All-Digital Clock Calibration for Source-Synchronous High-Speed I/O Links Based on Phase-to-Digital Conversion

Angeli, Nico

(2020)

All-Digital Clock Calibration for Source-Synchronous High-Speed I/O Links Based on Phase-to-Digital Conversion

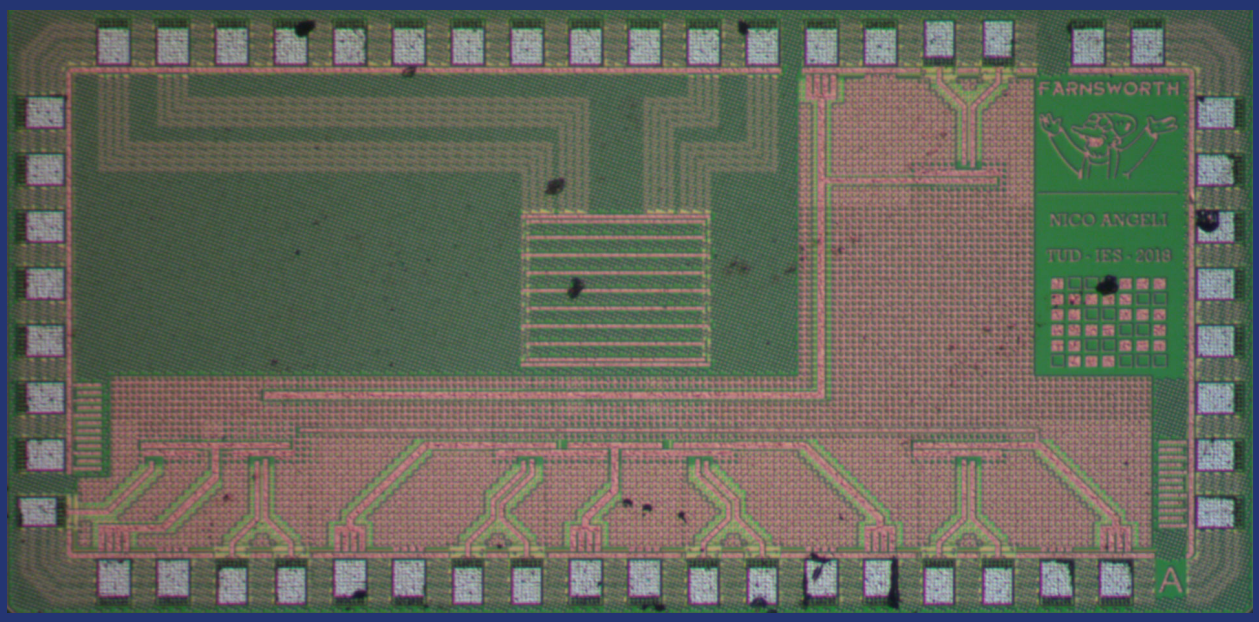
Digitale Taktsignal Kalibrierung für schnelle quellsynchrone I/O Links basierend auf Phasen-zu-digital Wandlung

Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation von M.Sc. Nico Angeli aus Würzburg
Tag der Einreichung: 13. März 2020, Tag der Prüfung: 01. Juli 2020
Darmstadt — D 17

1. Gutachten: Prof. Dr.-Ing. Klaus Hofmann
2. Gutachten: Prof. Dr.-Ing. Chihao Xu

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachgebiet
Integrierte Elektronische Systeme



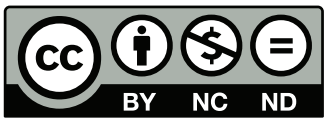
All-Digital Clock Calibration for Source-Synchronous High-Speed I/O Links Based on Phase-to-Digital Conversion
Digitale Taktsignal Kalibrierung für schnelle quellsynchrone I/O Links basierend auf Phasen-zu-digital Wandlung

Genehmigte Dissertation von M.Sc. Nico Angeli aus Würzburg

1. Gutachten: Prof. Dr.-Ing. Klaus Hofmann
2. Gutachten: Prof. Dr.-Ing. Chihao Xu

Tag der Einreichung: 13. März 2020
Tag der Prüfung: 01. Juli 2020

Darmstadt — D 17



Erklärungen zur Dissertation

§ 8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§ 8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§ 9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§ 9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 18. August 2020

.....
Nico Angeli

Abstract

Wire-linked high-speed interfaces play an important role in modern computing systems. They are required to transfer enormous amounts of data in a short time between processors, memories and periphery. Many efforts are put into the development of I/O links with higher data rates and lower power consumption in order to keep up with the increasing processing speed of CPUs and GPUs. Especially memory interfaces are struggling to reach the bandwidth required by the processors. Source-synchronous parallel I/O links that connect the processor and the memory with PCB interconnects are limited by the high-frequency characteristics of the PCB and the maximum number of parallel data links. New interface types with shorter interconnects and more parallelization based on a silicon interposer instead of a PCB have been introduced to tackle this problem, but due to the higher manufacturing cost these interfaces still haven't fully replaced traditional PCB based memory interfaces.

A common challenge in all parallel high-speed I/O links is the calibration of the ideal sampling time for each individual data link. The conventional method requires time consuming bit error rate measurements over a 180° wide range of possible sampling phase positions and is prone to timing center errors due to the nonlinearity of the phase generation. In this work a fast timing center calibration based on a phase measurement between the received signal and the internal clock is presented which reduces the calibration time significantly and minimizes the timing center error by compensating the nonlinearity of the phase generation. The phase measurement is performed by a fully synthesized phase-to-digital converter (PDC) offering precise phase and duty-cycle measurements of high-frequency clock signals with low power consumption. A calculation method for estimating the deviation of the PDC measurement result from the actual value due to clock jitter allows to predict the behavior of the PDC and can be used to generate jitter specifications for the clock signals.

The proposed PDC also offers new possibilities in the implementation of other high-frequency clock calibration circuits such as delay-locked loops and duty-cycle correctors. Three ASICs in a 65 nm CMOS process have been designed that demonstrate the feasibility in four different clock calibration scenarios of a prototype bidirectional asymmetric source-synchronous I/O link similar to the GDDR5 standard for GPU memories. The application of the PDC as a linear phase detector in delay-locked loops simplifies the design procedure and results in a small and power efficient circuit that scales well with technology thanks to the all-digital implementation. The capability of the PDC to measure the duty-cycle of high-frequency clock signals with high precision is used to correct the duty-cycle in the high-speed transceivers at different nodes with a low area overhead. In addition to the timing center calibration a voltage threshold calibration based on the duty-cycle measurement of a received clock pattern is proposed that further reduces the calibration time of single-ended I/O links and simplifies the calibration procedure.

Kurzfassung

Schnelle leitungsgebundene I/O Schnittstellen spielen eine wichtige Rolle in modernen Computersystemen. Sie werden benötigt, um enorme Datenmengen in kürzester Zeit zwischen Prozessoren, Speichern und Peripherie zu übertragen. Trotz stetiger Bemühungen die Datenrate und den Leistungsbedarf von I/O Schnittstellen zu verbessern, um mit der anhaltenden Steigerung der Prozessorleistung mitzuhalten, ist es vor allem bei Speicherschnittstellen schwierig die geforderte Bandbreite zu erreichen. Die Datenrate quellsynchroner paralleler I/O Links die zur Verbindung von Prozessor und Speicher über eine Leiterplatte verwendet werden ist vor allem durch die Hochfrequenzeigenschaften der Leiterplatte und die maximale Anzahl an parallelen Datenleitungen begrenzt. Neuartige Schnittstellen mit kürzeren Verbindungen und mehr Parallelisierung basierend auf einem Silizium Interposer anstelle einer Leiterplatte versuchen dieses Problem zu lösen, haben jedoch aufgrund der höheren Herstellungskosten klassische Speicherschnittstellen noch nicht vollständig abgelöst.

Eine allgemeine Herausforderung bei schnellen parallelen I/O Schnittstellen ist die individuelle Kalibrierung des idealen Abtastzeitpunkts für jede Datenleitung. Die konventionelle Methode basiert auf der Messung der Bitfehlerrate über einen 180° umfassenden Bereich möglicher Abtastzeitpunkte. Dieser Ansatz benötigt eine lange Zeit für die Kalibrierung und kann zu einem nicht optimalen Kalibrierungsergebnis aufgrund der Nichtlinearität der Phasenerzeugung führen. In dieser Arbeit wird eine schnelle Kalibrierungsmethode basierend auf einer Phasenmessung zwischen dem empfangenen Signal und dem internen Taktsignal vorgestellt. Diese reduziert die Kalibrierungszeit deutlich und minimiert zusätzlich den Phasenfehler durch eine Kompensation der Nichtlinearität der Phasenerzeugung. Die Phasenmessung erfolgt durch einen voll synthetisierten Phasen-zu-digital Wandler (PDC), welcher hochpräzise und energieeffiziente Phasen- und Tastgradmessungen von hochfrequenten Taktsignalen ermöglicht. Eine Berechnungsmethode zur Abschätzung der Messfehler des Wandlers aufgrund von Jitter erlaubt es das Verhalten des PDC vorherzusagen und Jitterspezifikationen für die Taktsignale abzuleiten.

Der vorgestellte PDC bietet zudem neue Möglichkeiten bei der Implementierung anderer Kalibrierungsschaltungen für hochfrequente Taktsignale wie z.B. Delay-Locked Loops und Tastgradkorrektoren. Um die Machbarkeit solcher Kalibrierungsschaltungen anhand einer beispielhaften bidirektionalen asymmetrischen quellsynchronen I/O Schnittstelle, ähnlich dem GDDR5 Standard für GPU Speicher, zu demonstrieren wurden insgesamt drei ASICs in einer 65 nm CMOS Technologie hergestellt. Der Einsatz des PDC als linearer Phasendetektor in Delay-Locked Loops beispielsweise vereinfacht den Entwurf und führt zu einer energieeffizienten Schaltung mit geringer Größe, welche dank der voll digitalen Implementierung gut mit der verwendeten Technologie skaliert. Die Fähigkeit des PDC den Tastgrad von hochfrequenten Taktsignalen mit hoher Genauigkeit zu messen wird des Weiteren dazu eingesetzt eine Tastgradkorrektur mit geringem zusätzlichen Flächenbedarf an verschiedenen Punkten in den Transceivern durchzuführen. Neben der Kalibrierung des Abtastzeitpunktes wird auch eine Kalibrierung der Schwellspannung für nicht differenzielle Schnittstellen vorgestellt, welche auf der Tastgradmessung eines empfangenen Taktsignals basiert und die Kalibrierungszeit weiter reduziert.

Contents

List of Figures

List of Tables

List of Abbreviations

ABUJ	**A**periodic **B**ounded **U**ncorrelated **J**itter
AC	**A**lternating **C**urrent
ASIC	**A**pplication-**S**pecific **I**ntegrated **C**ircuit
BER	**B**it **E**rror **R**ate
BUJ	**B**ounded **U**ncorrelated **J**itter
CDF	**C**umulative **D**ensity **F**unction
CDR	**C**lock and **D**ata **R**ecovery
CML	**C**urrent-**M**ode **L**ogic
CPU	**C**entral **P**rocessing **U**nit
CSI	**C**urrent-**S**tarved **I**nverter
CTLE	**C**ontinuous **T**ime **L**inear **E**qualizer
DAC	**D**igital-to-**A**nalog **C**onverter
DC	**D**irect **C**urrent
DCA	**D**uty-**C**ycle **A**djuster
DCC	**D**uty-**C**ycle **C**orrector
DCD	**D**uty-**C**ycle **D**istortion
DCO	**D**igitally **C**ontrolled **O**scillator
DDJ	**D**ata **D**ependent **J**itter
DDR	**D**ouble **D**ata **R**ate
DES	**DES**erializer
DFE	**D**ecision **F**eedback **E**qualizer
DJ	**D**eterministic **J**itter
DLL	**D**elay-**L**ocked **L**oop
DRAM	**D**ynamic **R**andom **A**ccess **M**emory
ESD	**E**lectro**S**tatic **D**ischarge
FFE	**F**eed-**F**orward **E**qualizer
FIR	**F**inite **I**mpulse **R**esponse
FPGA	**F**ield **P**rogrammable **G**ate **A**rray
GPU	**G**raphics **P**rocessing **U**nit
HBM	**H**igh **B**andwidth **M**emory
HT	**H**yper**T**ransport
IC	**I**ntegrated **C**ircuit
IIR	**I**nfinite **I**mpulse **R**esponse
INL	**I**ntegral **N**on**L**inearity
ISI	**I**nter**S**ymbol **I**nterference
LDO	**L**ow-**D**rop**O**ut **R**egulator
LFSR	**L**inear **F**eedback **S**hift **R**egister

LUT	**L**ook**U**p **T**able
MIM	**M**etal-**I**nsulator-**M**etal
MOM	**M**etal-**O**xide-**M**etal
MOS	**M**etal-**O**xide-**S**emiconductor
MSB	**M**ost **S**ignificant **B**it
MUX	**MU**ltiple**X**er
NRZ	**N**on-**R**eturn-to-**Z**ero
OCD	**O**ff-**C**hip **D**river
ODT	**O**n-**D**ie **T**ermination
PAM	**P**ulse-**A**mplitude **M**odulation
PCB	**P**rinted **C**ircuit **B**oard
PDC	**P**hase-to-**D**igital **C**onverter
PDF	**P**robability **D**ensity **F**unction
PHY	**PHY**sical layer
PI	**P**hase **I**nterpolator
PJ	**P**eriodic **J**itter
PLL	**P**hase-**L**ocked **L**oop
PRBS	**P**seudo-**R**andom **B**it **S**equence
PVT	**P**rocess, **V**oltage, **T**emperature
QAM	**Q**uadrature **A**mplitude **M**odulation
QFN	**Q**uad **F**lat **N**on-lead
QPI	**Q**uick **P**ath **I**nterconnect
RJ	**R**andom **J**itter
RMS	**R**oot **M**ean **S**quare
SCI	**S**hunt-**C**apacitor **I**nverter
SST	**S**ource-**S**eries **T**erminated
TDC	**T**ime-to-**D**igital **C**onverter
TIE	**T**ime **I**nterval **E**rror
UI	**U**nit **I**nterval
UPI	**U**ltra **P**ath **I**nterconnect
USB	**U**niversal **S**erial **B**us

List of Symbols

The following table lists all common symbols used in equations. The symbols can appear with different indices.

Symbol	Unit	Definition
a	m^2	Area
BER	–	Bit error rate
C	F	Capacitance
$\mathrm{CDF}(t)$	–	Cumulative density function
DJ	s	Deterministic jitter
f	Hz	Frequency
$H(s)$	–	Transfer function in the s-domain
$H(z)$	–	Transfer function in the z-domain
I	A	Current
L	H	Inductance
n	–	Integer number
P	W	Power
$\mathrm{PDF}(t)$	–	Probability density function
R	Ω	Resistance
RJ	s	Random jitter
t	s	Time
T	s	Clock period
TJ	s	Total jitter
V	V	Voltage
W	m	Gate width of a MOSFET
Z	Ω	Impedance
ε	–	Permittivity
μ	non uniform	Mean of a gaussian distribution
σ	non uniform	Standard deviation of a gaussian distribution
ϕ	rad	Phase
ω	rad/s	Angular frequency

The following table lists important symbols with multiple appearances across different chapters of this thesis and special symbols that differ from the definitions in the previous table.

Symbol	Unit	Definition
C_{ox}	F	Oxide capacitance of a MOSFET
g_{m}	S	Small signal transconductance of a MOSFET
L_{g}	m	Gate length of a MOSFET
L_{min}	m	Minimum gate length of a MOSFET in a given CMOS process
l_{w}	m	Length of a bondwire
M	–	PDC scaling factor
M_{w1w2}	H	Mutual inductance between two neighboring bondwires
N	–	PDC resolution
N_{W}	–	Scaling factor for the gate width in the double transmission gate shunt-capacitor topology

Symbol	Unit	Definition
$P_1(\Delta t)$	–	Probability for detecting a logic one at an edge position Δt in the PDC sampling stage
$P_{1,1}(\Delta t)$	–	Probability for an edge position Δt to be the first to detect a logic one in the PDC sampling stage
Q_{BER}	–	Factor for the estimation of the peak-to-peak jitter from the standard deviation of a jitter histogram with a limited number of samples
$t_{\mathrm{jit}}^{\mathrm{cc}}$	s	Cycle-to-cycle jitter
$t_{\mathrm{jit}}^{\mathrm{per}}$	s	Period jitter
$t_{\mathrm{jit}}^{\mathrm{tie}}$	s	Time interval error (accumulated jitter)
V_{DD}	V	Positive supply voltage
V_{t}	V	Threshold voltage of a MOSFET
V_{th}	V	Voltage threshold of a single-ended data receiver
Z_{11}	Ω	Impedance parameter (Z-parameter) for the reflected wave at port 1 of a two-port network

Part I

Motivation and Context for All-Digital Clock Calibration in High-Speed I/O Links

1 Introduction

1.1 Motivation

The ongoing shrinking of transistors in semiconductor technologies enabled a constant performance increase of modern computer systems over the last decades. Smaller process nodes allow for faster digital circuits and a higher integration density in integrated circuits (IC). The trend of doubling the number of components per IC every two years was stated by G. Moore and is known as “Moore’s Law”. Although Moore is already forecasting the end of this trend [7], there are still efforts to continue the shrinking of transistors down to the physical limits with technology nodes as small as 7 nm [8]. However, parallelization becomes more and more important to achieve further improvements in processing speed as the maximum clock frequency of digital ICs is stagnating. Figure 1.1 shows the development of the number of transistors, the clock frequency and the number of cores in microprocessors over the last decades [1]. It can be seen that as soon as the clock frequency reached its limits the number of cores started to increase in order to further improve the processor performance.

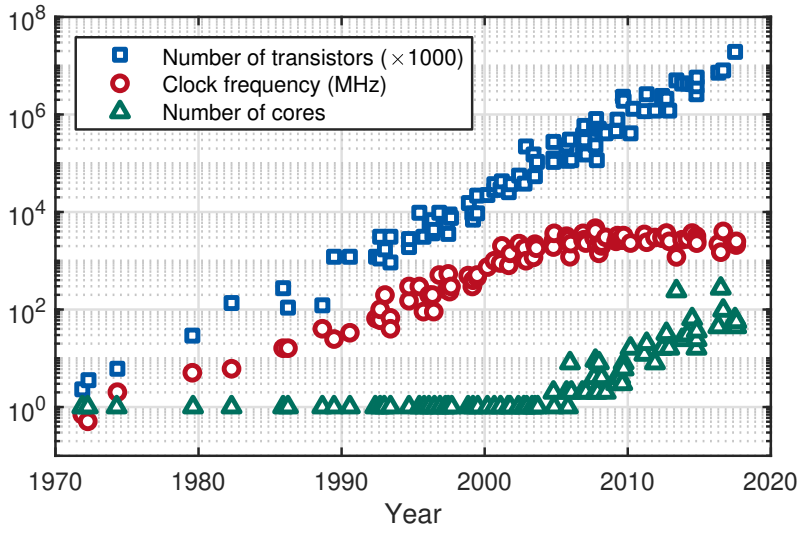
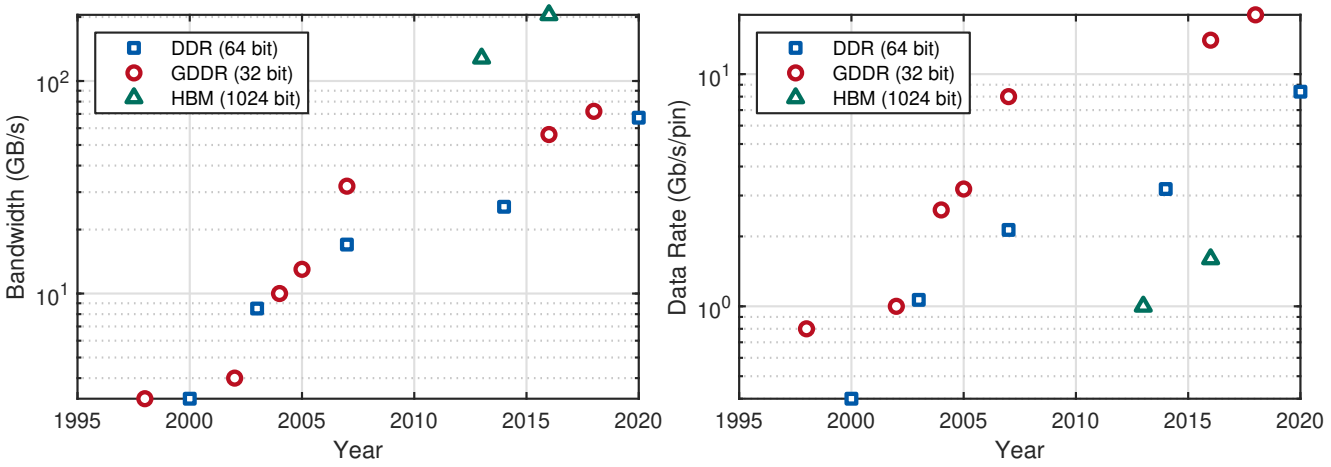


Figure 1.1: Development of the number of transistors, the clock frequency and the number of cores in microprocessors over the last decades [1].

