zoom to the rising edge (right).

In principle, the setup for such a measurement is very similar to that for the semi-dynamic ones. The main difference is that the PInjSig signal source is pulsed by dynamically controlling the InjectStrobe switch. In order to achieve a time resolution that is significantly shorter than the ADC's sampling period of 200 ns , the pulse is swept in time relative to the sampling point of the ADC¹.

The results for a pulse with a period of $6.4\mu\text{s}$, a width of $3.2\mu\text{s}$ and $7.3\mu\text{A}$ amplitude, sampled with the two ADCs C10-R0-L and C10-R15-L are shown in figure 7-12 (left). One can clearly see that - as expected - both react on the pulse in the same way. An important observation here is, however, that the rising and falling edges show significantly different speed. This cannot be explained with any characteristic of the ADC, but must be an artefact of the signal generation. The falling edge corresponds to the switch-on of the signal (InjectStrobe switch is closed), which leads to a slight change of the non-ideal signal source's operation point. The rising edge, in turn, is caused by the switch-off of the signal, where the InjectStrobe switch is simply opened and the current source's characteristic does not play any role. To this end, it is fair to judge the dynamic quality of the various analog DCDB sub-blocks based on the transition times caused by switching-off the input signal.

The right graph of figure 7-12 shows the same measurement as on the left side but with a zoom to the rising edge. Here, one finds that the signal rise is actually done in two steps. These are caused by the fact that in accordance to its algorithm the ADC samples the input signal successively using two different current memory cells. Hence, in order for the ADC to operate correctly the input signal is required to be sufficiently stable during the extended sample phase of 25 ns .

7.2.5 The Transimpedance Amplifier: DC Measurements

It is the purpose of the transimpedance amplifier (TIA) at the input node of a DCDB channel to receive and optionally amplify the incoming current signal. The most interesting and important characteristic of the TIA is certainly its dynamic performance.

1. This technique is called *Sequential Sampling* and is further explained in [58].

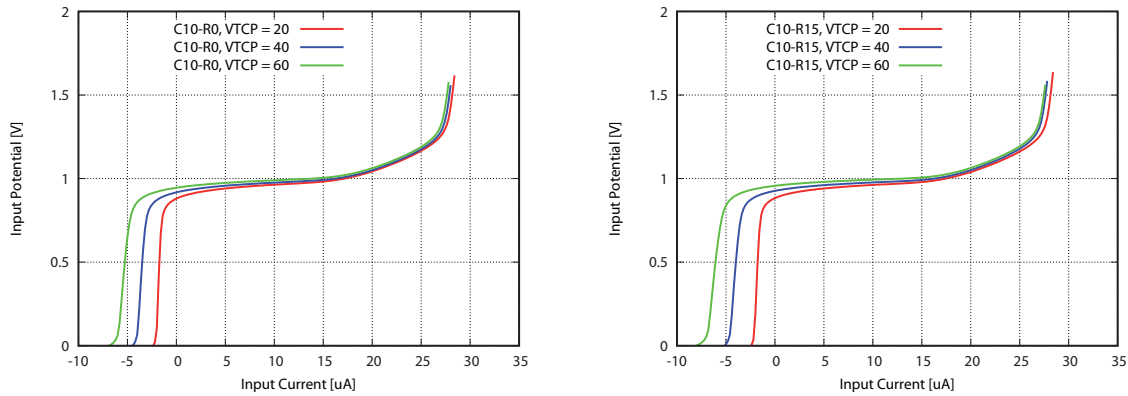


Figure 7-13 Transimpedance amplifier input characteristic for the two channels C10-R0 and C10-R15 depending on the VTCP bias setting.

Before that, however, a very basic feature needs to be tested, which is its capability to regulate the input node to a constant potential for the target input current range. This is $20\mu A$ in case of a TIA amplification factor of one, which is quite reasonable as it is a little more than the designed input range of the ADC.

In principle, the measurement setup is the same as used before for the determination of the current memory cell's input characteristic. An external SMU is connected to the input of the TIA via the DCDB's monitor bus and accordingly set switches. An input current is sourced by the SMU while the potential at the monitor pin is measured. The resistive effect of the involved wires and switches is excluded by calibration¹. The relevant configurations used for the measurements are listed in table 7-3. The results are plotted in figure 7-13.

Config / Supply	Value
VTCP	20
VTCCasc	60
VTCSFN	120
Cap	All four switches closed
CapL	All four switches closed
AmpSFON	Closed
En30	Closed
En60, En90, En120	Opened
VDDA	1.80V
AmpLow	360mV

Table 7-3 Configuration and supply values for the TIA input characteristic measurements.

1. The only difficulty here is introduced by the fact that actually a DEPFET matrix is attached to the DCDB, since the Golden Module is used for this test. Therefore, it is necessary to make sure that all the DEPFET transistors of the matrix are switched off in order to really feed the entire input current into the TIA.

One finds that for the nominal VTCP setting of 60 the dynamic range of the TIA at C10-R0 is $19.6\mu A$ wide, for the TIA at C10-R15 it is $20.6\mu A$. So both show expected performances. Reducing the current of the TCP source leads to a decrease of the dynamic range. For a VTCP setting of 20, the former shows $16.6\mu A$, while the latter has a dynamic range of $16\mu A$ in this case.

7.2.6 The Transimpedance Amplifier: Dynamic Measurements

The most important measurements concerning the TIA are certainly those determining its dynamic performance. This is because they can give the answer to the question whether its signal response time is fast enough for the target application. It is already

Config / Supply	Value
VTCP	60
VTCPL	5
VTCCasc	120
VTCSFN	120
Cap	Two of four switches closed
CapL	All four switches opened
AmpSFON	Closed
En30	Closed
En60, En90, En120	Opened
VAmpPBias	120
VFBPBias	120
VPSource	120
VPSource2	120
VPSourceCasc	40
VFBNCasc	0
VRefFB	60
VPDel	127
VNMOS	60
VDDA	1.80V (sensed)
AmpLow	420mV (sensed)
RefIn	1.16V (sensed)

Table 7-4 Configuration and supply values for the dynamic TIA measurements.

shown that the ADC can properly sample input signals that are stable for at least $25ns$. In order to allow for an overall sampling period of $100ns$ the transimpedance amplifier must therefore be able to show signal transition times of less than $75ns$.

The setup is similar to what is used for the ADC's dynamic measurement. The PInjSig current source is used to inject input signals to the TIA dynamically. The output signal is

sampled by either of the two connected ADCs. The time resolution is improved by sweeping the input signal relative to the ADC's sampling point. The used configuration values are summarized in table 7-4. With the same justification as for the dynamic ADC measurement, only the signal transition caused by opening the InjectStrobe switch is considered here¹. All measurements are done for both TIAs, that at C10-R0 and the one at C10-R15, in order to illustrate eventually existing differences between top and bottom edge of the chip's analog domain. The load capacitance at the TIA's input node introduced by the connected DEPFET detector is estimated to be roughly $3pF$.

In this setup the ADC serves as a measurement instrument for determining the TIA's performance. In first order, this is a very advantageous approach because no external instruments need to be connected by means of comparatively long wires that introduce additional RC elements and therefore distort the results. However, the TIA and the ADC do share common resources, primarily the supply voltages VDDA and AmpLow. As a consequence, the two blocks cannot be considered completely independent but interfere with each other. This is a particular problem in the context of parameter sweeps concerning a target parameter that influences both blocks. So if a malfunction is discovered it must be considered as interference of the device under test (TIA) with the instrument (ADC).

Supply Voltage Variation

The results obtained from varying the supply voltages are presented in figure 7-14. The VDDA potential is swept from $1.7V$ to $1.9V$. The involved ADCs are proven to be functional in this range, so no bad influence is expected from them. Nevertheless, it seems that a potential higher than $1.8V$ is necessary for the TIA to operate, especially for C10-R15, as there is still some gain reduction observable at $1.8V$. With higher potentials, like $1.85V$ or $1.9V$, the TIA's output signal tends to show slight undershoots, but the transition time gets shorter at the same time. For $1.9V$ both TIAs show a transition time of about $60ns$ ² and thus comply to the speed requirements!

By examining the TIA's behaviour for different AmpLow potentials, one finds that the optimal operation point for the TIA does not match to that of the ADC. While $360mV$ is best for the latter, the results here show a sharp optimum at $400mV$. There seems to be only little tolerance to even higher values, which can fairly be explained, however, by a less well-performing ADC. The transition time in the case of $400mV$ AmpLow potential is about $66ns$ for the C10-R0 and about $72ns$ for C10-R15.

The noise analyses confirm these results. For a variation of VDDA, there is no serious functionality in C10-R15 below $1.75V$ and the noise gets optimal towards $1.9V$ for both TIAs. Using an AmpLow potential of less than $400mV$ causes huge noise which points to unstable TIA working points. So for safety reasons spending $10mV$ more on AmpLow seems reasonable.

Without doubt, the good news about these results is that indeed a working point can be found which allows the DCDB to be operated at target speed. But in general, such a strong performance variation is not expected from simulations, neither for VDDA nor for

1. Since the TIA is an inverting amplifier, the ADC's output values show a falling edge in this case, which is in contrast to the results obtained from the dynamic ADC measurements.

2. Peaking time. Transition over the full height, ignoring the undershoot.

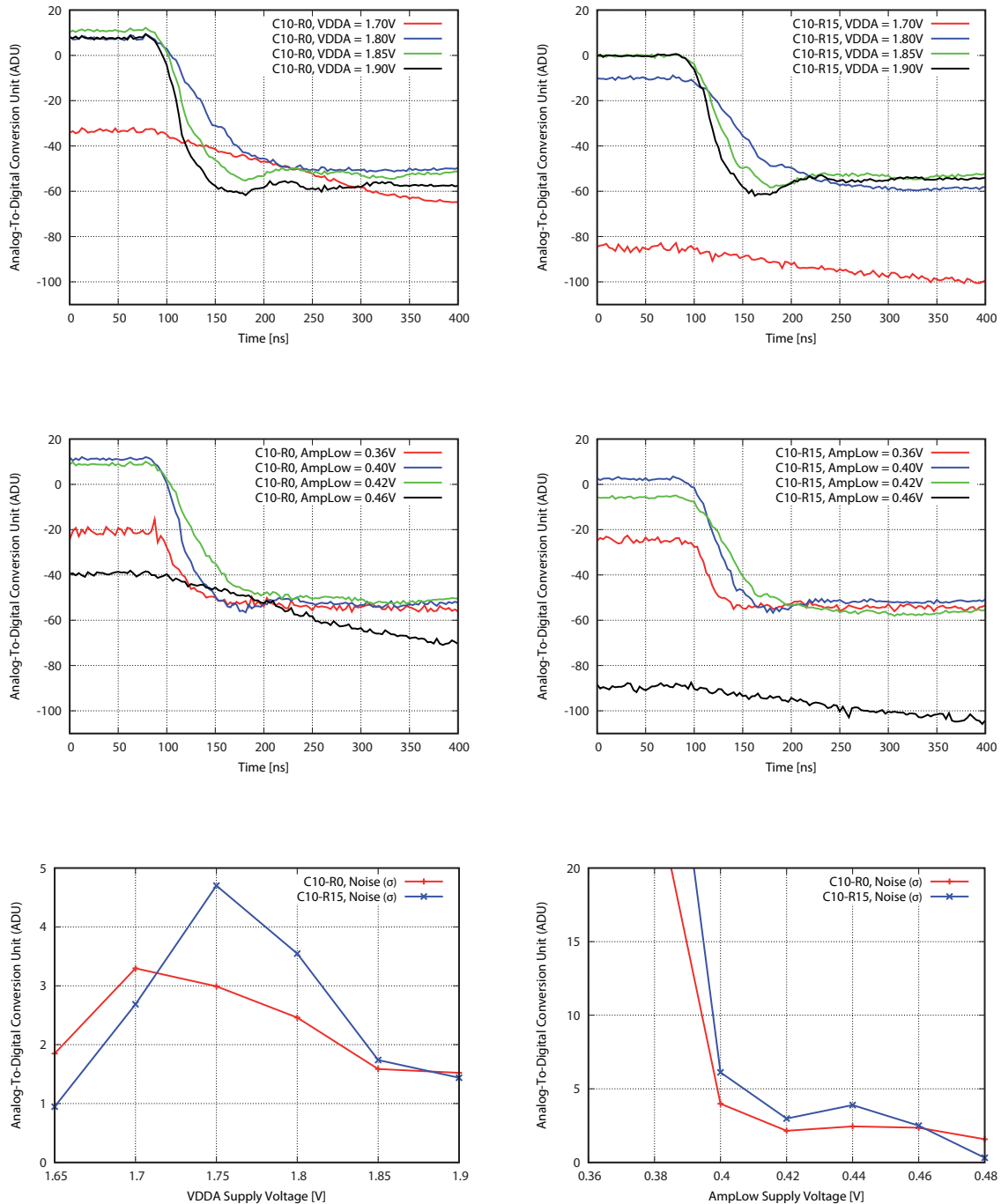


Figure 7-14 Dynamic TIA performance study for variations on the relevant supply potentials VDDA and AmpLow.

AmpLow. This is because the TIA operation relies on currents, which are generated by current sources that should be independent to a large extent from potential variations of the shown scale. So the most reasonable explanation is that the relevant bias circuits, which are placed at the top edge of the DCDB and thus the maximum distance away from the power pads, suffer from a significant on-chip voltage drop.