An important aspect that is not obvious from the data in Figure 1.1 is that the bandwidth of memory interfaces is also a limiting factor in computer systems. A fast data processing is only possible if a fast access to the data storage is provided. Today’s high-performance central processing units (CPU) and graphics processing units (GPU) have a dedicated dynamic random access memory (DRAM) of multiple gigabytes. Due to the large chip size and the special requirements to the semiconductor process the DRAM is fabricated as a separate IC and is connected via high-speed I/O links to the CPU or GPU. High-bandwidth memory interfaces have always been relying on parallelization, but there has been no increase of the bus width over the generations of main memory interface standards for CPUs and GPUs. The most common standards are double data rate (DDR) with a bus width of 64, graphics double data rate (GDDR) with a bus width of 32 and high-bandwidth memory (HBM) with a bus width of 1024. A bandwidth growth over the generations (Figure 1.2 (a)) is achieved by increasing the per pin data rate as shown in Figure 1.2 (b), but the progression is not sufficient to keep up with the processor speeds.



(a) Bandwidth.

(b) Data rate.

Figure 1.2: Maximum bandwidth (a) and per pin data rate (b) of the most common high-bandwidth memory interface standards.

The data rate of wire-linked high-speed interfaces is mainly limited by the transmission channel characteristics that degrade the signal integrity of the I/O link with increasing frequency. The interface standards have tackled this problem by improving the packages of processors and DRAMs to reduce signal reflections and by improving the transceiver circuits in order to compensate the negative channel characteristics. Today a variety of circuit techniques and calibration schemes is employed to reach data rates of 18 Gb/s and beyond. Although successful wire-linked data transmission at 112 Gb/s has already been demonstrated [9], commercial parallel high-speed interfaces currently cannot reach such high data rates. The key limitation at this point is the power consumption of the IC. CPUs and GPUs have a limited power budget due to the available cooling concepts and therefore need to compromise between core and interface power consumption. In mobile devices this is an even bigger issue since the cooling of the ICs is more difficult and an increased power consumption also reduces the battery runtime.

Another challenge at high frequencies is the determination of the ideal sampling time of the data. With increasing data rate, I/O links become more sensitive to timing skew between clock and data due to process and trace length mismatch. Recent memory interface standards (DDR4, GDDR5/5X/6) implement a per pin deskewing using local clock phase adjustment. The clock phase at each transceiver is calibrated prior to the data transmission for both, the read and the write operation (link training). Since the delay of the transceiver circuits also depends on the supply voltage and the ambient temperature, the calibration is repeated frequently in order to track variations of the operating conditions. The state-of-the-art methods for the link training are limited in accuracy and take a significant amount of time in which the interface is not available for data transmission. Optimizations to the link training and the internal calibrations of parallel high-speed I/O links have the potential to improve the link data rate, reduce the calibration time and reduce the power consumption of the interface.

1.2 Research Scope and Design Objective

This thesis presents a new approach to the phase and duty-cycle measurement of high-speed clocks and investigates its application in the link training and the internal clock calibration of high-speed I/O links. The concept is applied in four major calibration topics:

- **Phase detection in multiphase delay-locked loops** for area- and power-efficient solutions in parallel high-speed I/O links.
- **Duty-cycle detection** for precise duty-cycle correction circuits with low area overhead.

- **Timing center calibration** of bidirectional source-synchronous parallel I/O links with improved accuracy and reduced link training duration.
- **Voltage threshold calibration** of single-ended high-speed I/O links based on duty-cycle measurement for reduced link training duration.

Three custom application-specific integrated circuits (ASIC) have been designed to verify the concepts and analyze their performance. The main objective of the implementations is to improve the accuracy and reduce the power consumption of the calibration circuits. A second target is to minimize the area consumption of the circuits by using as few analog components as possible and implementing major parts in the digital domain. This constraint follows the trend towards "all-digital" implementations that can be seen in many different mixed-signal applications like phase-locked-loops, delay-locked-loops and duty-cycle correctors. Digital circuits scale better with technology than analog circuits and therefore provide a continuous area reduction with shrinking process nodes.

A source-synchronous I/O link based on the GDDR5/5X/6 standard is chosen as an example application, as it represents the highest performing commercially successful memory interface at the time of publication. The bidirectional asymmetric structure of the interface also reflects the constraint of moving most of the calibration circuits to one side of the memory controller IC in order to minimize the cost of the DRAM ICs.

1.3 Thesis Outline

The structure of this thesis is based on the calibration sequence of typical source-synchronous parallel high-speed I/O links illustrated in Figure 1.3 and is divided into three parts.

Part I: Motivation and Context for All-Digital Clock Calibration in High-Speed I/O Links

After the motivation and the research scope are presented in chapter 1, chapter 2 gives an overview of high-speed I/O links and common interface standards with a special focus on the bidirectional asymmetric source-synchronous interface architecture. The modeling techniques used in this work for the simulation of a chip-to-chip interface are presented and the considered metrics for the analysis of high-speed signals are elaborated. Finally, state-of-the-art methods for the calibration of source-synchronous high-speed

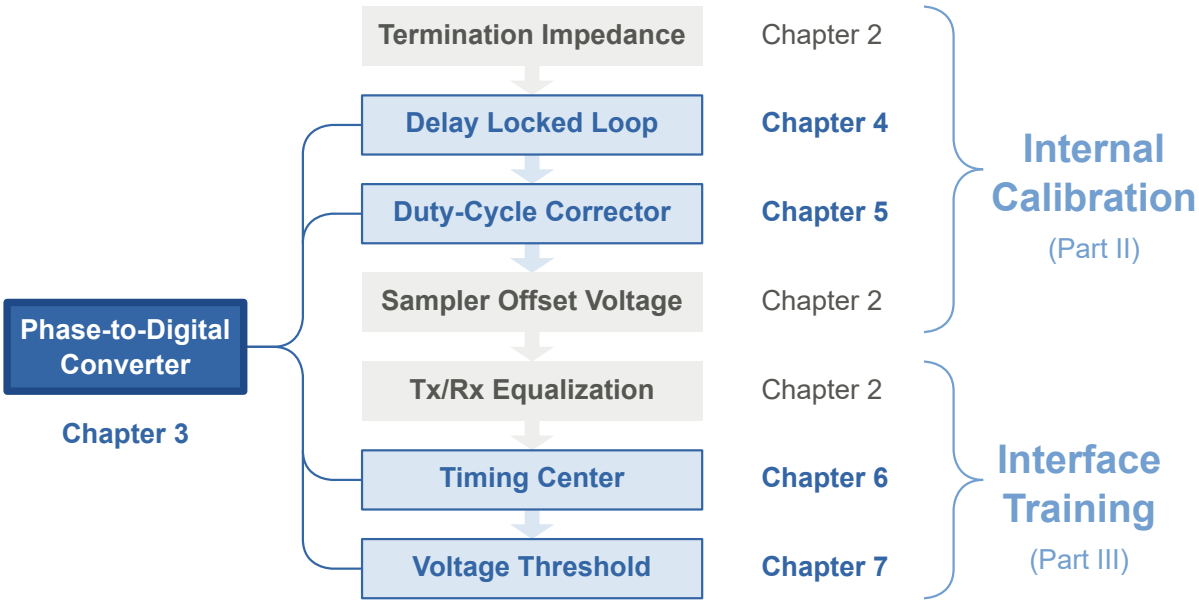


Figure 1.3: Calibration sequence of a typical source-synchronous parallel high-speed I/O link with PDC enhanced calibration steps annotated with the corresponding chapters of this thesis.

interfaces are explained together with implementation examples. The calibration steps are divided into internal calibrations and interface trainings as shown in Figure 1.3.

Part II: Design of Low-Power All-Digital High-Speed Clock Generation and Measurement Circuits

This work presents enhancements to four of the seven calibration steps through the use of a new phase-to-digital converter (PDC). Implementation details, as well as an in-depth analysis of the behavior of the PDC are discussed in chapter 3. The PDC can be fully synthesized from a standard-cell library and is verified on a field programmable gate array (FPGA) prior to the implementation in an ASIC.

Chapter 4 presents a new approach to the design of all-digital multiphase delay-locked loops (DLL). The use of a PDC as a linear phase detector and area efficient shunt-capacitors simplifies the loop filter design and minimizes the active area of DLLs. A second, current-starved inverter based multiphase DLL design is presented that also brings the jitter performance and power efficiency to a similar level as the best reported state-of-the-art designs.

In chapter 5 a duty-cycle corrector for high-speed I/O links using the PDC as a duty-cycle detector is described. The design provides a linear duty-cycle adjustment with a fine resolution over a wide frequency range and can share the PDC with other clock calibration circuits in the transceiver ASIC.

Part III: Phase-to-Digital Converter Based Fast Link Training of Asymmetric Bidirectional Source-Synchronous I/O Links

The timing center calibration based on a PDC measurement of the phase between the sampling clock and the received data is discussed in chapter 6. A differential 6.4 Gb/s/pin transceiver ASIC demonstrates the reduced calibration time of the proposed method compared to the conventional passing window approach. An additional INL compensation with optional duty-cycle correction also improves the accuracy of the timing center calibration and allows the use of lower power circuits for the phase generation that would lead to a significant timing error with the conventional passing window approach.

Chapter 7 investigates a method to perform the voltage threshold calibration of single-ended high-speed I/O links by measuring the duty-cycle of a received clock pattern. A single-ended 6.4 Gb/s/pin transceiver ASIC implements all concepts of the preceding chapters together with the proposed voltage threshold calibration method and demonstrates the feasibility of a fast 2 D eye detection.

In chapter 8 the research results are summarized and an outlook is given on possible improvements to the presented designs and calibration methods, as well as further applications of the PDC.

2 High-Speed I/O Links

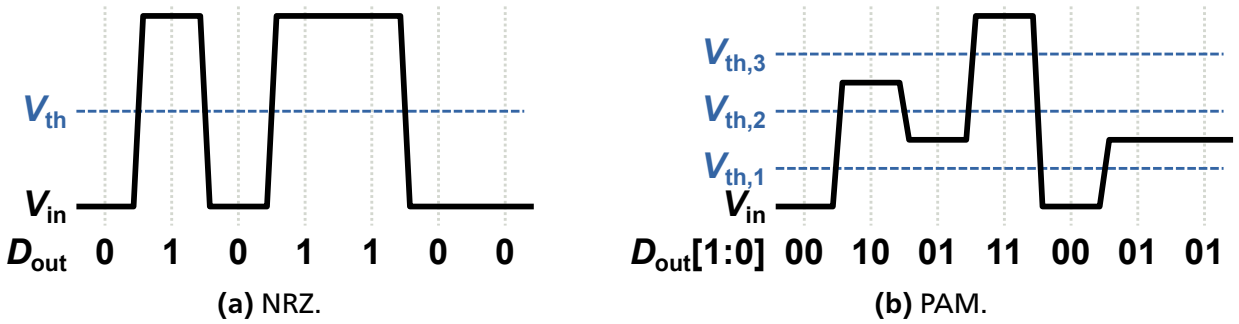
High-speed I/O links usually come into play when large amounts of data must be transferred between two or more integrated circuits in a very short time. The enormous amount of different applications that require high-speed data transmission have led to a variety of interface types with individual trade-offs optimized for each application. The intention of this chapter is to give an overview of the various types of high-speed interfaces (Section 2.1), explain the modeling techniques used in this work to simulate the behavior of high-speed I/O links (Section 2.3) and present commonly used techniques to (partially) overcome the effects that limit the data rate (Section 2.5). Most of those techniques are used in the two interface training test chips presented in Part III of this thesis.

2.1 Overview and Applications of High-Speed I/O Links

The total data transfer rate of an I/O link between integrated circuits is called the *bandwidth*. Since the bandwidth requirement of an application often surpasses the maximum possible data rate of a single serial data link, or the required power for such a single link would exceed the power budget, many high-bandwidth interfaces consist of multiple parallel data links. In this context the term *speed* relates to the data rate of a single data link within the interface. The term *high-speed* does not define a specific speed range. Instead it will be used in the scope of this thesis for interfaces that require one or more of the link training procedures presented in Section 2.5 for a reliable data transmission.

A first classification of high-speed I/O links can be done by the medium of transport used for the data transfer. The three main categories are wire-linked, wireless and optical. As the data is generated inside the integrated circuits as electrical signals it is an obvious approach to transfer them directly in form of a wire-linked data transmission. This type of link is usually quite reliable but limits the flexibility in the application due to the interconnect with an electrical conductor (cable or printed circuit board (PCB)). Also the distance between transmitter and receiver is limited at high data-rates by the attenuation of the electrical channel. In wireless communication systems the electrical signals are converted to electro-magnetic waves. The information is usually modulated onto a carrier signal by varying one or more properties of the periodic carrier waveform. Typical examples are frequency, phase or amplitude modulation or combinations of them like quadrature amplitude modulation (QAM). Typical optical high-speed communication systems transmit data with an optical transmitter (light-emitting diode or laser diode) through an optical fiber. The range of data transmission through optical fibers is usually larger than through electrical wires due to a lower attenuation of the optical channel. Therefore this type of I/O link is used for long distance data transmission in the range of hundreds or thousands of kilometers. This work focuses only on wire-linked high-speed I/O links and therefore does not go deeper into the details of wireless and optical communication.

In wire-linked communication interfaces the non-return-to-zero (NRZ) coding is the most common modulation technique. The signal has only two states (see Figure 2.1 (a)), high and low, which allows the use of a simple comparator to detect the signal state in the receiver front-end. A more complex coding that is used to increase the data rate without increasing the highest frequency of the data signal is pulse-amplitude modulation (PAM). With PAM the data signal has more than two voltage levels and therefore a single pulse contains more than one bit of information (see Figure 2.1 (b)). In the following only NRZ signals will be considered, but the concepts presented in this thesis can be in general also applied for PAM coded interfaces.



(a) NRZ. **(b)** PAM.

Figure 2.1: Amplitude coding schemes of wire-linked communication interfaces.

2.1.1 Source-Synchronous I/O Links

Wire-linked high-speed interfaces can be divided into two categories. In source-synchronous I/O links the clock that is used to generate the data stream is transmitted along with data signal (see Figure 2.2). Ideally both signal paths have equal flight times, so the timing relationship between clock and data is maintained and the data can be sampled in the receiver for retiming using the transmitted clock signal without further adjustment. In reality the flight times are different due to local process, voltage and temperature (PVT) variations and unequal trace lengths. Above a certain data rate a timing center calibration is needed to ensure proper sampling of the data signal. For multiple parallel data links only one clock signal is required, so the drawback of the additional pins for the clock is acceptable. When the frequency of the transmitted clock signal is equal to the data rate of a single data link the interface is called a single data rate (SDR) interface. In this case the data is captured on the rising (or falling) edges of the clock signal (see Figure 2.3 (a)). The drawback of SDR signaling is that the frequency of the clock is twice the maximum frequency of the data signal and therefore the channel requirements and the power consumption of the clock transmitter are higher than for the data channels. In order to balance the requirements of the clock and data links the double data rate (DDR) signaling has been introduced. The data is sampled at both, the rising and falling edges of the clock, resulting in two bits of information per clock cycle (see Figure 2.3 (b)). Some very high-speed interfaces use a clock signal that has a frequency of less than half the data rate. A common type is quad data rate (QDR) signaling with a clock frequency of one fourth of the data rate (see Figure 2.3 (c)). QDR requires either a frequency doubler or a quadrature phase generator in the receiver to generate clock edges for every bit of the data stream. Examples for source-synchronous interface standards with high bandwidths are given in Table 2.1. The bandwidths given in the table are gross bandwidths including any protocol overhead. The DDRx memory interface standards (e.g. DDR4) are used for the main memory of desktop computers. The standard allows to connect multiple modules in parallel. Together with the sockets for CPU and memory modules there are many impedance discontinuities in the signal paths limiting the link speed of this type of interface. By the use of a 64 bit wide bus it can still achieve a bandwidth of up to 25.6 GB/s. AMD's HyperTransport (HT) standard [10] as well as Intel's Quick Path and Ultra Path Interconnect

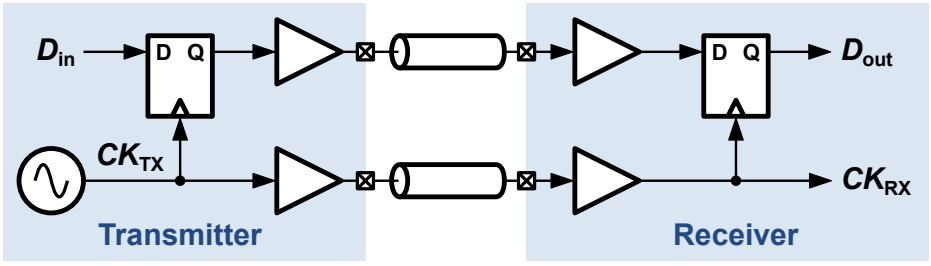
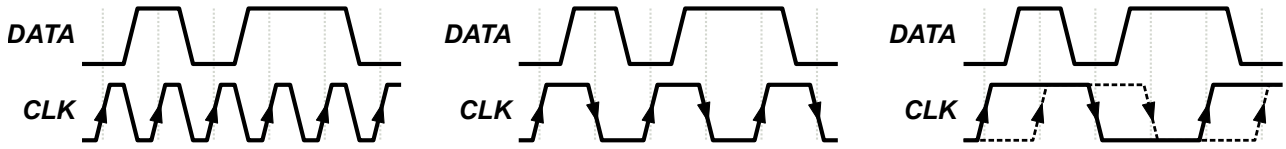


Figure 2.2: Block diagram of a typical source-synchronous I/O link.



(a) SDR. **(b)** DDR. **(c)** QDR.

Figure 2.3: Clock signaling schemes of source-synchronous I/O links.

Table 2.1: Performance and specifications of selected wire-linked source-synchronous high-speed interface standards.

Standard	Link Speed (Gb/s/pin)	Bus Width	Bandwidth (GB/s)
DDR3	0.8 - 2.133	64	6.4 - 17
DDR4	1.6 - 3.2	64	12.8 - 25.6
HyperTransport (HT) 3.0	5.2	32	20.8
HyperTransport (HT) 3.1	6.4	32	25.6
Quick Path Interconnect (QPI)	4.8 - 9.6	20	12 - 24
Ultra Path Interconnect (UPI)	9.6 - 10.4	20	24 - 26
GDDR5	3.6 - 8	32	14 - 32
GDDR5X	10 - 14	32	40 - 56
GDDR6	14 - 18	32	56 - 72
HBM1	1	1024	128
HBM2	1.6	1024	204.8

(QPI/UPI) standards [11] are intended for fast interconnects between CPUs in a multi-processor system. In contrast to memory interfaces, the HT and QPI/UPI standards have only two bus nodes with equal properties (e.g. two processors). The limitation for two bus nodes improves the channel characteristics and therefore enables higher data rates. For the memory of GPUs the GDDRx standards (e.g. GDDR5 [12]) have been the dominating interface standards for a long time. Since only one memory chip is connected to each channel and GPU and DRAM are mounted on the same PCB the interconnects provide a much better signal quality than DDRx memory interfaces. Therefore the link speed is much higher with up to 18 Gb/s for GDDR6 [13]. This allows bandwidths on a 32 bit wide bus of up to 72 GB/s. Recently a new interface standard for memories has been introduced which is now used on many GPUs by AMD. The HBM standard uses massive parallelization with 1024 data links and achieves the highest total bandwidth of the presented interface standards with up to 204.8 GB/s [14]. This is only possible because the interconnect between controller and memory is not realized using a PCB. Instead the dies are directly mounted on a silicon interposer, which provides very short and fine pitched metal interconnects, but is significantly more expensive than a PCB.

2.1.2 Source-Asynchronous I/O Links

Source-asynchronous I/O links do not require the transmission of the source's clock signal. Instead the clock is recovered from the data stream inside the receiver using a clock recovery circuit (see Figure 2.4). Thereby the number of interconnects is minimized and the need for trace length matching is avoided.

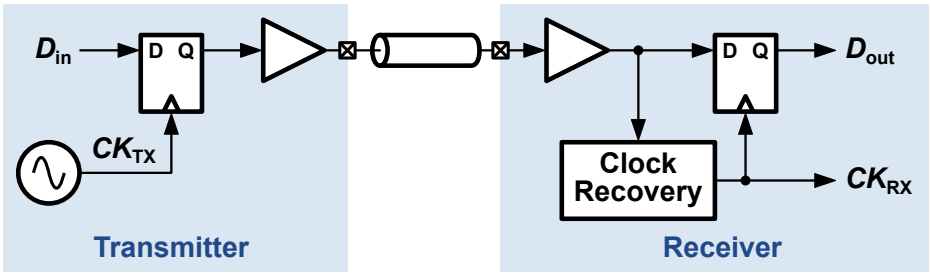


Figure 2.4: Block diagram of a typical source-asynchronous I/O link with clock and data recovery.

Table 2.2: Performance and specifications of selected wire-linked source-asynchronous high-speed interface standards.

Standard	Link Speed (Gb/s/pin)	Bus Width	Bandwidth (GB/s)
USB 3.0	5	1	0.5
USB 3.1	10	1	1.25
USB 3.2	10	2	2.5
Thunderbolt 2	10	2	2.5
Thunderbolt 3	20	2	5
NVLink 1.0	20	8	20
NVLink 2.0	25	8	25
PCIe 4.0	15.752	1 - 16	1.969 - 31.508
PCIe 5.0	31.504	1 - 16	3.938 - 63.015

The process of extracting the clock signal from an asynchronous data stream recovering the data is called *clock and data recovery* (CDR). Different CDR strategies for the extraction of the source clock from the data signal have been proposed which can be divided into open-loop and closed-loop architectures [15]. Both concepts require a coding of the data that ensures a sufficient number of transitions in the data signal for the circuits to operate correctly. An example for such a coding scheme is the 8b/10b coding, which maps each 8 bit data word to a 10 bit word adding 20 % of overhead, thus reducing the usable bandwidth of the interface [16]. In return, the number of ones and zeros in the resulting data stream is equal, by which AC coupling of the data link is enabled. Another CDR approach that does not require clock recovery is oversampling CDR. The data is sampled with multiple equally spaced phases of a clock in the receiver with approximately the same frequency as the transmitter clock. The correct bit value is obtained by a majority voting of the bits detected by the phase shifted samplers. Table 2.2 shows examples of source-asynchronous high-speed interface standards with CDR. Again all bandwidths are gross bandwidths including protocol overhead. The Universal Serial Bus (USB) [17] and Thunderbolt [18] standards are mainly used for short range cable connection of consumer electronics. In order to keep the cost of the cables low, only one or two data links are used with rather low data rates. Interface standards for PCB and backplane interconnects (e.g. NVLink [19] and PCI Express (PCIe) [20]) achieve higher link speeds and use more parallel data links (up to 16 with PCIe). Many other commercial CDR interfaces are based on the physical layer (PHY) of PCIe (e.g. Thunderbolt, SATA and M.2).

From the presented examples of source-synchronous and source-asynchronous interface standards it can be seen that source-synchronous interfaces are mainly used for short interconnects within a computing system (e.g. CPU to memory or CPU to CPU). The short distance and the integration on a PCB or in a multi-die package allow the use of many parallel I/O links. An additional clock signal can be easily transmitted along with the data to simplify the receiver design and minimize the power consumption. Source-asynchronous interfaces are typically used for I/O links between different systems (e.g. CPU to GPU or computer to periphery). Especially interfaces with a cable connection try to minimize the number of interconnects in order to keep the cables thin and cheap. Therefore only a few parallel data links are used and an additional clock signal is omitted. The link speeds are higher than in source-synchronous interfaces to compensate for the reduced parallelization.

2.2 Bidirectional Source-Synchronous Parallel I/O Links

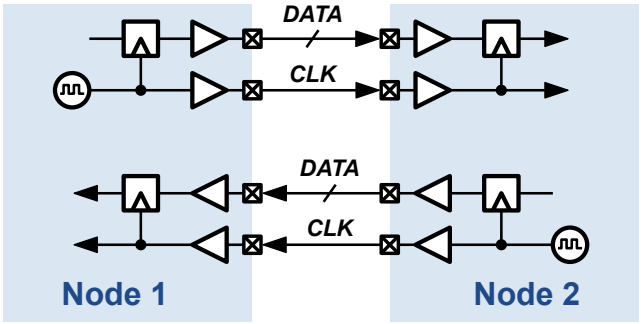
Bidirectional architectures found in the literature can be divided into three different categories, dual unidirectional, symmetric bidirectional and asymmetric bidirectional interfaces. The terms *symmetric* and *asymmetric* in this context are used to describe the implemented data sampling methods at the nodes of bidirectional interfaces. In fact, there are often many other asymmetries in both, symmetric and asymmetric interfaces. In most cases the interface nodes are fabricated with different processes from different manufacturers. The circuit topologies are usually not part of the interface standard and may therefore also differ. Certain circuits like equalizers or offset calibration may not be implemented on all nodes. Circuits for timing adjustment are frequently moved to a master device to reduce the complexity of the slave device(s) (e.g. in DRAM interfaces).

- **Dual unidirectional** interfaces consist of independent unidirectional source-synchronous interfaces for each direction with separate clock and data signals (Figure 2.5 (a)). Due to the independence of the two interfaces a simultaneous bidirectional data transmission is possible. The link timing can be matched separately for each direction. Examples are the HT, QPI and UPI interface standards.

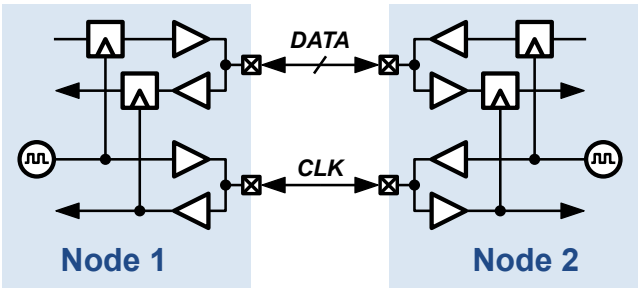
- **Symmetric bidirectional** interfaces are based on bidirectional clock and data interconnects (Figure 2.5 (b)). Each clock and data signal can only be operated in one direction at a time, so a simultaneous bidirectional data transmission of a bundle is not possible. Due to the symmetry of the interface and independent transmitter and receiver circuits on each side of the I/O Link the link timing is the same for both directions. Examples are the DDR3 and DDR4 interface standards.

- **Asymmetric bidirectional** interfaces consist of an unidirectional clock and bidirectional data interconnects (Figure 2.5 (c)). As in the symmetric case, each data signal can only be operated in one direction at a time, but within a bundle the data links do not necessarily transmit in the same directions. Practical implementations do not usually make use of this feature, but it is mentioned for completeness. The link timing for the direction from node 1 to node 2 is matched in the same way as with a unidirectional interface. For the direction from node 2 to node 1 the clock on the receiving side is not aligned to the data by default. A phase recovery is needed to adjust the timing center for the received data. This is sometimes also called a CDR, although the clock signal is not being extracted from the data stream using a clock recovery circuit as in Figure 2.4. Examples are the GDDR5, GDDR5X and GDDR6 interface standards.

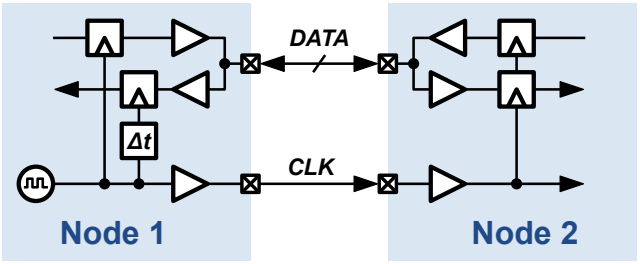
Source-synchronous interfaces can consist of one ore more *bundles*, where a *bundle* is the combination of a clock signal and the data links associated with that clock. Trace length matching is only required within a bundle and not between different bundles. This work focuses on asymmetric bidirectional interfaces since they have the highest requirements to the link training and the developed concepts can be applied for symmetric bidirectional and dual unidirectional interfaces as well.



(a) Dual unidirectional source-synchronous I/O link.



(b) Symmetric bidirectional source-synchronous I/O link.



(c) Asymmetric bidirectional source-synchronous I/O link.

Figure 2.5: Block diagrams of dual unidirectional, symmetric bidirectional and asymmetric bidirectional source-synchronous I/O links.

2.3 Channel Modeling and Simulation Setup

This section describes the modeling techniques applied for the simulation of the transmission channel in the context of this thesis. Quad Flat Non-lead (QFN) packages are used for the ASICs in combination with four layer PCBs. This type of package is often used for high-frequency ASICs with a low to medium number of I/Os. The ASIC is mounted on a center pad using conductive glue to provide a good connection of the substrate to ground. The pads of the ASIC are connected to the package pins using gold bondwires. As a Surface-Mounted Device (SMD) the QFN package is soldered on the top layer of the PCB and all high-speed signals are routed solely on the top layer in form of microstrip transmission lines with a ground plane underneath the signal trace. Figure 2.6 shows a cross section of the high-speed PCB interconnect and the modeling techniques used for the different sections of the transmission channel.

If not otherwise noted, the circuits inside the ASIC are simulated using schematics with annotated parasitic resistors and capacitors extracted from the layout using the software tool Assura QRC. An equivalent circuit model (Figure 2.7) for the bondwire including package parasitics, based on the models presented in [5] and [21], is used for the high-speed signal I/Os and the power and ground connections. For early simulations, the capacitor C_{pg} models the capacitances of the pad and the electrostatic discharge (ESD) protection diodes with a typical value of $\approx 1\,\mathrm{pF}$. After layout creation the extracted layout views of the pads and diodes are used instead. L_{W} and R_{W} model the inductance and resistance of the bondwire. The inductive coupling between bondwire and lead is modeled by the mutual inductances M_{L1W} and M_{L2W}. In order to adapt the models to the package and ASIC geometries, the length of the wire l_{w} is left as a parameter to calculate the component values of L_{W} and R_{W}. Table 7 in [21] with measured inductance values for the typical range of wire lengths between 0.5 mm and 5 mm is used to fit the linear equation

$$L_{\mathrm{W}} = 1.1538\,\frac{\mathrm{nH}}{\mathrm{mm}}\,l_{\mathrm{w}} - 0.3739\,\mathrm{nH}, \tag{2.1}$$

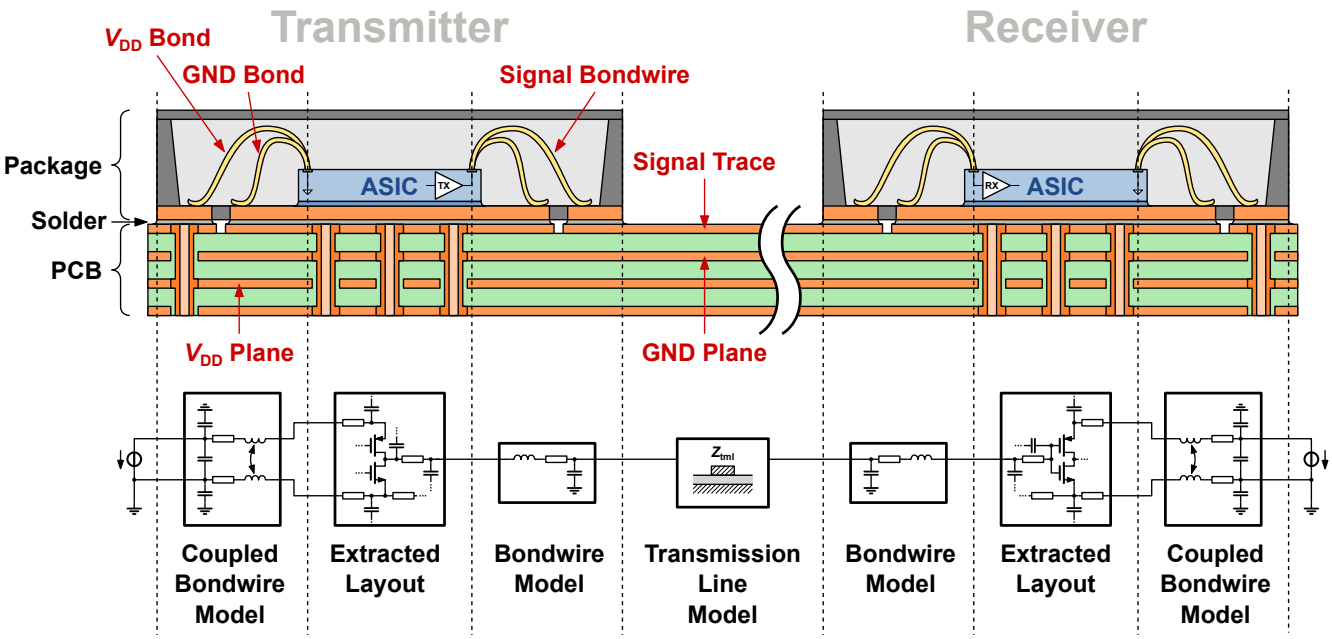


Figure 2.6: Cross section of a typical high-speed PCB interconnect and modeling of the different sections.

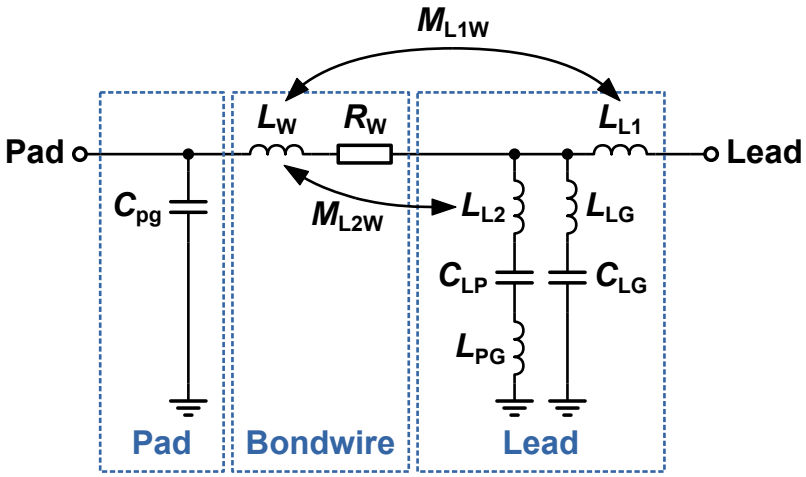


Figure 2.7: Equivalent circuit model of the bondwire and package parasitics of a QFN package.

for the wire inductance L_W. The resistance of the bondwire is estimated by the equation

$$R_W = 0.1 \frac{\Omega}{mm} l_w. \tag{2.2}$$

The lead parasitics are independent of the wire length and are modeled by L_{L1}, L_{LG}, M_{L1W} and C_{LG} for the path from the lead to the ground plane, as well as L_{L2}, M_{L2W}, C_{LP} and L_{PG} for the path from the lead to the center pad of the package. The values are taken from Table 1 in [5] and are listed in Table 2.3.

Apart from the model for a single bondwire, three additional models for the simulation of two, three and four coupled bondwires are implemented to simulate crosstalk effects of neighboring I/Os. These models include additional mutual capacitors and inductances between the single wire model of each bondwire. Figure 2.8 shows the equivalent circuit model for two coupled bondwires. The mutual inductance M_{W1W2} to an adjacent wire is modeled by the linear equation

$$M_{W1W2} = 0.7467 \frac{\text{nH}}{\text{mm}} l_w - 0.3506\,\text{nH} \tag{2.3}$$

Table 2.3: Component values of the parasitic elements in the equivalent circuit model for the lead of a QFN package taken from Table 1 in [5].

L_{L1}	L_{LG}	C_{LG}	L_{L2}	C_{LP}	L_{PG}	M_{L1W}	M_{L2W}
0.15 nH	0.01 nH	38.3 fF	0.24 nH	106.5 fF	0.1 nH	0.008 nH	0.496 nH

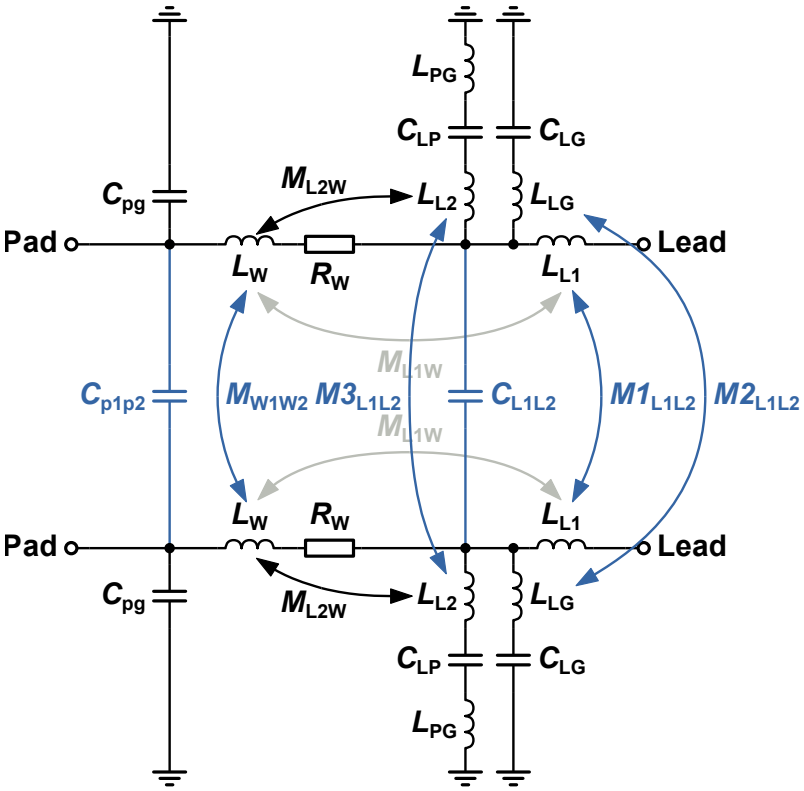


Figure 2.8: Equivalent circuit model of two coupled bondwires with package parasitics of a QFN package.

fitted to the measured values in Table 7 in [21]. The models for three and four bondwires also include the mutual inductance M_{W1W3} of non-adjacent wires

$$M_{W1W3} = 0.6098 \frac{\text{nH}}{\text{mm}} l_w - 0.3319\,\text{nH}. \tag{2.4}$$

The values for the mutual components of the leads are taken from Table 1 in [5] as listed in Table 2.4. Since the high-frequency characteristics strongly depend on the geometry of the bond, these models do not provide an accurate estimation of the actual performance. However, they help to understand and overcome the negative effects of the unavoidable impedance discontinuity introduced by the bondwire.

The PCB trace is simulated using the components "stackup" and "nclin" from the "rfTlineLib" library in Cadence Virtuoso. These components model a microstrip transmission line based on the geometry of the PCB trace. The "stackup" component defines the layer stack of the PCB in terms of signal trace and ground plane thickness h_s and h_g, as well as thickness h_d and relative permittivity ε_r of the dielectric layer (see Figure 2.9). The "nclin" component creates a PCB trace based on the layers defined in the "stackup" component and the width w_s and length l_s of the signal trace. The characteristic impedance Z_{tml} of a

Table 2.4: Component values of the mutual parasitic elements in the equivalent circuit model for the coupling of two neighboring leads of a QFN package taken from Table 1 in [5].

C_{L1L2}	$M1_{\mathrm{L1L2}}$	$M2_{\mathrm{L1L2}}$	$M3_{\mathrm{L1L2}}$
20.2 fF	0.008 nH	0.008 nH	0.008 nH

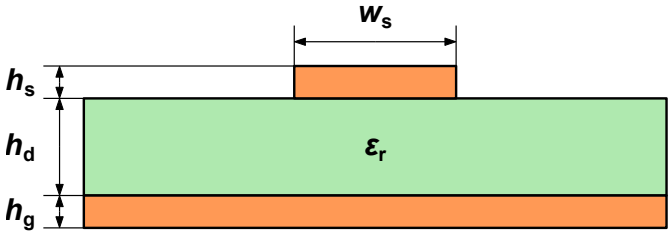


Figure 2.9: Relevant parameters of the PCB layer stack for the calculation of the characteristic impedance Z_{tml} of a microstrip transmission line.

microstrip transmission line can be estimated from ε_{r}, w_{s} and h_{d} and is independent of the length l_{s} of the signal trace. An approximation for Z_{tml} in the case of $\frac{w_{\mathrm{s}}}{h_{\mathrm{d}}} \geq 1$ is given in [22] by

$$Z_{\mathrm{tml}} = \frac{Z_0}{\sqrt{\varepsilon_{\mathrm{eff}}}\left[\frac{w_{\mathrm{s}}}{h_{\mathrm{d}}} + 1.393 + 0.667 \ln\left(\frac{w_{\mathrm{s}}}{h_{\mathrm{d}}} + 1.444\right)\right]}, \tag{2.5}$$

with the impedance of free space $Z_0 = \sqrt{\frac{\mu_0}{\varepsilon_0}} = 376.73\,\Omega$ and the effective dielectric constant

$$\epsilon_{\mathrm{eff}} = \frac{\varepsilon_{\mathrm{r}} + 1}{2} + \frac{\varepsilon_{\mathrm{r}} - 1}{2}\left(\frac{1}{\sqrt{1 + 12\,\frac{h_{\mathrm{d}}}{w_{\mathrm{s}}}}}\right). \tag{2.6}$$

Using these equations the required width of the PCB trace w_{s} can be calculated for a given PCB layer stack and a target characteristic impedance Z_{tml}. In the scope of this work all transmission lines are designed for $Z_{\mathrm{tml}} = 50\,\Omega$ for compatibility with the measurement equipment.