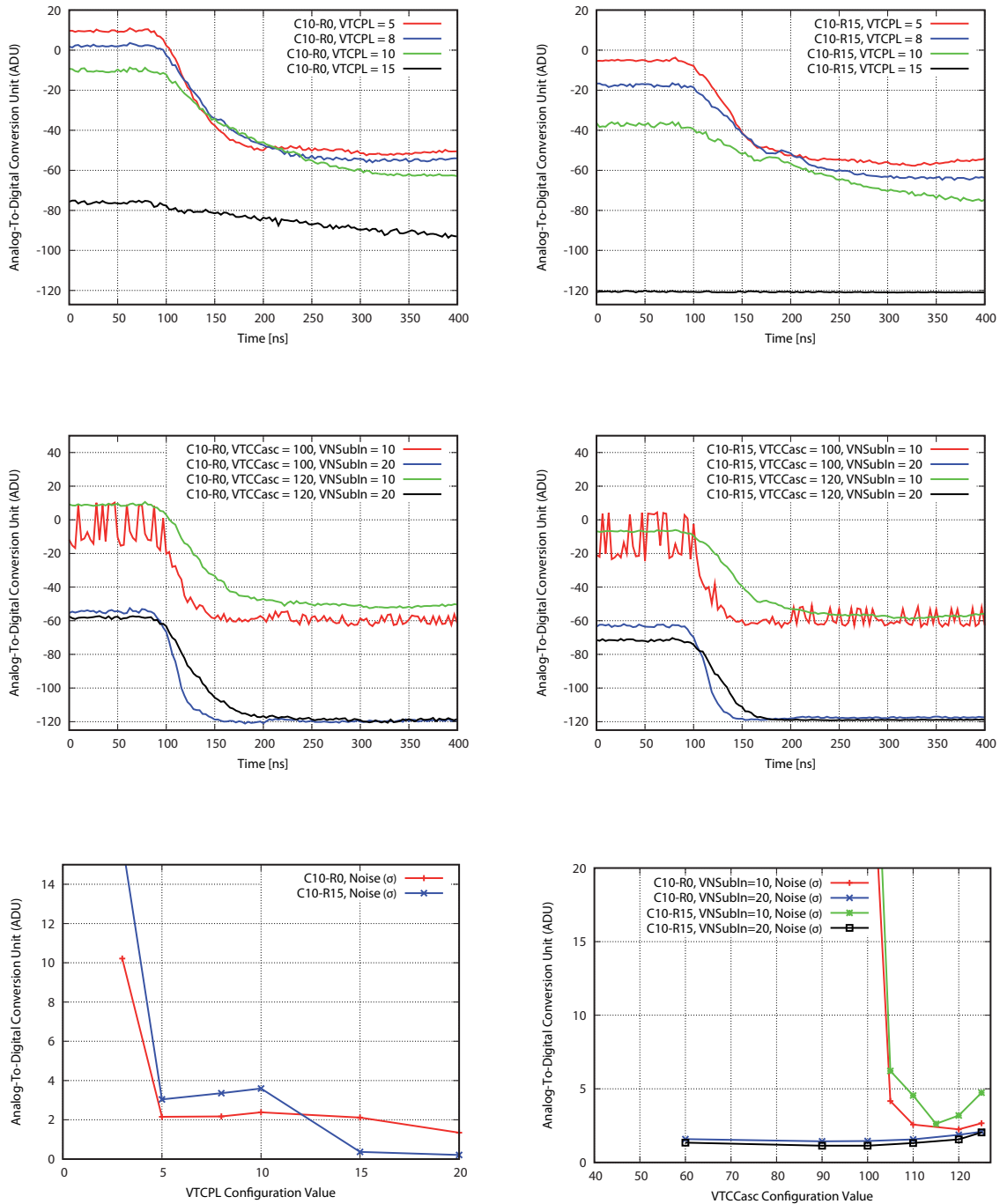


Figure 7-15 Dynamic TIA performance study for variations on the critical internal configurations VTCPL and VTCCasc.

Variations on Internal Configurations

The transimpedance amplifier's sensitivity on biasing can also be revealed directly as illustrated in figure 7-15. Slight changes in the configuration of the TIA's load current source TCPL has obviously a significant influence on gain and transition time. The optimal value for the VTCPL configuration is five. The corresponding noise plot,

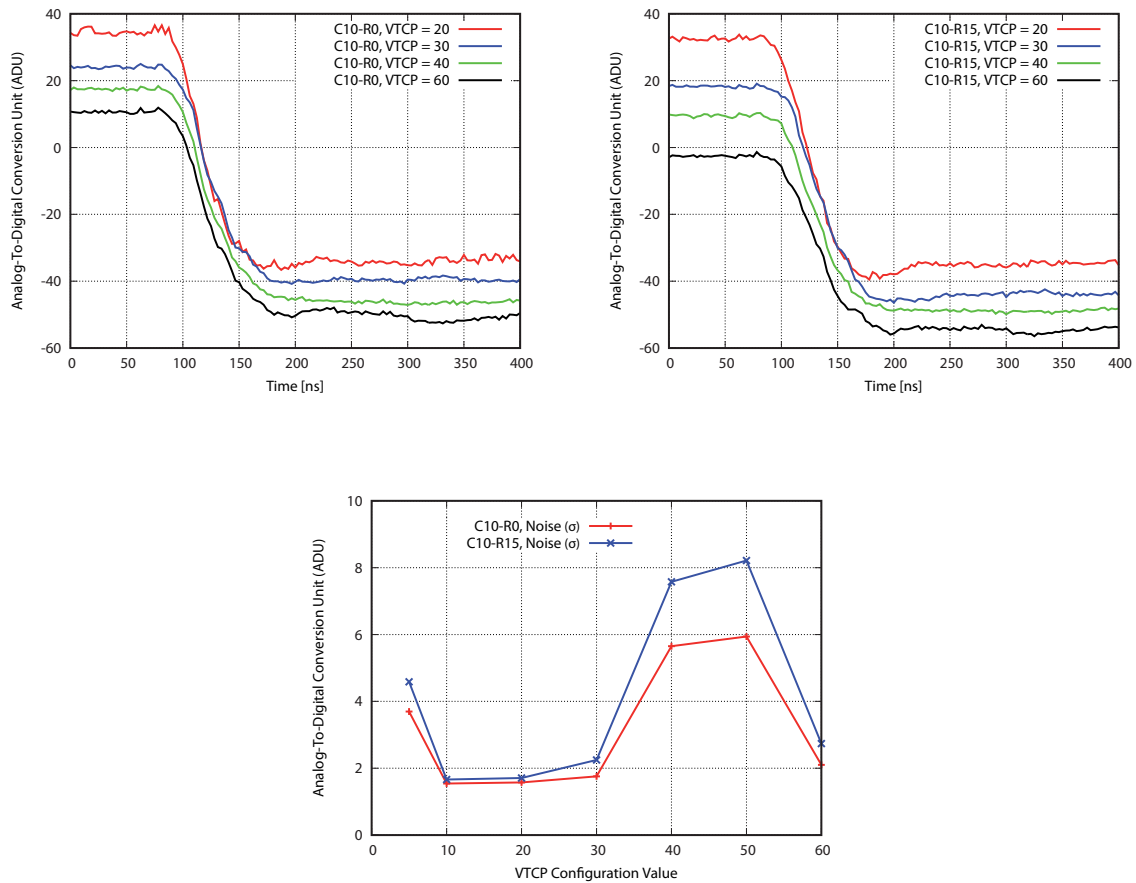


Figure 7-16 Dynamic TIA performance study for variations on the internal configuration VTCP.

however, shows that this is at the same time the minimal setting for which the system remains stable. That is why the VTCP configuration must be considered critical.

The bias potential for the cascode transistor of the TIA's input stage (TCCasc) turned out to be critical as well. For a stable regulation of the input node, the VTCCasc value must be set to at least 110, that is almost maximum, although a lower value resulting in a higher bias potential would be desirable for shorter transition times. In order to illustrate that behaviour, the NSubIn current source is used to manipulate the input signal in addition to the dynamic pulse, which leads to an offset variation. For a cascode bias configuration of 120, the regulation works fine, while changing the setting to 100 results in a significant instability for a certain range of input signals. This behaviour is observed for both examined TIAs and can also be reproduced in simulations of the circuit. Additional capacitors between input and output node of the TIA's input stage help to overcome this issue for subsequent versions of the design.

The effect of varying the current of the TCP source is shown in figure 7-16. Decreasing the VTCP configuration value and thus the current has advantageous but only moderate consequences. Comparing the results for the VTCP values of 60 and 20, the signal transition gets faster by roughly 30ns and even the noise improves. It has already been shown that a lower TCP current results in less dynamic input range, but concerning the speed there is no reason why not to choose the value of 20 to be the optimal setting for

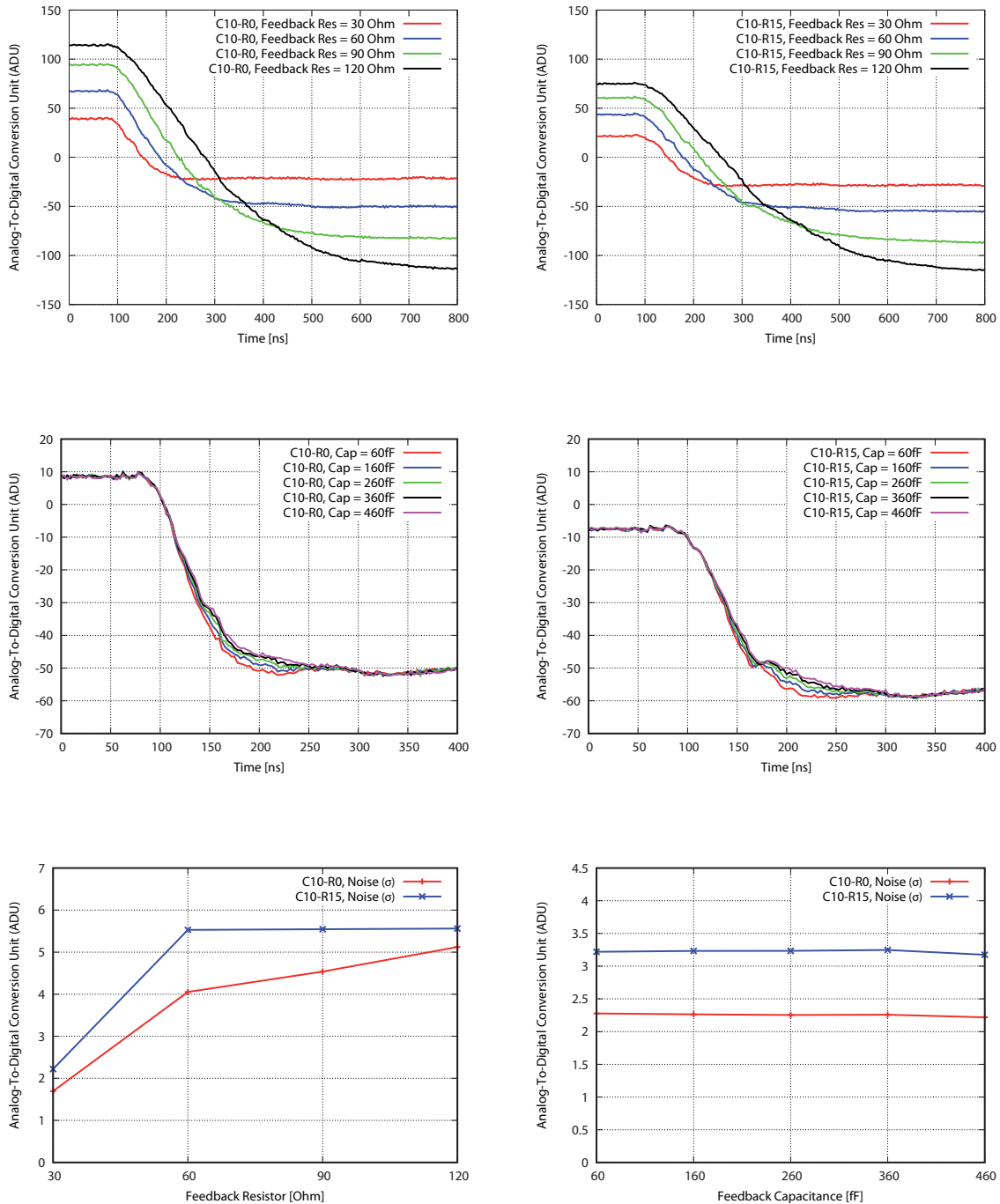


Figure 7-17 Dynamic TIA performance study for variations on the feedback path.

VTCP. In fact, there is another positive side effect in doing that, since the TCP current source is responsible for a substantial fraction of the DCDB's power consumption. Detailed information on how much power can be saved here is given in section 7.2.8.

Feedback Variation

The effects of feedback path variations are demonstrated in figure 7-17. Changing the resistor does not reveal any big surprises. With increasing resistance the signal transition

time gets slower and the gain scales linearly. This matches the expected behaviour. The noise, measured in ADU, is minimal for $30k\Omega$ and shows a plateau at about $5.5ADU$ for the other three settings. The conclusion of these results is the following. On the one hand, $30k\Omega$ is mandatory for full speed operation, since the signal transition time is too slow for higher resistances. On the other hand, for special measurements with relaxed requirements in terms of speed and dynamic range the DCDB can fairly be operated in a low noise mode. This is because for higher feedback resistance the noise remains constant while the gain rises. Therefore, the effective noise referred to the input signal current improves.

In contrast to the resistance, however, the effect of varying the feedback capacitor is a little surprising, as its scale is expected to be much larger. In fact, for the nominal setting of $30k\Omega$ there is only very little influence observable. At least the trend is as it should be, which means that the signal transition gets slower with increasing capacitance. Finally, the noise plot shows that the feedback capacitor has practically no effect on the noise of the system.

Config / Supply	Value
VTCP	20/60
VTCPL	5
VTCCasc	120
VTCSFN	120
Cap	2 switches closed
CapL	All four switches opened
AmpSFON	Closed
En30	Closed
En60, En90, En120	Opened
VAmpPBias	120
VFBPBias	120
VPSource	120
VPSource2	120
VPSourceCasc	40
VFBNCasc	0
VRefFB	60
VPDel	127
VNMOS	60
VDDA	1,9V (sensed)
AmpLow	410mV (sensed)
RefIn	1,16V (sensed)

Table 7-5 Optimal configuration values for the TIA and the ADC.

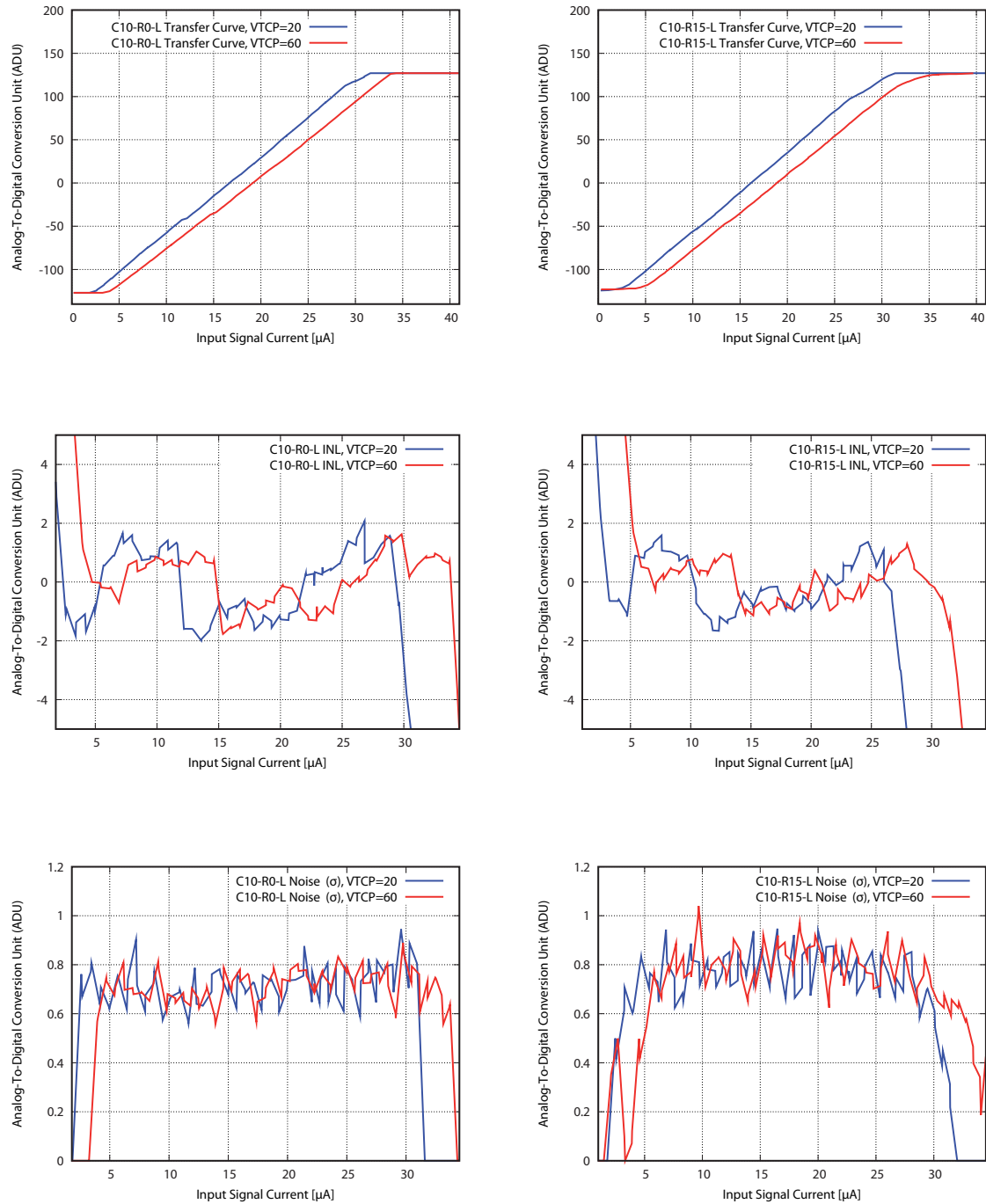


Figure 7-18 Combined performance of TIA and ADC with optimal configurations.

7.2.7 Overall Channel Performance

As a conclusion of all the previously shown analyses of the various building blocks of the analog-to-digital conversion channel, its overall performance is presented here. In the same semi-dynamic way as done before, an ADC transfer curve is measured and analysed in terms of noise and linearity. But this time the input signal current is injected

into the TIA instead of directly into the ADC. The configurations and supply voltages are set to the identified optimal values as summarised in table 7-5. The results are plotted in figure 7-18. For each input current 100 samples are taken.

The dynamic TIA analysis showed that based on these results only the decision whether to choose 60 or 20 to be the optimal setting for VTCP is not fully clear. So both options are still considered here.

At first sight, the dynamic input ranges do neither fit to the statically measured input range of the TIA, nor to that of the ADC, as it is simply too large. Concerning the TIA's input range, the explanation for the increased dynamic range is that other configuration changes, such as VTCCasc for example, have noticeable influence on this property. The second effect that plays a major role here is that the various switches of the circuit, most of all the SmpL/SmpR switches, have a significant resistance even in closed state. This adds up to the output series resistance of the TIA and thus results in an amplification factor of less than one when using the $30k\Omega$ feedback resistor. In this case the transimpedance amplifier acts as an attenuation.

Further analyses of the transfer curves in terms of noise and linearity reveals another two major observations. The first one is that, except from a clearly visible bend of the transfer curve close to the upper end of the dynamic range for the VTCP setting of 20, the two options do not differ significantly. In particular, the noise is the same. This is really a great result since this offers the opportunity to save power for the cost of a little less linearity. The second observation is that both, noise and linearity performance, have improved significantly compared to the results of the ADC-only measurements presented in section 7.2.2. The conclusion of this must be that the dominating source of noise and non-linearity for the former measurements was not the ADC itself. The reduction of noise can be explained by the low-pass filter effect of the TIA, which was not present for the ADC-only measurements. The linearity improvement is related to the TIA's regulation quality, since the regulation of the ADC's input node potential is much more robust if the TIA is enabled.

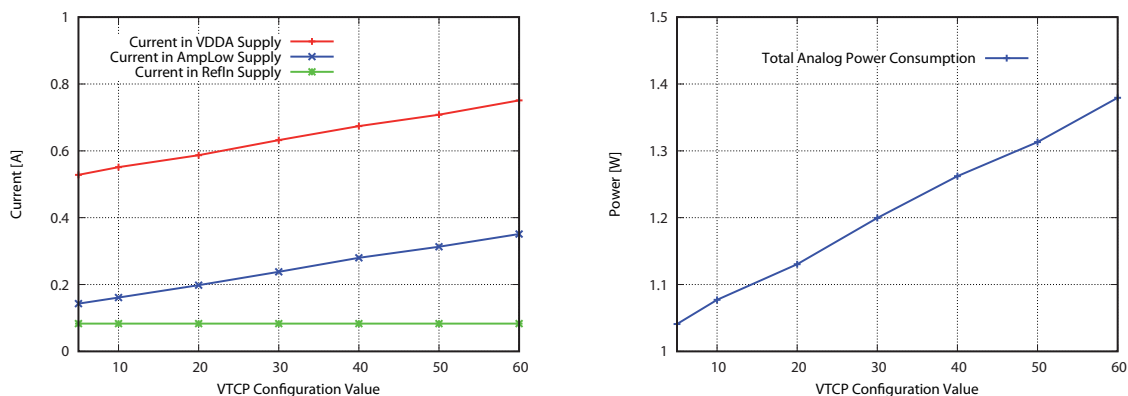


Figure 7-19 Currents at the various analog supplies and the resulting power consumption of the DCDB's analog domain as a function of the VTCP configuration.

7.2.8 Power Consumption of the Analog Channels

The power consumption of the DCDB's analog domain at optimal settings and as a function of the VTCP configuration is illustrated in figure 7-19. It shows also the currents at the three analog power supplies VDDA, AmpLow and RefIn. The RefIn supply is not directly used in the TIA, so the current there is constant at $83mA$ for variations on VTCP. In contrast, the currents in AmpLow and VDDA are, as expected, linearly rising with an increase of the TCP current. The total power consumption of the system's analog part is $1.13W$ for VTCP set to 20 and $1.38W$ for 60. In accordance to the powering scheme of the DCDB as illustrated in figure 6-2 on page 98, the power consumption is calculated by the following equation:

$$P = (I_{VDDA} - I_{AmpLow}) \cdot U_{VDDA} + I_{AmpLow} \cdot (U_{VDDA} - U_{AmpLow}) + I_{RefIn} \cdot U_{RefIn}$$

7.3 Multi-Channel Measurements

In the previous section, the presented measurements focused on two specific conversion channels and the ADC circuits within them. But even if the performances of these channels were totally perfect, the chip would still be useless in case they were the only

Golden Module: All ADC Transfer Curves

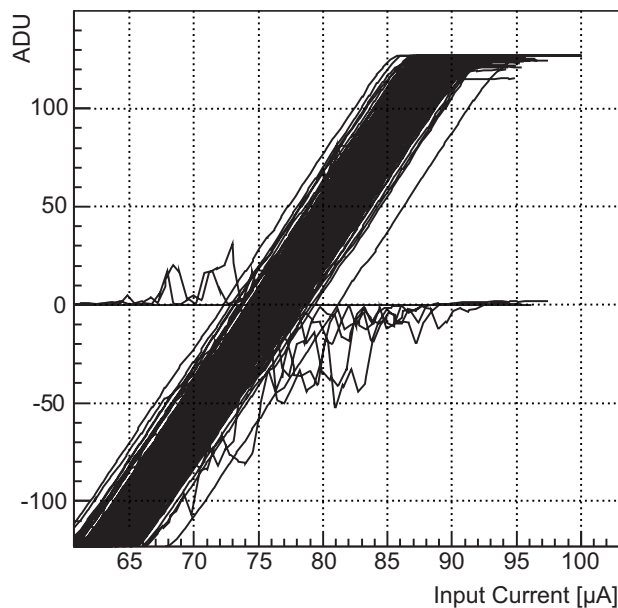


Figure 7-20 Transfer curves of all accessible analog-to-digital converters of the *Golden Module*.

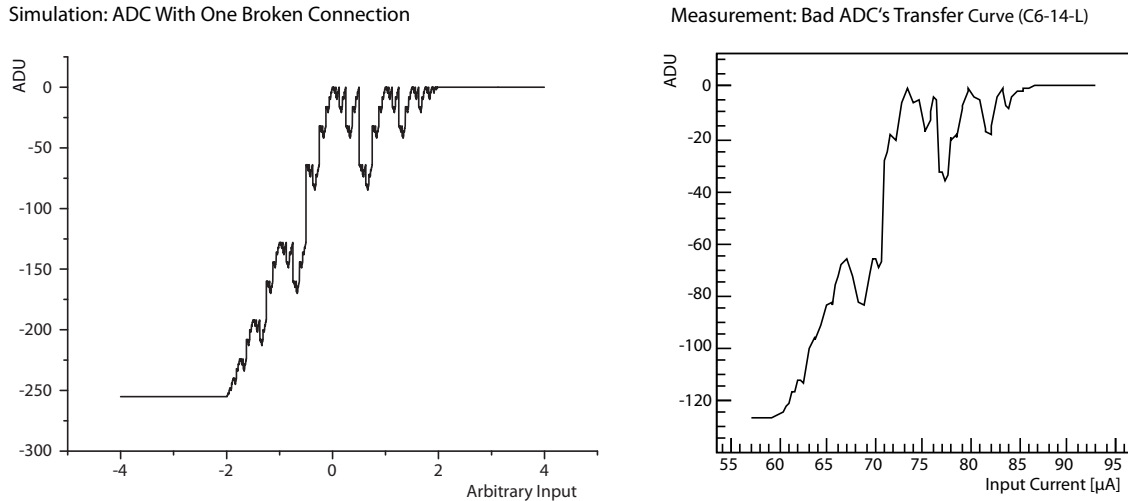


Figure 7-21 Bad ADC: comparison of simulation (left) and measurement (right). The simulation shows a nine-bit algorithmic ADC with one of the two connections to the digital processing logic being broken [70].

working ones of the entire design. It is therefore right as important to make sure that the other channels are functional as well and show similar performance. To this end, this section provides results that compare the measurements of the various channels.

Figure 7-20 shows a set of transfer curves obtained by semi-dynamically measuring all the 256 accessible ADCs of the Golden Module using the optimal settings as listed in table 7-5 (with VTCP set to 60) and the following scenario. In order to make it as realistic as possible the DEPFET matrix on the hybrid board is used as an offset current source. That means a single row of the matrix is constantly activated resulting in offset currents that flow through the pixels into the DCDB's analog inputs. Inside the channels this offset is subtracted by the NSubIn current source, which is set to subtract the maximum possible current. The bias voltages of the matrix are adjusted to make the produced offset current on average meet the maximum subtraction capability of the NSubIn sources. On top of that, the PInjSig sources are used to produce configurable signal current in order to generate the transfer curves. 100 samples are taken for each input signal and the PInjSig source is calibrated for each channel individually.

7.3.1 Bad ADCs

The first thing one recognizes when looking at the set of transfer curves is that obviously there are several ADCs not working properly. This means some of the curves are not going straight from the lower to the upper end of the dynamic range, but are rather converting to the ADU value '0'. In fact, the total number of obviously misbehaving ADCs on the accessible half of the Golden Module's DCDB is eight, which corresponds to a fraction of ~3%.

Analysing this problem leads to the following facts. First of all, it is important to notice that the failure must be related to a single ADC and its individual data processing. This is because for all broken ADCs there is a neighbouring ADC within the same channel that works fine. Secondly, for some of the broken ADCs it can be proven by means of the

digital test signal injection that the digital data processing is functional as well¹. Promising candidates for being the cause of these malfunctions are the metal wires connecting the ADCs to the digital data processing logic. Even for those ADCs that are located comparatively close to the digital domain, their interconnection lines are mostly several hundred or even a few thousands of micrometers long. Metal lines of that length are potentially critical elements during the production of a chip because of the so-called *antenna effect*. If inadequately protected transistor gates are connected to such long wires, charge that accumulates on the wires during production can destroy them.