2.4 Definition of Considered Metrics

A common method for the analysis of the signal quality in a high-speed interface is to measure or simulate the *eye diagram* of a data link. The eye diagram is obtained by overlaying the signal transients of each *unit interval* (UI) of the data stream as shown in Figure 2.10. One UI is the duration of a single bit in the data stream defined as

$$1\,\mathrm{UI} = T_{\mathrm{bit}} = \frac{1}{f_{\mathrm{bit}}}, \tag{2.7}$$

with the data rate of the link f_{bit}. UI is often used as a unit of measure for eye diagrams and bathtub curves instead of time. The independence of the frequency makes it a useful measure to compare the quality of transceiver circuits that operate at different data rates. Figure 2.11 shows an example eye diagram measured on a differential high-speed link at 6.4 Gb/s and the typical measures to determine the signal quality. The *data eye* describes the area in the center of the eye which is not crossed by any signal transients. Anywhere inside this area the data can be sampled correctly without bit errors. The maximum distance between the innermost transitions at −0.5 UI and +0.5 UI is defined as the *horizontal*

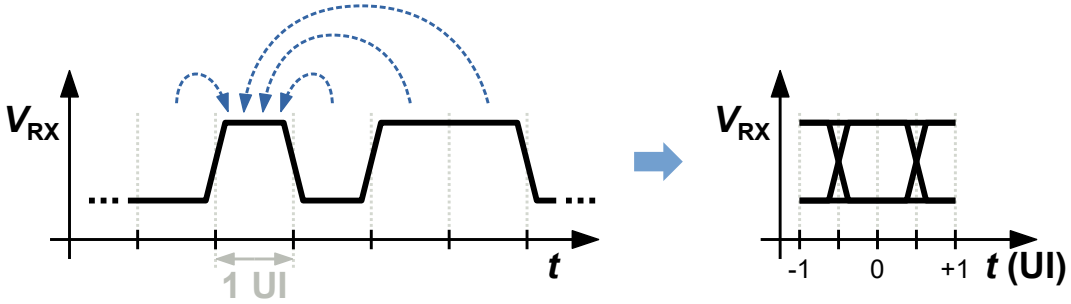


Figure 2.10: Illustration of how to obtain an eye diagram from a transient data signal.

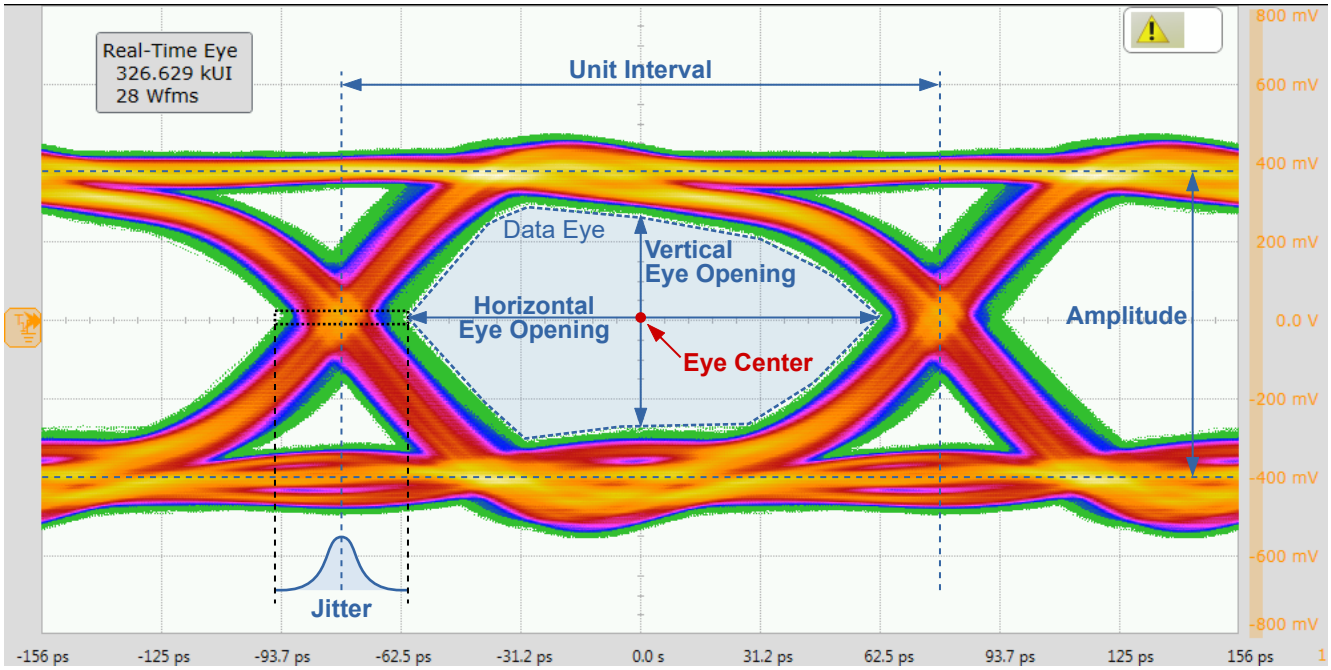


Figure 2.11: Example eye diagram of a 6.4 Gb/s differential I/O link annotated with typical measures.

eye opening or *clock phase margin*. The distance between the worst case high and low voltage levels at the timing center of the eye is defined as the *vertical eye opening* or *voltage margin*. A common way to constrain the required signal quality for high-speed transceivers is to define minimum values for clock phase margin and voltage margin. Some interface standards use polygons instead to constrain a minimum area for the data eye. The voltage threshold and the sampling time of the data receiver are ideally located at the *eye center* in order to maximize the robustness against voltage and timing variations due to external influences. The *jitter* of the data signal is measured by taking a histogram of the signal crossings at the voltage threshold V_{th}. In the given example of a differential signal $V_{\text{th}} = 0\,\text{V}$.

Different jitter metrics exist for the characterization of pure clock signals that are obtained by comparing the clock signal to an ideal clock as illustrated in Figure 2.12. Frequently used metrics are:

- **Period jitter** is the deviation

$$t_{\text{jit},i}^{\text{per}} = T_i - T_{\text{ideal}} \tag{2.8}$$

  of each clock period T_i from the ideal clock period T_{ideal}.

- **Cycle-to-cycle jitter** is the deviation

$$t_{\text{jit},i}^{\text{cc}} = T_{i+1} - T_i \tag{2.9}$$

  of the period T_i of adjacent clock cycles.

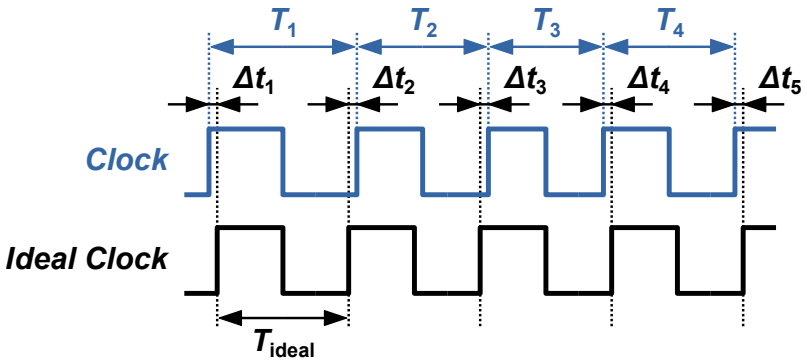


Figure 2.12: Illustration of the metrics used to define different types of clock jitter.

- **Time interval error**, also called **accumulated jitter**, is the deviation

$$t_{\text{jit},i}^{\text{tie}} = \Delta t_i \tag{2.10}$$

  of the clock edge position from the ideal clock edge position.

The time interval error (TIE) is the most common metric for the characterization of clock signals and is equivalent to the jitter metric used for data signals in eye diagram analysis. In the following the term *jitter* will always refer to the TIE metric. The total jitter (TJ) of a signal is composed of various jitter components with different root causes that can be divided into two main categories:

- **Random jitter (RJ)** stems from device noise such as shot noise, burst noise, $1/f$ noise and thermal noise. The distribution of random jitter is gaussian and unbounded and is typically characterized by its standard deviation σ_{RJ}.
- **Deterministic jitter (DJ)** is introduced through characteristics of the circuits and interconnects. Typical sources are losses, crosstalk, supply noise, mismatch and reflections. The bounded deterministic jitter is characterized by the peak-to-peak width of the jitter distribution and can be divided into two subcategories:
  - **Bounded uncorrelated jitter (BUJ)** is caused for example by supply noise or crosstalk from independent circuits in the same system. It is composed of periodic jitter (PJ) and aperiodic bounded uncorrelated jitter (ABUJ).
  - **Data dependent jitter (DDJ)** stems from non-idealities in the signal transmission like intersymbol interference (ISI) due to reflections and channel loss, as well as duty-cycle distortion (DCD) through circuit mismatch.

Since the total jitter is a superposition of the different jitter types an exact determination of the contribution of each individual component is not easily possible. The *dual-dirac model* [6, 23] is used to find approximate values for the random and deterministic jitter components by fitting two gaussian distributions to the left and right tails of the jitter histogram as illustrated in Figure 2.13. The total jitter estimated for a given bit error rate (BER)

$$TJ(BER) = 2Q_{\text{BER}}\sigma_{\text{TJ}} + DJ(\delta\delta) \tag{2.11}$$

is calculated from the average standard deviation of the two gaussian distributions

$$\sigma_{\text{TJ}} = \frac{\sigma_{\text{L}} + \sigma_{\text{R}}}{2} \tag{2.12}$$

and the distance of the averages

$$DJ(\delta\delta) = \mu_{\text{R}} - \mu_{\text{L}}. \tag{2.13}$$

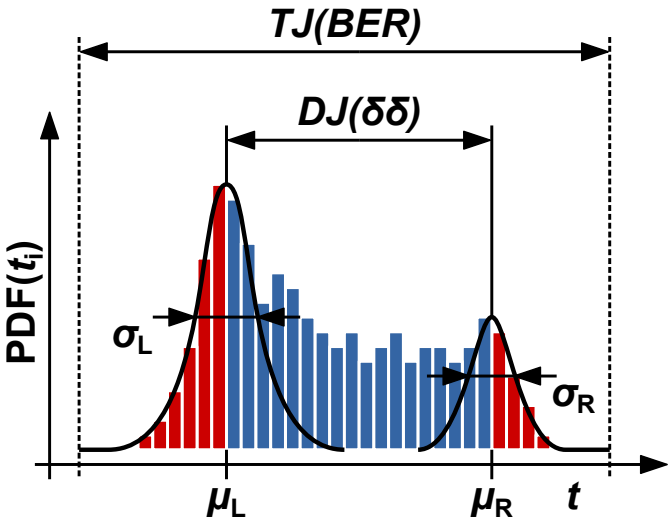


Figure 2.13: Illustration of the fitting of two gaussian distributions to the tails of a jitter histogram for the extraction of the jitter components using the dual-dirac model.

Table 2.5: Example values of Q_{BER} at different values of BER [6].

BER	Q_{BER}
10^{-10}	6.4
10^{-11}	6.7
10^{-12}	7.0
10^{-13}	7.3
10^{-14}	7.6

The factor Q_{BER} is calculated from the complementary error function to estimate the peak-to-peak width of the histogram of $n_{\mathrm{BER}} = 1/BER$ normally distributed samples. Example values of Q_{BER} at different bit error rates BER are shown in Table 2.5. The BER is defined as

$$BER = \frac{n_{\mathrm{errors}}}{n_{\mathrm{BER}}} \tag{2.14}$$

with the number of bit errors n_{errors} that occur when transmitting n_{BER} consecutive bits. More jitter components like PJ and DCD can be extracted using advanced jitter decomposition methods [24] that are out of the scope of this thesis.

Today most oscilloscopes provide a tool for jitter extraction using the dual-dirac model or advanced methods like spectral jitter decomposition [25]. However, many publications based on older measurement equipment only provide measurements of the total jitter in terms of root mean square (RMS) jitter

$$TJ_{\mathrm{RMS}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(t_{\mathrm{jit},i} - t_{\mathrm{jit,avg}})^2} \tag{2.15}$$

and/or peak-to-peak jitter

$$TJ_{\mathrm{pk-pk}} = \max(t_{\mathrm{jit},i}) - \min(t_{\mathrm{jit},i}) \tag{2.16}$$

calculated from n crossing times t_i and the average crossing time $t_{\mathrm{jit,avg}} = \overline{t_{\mathrm{jit},i}}$. The peak-to-peak jitter is frequently published without mention of the number of samples recorded. Since the random jitter

component is unbounded, its peak-to-peak value depends on the number of samples and therefore $TJ_{\mathrm{pk-pk}}$ cannot be compared without knowledge of the sample size. For this reason TJ_{RMS} is preferred for the comparison of total jitter measurements.

Another method for the analysis of the quality of a clock signal is the phase noise measurement. The phase noise is defined as the noise power of a signal spectrum in a 1 Hz bandwidth divided by the total signal power. The most common representation is the single sideband phase noise $S_\phi(f)$ in dBc/Hz at an offset f from the carrier frequency [26]. This type of measurement is typically performed with a spectrum analyzer, but can also be realized with an oscilloscope using the Fourier transform of the clock TIE. The phase noise analysis is mostly applied to PLLs and DLLs and can give deeper insights into the circuit performance, such as the loop bandwidth. It also helps to identify sources for clock jitter based on their dominant frequencies. The RMS jitter of a clock signal can be estimated from the phase noise by integrating $S_\phi(f)$ [26]. Common metrics are the phase noise at a certain offset from the carrier (typically 1 MHz) and the integrated RMS jitter over a certain frequency offset range (e.g. 10 kHz to 100 MHz).

2.5 Internal Circuit Calibration and Interface Training

In this section the typical link training steps of high-speed I/O links are explained and the most common approaches are presented. A high-speed link does not necessarily employ all of these procedures, e.g. differential links do not require a voltage threshold calibration. The goal of each link training step is to improve the performance of the link in terms of increased bandwidth and optimal sampling of the data. The calibration steps are presented in the order in which they are usually executed.

2.5.1 Calibration of Transmitter and Termination Impedance

The first link training step in almost any high-speed I/O link is the calibration of the transmitter and termination impedance. For high frequencies the link channel is treated as a transmission line and is designed for a characteristic impedance Z_{tml}. In order to avoid ISI due to reflections at the ends of the transmission line the source impedance Z_{S} and the termination impedance Z_{T}, connected to the termination voltage V_{TT}, should be equal to Z_{tml} as illustrated in Figure 2.14. A simple approach for the termination of a microstrip transmission line is to use external SMD resistors mounted on the PCB as close as possible to the package pins. By using resistors with low manufacturing tolerances a reliable impedance matching can be achieved. The drawback of this approach is that the bondwires of the chips are not included in the terminated transmission line and can cause unwanted reflections between termination and transmitter or receiver circuits. Also the layout space on the PCB and the system cost are increased by the additional components and the uncorrelated PVT variations of the on-chip circuits limit the accuracy of the termination.

Today, most integrated high-speed transceivers implement an *on-die termination* (ODT) close to the signal pad using polysilicon resistors and/or MOSFETs. The polysilicon resistance and the MOSFET

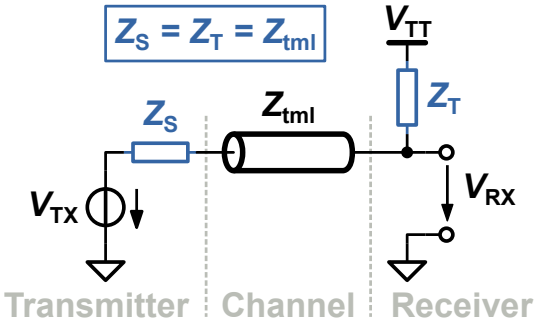
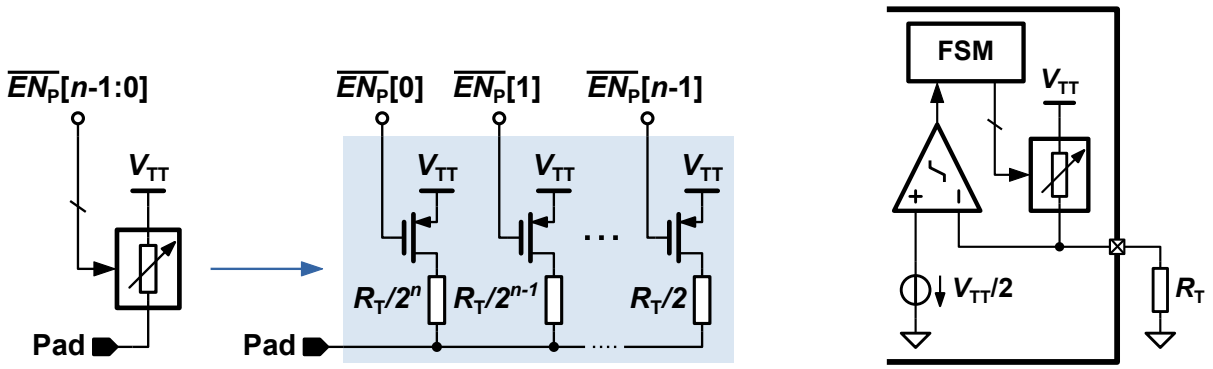


Figure 2.14: Ideal termination of a transmission line to avoid signal reflections.



(a) ODT circuit. **(b)** Calibration circuit.

Figure 2.15: Example circuits for a digitally adjustable ODT (a) and an ODT calibration circuit using an external resistor (b).

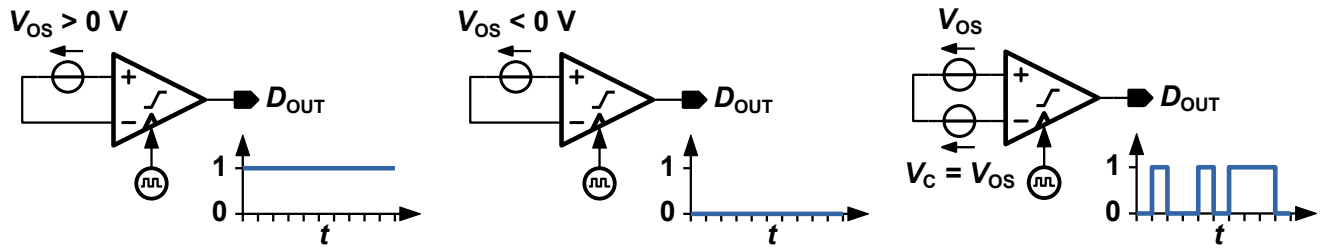
characteristics in a CMOS process show significant deviations from their nominal values due to process tolerances, as well as a noticeable voltage and temperature dependency [27]. Therefore precise termination circuits require a calibration to cover these PVT variations. This is typically realized using a replica of the termination circuit and an external precision resistor. The calibration can be either a continuous analog adaptation or an occasional digital calibration. The later has the advantage of reduced power consumption as the replica circuits can be switched off after calibration. In bidirectional transceivers the calibrated transmitter usually serves as ODT for the receiver. Figure 2.15 (a) shows an example for an n bit digitally adjustable ODT circuit using binary scaled polysilicon resistors and PMOS transistors. A simple calibration circuit using an external precision resistor, a comparator and an FSM is given in Figure 2.15 (b).

Advanced approaches to improve the impedance matching, and therefore increase the link bandwidth, make use of on-chip inductors between the terminated transmitter and the signal pad. Either single inductors [28] or transformers (T-coils) [29] are used. The drawbacks of these methods are the increased layout area and the limited availability and complex modeling of integrated inductors in CMOS processes.

2.5.2 Calibration of the Sampler Offset Voltage

The typical receiver front-end consists of a preamplifier and one or more data samplers, depending on the sampling scheme (SDR, DDR, QDR etc.). The transceivers designed in this work are operated using the DDR sampling scheme and therefore implement two data samplers clocked with the positive and the negative signal of the differential half-rate clock CK. After impedance calibration the input referred offset voltage of the data samplers is compensated in order to have the same threshold on all samplers. For links with sufficient voltage margin the offset compensation is sometimes omitted. The offset voltage V_{OS} of a data sampler (clocked comparator) can be determined by shorting the inputs and observing the output signal D_{OUT} of the sampler. In case of a positive V_{OS} the output is a static one (Figure 2.16 (a)) and for a negative V_{OS} it is a static zero (Figure 2.16 (b)). When a compensation voltage $V_C = V_{OS}$ is introduced, the output toggles between one and zero with equal probability. The offset calibration scheme applied in high-speed transceivers counts the number of ones and zeros at the output of the sampler and adjusts the compensation until an equal number is observed.