In order to verify this theory, figure 7-21 shows a comparison of the transfer curve of one of the Golden Module's bad ADCs and a simulated curve where one of the two interconnection lines between the ADC and its digital processing logic is missing. In both cases the curve starts at the lower end² of the codomain and converts to '0'. The shapes of the two curves match perfectly!

This type of failure fits to at least five others of the remaining misbehaving ADCs of this particular DCDB chip. Consequently, it is essential for future revisions of the chip to put enhanced effort on the protection of long wires, such as the interconnection between ADCs and digital domain, against yield issues.

7.3.2 Offset Analysis

Fortunately, the large majority of conversion channels can be considered reasonably functional. Nevertheless, a sizeable offset variation among the transfer curves shown in figure 7-20 can be observed. The maximum distance is measured to be $86.8ADU$. This offset, however, is not only caused by the DCDB's analog channels themselves. It is dominated by the overlaying variation in offset currents of the various DEPFET transistors of the connected detector matrix. In that spirit, this measurement gives a first idea how necessary an additional dynamic offset compensation really is.

A second observation concerns the strength of the NSubIn current sources. Here, the average constant offset subtraction capability of these sources is about $60\mu A$. In order to operate the DEPFET transistors at their target working point, however, an offset current of rather $100\mu A$ has to be faced. That means, with the present DCDB it might be possible that the connected DEPFET transistors cannot be operated with full gain. This comparatively simple issue has to be solved in subsequent revisions of the design.

7.3.3 Gain Analysis

More detailed analyses of the Golden Module's well operating ADCs are provided in figure 7-22. First of all, the two graphs on the top focus the gain as measured from the transfer curves. In fact, the gain is the reciprocal of the slope of the transfer curve and thus expressed in nA/ADU . It is given as a function of an arbitrary index of all accessible ADCs as well as sorted by the respective position on the chip.

-
1. The reason why not all digital processing units of broken ADCs can be verified is simply related to the generated test pattern, which is not sufficiently meaningful for some channels to make a statement.
 2. Without loss of generality, the output of the high-threshold comparator is chosen to be broken here. If the low-threshold comparator's output was broken instead, the transfer curve would by failure start at zero and head towards the positive end of the codomain.

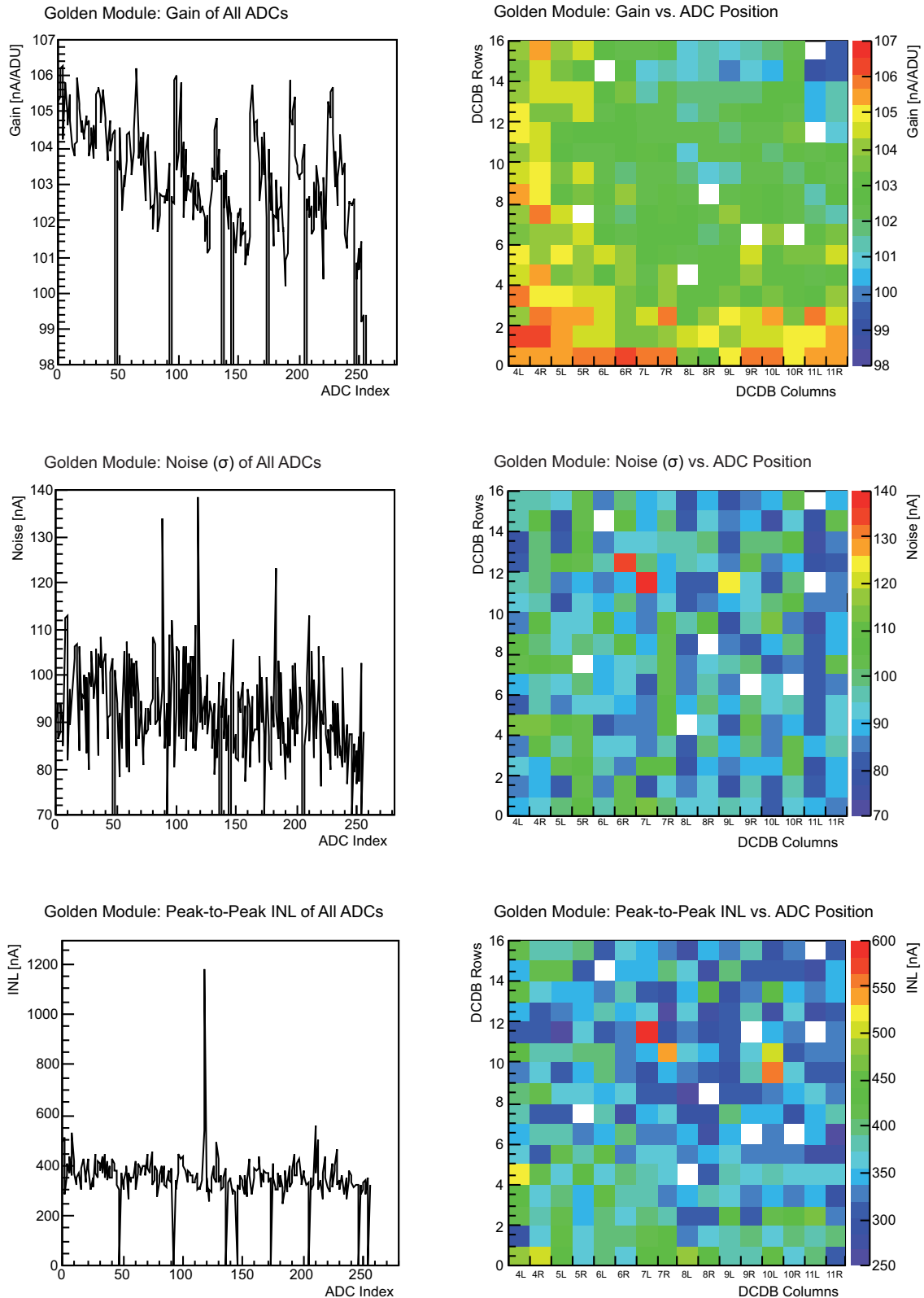


Figure 7-22 Detailed analyses of the measured transfer curves. Gain, noise and linearity are given as a function of an arbitrary ADC index as well as sorted by the absolute position on the chip. The bad ADCs are masked by setting their values to zero.

The gain is roughly in the range of $98 - 107 nA/ADU$, with the RMS (Root Mean Square) value at $103.3 nA/ADU$. The map reveals clearly the existence of a gradient within the measured gain values over the entire chip. The ADCs showing the highest gain are located in the lower left corner of the analog domain, while the lowest gain is obtained from those in the upper right corner. Although the DCDB is a quite large chip design and thus the effects of process variations during the production might be visible, voltage drop on the various supply rails is believed to be the dominating reason. The power pads of VDDA and AmpLow are arranged in a row, located just underneath the channel row #0. So the vertical fraction of the gradient can be caused by these supplies. An even more suitable candidate is the RefIn supply. Since the current is significantly lower there, only two pads are used for this. Indeed, these two pads feed the corresponding distribution network of the channel matrix from the lower left corner. Hence, the observed gradient is believed to be an interference of voltage drops on all three analog power supply rails.

7.3.4 Noise Analysis

The next analysis step focuses on the measured noise. It is again given as a function of an arbitrary index and sorted by the position on the chip. The amplitudes are translated to an equivalent input current by means of the corresponding gain factors. One finds that the noise of the majority of the ADCs is distributed in the range of $70 nA - 120 nA$ with the RMS value at $93.7 nA$. Three ADCs exceed this range. The worst noise is shown by the ADC at position C7-R11-L.

Again, an interesting feature is revealed by examining the results versus the absolute positions on the chip. This time, it is not a gradient but a stripe pattern. Although the effect is only very small, it seems that the left ADCs within a channel tend to show a lower noise than the right ones. A possible explanation for that observation is rooted in the layout of the ADCs. Left and right ADCs are not just identical instances (copies) of a master design. In fact, they are mirror images to each other. It is therefore believed that this structure is caused by mismatch effects. Thus, it is a production artefact.

7.3.5 Integral Non-Linearity Analysis

The last one of the detailed analyses is concerning the linearity. Obviously, the results are prepared in the same way as done before for the gain and the noise. The RMS value of all measured non-linearities (peak-to-peak) is $368 nA$, which is almost four times the noise.

Again, there is one ADC showing a noticeably bad linearity. In fact, it is once more the ADC at position C7-R11-L that has already been identified to have a bad noise performance as well. So it is quite interesting to see what the reason is. To this end, figure 7-23 presents an isolated view to the corresponding transfer curve and its linearity analysis. Obviously, the transfer curve is not a proper straight but has several steps in it that indicate missing and wide output codes. These steps cause deviations from the straight line fit and thus result in large peaks in the non-linearity plot. The fact that only very few ADCs within this particular DCDB chip show such a behaviour rather points to another yield issue than a badly adjusted general operation point to be the root cause of this problem.

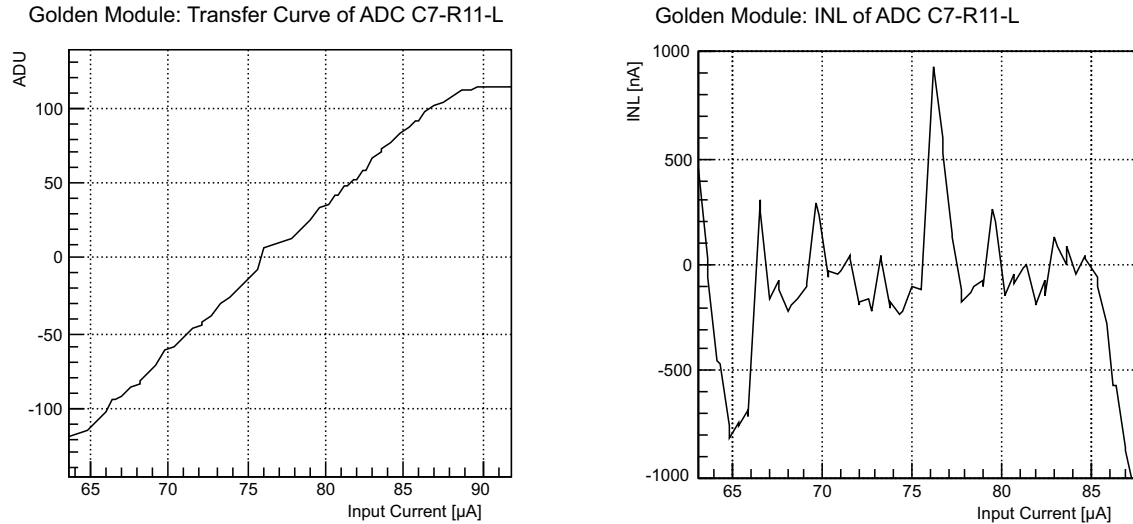


Figure 7-23 Transfer curve and linearity analysis of the Golden Module's ADC C7-R11-L.

7.4 Conclusions

The results and analyses presented in this chapter can be concluded as follows. On the one hand, the digital logic of the DCDB is proven to be functional. The measured performance parameters meet the expectations, particularly in terms the operation speed and power consumption. The interaction with the ADC channels works well. That means the steering of the ADC operation and the sampling of the generated values is properly synchronized.

On the other hand, the ADC performance in terms of noise and linearity in comparison to that obtained with the DCD2 is reasonable. This is fairly good news, since many implementation details changed from DCD2 to DCDB. Not least it is the number of channels that increased enormously and is now at the required level. Nevertheless, improvements were expected, especially for the noise performance, which is still off target by a factor of two [9]. Moreover, the detailed measurements revealed that the optimal operation point is very tight. For some settings, like the RefIn potential for example, there is only very little margin. The transimpedance amplifier is a new development for the DCDB, thus there is now direct comparison to the DCD2. It does meet the speed requirements, but again with very little margin. Beyond that, it even shows stability issues. These aspects need to be improved in subsequent versions of the design. To this end, a smaller test chip with a reduced number of channels (so-called *DCDB-TC*) as well as another full size chip (so-called *DCDBv2*) have already been submitted. The following changes are realized there:

- The switch transistors inside the ADC as well as the two sample switches *SmpL* and *SmpR* are implemented using low-threshold-voltage transistors in order to improve both performance measures, the noise and the linearity.

- The bulks of these switch transistors used to be biased with RefIn. Now they are biased independently from RefIn, which should improve crosstalk immunity.
- The capacitive feedback path of the transimpedance amplifier is adjusted to improve its stability.
- The bias generator for the switch steering delay element is designed for higher output voltages and thus shorter delay. For quite some time during first chip tests, this delay was considered the best candidate to be blamed for a significant reduction of the maximum operation speed [40].
- The AmpLow potential can optionally be generated by an on-chip regulator circuit.
- The NSubIn current sources for static pedestal current subtraction are doubled in order to allow for higher pedestal currents.
- Since voltage drops on the chip are discovered indirectly in many results, the supply potentials at various points on the chip are measurable directly now.
- At those points in the design where it is possible, vias are doubled for yield improvement.
- Antenna diodes are attached to long wires in order to avoid ADC failures like those shown in section 7.3.1.
- The density on the top metal layer is reduced to meet the design rules.
- There is even a new functionality implemented in the DCDBv2: an analog correction for common mode noise is optionally available [69].

Indeed, first preliminary and not optimised measurements with the DCDB-TC allow for optimism [70].

The Detector Prototype Operation

Abstract:

The present chapter focuses on the DCDB's operation within the DEPFET detector prototype system. First, a simulation setup is proposed that is meant to serve as environment for simulating the entire front-end readout chain. Afterwards several measurements with the real system are presented. The highlights are the measurement of a cadmium-109 spectrum and first experiences in a beam test experiment at CERN. Beyond that, the effectiveness of the DCDB's dynamic pedestal compensation mechanism is demonstrated.

According to the results presented in the previous chapter, the DCDB is proven to be able to serve as the front-end readout ASIC for the BELLE-II PXD detector system. It is therefore very natural to proceed to a detector prototype system in order to determine the combined performance of DCDB and DEPFET detector. This development milestone is taken in two steps. Firstly, the devices are integrated into a common simulation environment. In the second phase the real chips are put together. The setup is already presented in chapter 6. Here, some results are illustrated that have been obtained with it.

8.1 System Simulation

8.1.1 Motivation

The PXD detector for BELLE-II is in every respect a highly complex system. This is not least true for the front-end readout chain, as it consists of many non-standard components, each of them individually developed for this application. A large variety of technologies in microelectronics and interconnections are used. The development effort is distributed over several institutes of the collaboration. Of course, the interfaces between the respective parts are well defined. Nevertheless, having a common simulation platform where the various elements can be verified together certainly helps to prevent from unnecessary development iterations and thus saves time and money.

The presented simulation environment aims to be the common front-end readout chain simulation platform for the DEPFET collaboration. It is able to combine designs of different technologies as well as both, digital and analog world, to one entity. It is meant to be a tool for the various groups of developers to put their designs in and check the proper interaction with other elements.

8.1.2 Simulation Setup

The simulation is set up using the *Cadence Incisive Unified Simulator 9.2* software in analog mixed-signal mode¹. Since this tool is actually a digital-only simulation environment, the *Spectre* analog simulation engine is required in addition. Digital blocks are added to the environment by means of their textual descriptions. This can be both, a standard cell netlist or a high level code written in Verilog or VHDL. Analog elements must be represented by a netlist using the Spectre syntax. The analog models of transistors, resistors, capacitors etc. for all used technologies must be available as well.

The simulation environment in its current expansion stage is illustrated in figure 8-1. Besides a stimuli generator, it comprises mainly models of the SwitcherB, the DCDB and the DEPFET detector. Even though there is no general restriction in the number of detector pixels and according steering and readout channels in the simulation, the detector matrix consists of only three DEPFET transistors (pixels), organized as three rows and one column, in order to keep the simulation effort reasonable. The matrix is

1. Choosing this software package results in a digital-on-top simulation flow. Using the *Virtuoso Analog Design Environment* instead is perfectly possible as well. In this case the simulation setup would be different, namely according to the analog-on-top flow.

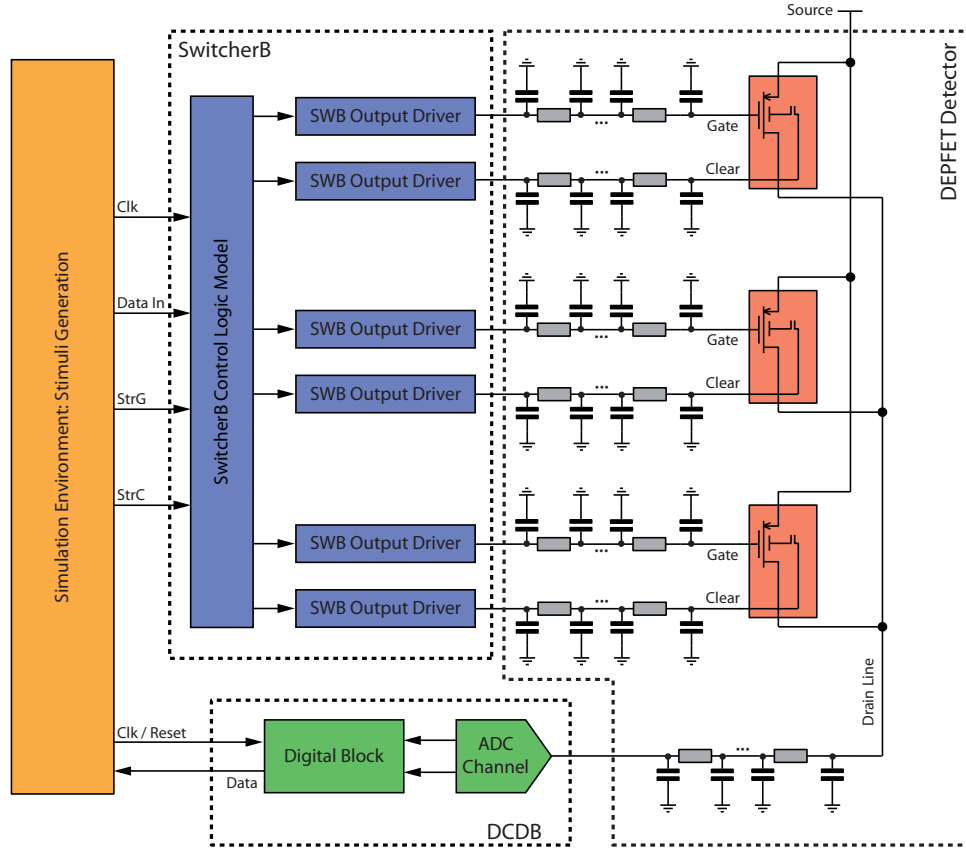


Figure 8-1 Schematic of the DEPFET detector prototype system simulation.

steered by a SwitcherB instance, which is equipped with the real analog output driver circuits and a simplified model of the digital controlling logic. The number of simulated output channels is adopted to the three row detector matrix. The DCDB model is entirely identical to the circuitry that is produced as DCDBv1. The number of input channels is reduced to one, however, fitting the single column of the detector model. The electrical properties of the detector's gate, clear and drain lines are approximated by means of cascaded RC elements. Gate and clear lines are both attached with 40Ω sum resistance and $50pF$ sum capacitance each. For the drain line there are 300Ω sum resistance and $50pF$ sum capacitance. So relative to the readout devices, these parameters correspond to pixels residing in the farthest corner of the half-ladder sized detector.

A special feature of this simulation environment is the very accurately modelled DEPFET transistor. Since DEPFET is a non-standard and non-commercial technology, the simulation model is developed by the technologists at the MPI Semiconductor Laboratory (Munich) [71]. Beside the bare transistor characteristics, it includes all relevant parasitic parameters and the modulation of signal electrons residing in the internal gate to the channel current. Moreover, the clearing process is implemented as well. The model is created using the *Compiled Model Interface* of the Spectre simulator (SpectreCMI) [72]. Currently, the model corresponds to a very early design iteration of DEPFET transistors. An updated version representing the latest design of PXD6 is currently being developed [73].

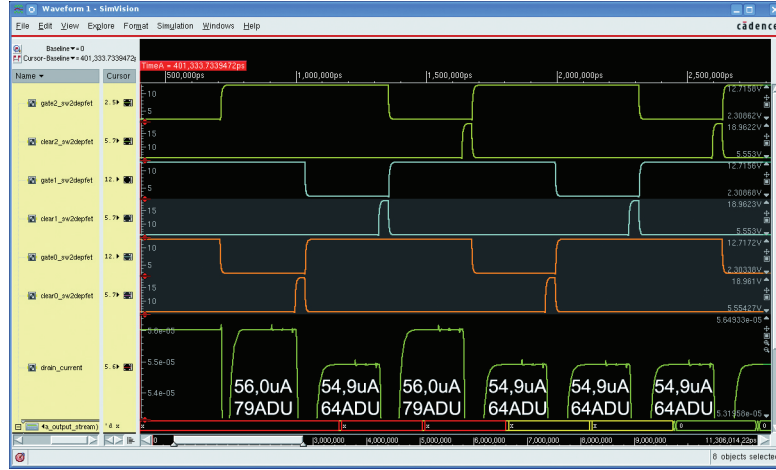


Figure 8-2 Simulation screenshot showing the gate and clear signals of the three detector rows as well as the drain current. The current value together with the digitization result is indicated for convenience.

8.1.3 Simulation Result

An exemplary simulation result obtained with this environment is provided in figure 8-2. It shows the three pairs of gate and clear lines that steer the detector in the rolling shutter mode using single sampling. Initially, the pixels in the rows zero and two are loaded with 4000 signal electrons, while the internal gate of the pixel in row one is left empty. Reading the pixels shows a current of $56\mu A$ for the loaded pixels, which is digitized by the DCDB to 79ADU. The unloaded pixel in row one gives $54.9\mu A$ and 64ADU respectively. According to the readout principle, the pixels are cleared after reading. Thus a second run through the detector gives the unloaded current of $I_{Sig} = 54.9\mu A = 64ADU$ for all pixels, as expected.

8.1.4 Concluding Remarks

The presented simulation result is an impressive example that shows the possibility to combine designs of entirely different technologies into one environment. So from the tooling point of view it can fairly be regarded as a prove of principle for that kind of system level simulation. Nevertheless, it is obvious that the model accuracy, in particular concerning those of the DEPFET transistor and the detector parasitics, must still be enhanced. This becomes clear in the poor agreement with the detailed detector readout performance measurements presented in [58].

Although there is no doubt about the general worthiness of this simulation environment for the PXD project, there is also some inconvenience about it. First, setting up the simulation environment is not straightforward. It requires advanced knowledge in handling the simulation software. Second, in order to run the simulation at all, access to all kinds of used microelectronic technologies in terms of simulation models and latest versions of the simulation software is mandatory. This fact might preclude the simulation

environment from being spread over the collaboration due to license issues. Furthermore, a drawback of the current SpectreCMI-based DEPFET transistor model implementation is that it does not support multi-threaded analog simulators, such as *Spectre APS*. Since the simulation environment is intended to grow as the PXD project development proceeds, using multi-threaded simulators will become indispensable. A solution to that issue is either to further extend the existing model or to rebuild it using the analog hardware description language Verilog-A.

8.2 Reducing Pedestal Fluctuations

As explained in section 3.4.6, the DCDB offers the feature to reduce the relative pedestal current fluctuations among the various DEPFET transistors of the detector by dynamically adding compensation currents to the input signals. The quality of this feature is analysed here, first by calculating the theoretical benefit and afterwards by presenting appropriate measurements.

8.2.1 Theoretical Benefit

The functional principle of using the DAC circuit at the input of each analog channel to reduce the relative pedestal current fluctuations is illustrated in figure 8-3. It is assumed that the pedestal fluctuations are normally distributed. Metaphorically speaking, the basic idea is to squeeze the distribution towards the right side (high positive values), by adding current to those signals that show a comparatively low pedestal current. Since the DAC circuit is designed with a two bit resolution, the distribution is virtually segmented into four bins with the bin width being defined by the unit current setting (PDAC). Those pixels having a pedestal current fitting the leftmost bin are boosted by three time the unit current. The higher the pedestal current, the less current is added. Pixels with a pedestal current fitting the rightmost bin are not boosted at all. Mathematically, this process can

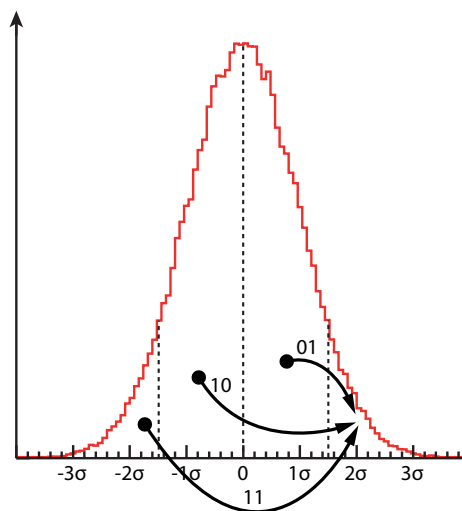


Figure 8-3 Strategy of reducing the relative pedestal current fluctuations using the DAC circuits.

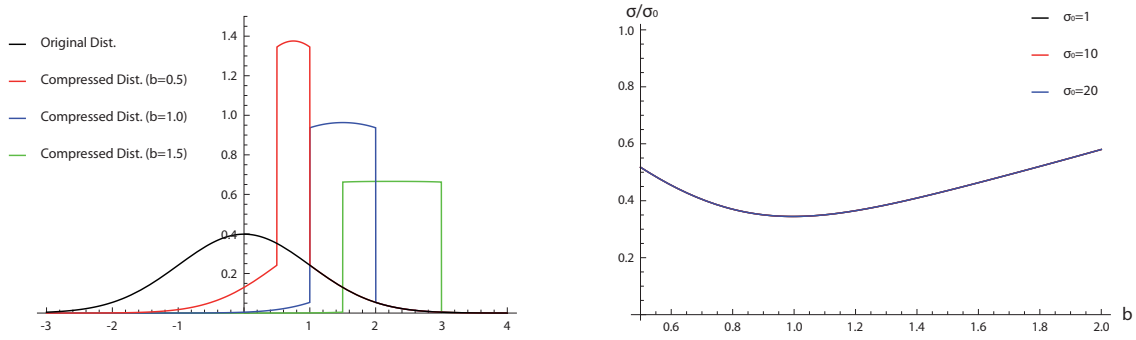


Figure 8-4 Analysis of the pedestal current distribution. The Graph on the left hand side illustrates the resulting distributions for various values of b . The one on the right hand side shows the standard deviation of the compressed distribution as a function of b for various values of σ_0 .

be expressed in the following way. Assuming the original distribution to be described by the gaussian function $g(x)$ with mean value μ_0 and standard deviation σ_0 :

$$g(x) = \frac{1}{\sigma_0 \cdot \sqrt{2\pi}} \cdot e^{-\frac{1}{2} \cdot \left(\frac{x - \mu_0}{\sigma_0}\right)^2}$$

Then the resulting compressed distribution as a function of the bin width $b \cdot \sigma_0$ is:

$$c(x, b) = \begin{cases} g(x - 3b\sigma_0) & x < b\sigma_0 \\ g(x) + g(x - b\sigma_0) + g(x - 2b\sigma_0) + g(x - 3b\sigma_0) & b\sigma_0 \leq x < 2b\sigma_0 \\ g(x) & 2b\sigma_0 < x \end{cases}$$

Hence, the standard deviation $\sigma(b)$ of the compressed distribution is:

$$\sigma(b) = \sqrt{\int_{-\infty}^{\infty} (x - \mu(b))^2 \cdot c(x, b) dx} \quad \text{with} \quad \mu(b) = \int_{-\infty}^{\infty} x \cdot c(x, b) dx$$

$\sigma(b)$ as well as exemplary shapes of the resulting compressed distributions are illustrated in figure 8-4. The function shows an optimum of $\sigma/\sigma_0 \approx 0.34$ for $b \approx 1$. In other words, the best achievable reduction of the pedestal current fluctuation in terms of the standard deviation is 34%. The influence of σ_0 on the reduction factor is negligible.