The schematic of a typical DDR receiver front-end with offset calibrated samplers and a CML preamplifier is shown in Figure 2.17. Shorting of the sampler inputs is realized by cutting off the current source of the CML preamplifier using the transistor $M_{N,0}$. When no current is flowing through the differential pair, the outputs of the preamp are both tied to V_{DD} by the load resistors R_L, effectively shorting the inputs of the samplers. An offset calibration FSM observes the deserializer (DES) output and adjusts the sampler offset using the correction signals OS_E and OS_O. Figure 2.18 (a) shows the schematic



(a) Positive offset voltage. **(b)** Negative offset voltage. **(c)** Compensated offset voltage.

Figure 2.16: Output signal of a clocked comparator with positive (a), negative (b) and compensated (c) offset voltage.

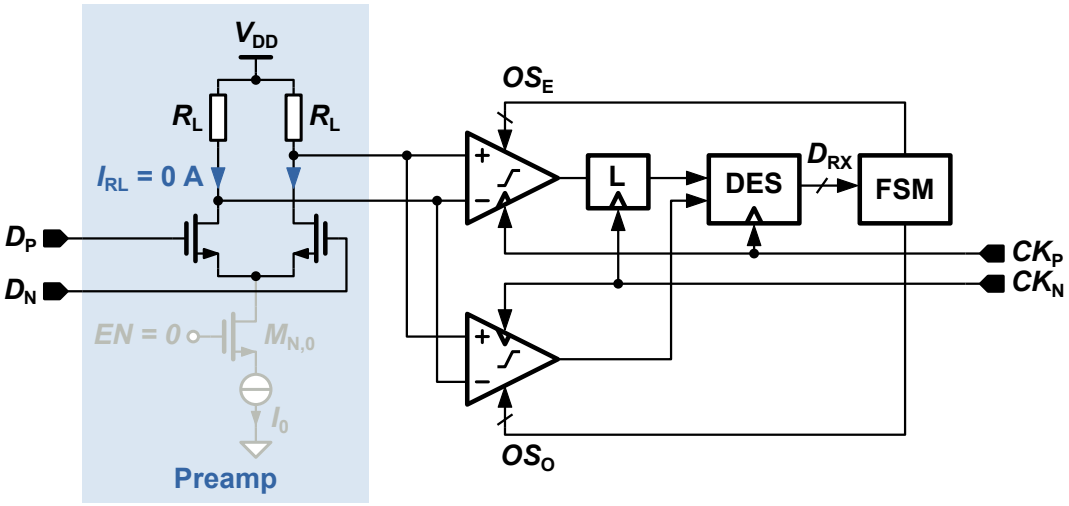
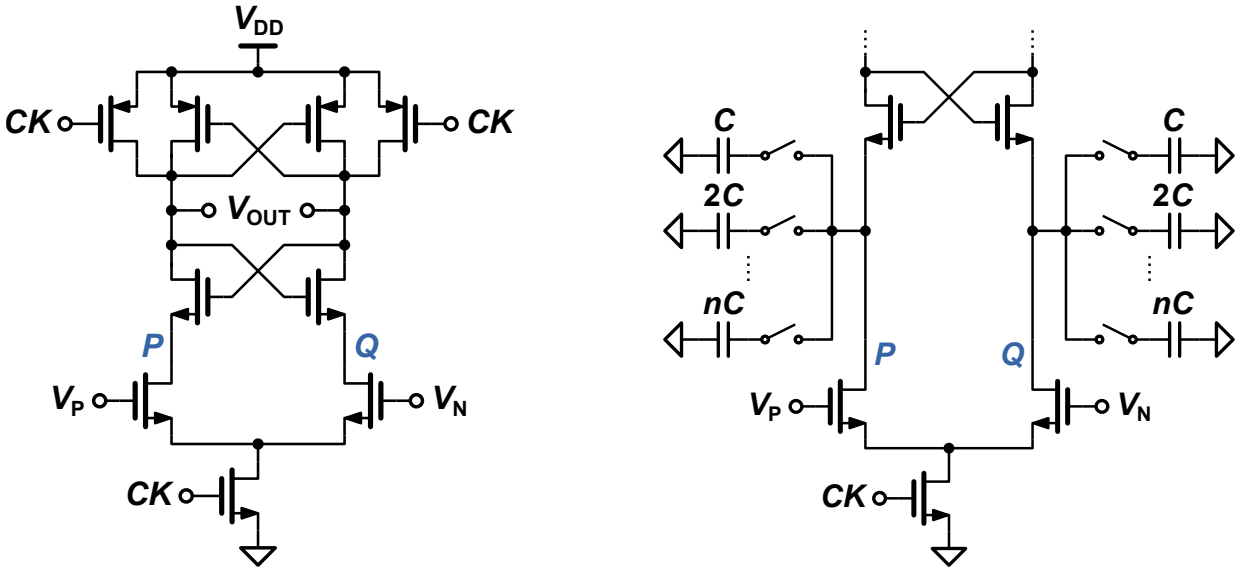


Figure 2.17: Schematic of a typical receiver front-end with sampler offset calibration for DDR sampling high-speed I/O links.



(a) StrongARM latch. **(b)** Offset compensation circuit.

Figure 2.18: Schematics of a typical data sampler based on the strongARM latch topology (a) and a capacitive offset compensation (b).

of a typical data sampler based on the strongARM latch topology [30]. The input referred offset voltage is caused by mismatch of the transistors in the differential circuit, mainly the input differential pair. The offset correction of clocked comparators is typically realized by digitally adjustable capacitors at nodes P and Q of the strongARM latch [31]. Alternative approaches use current injection at the output nodes [32] or body bias control [33].

2.5.3 Calibration of Transmitter and Receiver Equalizers

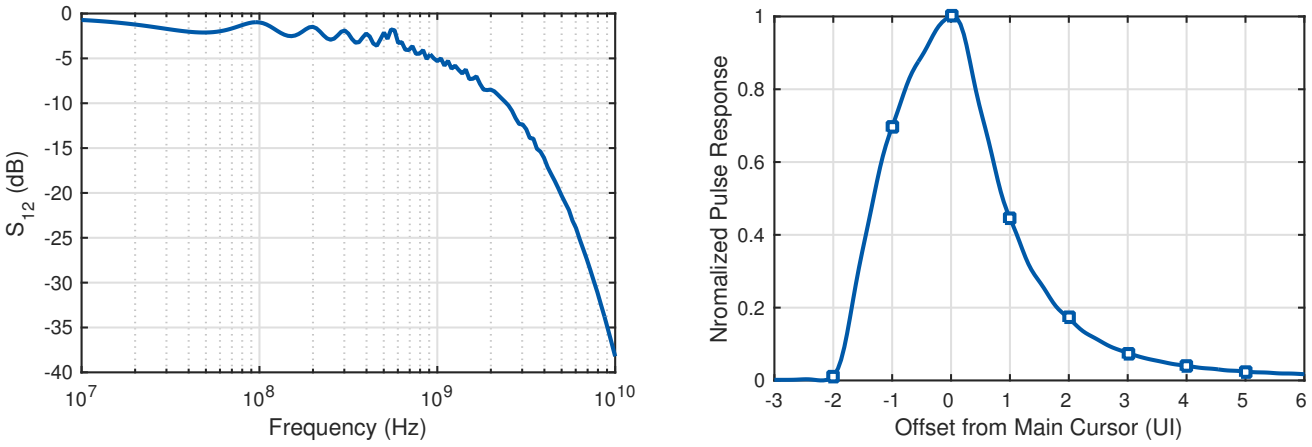
The frequency dependent attenuation of the transmission channel is limiting the maximum possible data rate of high-speed I/O links. The attenuation causes ISI and can even result in a fully closed data eye at the receiver. A common approach to counter this effect is the integration of *equalizers* at the transmitting and/or receiving end of the interface. An analysis of the channel characteristics is usually performed prior to deciding which equalization circuits to implement in an application. This analysis is based on two measurements (or simulations), the S-parameter measurement and the pulse response measurement. Figure 2.19 (a) shows the simulated transfer function of a single-ended transmission channel consisting of an 800 mm microstrip transmission line and a package, modeled using the techniques presented in section 2.3, in form of the S_{12}-parameter. A 6.4 Gb/s interface using this transmission channel experiences a maximum attenuation of ≈ 13 dB at 3.2 GHz. The pulse response in Figure 2.19 (b) shows significant ISI in form of an influence of the transmitted bit (main cursor) on the preceding bits (precursor) and the following bits (postcursor).

There are three kinds of equalizers that are typically used in high-speed I/O links. A *continuous time linear equalizer* (CTLE) can be added at the receiver side as a first amplification stage. The idea of a CTLE is to compensate the channel attenuation with an inverse transfer function. Typical CTLE implementations [34, 35, 36] have a nonlinear transfer function with a gain peaking around the operating frequency of the interface. The schematics of a conventional CML preamplifier and a CTLE based on source degeneration are given in Figure 2.20. In the CTLE the bias current source is split into two equal sources connected via the resistor R_S and the capacitor C_S at the source terminals of the input differential pair. R_S and C_S introduce a zero

$$\omega_z = \frac{1}{R_S C_S} \tag{2.17}$$

and a pole

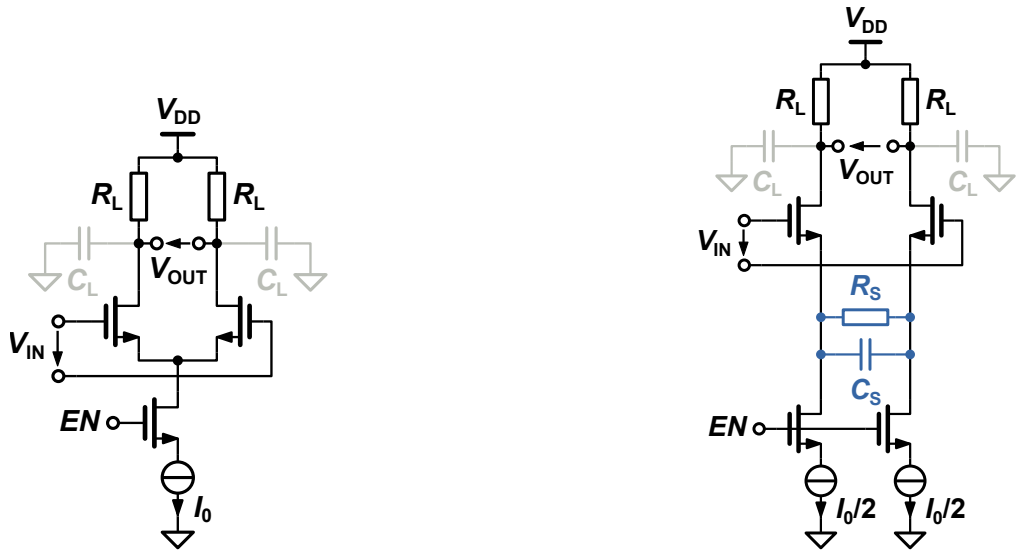
$$\omega_{p2} = \frac{1 + g_m \frac{R_S}{2}}{R_S C_S} \tag{2.18}$$



(a) S_{12} parameter.

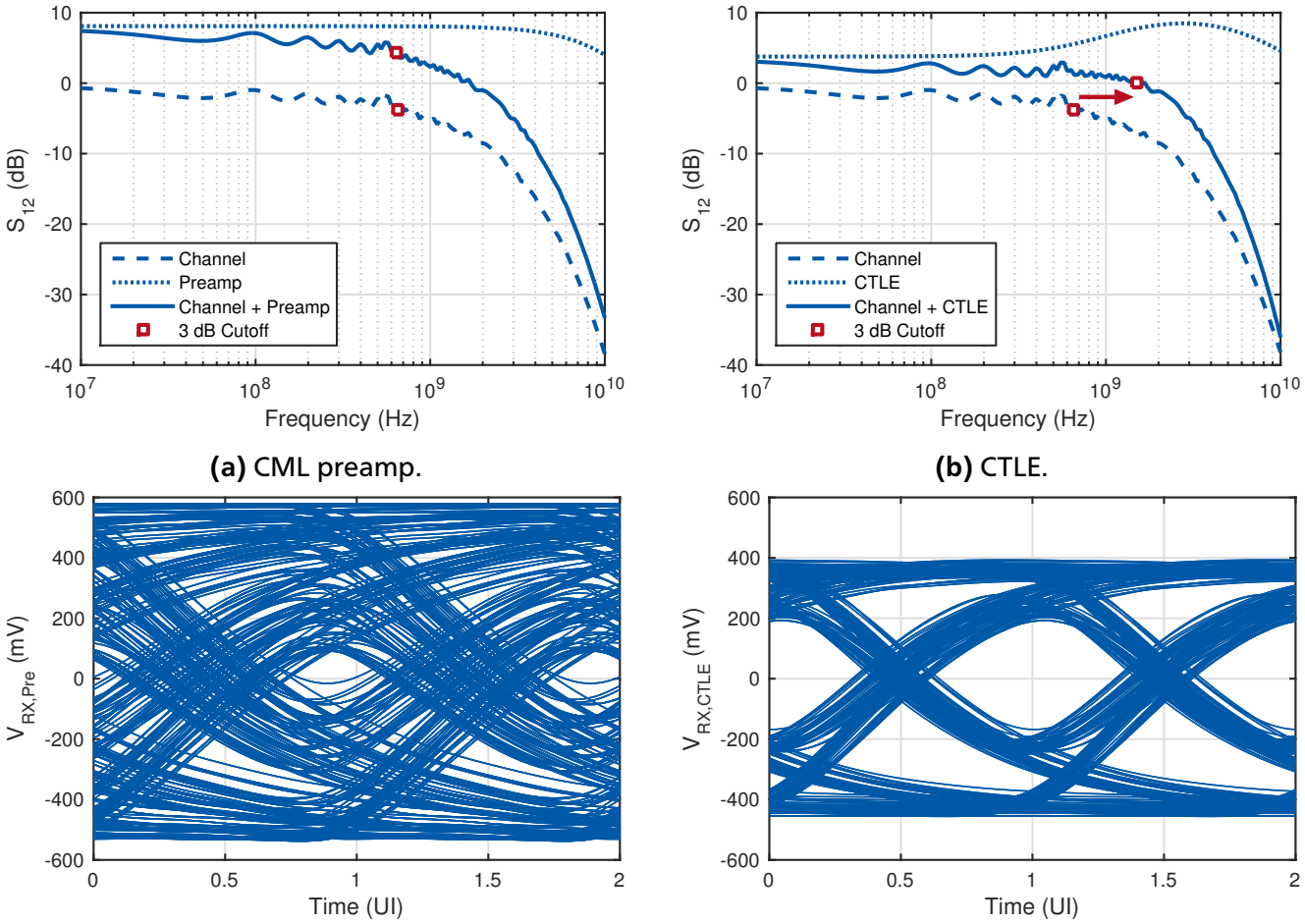
(b) Pulse response.

Figure 2.19: Simulated S_{12} parameter (a) and pulse response (b) of a single-ended transmission channel.



(a) CML preamp. **(b)** CTLE.

Figure 2.20: Schematics of a CML preamplifier (a) and a CTLE based on source degeneration (b).



(a) CML preamp. **(b)** CTLE.

(c) CML preamp. **(d)** CTLE.

Figure 2.21: Simulated S_{12} parameters (top) and eye diagrams (bottom) of a transmission channel with a conventional CML preamp (left) and a CTLE (right).

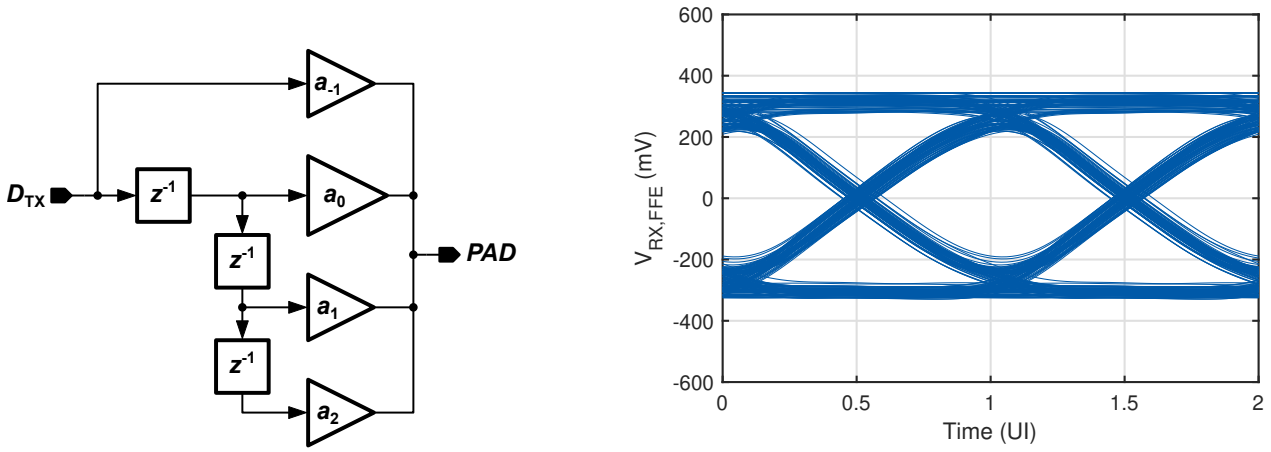
in the small signal transfer function of the CTLE in addition to the pole

$$\omega_{\mathrm{p1}} = \frac{1}{R_{\mathrm{L}}\, C_{\mathrm{L}}} \tag{2.19}$$

created by R_{L} and the load capacitance C_{L}, with the small signal transconductance g_{m} of the input differential pair [37]. Figure 2.21 shows a comparison of the simulated S_{12} parameters and the resulting eye diagrams of the example transmission channel at 6.4 Gb/s with a conventional CML preamplifier and a CTLE as the first amplification stage at the receiver side. Although the conventional CML preamplifier has a higher DC gain the eye diagram is nearly fully closed. The gain peaking of the CTLE moves the 3 dB cutoff frequency from 650 MHz to 1.5 GHz resulting in an acceptable eye diagram with sufficient voltage and timing margin for proper data transmission.

The other two equalizer types are *discrete time* equalizers, that means they provide a correction of the signal amplitude at the sampling time rather than a continuous equalization of the signals frequency content. *Feed-forward equalizers* (FFE) are finite impulse response (FIR) filters implemented on the transmitter side of the data link and can compensate both precursor and postcursor ISI. Figure 2.22 (a) shows the block diagram of a 4-tap FFE with the filter coefficients a_{-1}, a_0, a_1 and a_2. The coefficients can be derived from the pulse response of the transmission channel. A 2-tap FFE with only precursor compensation already provides significant ISI reduction as can be seen from the simulated eye diagram in Figure 2.22 (b) in comparison to Figure 2.21 (c) using the example transmission channel with a conventional CML preamplifier. *Decision feedback equalizers* (DFE) are infinite impulse response (IIR) filters implemented at the receiver side and can only compensate postcursor ISI. Due to the increased implementation effort DFEs are only implemented if FFEs and CTLEs are not sufficient to achieve an acceptable eye opening. DFE implementations can be found in the literature ([32, 36]) and are not presented in this work.

In systems with a well known transmission channel (e.g. a graphics card with both, the GPU and the DRAM soldered to the same PCB) the coefficients of the equalizers are often predefined by the manufacturer through simulation and/or measurement of the interface and are therefore static. When the channel characteristics are unknown or subject to change (e.g. cable or backplane interfaces) adaptive equalizers [38] are employed that can determine the ideal equalizer coefficients for the interface. More information on adaptive equalizer training can be found in [39]. The transceiver ASICs implemented in the scope of this work provide only static transmit equalization using a 2-tap FFE (see section 6.3.3).



(a) 4-tap FFE. **(b)** Eye diagram.

Figure 2.22: Block diagram of a 4-tap FFE (a) and simulated eye diagram of the example transmission channel with a 2-tap FFE (b).

2.5.4 Calibration of the Timing Center

The ideal timing center of an I/O link is the sampling time with the maximum margin to the points at which the BER exceeds a certain threshold (target BER). The ideal sampling time can be obtained by measuring the *bathtub curve* at the receiver of the I/O link and determining the timing center at the target BER [40]. Figure 2.23 shows an example for a bathtub curve in correlation with an eye diagram. The bathtub curve plots the BER in logarithmic scale over the offset from the eye center from -0.5 UI to +0.5 UI. This plot is obtained by sweeping the receiver clock phase and measuring the BER of a pseudo-random bit sequence (PRBS) transferred at every phase position. The measurement duration depends on the lowest BER to be measured because at least $1/BER_{\text{min}}$ bits need to be transmitted in order to measure a BER of BER_{min}. The target BER of high-speed I/O links is usually very low such as $BER_{\text{target}} = 10^{-12}$. The measurement duration for such a low BER is quite long (e.g. 2.6 minutes per phase position at 6.4 Gb/s data rate) and therefore this method is unpractical for the link training.