8.2.2 Optimization Algorithm

In the following, an algorithm is proposed that allows to determine the unit current setting (PDAC) as well as the two-bit steering parameters for every pixel of the attached detector matrix¹.

In the first step, the algorithm defines the mean value $\mu(b)$ (in ADU) of the target compressed distribution for a given bin width. This is considered the optimal value to be measured for every pixel of the detector after applying the correction. According to the theoretical considerations, this value should be the centre of the rightmost bin. Thus, in order to define the bins, the raw² pedestal current distribution must be measured. However, it may fairly be the case that this distribution does not fit entirely into the overall dynamic range of the DCDB. In this situation, the mean and the standard deviation of the distribution can only be obtained by calculating a gaussian fit³, as long as a sufficient fraction of the distribution can be measured.

In the second phase, the PDAC configuration value and the steering parameters for the pixels are chosen such that the standard deviation of the resulting distribution around $\mu(b)$ is minimal. In principle, this can be done in two ways. The first one is to calibrate all ADCs as well as all PDAC sources and calculate the optimal values based on these measurements. The second possibility is to simply do a scan over all PDAC settings and all multiplication factors for every pixel. Although the first option is certainly the more elegant one, it is advantageous only if the captured calibration can be used more than once. During chip testing and system development, however, this is often not the case. Thus, the more flexible scan approach is implemented.

The scan is done in the following way. For a fixed PDAC setting, entire matrix frames are captured for all four possible dynamic compensation values. That means, afterwards there is a resulting pedestal value (in ADU) available for every pixel of the detector and every of the four dynamic compensation values. Based on that data, the best of the four dynamic compensation values is picked for every matrix pixel. The measure for this selection is the distance of the captured resulting pedestal value to the target mean $\mu(b)$, which has to be minimized. The hereby calculated set of compensation values is considered the *optimal compensation matrix* for this particular PDAC setting. Then, the PDAC setting is rated by calculating the standard deviation of the virtual pedestal distribution, which would be obtained if the optimal compensation matrix was applied. This procedure is carried out for every value of PDAC. Finally, the optimal PDAC value together with its optimal compensation matrix is selected.

-
1. A general and obvious boundary condition for allowing any optimization at all is that no detector pixel produces a current that exceeds the dynamic range of the associated ADC channel after static offset subtraction but without dynamic compensation. This has to be ensured by the measurement setup, i.e. the bias potentials of the detector.
 2. „Raw“ in this context refers to the dynamically uncorrected pedestal current. However, the static current subtraction may be applied.
 3. Currently, the gaussian fit is not done automatically, but the ROOT environment of the DAQ Monitor software provides the required functionality to initiate the fit manually. The resulting parameter can then be provided to the algorithm.

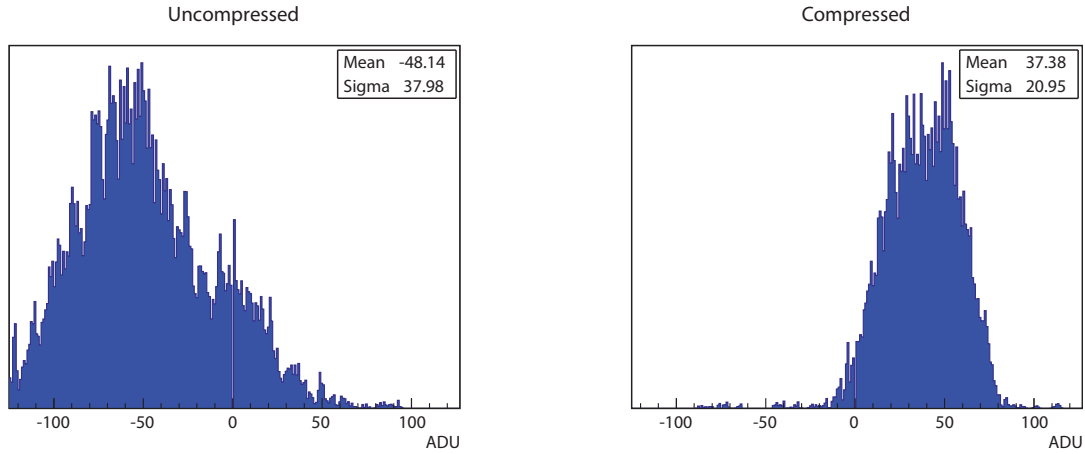


Figure 8-5 Illustration of the DCDB's pedestal current compression mechanism. Uncompressed distribution on the left hand side, compressed one at the right hand side. Note: the number of entries in the histograms is considered arbitrary.

8.2.3 Measurement Results

The algorithm presented above is then used to compensate the pedestal current dispersion as it appears in the prototype system. Figure 8-5 illustrates its effect. On the left hand side it shows the uncompressed distribution with a mean value of $-48.14ADU$ and a standard deviation of $37.98ADU$. The distribution on the right hand side shows the compressed pedestal distribution with the compensation being calculated for $b = 1.5$. The mean value of the compressed distribution fits perfectly to the expectation:

$$\mu_0 + 1.5 \cdot b \cdot \sigma_0 = 37.315$$

This result does not only verify the correct functionality of the algorithm, but also proves the proper operation of the DCDB's dynamic pedestal compensation circuitry. The second observation concerning this result is that the compressed distribution's standard deviation is not reduced by the expected 44% but only by 55%. This, however, can be explained by the fact, that the uncompressed distribution is not exactly gaussian.

8.3 Detector Operations

8.3.1 First Imaging Measurement

The first measurement that impressively demonstrates the functionality of the DCDB-based DEPFET prototype system is presented in figure 8-6. It shows the spot of a laser pointer hitting the edge of the detector.

From the system aspects of the prototype device, this measurement serves as first order verification of the software-applied mapping schemes concerning the rows and the

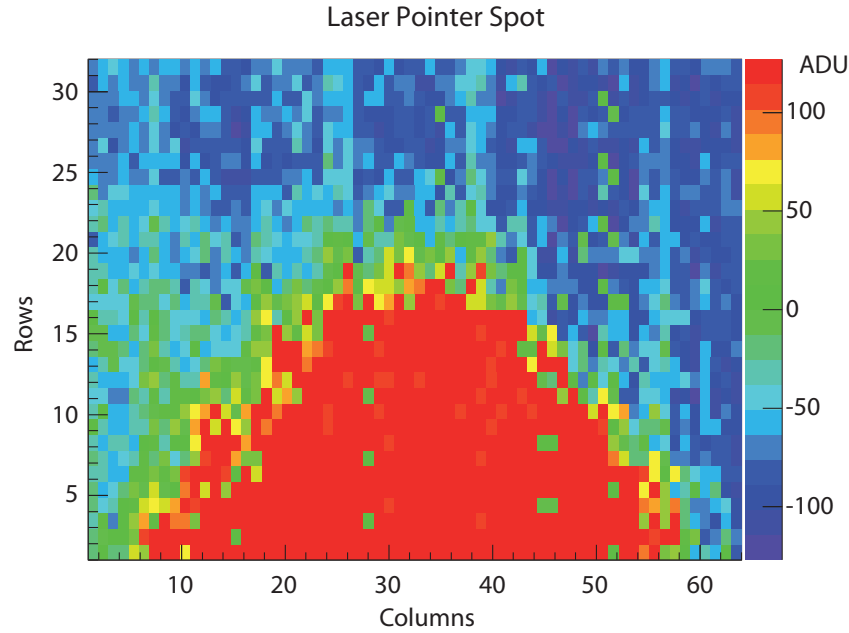


Figure 8-6 Laser pointer spot on the edge of the accessible part of the DEPFET detector matrix.

columns of the matrix. In particular, the mapping of the matrix pixels via DCDB channels and the various serialization steps inside the DCDB and the FPGA to the displaying software is by far a non-trivial task. In this context, being able to capture such a picture is a great success. Nevertheless, more detailed and fine-grained measurements are necessary in order to ultimately prove the mapping scheme. Such measurements are successfully done by the group of Prof. Dr. Wermes at the University of Bonn. Fortunately, the required equipment to focus a laser spot onto a single pixel is available there [74].

8.3.2 Radioactive Source Measurement

The measurement shown above demonstrates that the presented prototype system is basically working. This means mainly that the detector is reasonably biased as well as the readout chain is operating and synchronized. The ultimate test to show its capabilities as an instrument for particle physics application is now to measure the signal of incident particles.

The measurement is carried out using a radioactive material, cadmium-109 (Cd) in this case, as signal source. It is placed as close as possible (about 6 mm) above the top side of the detector. The detector is read out continuously with an increased row readout period of 320 ns , resulting in a frame readout period of $5.12\text{ }\mu\text{s}$. This is because of a reduced maximum operation speed of the used SwitcherB for clearing the detector due to a malfunction. Although the system is prepared for a triggered readout in order to forward only those frames to the host PC that are supposed to contain a hit signal, this scheme is not used for the sake of the setup's simplicity. Technically, this means that the system is triggered randomly without any correlation to incident particles. This results in frame read rate by the host PC of about 200 Hz , thus only roughly 0.1% of the physically

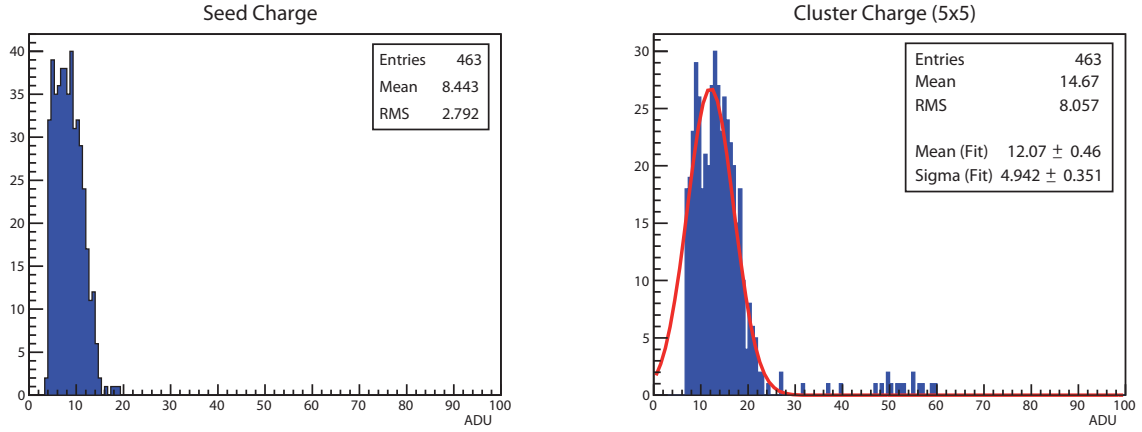


Figure 8-7 Spectrum of a radioactive Cd-109 source. The histogram on the left shows the seed pixel charge distribution, the one on the right gives the charge distribution of 5×5 pixel clusters as well as a gaussian fit (red curve).

captured frames are used for the analysis. The setup is darkened as accurately as possible in order to avoid hit signals induced by environmental light.

The captured data is analysed in the following way. After pedestal and common mode correction, the frames are scanned for so-called *seed pixels*. Assuming that the charge cloud in the detector material induced by incident particles spreads over several pixels, the pixel showing the largest signal is considered being the seed pixel. In order to avoid fake hits, the requirement for such a seed pixel is to show a signal that is larger than eight times its noise. A certain amount of pixels surrounding the seed pixel is analysed as well in order to determine the full signal. Here, clusters of 5×5 pixels are considered and a cut of five times the respective noise is applied.

The result of the measurement is shown in figure 8-7. The decays of Cd-109 produce photons, predominantly with an energy of 22.1 keV . By absorption in silicon, these photons create 6139 electron-hole pairs on average, assuming an ionisation energy of 3.6 eV . Hence, this corresponds approximately to the signal of a minimum ionising particle (MIP) in a DEPFET detector with a thickness as it is proposed for the PXD of BELLE-II. Based on that, the total gain of the system as well as the gain of the detector's DEPFET transistors can be calculated. The gaussian fit of the cluster charge gives a mean value of 12.07 ADU per cluster, which corresponds to a total gain of $508.6 e^-/\text{ADU}$. The gain (RMS) of the DCDB channels is $103.3 \text{ nA}/\text{ADU}$ as determined in section 7.3.3, so the measured gain of the DEPFET transistors is $203 \text{ pA}/e^-$. The latter result is only in rather poor agreement with the nominal DEPFET transistor gain for this detector type of about $300 \text{ pA}/e^-$, which was determined using 2008 beam test data [66]. This is, however, easily explainable with the non-optimal biasing of the detector due to the too weak static pedestal current subtraction source (NSubIn) of the DCDB. The mean noise of the various pixels in this measurement is 0.53 ADU , which is even less than what is reported in section 7.3.4, due to the reduced readout speed. Thus, the measured noise corresponds to 270 signal electrons in the detector and the measured signal-to-noise ratio is:

$$\frac{S}{N} = \frac{12.07 \text{ ADU}}{0.53 \text{ ADU}} = 22.8$$

These results are of course going to further improve once a more powerful NSubIn source of the DCDB allows for optimal detector operation conditions. More spectrum measurements using this readout system at full speed together with a thinned PXD6 (BELLE-II-type prototype) detector in a more sophisticated setup have already been successfully performed by the team at the Semiconductor Laboratory (MPI, Munich) and published in [75]. They report a signal-to-noise ratio of 17 for a strontium spectrum¹.

8.3.3 Clear Efficiency Studies

Clear efficiency studies using the trailing frames readout mode as described in section 6.2.3 have been performed and published in [77]. Here, a Cd-109 spectrum is measured and for every particle hitting the detector four consecutive frames are read. If the clear potentials are adjusted correctly, only the first frame in a set of four should have the hit in it, while the trailing three frames should be empty due to the clearing of the signal. However, for too low clear potentials shadows appear in the trailing frames because of an incomplete clearing process. As a conclusion of these measurements, the clear efficiency as a function of the clear potential is derived for a PXD5 detector as illustrated in figure 8-8. The results correspond to a clear pulse width of approximately 40ns.

8.3.4 Beam Test Period at CERN

It has become a nice tradition of the DEPFET collaboration over the past years to more or less regularly visit CERN, the *European Organization for Nuclear Research*, for comprehensively testing new detector and readout component developments in a beam test experiment. Remarkable results [78] were obtained, for instance, with the so-called

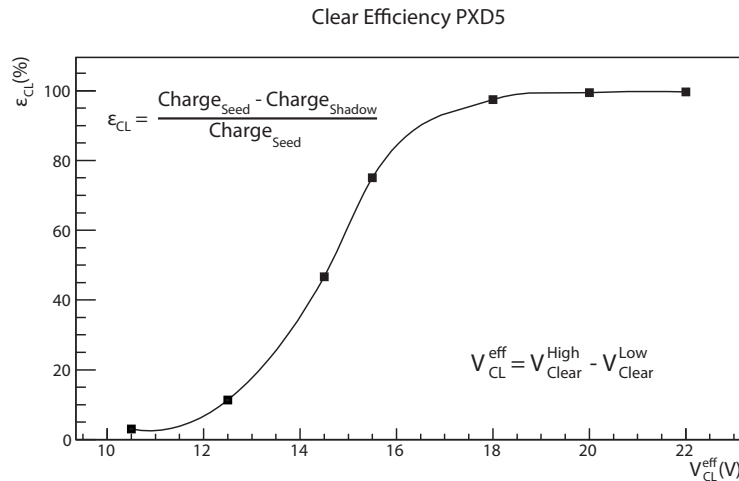


Figure 8-8 Clear efficiency of the PXD5 detector as a function of the clear potential. The clear pulse width is approximately 40ns [77].

1. Electrons with a maximum energy of 2.283MeV are emitted via the decay chain $^{90}\text{Sr} \rightarrow ^{90}\text{Y} \rightarrow ^{90}\text{Zr}$ [76]. These correspond approximately to a minimum ionizing particle and they are detected with a signal-to-noise ratio of 17.

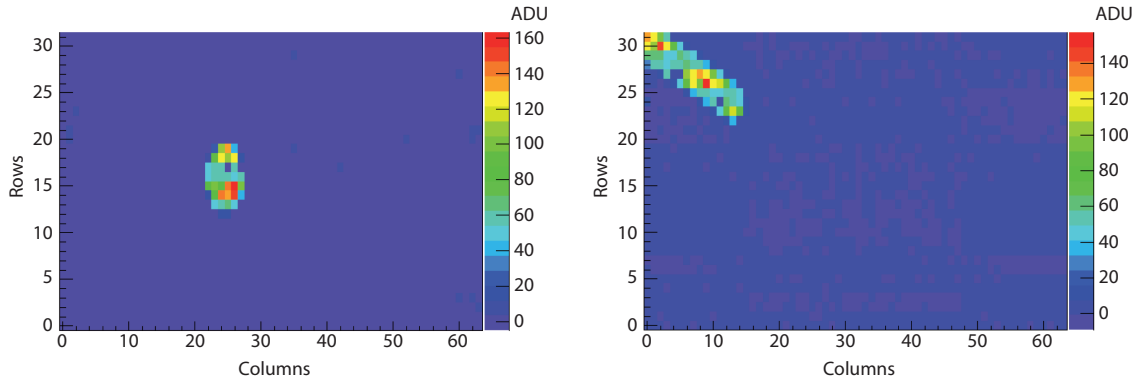


Figure 8-9 Nuclear interactions observed during the beam test experiment with the DCDB-based DEPFET prototype system at CERN SPS [80].

S3B system [67] based on the Switcher3 [79] and the CURO [21] chips, which is the predecessor setup of the one presented herein.

Continuing this tradition, the DEPFET collaboration went to CERN in November 2010 for a beam test experiment with the DCDB-based DEPFET prototype system at the SPS (Super Proton Synchrotron) accelerator. In fact, it was exactly that module including all chips as well as the detector which is introduced in section 6.1. The system was integrated as DUT (Design Under Test) into the EUDET telescope.

Several tests and scans were carried out on the device during the beam test experiment in order to determine the system's performance parameters. These are, for example, scans over several bias voltages of the detector for comparison with earlier results. DCDB configuration parameter scans and SwitcherB steering sequence variations were performed as well expecting to learn more about these devices that were used in a beam test for the very first time.

Unfortunately, the system to be tested got ready only very shortly before the scheduled beam time. So there was only very limited time to calibrate the system in the laboratory beforehand. Consequently, it was after the beam test when it became evident that the bonding of the two SwitcherB chips to the detector had been misaligned. This led to the situation that different rows of the detector were addressed for reading and clearing, which is of course dramatically degrading the performance of the system. The captured data is therefore only of limited usefulness for physics studies. Nevertheless, the beam test gave the great opportunity to test all aspects of the readout system, from the operation of the front-end readout chips up to the integration into the analysis framework provided by EUDET, with all experts at one place. Moreover, a few nice pictures of nuclear interactions within the detector material, like those shown in figure 8-9, could be observed.

The next beam test period at CERN with the DCDB-based prototype system is already scheduled for October 2011. This time, it is even planned to use the first thinned BELLE-II type PXD6 detector prototype, which is going to bring the results closer to the final performance of the PXD sub-detector for the BELLE-II experiment.

Conclusion & Outlook

Abstract:

This closing chapter provides a retrospection of the presented work and summarizes the major results as well as the author's contributions. Moreover, it gives an outlook to the upcoming milestones of the PXD detector system's readout chain in the near future.

9.1 Conclusion

After introducing the DEPFET principle and the design parameter of BELLE-II as well as their implications to the readout system, the presented work focused on the development, characterization and operation of the DCDB chip, the front-end readout ASIC for the PXD detector system of BELLE-II. It is supposed to sequentially sample and digitize current-modulated signals from the pixels of a DEPFET detector. The specifications for the DCDB are derived from physics aspects in chapter 2. The most important requirements are a large number of channels (at least 250), a sampling period of 100ns and a noise of 150nA . Signals per pixel are expected up to $8\mu\text{A}$, while pedestal fluctuations in the same order need to be handled. The conversion should have a resolution of 8 bit.

The DCDB is designed to meet these requirements. It provides 256 input channels in total. Each consists of an input current receiver based on a transimpedance amplifier and two cyclic analog-to-digital converter circuits. The implementation details are described in chapter 3. Special emphasis is placed on the development of the chip's digital domain, which implements a data format conversion and output serialization. The DCDB is the first front-end readout device for DEPFET detectors that is not entirely a full-custom design but makes use of advanced digital design techniques. The details are discussed in chapter 4.

In order to test the chip and to verify its performance, a FPGA-based test environment was developed as described in chapter 5. Large effort is put on the design of the FPGA's firmware, since the major intelligence for operating the DCDB is implemented there. The tests and the results are exhaustively discussed in chapter 7. The most important outcomes are summarized as follows. Firstly, the DCDB's digital logic is functional and operates even beyond the target speed of 320MHz at a reasonable power consumption. This is a big and remarkable success as it proves the know-how of the entire digital design process! Next, the chip's analog circuits are examined showing that almost all requirements are met. The combined speed of the transimpedance amplifier's signal transitions and ADC's sampling stays below the specified 100ns . The noise referred to the input current is 93.7nA (RMS) and thus fairly less than the required 150nA . The linearity is reasonable and with a dynamic range of roughly $25\mu\text{A}$ in conjunction with the static and dynamic pedestal current compensation mechanisms, the DCDB is sufficiently prepared for being operated together with a DEPFET detector. Nevertheless, the noise performance was expected to be even better by a factor of two. Indeed, on the digital side, the noise corresponds to 0.89ADU , which reduces the effective signal resolution measured in bit. Successive versions of the design with possible issues being fixed are already submitted to production.

The DCDB is embedded into a FPGA-based DEPFET prototype system, together with a detector matrix and SwitcherB steering chips. The FPGA is again the heart of the system as it is used to operate and synchronize the various chips. Thus, there is again large effort spent on implementing its firmware as described in chapter 6. Its successful operation is demonstrated in chapter 8. The most significant milestones are the proven effectiveness of the DCDB's dynamic pedestal current compensation, the spectrum measurement of a radioactive material (Cd-109) and the operation in a beam test experiment at CERN. The spectrum measurement is of outstanding importance, as it shows a signal-to-noise ratio

of 22.8 for a signal that is in the order of what is expected for a minimum ionizing particle at the final BELLE-II PXD detector geometries.

After all, the DCDB in version one is a fairly good baseline design for being used in the PXD detector for BELLE-II. New improved versions that are expected to overcome the limitations of that first DCDB design are on the way.

9.2 Summary of Own Contributions

The following list summarizes the contributions of the author, Jochen Knopf, as a member of the DEPFET collaboration to the progress of the PXD sub-detector development for BELLE-II:

- Integration of the Switcher3 chip into the existing S3B detector prototype readout system. Update of the firmware for the S3B system's FPGA as a preparatory work for the DEPFET collaboration's beam test period in 2009.
- Creation of a digital standard cell library based on mostly existing cell designs for the use in the physical implementation of the DCDB's digital domain. This includes the development of two test chips for the verification of the library and the physical implementation procedure.
- Development of the entire digital domain of the DCDBv1, the DCDB-TC and the DCDBv2. This includes the logic's functional description (except the JTAG interface), its simulation as well as its physical implementation using state-of-the-art software tools and methodologies.
- Development of the FPGA firmware as well as the appropriate software program for the DCDB test environment.
- Development of the FPGA firmware and the hardware-interacting software layer for the DCDB-based detector prototype system. This includes the system's first operation.
- First operation of the DCDB and realization of its characterization. This includes both, the digital functionality checks as well as the detailed investigation of the design's analog performance parameters.
- Setup of the system level simulation of the PXD detector system's front-end readout chain based on existing models of the involved ASICs and the DEPFET transistor.
- Development of an algorithm for minimizing the pedestal current fluctuations by means of the DCDB's dynamic pedestal compensation feature.
- Setup and realization of the Cd-109 spectrum measurement with the DCDB-based detector prototype system.
- Active contributions to realizing the DEPFET collaboration's beam test periods in the years 2009 and 2010 as person in charge for the hardware operation. Further active contribution in the upcoming 2011 beam test period is planned.

9.3 Outlook

In the near future, that is within the remaining part of 2011, the development of the front-end readout electronics for the BELLE-II PXD detector will be pushed forward by four main events. First of all, this will be the availability of the DCDBv2. Although the manufacturer already announced that production problems occurred, the chip is expected to be delivered still within 2011. Characterizing this ASIC will provide new knowledge about the design and most probably show improvements of major design performances. Secondly, the upcoming beam test experiment at CERN in October 2011 will be the premiere for the thinned BELLE-II type PXD6 prototype detector. Its operation together with the SwitcherB and the DCDB will provide performance results that are very close those of the final detector. The third event will be the return of the first full size DHP chip from production. By having some problems fixed, this chip is expected to be fully compliant with the DCDB design. Finally, a big step forward towards the final system setup will be taken by the tests with an electrical module of the half-ladder. This module is a half-ladder design comprising everything but the DEPFET detector structures. This allows to test the interactions of the various chips under real conditions in terms of voltage drops, signal integrity and so on. Thermal studies with real devices instead of dummies will be possible as well.

Obviously, the various parts of the PXD readout chain evolve quite nicely towards final designs. Nevertheless, a major design change is currently discussed, triggered by the need to change the technology for future versions of the DHP. Up to now, the DCDB is developed in a $180nm$ technology node, the DHP is implemented using $90nm$ minimal feature size. Transferring the latter design to a $65nm$ technology offers free space that can be used to merge the DCDB and the DHP into one ASIC. On the one hand, this is a big chance to simplify the enormously complex setup on half-ladder level and eventually even to improve the combined performance of the two designs. On the other hand, taking such a major design change in the already late state of the project is doubtlessly a risk. Not so much for the digital design parts, but the development of the analog blocks needs to start almost from scratch. First test structures of DCDB's analog design elements in a $65nm$ UMC technology have already been submitted to production [69]. Characterizing this chip once it returns will provide a good basis for discussions on whether to go for this solution or not. In principle, the expected slip of the BELLE-II schedule by roughly one year could provide the required time budget. But nevertheless, time is short, since the decision on the chip footprint and the interconnection on the final half-ladder module will have to be fixed very soon in order to start their final production.