The conventional approach for the calibration of the timing center is to measure the BER at a higher BER_{min} resulting in a shorter measurement time. Typical values for BER_{min} of the conventional calibration method are 10^{-3} to 10^{-4}. The result of the measurement is a simple pass/fail decision where pass means that no bit errors have occurred. This way a *passing window* is determined and the timing center is set to the middle of the boundaries between the pass and fail regions [41]. Due to asymmetries of the eye diagram the calibrated timing center can be offset from the ideal center [40] as illustrated in Figure 2.23. Also, nonlinearity of the phase generation can lead to a clock centering error. Section 6.1 takes a deeper look at the passing window approach and other state-of-the-art methods for the timing center calibration.

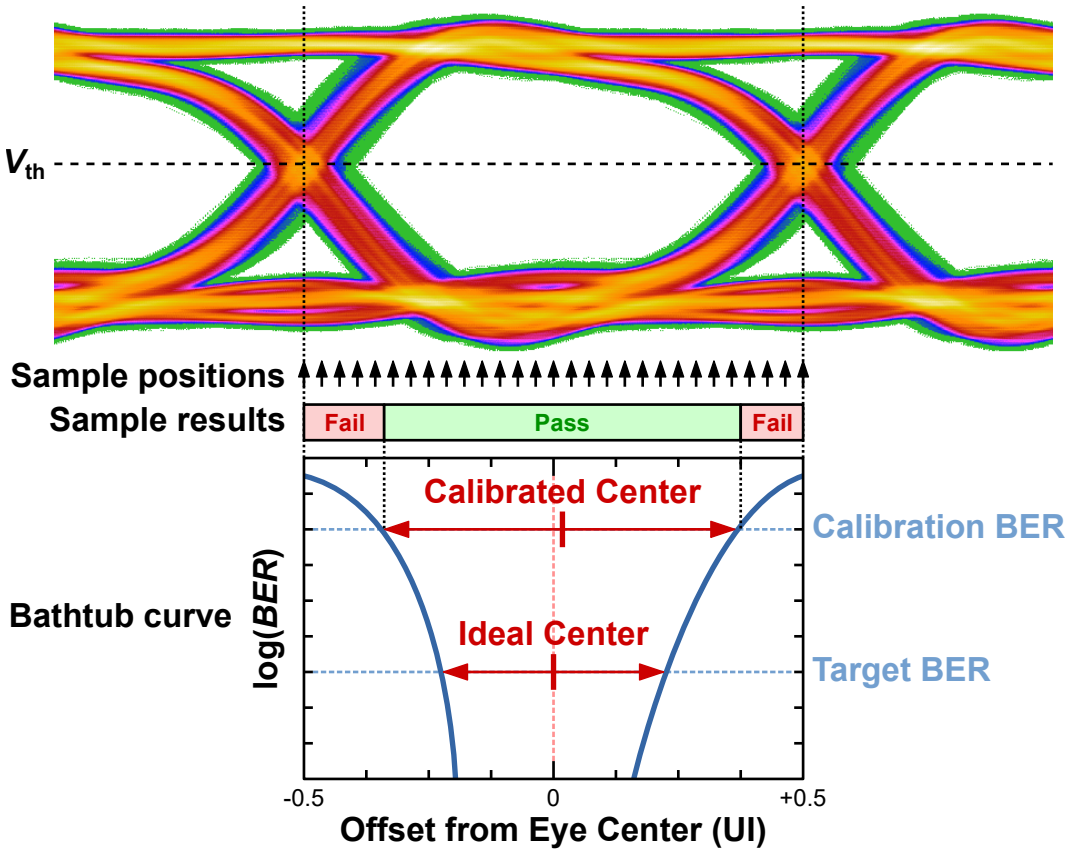
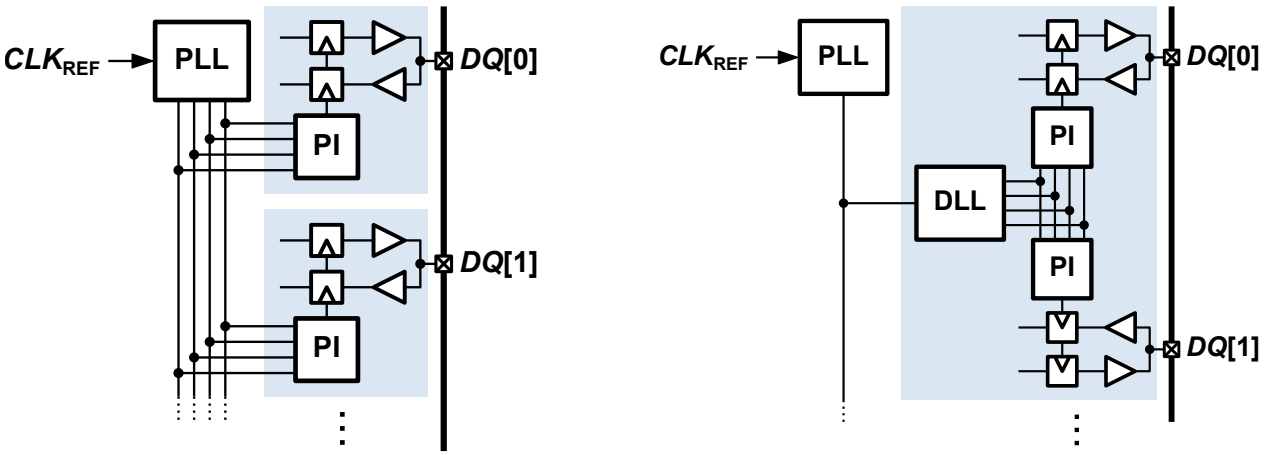


Figure 2.23: Illustration of the conventional timing center calibration based on the passing window approach.



(a) Global multiphase distribution.

(b) Local multiphase generation.

Figure 2.24: Block diagrams of a bidirectional parallel transceiver with individual clock phase adjustment using global multiphase distribution (a) and local multiphase generation (b).

The basic concept for the individual phase adjustment of the receiver sample clock in parallel high-speed I/O links is to generate fine phase steps from a multiphase clock signal using a phase interpolator (PI). The multiphase clock is either generated globally using a phase-locked loop (PLL) or a multiphase delay-locked loop (DLL) and is distributed to each transceiver (Figure 2.24 (a)) or it is generated locally using a multiphase DLL that is often shared between neighboring transceivers (Figure 2.24 (b)). The power consumption of the clock distribution takes up a significant part of the overall power consumption with increasing frequencies [34] and therefore the local multiphase generation is often preferred in parallel high-speed I/O links. Circuit examples for the multiphase DLL are presented in section 4.1.

2.5.5 Calibration of the Voltage Threshold

Single-ended high-speed I/O links require a threshold voltage V_{th} for the receiver circuits to distinguish between the high and the low voltage levels of the data signal. Due to channel and circuit non-idealties the ideal threshold can differ from the nominal common mode voltage of the transmitted signal. In order to maximize the voltage margin at the receiver, a voltage threshold calibration is performed. Figure 2.25 shows the schematic of a typical single-ended receiver front-end with a voltage digital-to-analog converter (DAC) for the generation of V_{th}. The received signal is coupled to the power supply networks through the parasitic capacitances of the ESD protection diodes. The resulting supply noise on the data signal alters the ideal voltage threshold. A supply noise tracking circuit consisting of C_{P} and C_{N} is used to couple V_{th} to the supply networks with a similar impedance as the ESD protection diodes. This way the voltage threshold follows the supply noise induced on the received data signal.

The voltage threshold calibration works similar to the timing center calibration. After the timing center is calibrated at a predefined nominal voltage threshold, the DAC sweeps V_{th} and a BER measurement is performed at each V_{th} level. Again, a passing window for the voltage threshold is determined by the boundaries of the pass and fail regions of the BER measurement and the calibrated threshold is set to the middle of the passing window. Figure 2.26 illustrates the voltage threshold calibration based on a vertical bathtub curve correlated with an eye diagram. As in the timing center calibration, the calibrated voltage threshold can be offset from the ideal threshold due to an asymmetry of the eye diagram and a measurement at a higher BER_{min} than the target BER. The voltage DAC typically has a very linear characteristic resulting in a neglectable calibration error due to nonlinearity.

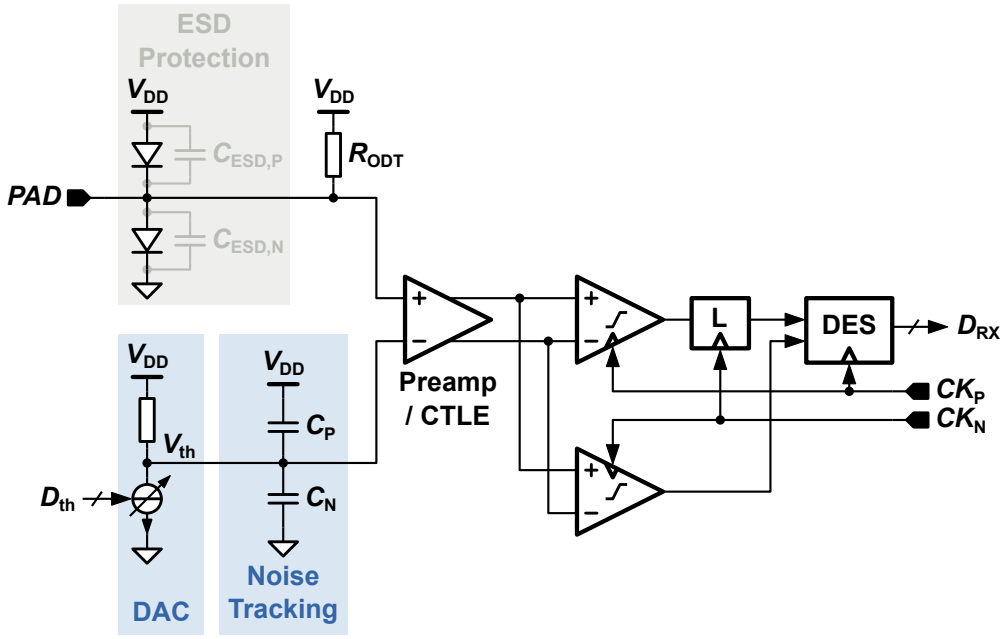


Figure 2.25: Schematic of a single-ended receiver front-end with a V_{th} generation DAC and a supply noise tracking circuit.

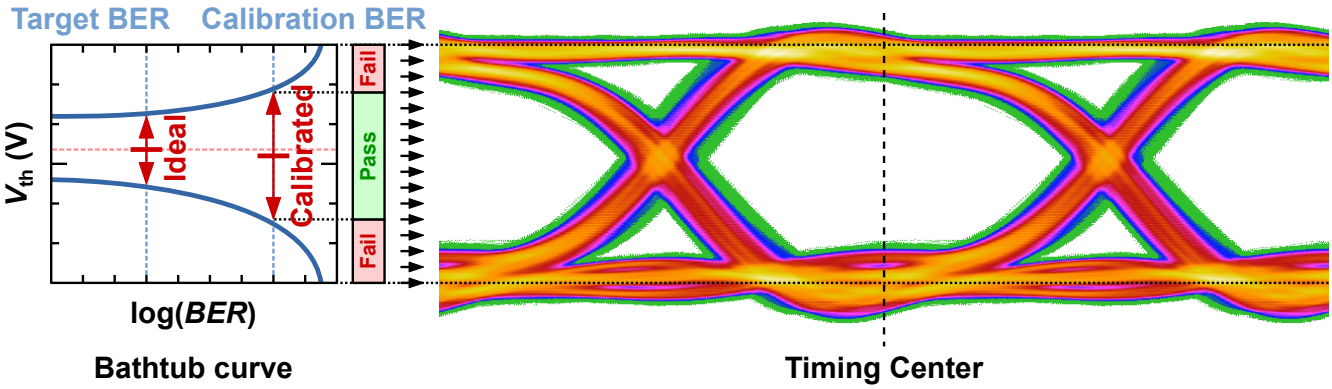


Figure 2.26: Illustration of the conventional voltage threshold calibration based on the passing window approach.

2.6 Conclusion

High-speed I/O links have a wide range of applications in integrated electronics. The diversity of the requirements in these applications have lead to a variety of interface standards. This chapter has presented an overview and a classification of the most common interface standards. Wire-linked communication interfaces are divided into source-synchronous I/O links with a forwarded clock and source-asynchronous I/O links with a clock and data recovery in the receiver. This work focuses on bidirectional asymmetric source-synchronous I/O links with a unidirectional clock and a bidirectional data transmission. The typical internal calibration procedures and interface trainings of such I/O links have been presented together with common implementation examples. The following chapters discuss how the conventional calibration methods can be enhanced by the use of a phase-to-digital converter for precise and efficient phase and duty-cycle measurements of high-speed clocks.

Part II

Design of Low-Power All-Digital High-Speed Clock Generation and Measurement Circuits

3 Scalable Synthesized Phase-to-Digital Converter

A synthesized phase-to-digital converter (PDC) is proposed that offers precise phase and duty-cycle measurements of high-speed clocks with very low area overhead and scalable resolution and sample rate. The PDC is intended to replace conventional phase and duty-cycle detection circuits in DLLs and duty-cycle correctors (DCC) for high-speed I/O links and to enable new calibration concepts for improved link performance. The underlying measurement concept is based on the asynchronous sampling of high-frequency clocks. It significantly reduces the area and power consumption over previously published implementations and offers a fully digital design with excellent scalability. The proposed method was published in [103] and filed as a patent [106].

3.1 Related Work

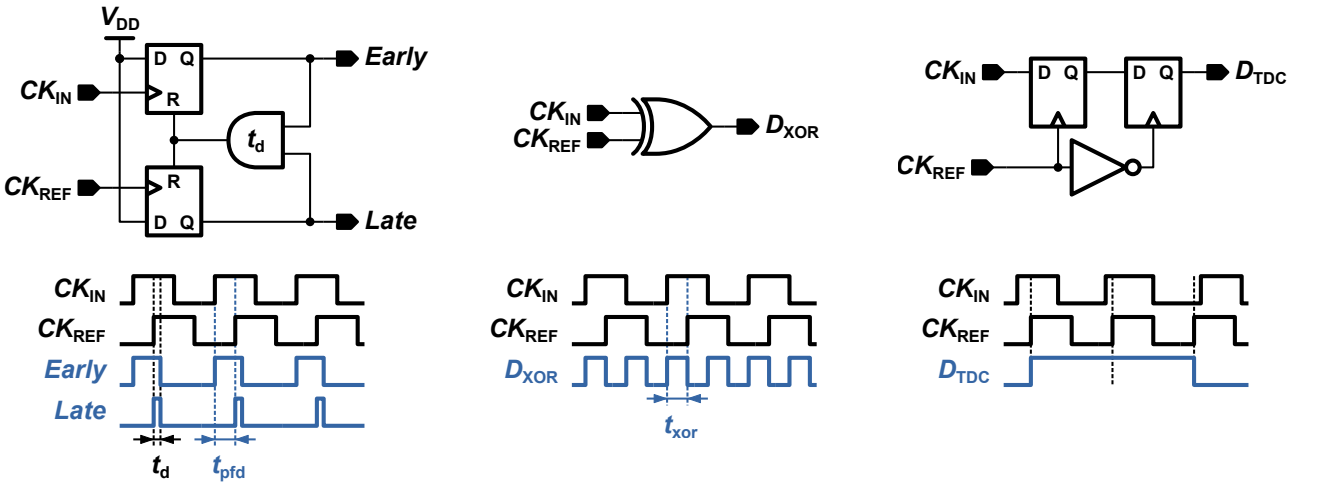
Circuits for the phase measurement of high-speed clock signals found in the literature can be divided into three categories. Binary phase detector based implementations are mainly used in PLLs and DLLs [42, 43, 44]. The output is proportional to the phase difference of the input clocks, but only offers a quantitatively precise result at 0°. Time-to-digital converters (TDC) can measure the time delay between arbitrary edges of a digital signal [45]. They are frequently used as phase detectors in PLLs, especially for fractional feedback ratios [46]. Asynchronous sampling based phase measurement circuits are emerging for high-speed applications as they offer a precise measurement and an all-digital implementation [2, 3].

The concepts for the duty-cycle measurement of high-speed clocks reach from fully analog to fully digital designs. The main application are DCCs for DDR sampling transceiver circuits. Methods for the digital conversion of the duty-cycle of a clock signal include analog integrator based implementations [47, 48, 49], oscillator based implementations [50, 51], TDCs [52, 53] and asynchronous sampling based methods [54]. Selected relevant concepts for efficient digital phase and duty-cycle measurements are presented in the following.

3.1.1 Binary Phase Detectors

Binary phase detectors generate a digital signal that indicates whether a clock signal leads or lags a reference clock signal with the same frequency. Three basic example circuits of common implementations are shown in Figure 3.1. The phase-frequency detector based on two D-flip-flops and a NAND gate is widely used in analog PLLs and DLLs and generates two output signals [42]. One that indicates whether the rising edge of the input clock CK_{IN} appears earlier than the rising edge of the reference clock CK_{REF} and one that indicates whether the edge appears later. The pulse width of the signals is proportional to the time delay t_{pfd} between the two clock signals with a constant offset of the delay through the NAND gate t_{d}. The XOR based phase detector generates only one output signal with a pulse width proportional to the time delay between the two input clock signals [55]. The pulses are generated on both, the rising and the falling edges, but it is not possible to distinguish between a positive and a negative delay between the signals. The 1 bit TDC generates a binary signal without pulse width modulation [43]. As long as CK_{IN} leads CK_{REF} the output is a logic one and otherwise it is a logic zero.

In most applications the binary output signals of the phase detectors are converted to either an analog voltage using a charge-pump [56] or a low-pass filter [55], or to a digital word using a digital accumulator [43, 44, 42]. The major drawback of a nonlinear conversion limits the possible applications



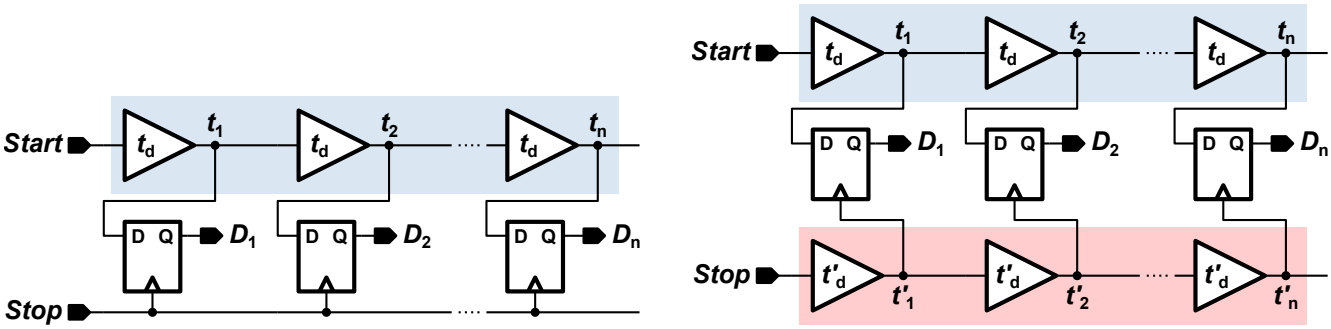
(a) Phase-frequency detector. **(b)** XOR phase detector. **(c)** 1 bit TDC.

Figure 3.1: Example schematics and timing diagrams of a phase-frequency detector (a), an XOR phase detector (b) and a 1 bit TDC (c).

of binary phase detectors. They are mostly used in PLLs and DLLs where a feedback system enforces a zero phase difference between two clock signals. The small area and the PVT robustness when combined with a digital accumulator make them the preferred choice for all-digital implementations. Binary phase detection can also be found in the synchronization of the data clock and the command clock in the GDDR5/5X/6 memory interface standards [12, 13]. Another popular application is the compensation of PVT variations by tracking the relative change of the phase between two clock signals with different circuit paths [55].

3.1.2 Time-to-Digital Converters

The basic idea of TDCs is to measure the time delay between two signal edges. This method can be used for both, phase and duty-cycle measurements by either measuring the delay between the same edges of two phase shifted clock signals or by measuring the delay between the rising and the falling edge of a single clock signal. Various concepts have been proposed to implement such a delay measurement. The simplest approach is the delay-line based TDC [57] that delays a start signal n times by a fixed delay of t_d. The intermediate signals at each delay stage are fed to a D-flip-flop clocked by a stop signal. At the moment when a rising edge of the stop signal occurs, the propagation status of the start signal through the delay line is reflected in the output signals of the D-flip-flops. For a phase measurement the two clock signals to be measured are used as the start and the stop signal of the TDC. For a duty-cycle



(a) Delay-line TDC. **(b)** Vernier delay-line TDC.