Bibliography

- [1] **J. H. Christenson et al.**, "*Evidence For The 2π Decay of The K_2^0 Meson*", Phys. Rev. Lett. 13 (1964) pp. 138-140.
- [2] **M. Kobayashi and T. Maskawa**, "*CP-Violation in the Renormalizable Theory of Weak Interaction*", Progr. Theor. Phys. 49 (1973) pp. 652-657.
- [3] **S. W. Herb et al.**, "*Observation of a Dimuon Resonance at 9.5GeV in 400GeV Proton - Nucleus Collisions*", Phys. Rev. Lett. 39 (1977) pp. 252-255.
- [4] "*The Nobel Prize in Physics 2008 - Scientific Background*", Nobelprize.org. 9 Nov 2010
http://nobelprize.org/nobel_prizes/physics/laureates/2008/sci.html
- [5] **A. D. Sakharov**, "*Violation of CP Invariance, C Asymmetry, and Baryon Asymmetry of the Universe*", Pis'ma Zh. Eksp. Teor. Fis. 5 (1967) pp. 32-35 [JETP Lett. 5 (1967) pp. 24-27].
- [6] **K. Sakai et al.**, "*Search for CP-violating charge asymmetry in $B^\pm \rightarrow J/\psi K^\pm$ decays*", arXiv:1008.2567v3 [hep-ex] (2010).
- [7] **A. G. Akeroyd et al.**, "*Physics at Super B Factory*", arXiv:1002.5012v1 [hep-ex] (2010) .
- [8] **sBelle Design Group, I. Adachi et al.**, "*sBelle Design Study Report*", arXiv:0810.4084v1 [hep-ex] (2008).
- [9] **Belle-II Collaboration, T. Abe et al.**, "*Belle-II Technical Design Report*", arXiv:1011.0352v1 [physics.ins-det] (2010).
- [10] **K. Kleinknecht**, "*Uncovering CP Violation - Experimental Clarification in the Neutral K Meson and B Meson Systems*", Springer (2003), ISBN 3-540-40333-7.
- [11] **P. Müller**, "*Investigation on Radiation Hardness of DEPFET Pixel-Detectors for Belle II*", Diploma Thesis, Ludwig-Maximilians-University Munich (2010).
- [12] **T. E. Browder et al.**, "*New Physics at a Super Flavor Factory*", arXiv:0802.3201v2 [hep-ph] (2008).

-
- [13] **Particle Data Group, C. Amsler et al.**, "*Review of Particle Physics*", Physics Letters B667, 1 (2008).
- [14] **K. Inami**, "*Development of a TOP counter for the Super B factory*", Nucl. Instrum. Meth. A595 (2008) pp. 96-99.
- [15] **J. Kemmer and G. Lutz**, "*New Structures For Position Sensitive Semiconductor Detectors*", Nucl. Instrum. Meth. A273 (1988), pp. 588-598.
- [16] **J. Kemmer, G. Lutz et al.**, "*Experimental Confirmation of a New Semiconductor Detector Principle*", Nucl. Instrum. Meth. A288 (1990), pp. 92-98.
- [17] **P. Klein et al.**, "*Study of a DEPFET pixel matrix with continuous clear mechanism*", Nucl. Instrum. Meth. A 392 (1997), pp. 254-259.
- [18] **P. Fischer et al.**, "*First Operation of a Pixel Imaging Matrix based on DEPFET pixels*", Nucl. Instrum. Meth. A 451 (2000), pp. 651-656.
- [19] **P. Klein et al.**, "*A DEPFET Pixel Bioscope for the use in autoradiography*", Nucl. Instrum. Meth. A 454 (2000), pp. 152-157.
- [20] **P. Fischer et al.**, "*A DEPFET Pixel Vertex Detector for TESLA. Proposal and Prototyping Report*", DESY PRC Report 4/2003.
- [21] **M. Trimpl et al.**, "*A DEPFET Pixel Matrix System for the ILC Vertex Detector*", Nucl. Instrum. Meth. A 560 (2006), pp. 21-25.
- [22] **M. Porro et al.**, "*Large Format X-ray Imager with Mega-Frame Readout Capability for XFEL, based on the DEPFET Active Pixel Sensor*", Nuclear Science Symposium Conference Record, 19-25 Oct. 2008. IEEE, N14-7, pp. 1578-1586.
- [23] **P. Fischer et al.**, "*Readout Concepts for DEPFET Pixel Arrays*", Nucl. Instrum. Meth. A 512 (2003), pp 318-325.
- [24] **H. Moser**, "*Summary of PXD Session*", 1st Open Meeting of the SuperKEKB collaboration (2008).
- [25] **D. Koetke**, "*A Measurement of the Z^0 Hadronic Branching Fraction to Bottom Quarks and the Charged Multiplicity of Bottom Quark Events Using Precision Vertex Detectors at $E_{cm} = 91\text{ GeV}$* ", SLAC-396, Stanford University (1992).
- [26] **M. Battaglia**, "*Vertex Tracking at a Future Linear Collider*", doi:10.1016/j.nima.2010.12.110.

-
- [27] **S. Tanaka**, *"IR Status and Schedule"*, 7th International Workshop on DEPFET Detectors and Applications (2011).
- [28] **B. Reisert et al.**, *"Performance Studies"*, 5th Open Meeting of the Belle II Collaboration (2010).
- [29] **A. Moll**, *"PXD Occupancy Studies from Machine and QED Background"*, 7th International Workshop on DEPFET Detectors and Applications (2011) .
- [30] **L. Rossi et al.**, *"Pixel Detectors - From Fundamentals to Applications"*, Springer (2006), ISBN 3-540-28332-3.
- [31] **M. Ritter et al.**, *"Mechanics & Glue Tests"*, 5th Open Meeting of the Belle II Collaboration (2010).
- [32] **Z. Drásal et al.**, *"Optimization Studies of Pixel Dimensions"*, 2nd International Workshop on DEPFET Detectors and Applications (2010).
- [33] **Z. Drásal et al.**, *"PXD Performance at High QED Occupancies"*, 6th International Workshop on DEPFET Detectors and Applications (2011).
- [34] **M. Ritter et al.**, *"Module Mechanics: Update on Dimensions and Half-Module Assembly"*, 7th International Workshop on DEPFET Detectors and Applications (2011).
- [35] **L. Andricek et al.**, *"The MOS-type DEPFET pixel sensor for the ILC environment"*, Nucl. Instrum. Meth. A 565 (2006), pp.165-171.
- [36] **C. Kreidl**, *"Steering electronics, module design and construction of an all silicon DEPFET module"*, Doctoral Thesis, University of Mannheim (2011).
- [37] **K. Prothmann et al.**, *"Effects of the PXD Sensor Thickness on the Impact Parameter Resolution"*, 4th International Workshop on DEPFET Detectors and Applications (2010).
- [38] **B. Reisert et al.**, *"Benchmark Physics Performance - Sensor Thickness"*, 4th International Workshop on DEPFET Detectors and Applications (2010).
- [39] **P. Fischer et al.**, *"Switcher-B Reference Manual"*, Group of Circuits and Simulation, University of Heidelberg (November 2010), <http://twiki.hll.mpg.de/twiki/pub/DepfetInternal/DesignResourcesSwitcher-B/SwitcherB-ReferenceManual-v1.1-05112010.pdf>.
- [40] **J. Knopf et al.**, *"A 256 Channel 8-Bit Current Digitizer ASIC for the Belle-II PXD"*, JINST 6 C01085 doi: 10.1088/1748-0221/6/01/C01085 (2011).

-
- [41] **M. E. Waltari and K.A.I. Halonen**, *"Circuit Techniques for Low-Power and High-Speed A/D Converters"*, Kluwer Academic Publishers, (2002).
- [42] **I. Peric et al.**, *"DCD - The Multi-Channel Current-Mode ADC Chip for the Readout of DEPFET Pixel Detectors"*, IEEE Trans. Nucl. Sci. 57 (2010), pp 743-753.
- [43] **H. Krüger**, *"Front-end electronics for DEPFET pixel detectors at SuperBelle (BELLE II)"*, Nucl. Instrum. Meth. A 617 (2010), pp. 337-341.
- [44] **H. Krüger et al.**, *"DHP 0.2 Status"*, 7th International Workshop on DEPFET Detectors and Applications (2011).
- [45] **C. Frougny**, *"On-the-Fly Algorithms and Sequential Machines"*, Proc. 13th IEEE Symp. Computer Arithmetic (1997), pp. 260-265.
- [46] **I. Peric et al.**, *"DCD-B Reference Manual"*, Group of Circuits and Simulation, University of Heidelberg (March 2010), <http://twiki.hll.mpg.de/twiki/pub/DepfetInternal/DesignResourcesDCDB/DCD-B-ReferenceManual.pdf>.
- [47] **IEEE Computer Society**, *"IEEE 1149.1-2001, IEEE Standard Test Access Port and Boundary-Scan Architecture"*, The Institute of Electrical and Electronics Engineers, Inc. (2001), ISBN 0-7381-2944-5.
- [48] **OVM Website**, <http://www.ovmworld.org> , accessed on February, 15th, 2011.
- [49] **J. Kinzel**, *"Design, Implementation and Verification of a High Performance NAND Flash Based Storage System with HyperTransport Interface"*, Diploma Thesis, Chair of Computer Architecture, University of Mannheim (2008).
- [50] **UVM Website**, <http://www.uvmworld.org> , accessed on February, 15th, 2011.
- [51] **S. Redant et al.**, *"Radiation Test Results on First Silicon in the Design Against Radiation Effects (DARE) Library"*, IEEE Transactions on Nuclear Science, Vol. 52, No. 5 (2005).
- [52] **M. Bruder**, *"Design und Implementierung einer strahlenharten Standardzellbibliothek"*, Diploma Thesis, Chair of Circuit Design and Simulation, University of Mannheim (2006).
- [53] **Cadence Design Systems, Inc.**, *"Using Encounter RTL Compiler, Product Version 8.1.202"* (April 2009).
- [54] **Cadence Design Systems, Inc.**, *"Encounter User Guide, Product Version 8.1.3"* (September 2009).

-
- [55] **Cadence Design Systems, Inc.**, "*Using Encounter RTL Compiler, Product Version 10.1*" (February 2011).
- [56] **Cadence Design Systems, Inc.**, "*Encounter Digital Implementation System User Guide, Product Version 10.1.1*" (March 2011).
- [57] **IDESA**, "*Advanced Digital Physical Implementation Flow*", Lecture Notes (2008), <http://www.idesa-training.org>, accessed on April, 11th, 2011.
- [58] **M. Koch**, "*Development of a Test Environment for the Characterization of the Current Digitizer Chip DCD2 and the DEPFET Pixel System for the Belle-II Experiment at SuperKEKB*", Doctoral Thesis, University of Bonn (2011).
- [59] **V4Board Website**, <http://pi.physik.uni-bonn.de/~koch/pcbs/virtex4-pcb.html>, accessed on April, 17th, 2011.
- [60] **Xilinx Inc.**, "*Virtex-4 Family Overview, DS112 (v3.1)*", (August 2010).
- [61] **D. Cussans**, "*Description of the JRA1 Trigger Logic Unit (TLU), v0.2c*", EUDET-Memo-2008-50-2 (2008).
- [62] **Xilinx Inc.**, "*Virtex-4 FPGA Data Sheet: DC and Switching Characteristics, DS302 (v3.7)*", (September 2009).
- [63] **Qt Website**, "<http://qt.nokia.com>", accessed on April, 22nd, 2011.
- [64] **C. Koffmane**, "*Hybrid Boards for PXD6*", 5th International Workshop on DEPFET Detectors and Applications, Valencia (2010).
- [65] **M. Lemarenko**, "*DHP emulator*", 5th International Workshop on DEPFET Detectors and Applications, Valencia (2010).
- [66] **L. Reuen**, "*Analysis of pixel systematics and space point reconstruction with DEPFET PXD5 matrices using high energy beam test data*", Doctoral Thesis, University of Bonn (2011).
- [67] **S. Furletov**, "*A system for characterization of DEPFET silicon pixel matrices and test beam results*", Nucl. Instrum. Meth. A 628 (2011), pp. 221-225.
- [68] **J. Furletova and L. Reuen**, "*JRA1 - The DEPFET sensor as the first fully integrated DUT in the EUDET pixel telescope: The SPS test beam 2008*", EUDET-Memo-2008-34 (2008).

-
- [69] **I. Peric et al.**, "*Chips submitted in March and April*", 7th International Workshop on DEPFET Detectors and Applications (2011).
- [70] **J. Knopf et al.**, "*DCDB Performance and Operation Updates*", 7th International Workshop on DEPFET Detectors and Applications (2011).
- [71] **A. Wassatsch and R. Richter**, "*DEPFET Simulation Model*", 2nd International Workshop on DEPFET Detectors and Applications (2009).
- [72] **Cadence Design Systems, Inc.**, "*Compiled-Model Interface Reference, Product Version 4.0*", (November 2004).
- [73] **C. Koffmane**, "*DEPFET - Parasitic Parameter Extraction*", Belle-II PXD / DEPFET Meeting (2010).
- [74] **K. Schmieden**, "*Charakterisierung einer neuen Generation von DEPFET-Sensoren mit Hilfe eines Lasermesssystems*", Diploma Thesis, University of Bonn (2008).
- [75] **C. Koffmane**, "*PXD6 Matrix Tests, Hybrid Production*", 9th Open Meeting of the Belle II Collaboration (2011).
- [76] **T. Jindra**, "*Development of semiconductor detectors for high energy physics experiments*", Bachelor Thesis, Prague University (2010).
- [77] **G. Eneas Timón Grau**, "*DEPFET pixel stability*", 7th International Workshop on DEPFET Detectors and Applications (2011).
- [78] **L. Andricek et al.**, "*Intrinsic resolution of DEPFET detector prototypes measured at beam tests*", Nucl. Instrum. Meth. A 638 (2011), pp. 24-32.
- [79] **P. Fischer et al.**, "*Steering and Readout Chips for DEPFET Sensor Matrices*", Proceedings of the TWEPP 2007 (2007).
- [80] **B. Schwenker**, "*Analysis of Test Beam and Source Measurements with Hybrid 4.1*", 6th International Workshop on DEPFET Detectors and Applications (2011).

Acknowledgements

First of all, special thanks go to my wife Andrea and my parents. I am deeply grateful for the support they provided during the years of my doctoral studies. They have always been my source of motivation and mental strength.

I would like to thank my doctoral advisor Prof. Dr. Peter Fischer. He gave me the great opportunity to work in the fascinating science of experimental particle physics. I really appreciate his active support and encouragement to extend my knowledge in this field.

Also, I would like to thank Dr. Ivan Perić, my second advisor. He supported me with his outstanding knowledge of microelectronics and ASIC development, especially in the context of particle physics applications.

Many thanks go to Annette Knopf and John Smith-Malzfeldt for proofreading this thesis. I am very grateful for their support.

Last but not least, I would like to thank my colleagues and friends at the Circuit Design Group and the DEPFET collaboration for their warm welcome, the excellent teamwork, the constructive discussions and the fun we had.