Figure 3.2: Schematics of a delay-line TDC (a) and a Vernier delay-line TDC (b).

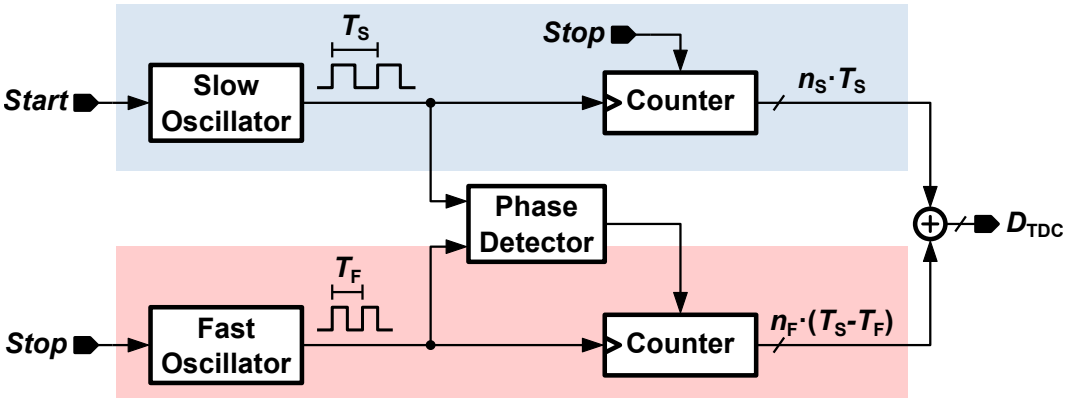


Figure 3.3: Block diagram of a Cyclic-Vernier TDC.

measurement the start signal is the rising edge of a clock signal and the stop signal is the falling edge. The resolution of this approach is limited by the minimum possible buffer delay t_d and the range is given by $n \cdot t_d$. A better resolution can be achieved by using a Vernier delay-line TDC [58] with a second delay-line for the stop signal. The delay elements in the second delay-line have a slightly lower delay $t'_d < t_d$ than the main delay elements and the resulting resolution of the TDC is defined by the difference $\Delta t_d = t_d - t'_d$ of the buffer delays. The 2 D Vernier delay-line architecture [59] improves the efficiency and provides an increased range covering also negative delays between the start and the stop signal. A two-dimensional array of D-flip-flops is used to detect delay steps in any combination of slow and fast delay buffers $t_{n,m} = n \cdot t_d - m \cdot t'_d$.

Oscillator based TDCs have the advantage of a small active area and a high resolution. The basic idea is to count the cycles of an oscillator during the time between a start and a stop pulse. The Cyclic-Vernier TDC architecture [60] works on a similar principle as the Vernier delay-line TDC. A slow oscillator with a period of T_S is enabled by the start signal and a counter counts the number of cycles n_S until the stop signal is detected (see Figure 3.3). At this point the fast oscillator with a period of T_F is enabled and a second counter counts the number of fast oscillator cycles n_F until a phase detector indicates that the slow and the fast oscillator edges appear exactly at the same time. The result of the time delay measurement is

$$t_{TDC} = n_S \cdot T_S + n_F \cdot (T_S - T_F) \tag{3.1}$$

with a resolution of $\Delta t = T_S - T_F$. Another advantage of this approach is that the range of the Cyclic-Vernier TDC can be easily extended by increasing the size of the counter for the slow oscillator cycles.

The gated ring oscillator TDC [61] in Figure 3.4 works with a single oscillator but uses multiple phases of the generated clock signal to increase the resolution of the TDC. Each clock phase drives a separate counter and the counter outputs are summed for the final conversion result. The design is intended for phase measurements of clock signals and can achieve an increased precision by repeatedly measuring the time delay between the clock edges and averaging the results.

A common way to increase the range and the conversion rate of TDCs is to use a coarse resolution TDC followed by a fine resolution TDC in a pipelined architecture [62]. The two TDCs can be based on different conversion concepts and thereby combine their properties. Coarse-fine TDCs have the drawback of a large area and power consumption. Many other TDC concepts have been proposed, e.g. pulse shrinking TDCs [63] or $\Delta\Sigma$ TDCs [64]. A significant drawback of all TDC architectures is the strong dependence on PVT variations that influences buffer delays and oscillator frequencies. As a result, most TDC concepts require a calibration or PVT compensation to generate precise results with a high and reliable resolution.

The performance comparison of different TDC architectures in Table 3.1 shows that a resolution of 1 ps can be achieved by most TDC concepts. Oscillator based TDCs have the smallest active area and

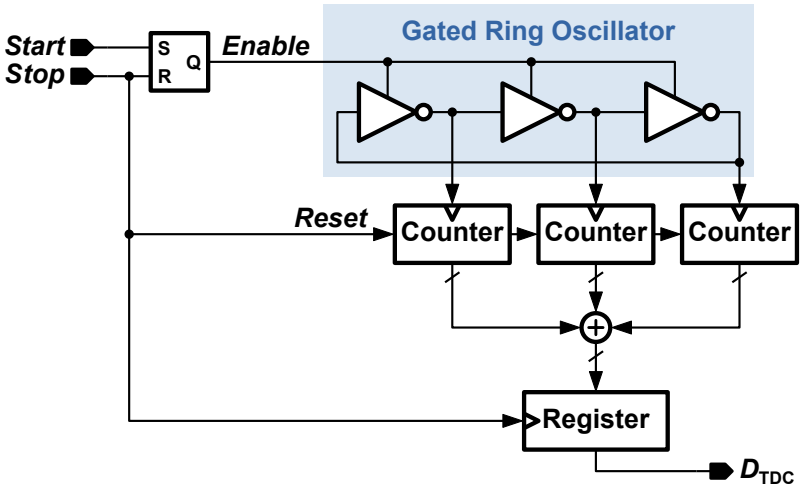


Figure 3.4: Block diagram of a gated ring oscillator TDC.

Table 3.1: Performance comparison of different TDC architectures.

Architecture	Process	Resolution	Conversion rate	Power	Area
Delay-line [57]	40 nm	9.8 ps	300 MS/s	N/A	N/A
Vernier delay-line [58]	130 nm	8 ps	15 MS/s	7.5 mW	0.26 mm^2
Pulse shrinking [63]	180 nm	1.8 ps	4.4 MS/s	3.4 mW	0.07 mm^2
Coarse-Fine [62]	65 nm	1.12 ps	250 MS/s	15.4 mW	0.14 mm^2
2D-Vernier [59]	130 nm	1 ps	10 MS/s	2.4 mW	0.06 mm^2
Cyclic-Vernier [60]	65 nm	1 ps	1 MS/s	0.01 - 0.15 mW	0.001 mm^2
Gated ring oscillator [61]	130 nm	1 ps	50 MS/s	2.2 - 21 mW	0.04 mm^2

can achieve a low power consumption when operated at a low conversion rate. Also, fully synthesized implementations of oscillator based TDCs have been proposed that scale well with technology.

3.1.3 Asynchronous Sampling Based Phase Measurement

A few phase and duty-cycle measurement circuits based on asynchronous sampling of high-speed clock signals have been proposed for clock calibration and eye detection in high-speed I/O links over the last years [3, 2, 65]. The basic concept in all these publications is to sample the input clock signal(s) with a fully asynchronous random clock and statistically evaluate the sampling results to determine the phase shift between two clocks or the duty-cycle of a single clock signal. In order to ensure the randomness of the asynchronous sample clock, a digitally controlled oscillator (DCO) is used and the frequency is altered by a linear feedback shift register (LFSR) in a pseudo-random manner. The circuits can be fully synthesized and therefore offer fast implementation with good scalability. The phase measurement circuit in [2] occupies an area of only 0.001 mm^2 in a 32 nm CMOS process and achieves a resolution of 250 fs. The resolution depends on the measurement duration and therefore represents a trade-off with the sample rate of the measurement. The block diagram in Figure 3.5 shows the asynchronous clock generator based on a DCO and an LFSR with additional clock dividers to enable autonomous operation at the desired frequency. The phase detector generates a pulse for each DCO cycle detected in the delay time t_d between the input clocks CK_{REF} and CK_{IN}. Two counters are used to count the number of detected DCO edges

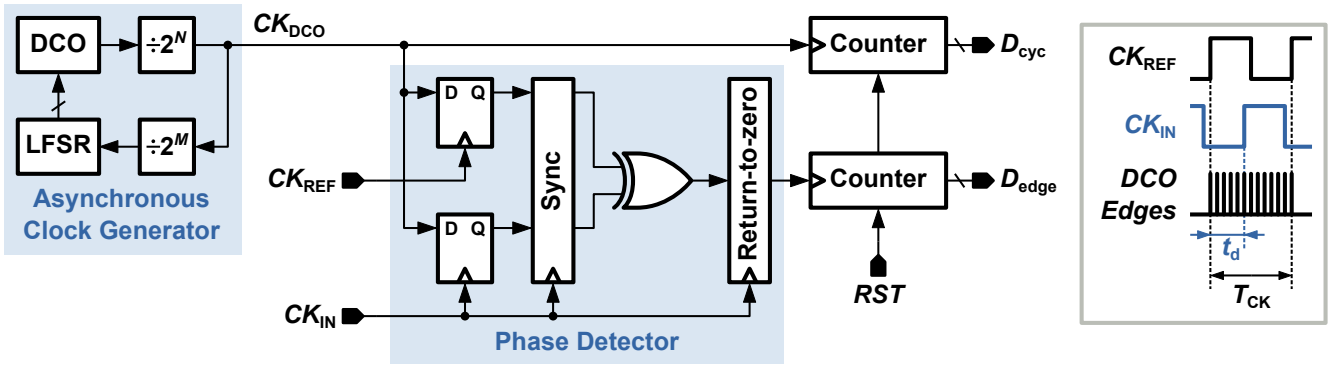


Figure 3.5: Block diagram of the asynchronous sampling based phase measurement circuit in [2].

n_{edge} during t_{d} and the number of DCO edges n_{cyc} during a complete clock cycle T_{CK}. The phase shift is then determined as the normalized delay

$$t_{\text{d,norm}} = \frac{t_{\text{d}}}{T_{\text{CK}}} = \frac{n_{\text{edge}}}{n_{\text{cyc}}}. \tag{3.2}$$

Multiple measurements are averaged to improve the accuracy and compensate the influence of jitter on the measurement result.

In [3] a timing skew compensation for source-synchronous I/O links is described based on the asynchronous sampling phase measurement circuit illustrated in Figure 3.6. Similar to [2] a DCO in combination with an LFSR is used to generate an asynchronous random clock signal. Cascaded D-flip-flops avoid metastability issues by the asynchronous sampling of the input clock signals CK_{REF} and CK_{IN}. There are four possible states of the sampled signals with both being zero, both being one and only one of them being one, described by a 2 bit region code $D_{\text{reg}}[1{:}0]$. One counter counts the number of occurrences n_{meas} of a selected region code and a second counter counts the total number of DCO clock cycles n_{total} during the measurement time. The phase shift between CK_{REF} and CK_{IN} can be determined by the ratio $n_{\text{meas}}/n_{\text{total}}$ of the two counter values. The accuracy depends on the total number of DCO clock cycles observed and again represents a trade-off with the sample rate of the measurement. In [66] the same concept is implemented using four counters to capture the occurrence of every possible state of the sampled signals instead of just one. Both designs have been verified on FPGAs and [3] estimated

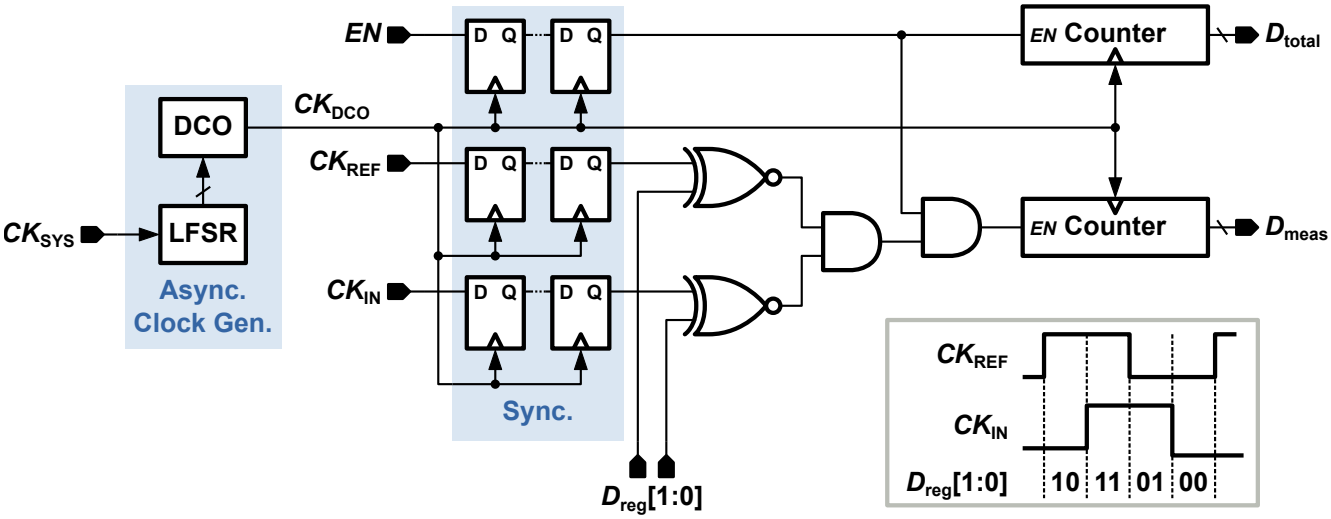


Figure 3.6: Block diagram of the asynchronous sampling based phase measurement circuit in [3].

the active area of an ASIC implementation with two 16 bit counters in a 130 nm CMOS process to be 0.0034 mm^2.

A simplified version of the circuit in [2] is used in [65] as a duty-cycle detector. Only one clock signal is sampled and the number of ones detected is divided by total number of cycles to determine the duty-cycle. The design uses single ended D-flip-flops as samplers and therefore the measurement is performed on the positive as well as the negative signal of a differential clock in order to compensate non-idealities of the samplers. Similarly the concept from [3] is modified in [54] for duty-cycle measurement.

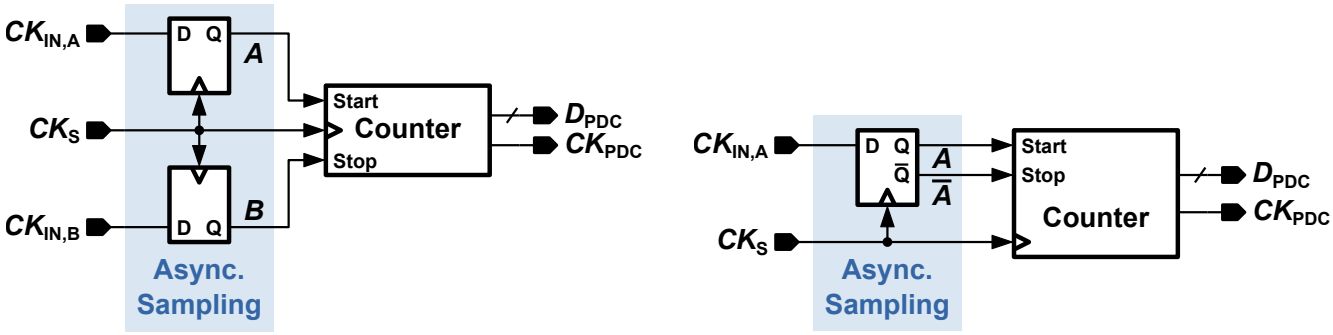
Asynchronous sampling based eye opening monitors [67, 68, 69] offer advanced analysis of on-chip high-speed signals and can also operate on PRBS signals instead of clock signals. The drawbacks are increased area and power consumption and therefore the designs are not suitable for the eye centering of massively parallel high-speed I/O links.

3.2 Proposed Design and Theory of Operation

The operation principle of the proposed PDC differs from the state-of-the-art asynchronous sampling based phase and duty-cycle measurement circuits and offers several advantages, such as a smaller design and a simple scalability of resolution and sample rate. Additionally, the design constraints can be derived from the desired performance of the PDC using the models for the influence of clock jitter presented in section 3.2.2.

3.2.1 Working Principle

In contrast to previously published phase and duty-cycle measurement circuits based on the asynchronous sampling method, the proposed PDC uses a fixed frequency ratio between sample clock and input clock instead of a fully random asynchronous clock. This simplifies the design of the measurement circuit, but requires the sample clock to be generated by a PLL locked to the same reference as the input clock. The frequency of the asynchronous sample clock CK_S is chosen in a way that the sampling generates lower frequency representations of the input clocks CK_{IN} with the same phase relationship and duty-cycle as the original signals. The operation is similar to the down conversion of radio frequency signals using analog mixers in wireless receiver circuits. The two properties phase and duty-cycle can then be measured using a single counter that counts the number of CK_S cycles between the rising (or falling) edges of the two input clocks in case of a phase measurement or between the rising and falling edges of a single input clock in case of a duty-cycle measurement. Due to the fixed frequency ratio, the total number of CK_S cycles for a full period of CK_{IN} is fixed and therefore a division by a second counter value, as in the conventional asynchronous sampling based methods, is not required. Basic block diagrams of the PDC for phase and duty-cycle measurement are shown in Figure 3.7.



(a) Phase measurement. **(b)** Duty-cycle measurement.

Figure 3.7: Basic block diagrams of the proposed PDC for phase (a) and duty-cycle (b) measurement.

The frequency of the asynchronous sample clock

$$f_S = \frac{N}{N\,M+1} f_{IN} \tag{3.3}$$

is calculated from the input clock frequency f_{IN} using two parameters N and M. The integer parameter $N > 1$ defines the resolution of the PDC measurement in terms of phase steps within a full clock period. By choosing the resolution to be a power of two ($N = 2^{n_{bit}}$) D_{PDC} is automatically an n_{bit} bit wide binary expression of the measurement result that simplifies further processing. The second parameter $M \geq 1$ is an integer scaling factor that can be used to reduce the sample clock frequency f_S without influence on the resolution of the measurement. An increase of M (and therefore a decrease of f_S) results in a decreased power consumption of the PDC due to the well known proportionality

$$P_{gate} \propto C_L\, V_{DD}^2\, f_{clk}, \tag{3.4}$$

with the power dissipation of a CMOS logic gate P_{gate}, the load capacitance C_L, the supply voltage V_{DD} and the operating frequency f_{clk}. As a trade-off the sample rate of the PDC

$$f_{PDC} = \frac{f_S}{N} = \frac{f_{IN}}{N\,M+1} \tag{3.5}$$

is reduced by increasing M.

Figure 3.8 shows a timing diagram of a PDC based on the block diagram in Figure 3.7 (a) with a resolution of $N = 2^3$ and a scaling factor of $M = 1$ for an input clock phase difference of $\phi_{A,B} = 225°$. The sampled signals A and B have the same phase difference at a lower frequency and control the operation of the counter D_{CNT} at their rising edges. The measurement result

$$D_{PDC}[3:0] = \left\lfloor \frac{\phi_{A,B}}{360°} N \right\rfloor = \left\lfloor \frac{225°}{360°} 8 \right\rfloor = 5 \tag{3.6}$$

reflects the input phase difference as a 3 bit binary number.

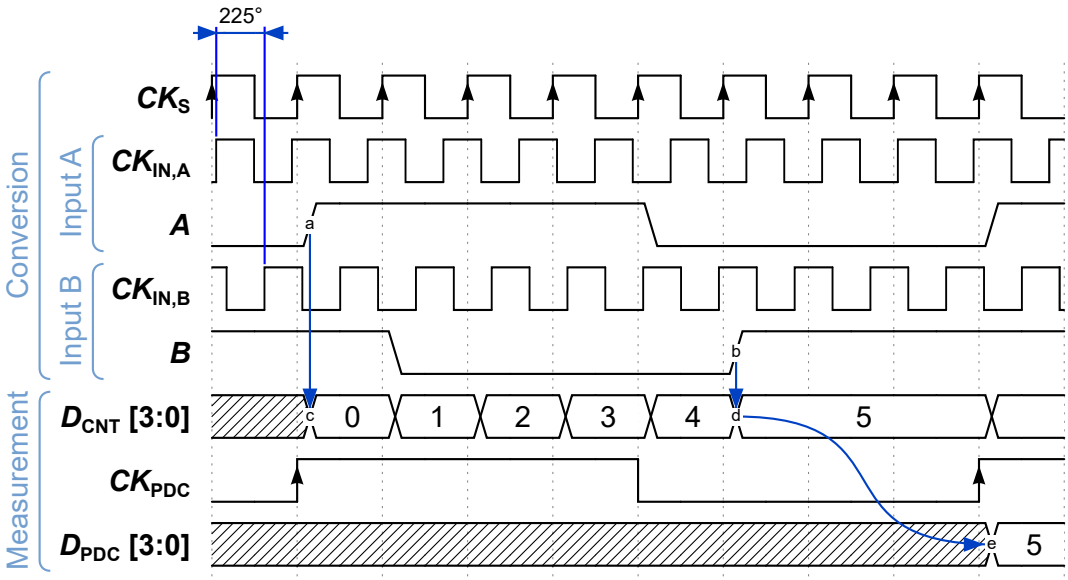


Figure 3.8: Timing diagram of a PDC based on the block diagram in Figure 3.7 (a) with $N = 2^3$ and $M = 1$ for an input clock phase difference of $\phi_{A,B} = 225°$.

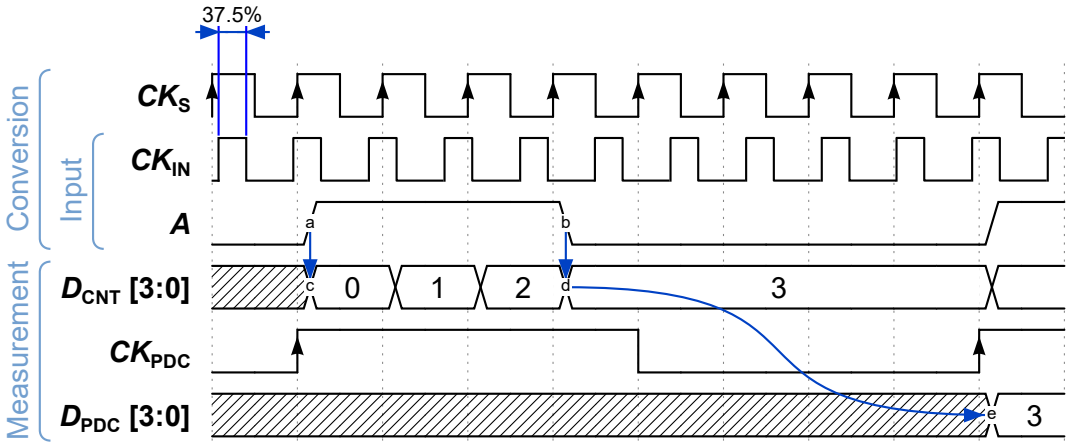


Figure 3.9: Timing diagram of a PDC based on the block diagram in Figure 3.7 (b) with $N = 2^3$ and $M = 1$ for an input clock duty-cycle of 37.5 %.

A timing diagram of a PDC for duty-cycle measurement based on the block diagram in Figure 3.7 (b) with the same configuration is shown in Figure 3.9. Here, the rising and falling edges of the sampled signal A control the operation of the counter and the input duty-cycle of 37.5 % is converted to a binary number

$$D_{\mathrm{PDC}}[3:0] = \lfloor 0.375 \cdot 8 \rfloor = 3 \tag{3.7}$$

by the PDC.

3.2.2 Modeling of the Impact of Clock Jitter on the Measurement Result

In real circuit implementations, the clock signals CK_{IN} and CK_{S} have a certain amount of jitter. This jitter has an impact on the measurement result of the PDC that can be estimated using statistical models. As a positive side effect, the influence of clock jitter can lead to an improved resolution of the measurement beyond the nominal resolution N when multiple PDC samples are averaged. The underlying assumption for the following calculations is that the jitter on all clock signals follows a normal distribution. This assumption is only true for signals dominated by random jitter sources (see section 2.4), which is the case for a majority of high-frequency clock generators, or at least a sufficient approximation.

As CK_{IN} and CK_S are asynchronous, they have to be generated from independent oscillators and therefore their jitter is assumed to be uncorrelated. The probability density function (PDF) of the standard normal distribution

$$\mathrm{PDF}(t_{\mathrm{s}}) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1}{2}\left(\frac{t_{\mathrm{s}} - t_{\mathrm{edge}}}{\sigma}\right)^2} \tag{3.8}$$

describes the probability of the appearance of a clock edge at an offset t_{s} from the ideal position t_{edge} based on the RMS jitter σ. For every position t_{s} of the sampling clock rising edge the probability of the rising edge of CK_{IN} being at an earlier position $t_{in} \leq t_{\mathrm{s}}$ can be calculated by the cumulative density function (CDF)

$$\mathrm{CDF}(t_{\mathrm{s}}) = \int_{-\infty}^{t_{\mathrm{s}}} \mathrm{PDF}(t_{\mathrm{s}})\,\mathrm{d}t_{\mathrm{s}} = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{t_{\mathrm{s}} - t_{\mathrm{edge}}}{\sigma\sqrt{2}}\right)\right]. \tag{3.9}$$

As only the relative position of the input clock edge distribution to the sample clock edge distribution matters for the calculation, $t_{\mathrm{edge,s}} = 0$ is chosen and $\Delta t_{\mathrm{edge}} = t_{\mathrm{edge,in}} - t_{\mathrm{edge},2}$ is used to describe the offset. The product of the probability density of the sample clock edge $\mathrm{PDF}_{\mathrm{S}}(t_{\mathrm{s}})$ and the cumulative probability

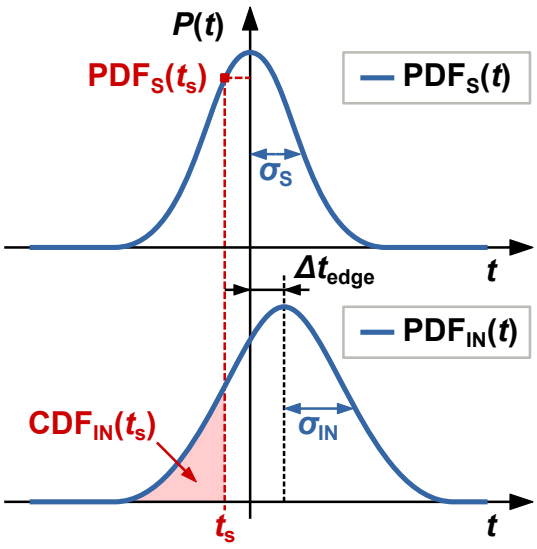


Figure 3.10: Illustration of the method for the calculation of the probability to detect a logic one $P_1(\Delta t_{\text{edge}})$ when sampling a clock signal with an RMS jitter of σ_{IN} using a sample clock with an RMS jitter of σ_{S} at an offset Δt_{edge} between the ideal positions of the edges of the two clock signals.

of the input clock edge $\text{CDF}_{\text{IN}}(t_{\text{s}})$ integrated over all possible edge positions t_{s} gives the total probability of detecting a logic one at a certain offset Δt_{edge}

$$P_1(\Delta t_{\text{edge}}) = \int_{-\infty}^{\infty} \text{PDF}_{\text{S}}(t_{\text{s}}) \cdot \text{CDF}_{\text{IN}}(t_{\text{s}})\, \text{d}t_{\text{s}}. \tag{3.10}$$

The resulting integral can be written as

$$P_1(\Delta t_{\text{edge}}) = \int_{-\infty}^{\infty} \frac{1}{\sigma_{\text{S}}\sqrt{2\pi}} e^{\frac{1}{2}\left(\frac{t_{\text{s}}}{\sigma_{\text{S}}}\right)^2} \cdot \frac{1}{2}\left[1 + \text{erf}\left(\frac{t_{\text{s}} - \Delta t_{\text{edge}}}{\sigma_{\text{IN}}\sqrt{2}}\right)\right] \text{d}t_{\text{s}} \tag{3.11}$$

using the RMS jitter values σ_{S} and σ_{IN} of CK_{S} and CK_{IN}, respectively. The derivation of $P_1(\Delta t_{\text{edge}})$ is illustrated in Figure 3.10 for one possible position t_{s} of the sample clock edge.

Equation (3.11) can be solved either analytically using taylor series or numerically. In the following the numerical approach is used, requiring finite integration boundaries. One practical way to find appropriate boundaries is to use the peak-to-peak jitter value of a real measurement or estimate the peak-to-peak jitter from the RMS jitter value. The JEDEC standard JESD65B [70] defines a sample size of 10000 for period jitter measurements. Thus the peak-to-peak value can be estimated as the boundaries of the normal distribution in which 9999 of the 10000 samples are contained (99,99 %). This is the case for approximately $\pm 3.72\,\sigma$, so the peak-to-peak jitter is

$$TJ_{\text{pk-pk}} \approx 2 \cdot 3.72\,\sigma = 7.44\ TJ_{\text{RMS}}. \tag{3.12}$$

Using this equation the integration boundaries for the calculation of $P_1(\Delta t_{\text{edge}})$ can be set to $[-3.72\,\sigma_{\text{S}}, 3.72\,\sigma_{\text{S}}]$. Outside of these boundaries $\text{PDF}_{\text{S}}(t)$ is approximately zero, so the product with $\text{CDF}_{\text{IN}}(t)$ is also zero and does not contribute to the total probability. The numerical integration formula is then

$$P_1(\Delta t_{\text{edge}}) = \sum_{t_{\text{s}}=-3.72\,\sigma_{\text{S}}}^{3.72\,\sigma_{\text{S}}} \frac{1}{\Delta t_{\text{s}}} \frac{1}{\sigma_{\text{S}}\sqrt{2\pi}} e^{\frac{1}{2}\left(\frac{t_{\text{s}}}{\sigma_{\text{S}}}\right)^2} \cdot \frac{1}{2}\left[1 + \text{erf}\left(\frac{t_{\text{s}} - \Delta t_{\text{edge}}}{\sigma_{\text{IN}}\sqrt{2}}\right)\right] \tag{3.13}$$

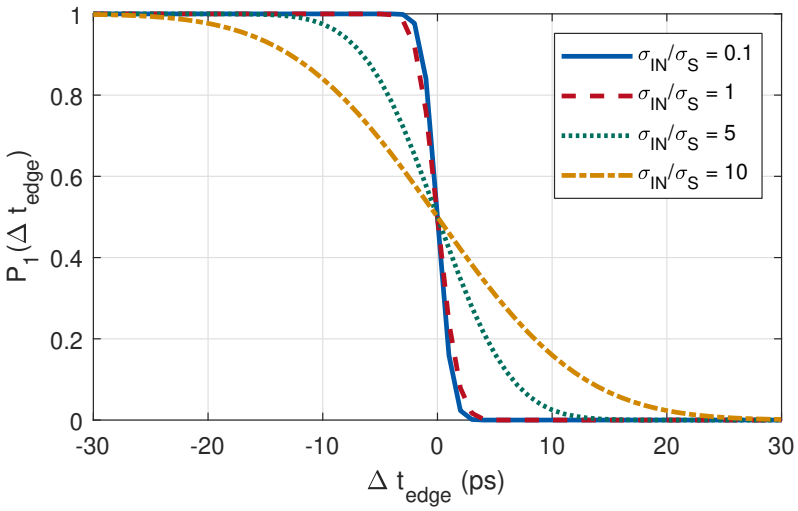


Figure 3.11: Example of $P_1(\Delta t_{\text{edge}})$ for different RMS jitter ratios $\sigma_{\text{IN}}/\sigma_{\text{S}}$ between input clock and sample clock.

with the step size

$$\Delta t_{\text{s}} = \frac{7.44\,\sigma_{\text{S}}}{n_{\text{steps}}} \tag{3.14}$$

and the total number of integration steps n_{steps}. $P_1(\Delta t_{\text{edge}})$ depends on the ratio of the input clock jitter to the sample clock jitter $\frac{\sigma_{\text{IN}}}{\sigma_{\text{S}}}$ as shown in Figure 3.11. Using finite boundaries based on equation (3.12) for both clock signals $P_1(\Delta t_{\text{edge}})$ can be limited by

$$P_1(\Delta t_{\text{edge}}) = \begin{cases} 1, & \Delta t_{\text{edge}} < -(3.72\,\sigma_{\text{S}} + 3.72\,\sigma_{\text{IN}}) \\ P_1(\Delta t_{\text{edge}}), & -(3.72\,\sigma_{\text{S}} + 3.72\,\sigma_{\text{IN}}) \leq \Delta t_{\text{edge}} \leq (3.72\,\sigma_{\text{S}} + 3.72\,\sigma_{\text{S}}) \\ 0, & (3.72\,\sigma_{\text{S}} + 3.72\,\sigma_{\text{IN}}) < \Delta t_{\text{edge}} \end{cases}, \tag{3.15}$$

so for large negative offsets Δt_{edge} the sampling flip-flop always samples a one and for large positive offsets it always samples a zero.

Due to the asynchronous sampling the relative position of the sample clock edges to the input clock edges Δt_{edge} changes with every cycle of CK_S. Since the frequency ratio is fixed, the values of Δt_{edge} repeat every N cycles of CK_S. The increment of Δt_{edge} with every cycle of CK_{S} is equal to the difference of the clock periods at $M = 1$

$$T_{\text{S}}^{M=1} - T_{\text{IN}} = \frac{1}{(N+1)}\,T_{\text{S}} =: \Delta T. \tag{3.16}$$

Increasing M has an effect on T_{S} but not on the increment ΔT to keep the resolution of the PDC unchanged. The quantized edge positions

$$\Delta t_{\text{edge}}^{n} = n\,\Delta T + t_{\text{offset}} \tag{3.17}$$

can have an offset of $0 \leq t_{\text{offset}} < \Delta T$ due to the arbitrary phase relationship between CK_{S} and CK_{IN}. For the estimation of the influence of jitter the scenario $t_{\text{offset}} = 0$ represents the worst case since at the edge position $n = 0$ the rising edges of CK_{S} and CK_{IN} appear exactly at the same time and the smallest amount of jitter leads to an equal probability of sampling CK_{IN} before or after the rising edge. Otherwise the jitter needs to exceed the offset

$$TJ_{\text{pk-pk,S}} + TJ_{\text{pk-pk,IN}} > \min(t_{\text{offset}}, \Delta T - t_{\text{offset}}) \tag{3.18}$$

before it has an impact on the sampling result. Therefore, $t_{\text{offset}} = 0$ is assumed in the following. Ideally the zero to one transition of the sampled input clock signal happens when $\Delta t_{\text{edge}} = 0$. Due to the jitter on CK_{S} and CK_{IN} the transition can also happen at $\Delta t_{\text{edge}} < 0$ or $\Delta t_{\text{edge}} > 0$ with a certain probability. For a high relative jitter $\frac{\sigma}{T}$ on one or both clocks there can also be multiple zero to one transitions in one cycle of the sampled input clock signal. These glitches need to be handled in some way in order to properly decide on a counter value. A method to remove these glitches from the sampled signals using a filter is proposed in section 3.3.1.

In the following the first zero to one transition of the sampled input clock signal is used for the counter evaluation. In order to cover all possible transition positions a range of $[-\frac{N}{2}\,\Delta T, \frac{N}{2}\,\Delta T]$ for Δt_{edge} is considered. Based on equation (3.15) the range can be further narrowed, so the offsets that need to be evaluated are

$$\Delta t^{n}_{\text{edge}} = n\,\Delta T,\ n \in [-n_{\max}, n_{\max}] \tag{3.19}$$

with a maximum offset of

$$n_{\max} = \left\lfloor \frac{\frac{N}{2}\,\Delta T}{3.72\,\sigma_{\text{S}} + 3.72\,\sigma_{\text{S}}} \right\rfloor. \tag{3.20}$$

For simplicity and because of typical RMS jitter values being small enough, it is assumed that the considered range of the PDFs of the rising edge and the falling edge of CK_{IN} never overlap with the PDF of CK_{S} at the same time.

The probability $P_{1,1}(\Delta t^{n}_{\text{edge}})$ for an edge position n to be the first one to detect a one is the product of the probability of the edge n to detect a one and the product of the probabilities of all previous edges to detect a zero

$$P_{1,1}(\Delta t^{n}_{\text{edge}}) = P_1(\Delta t^{n}_{\text{edge}}) \prod_{x=n+1}^{n_{\max}} [1 - P_1(\Delta t^{x}_{\text{edge}})]. \tag{3.21}$$

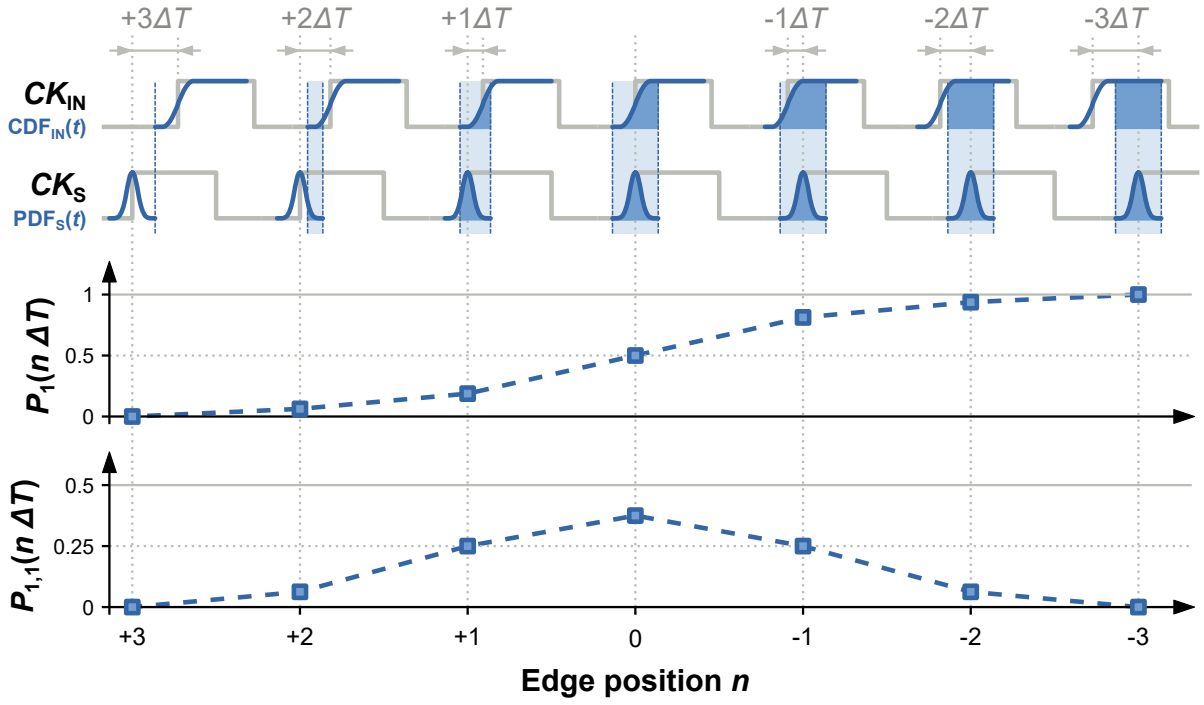


Figure 3.12: Illustration of the procedure to determine the probability distribution $P_{1,1}$ of the PDC measurement phase error due to jitter.

Figure 3.12 illustrates the calculation of $P_1(\Delta t^n_{\text{edge}})$ and $P_{1,1}(\Delta t^n_{\text{edge}})$ using a timing diagram of CK_{S} and CK_{IN} with annotated $CDF_{\text{IN}}(t)$ and $PDF_{\text{S}}(t)$. $P_{1,1}(\Delta t^n_{\text{edge}})$ describes the histogram of the detected position of the zero to one transition in terms of PDC phase steps ΔT. For both, phase and duty-cycle measurements the difference between a start and a stop transition time is evaluated as the final PDC measurement result. The two transition times are assumed to be statistically independent. The PDF of the difference of two independent random variables A and B can be calculated by the convolution of the PDFs of A and B

$$\text{PDF}_{\text{B–A}}(z) = \int_{-\infty}^{\infty} \text{PDF}_{\text{B}}(x)\,\text{PDF}_{\text{A}}(x-z)\,\text{d}x. \tag{3.22}$$

Using the discrete probability densities $P^a_{1,1}(n)$ and $P^b_{1,1}(n)$ of the independent input signals the resulting PDC output probability density is

$$\text{PDF}_{\text{PDC}}(\Delta t^m_{\text{edge}}) = \sum_{m=-2n_{\max}}^{2n_{\max}} \frac{1}{4n_{\max}} P^b_{1,1}(\Delta t^n_{\text{edge}})\, P^a_{1,1}(\Delta t^{n-m}_{\text{edge}}). \tag{3.23}$$

$\text{PDF}_{\text{PDC}}(\Delta t^m_{\text{edge}})$ describes the histogram of the output values of the PDC. The standard deviation

$$\sigma_{\text{PDC}} = \sqrt{\sum_{m=-2n_{\max}}^{2n_{\max}} \left(\Delta t^m_{\text{edge}} - \overline{\Delta t^m_{\text{edge}}}\right)^2 \text{PDF}_{\text{PDC}}(\Delta t^m_{\text{edge}})}. \tag{3.24}$$

of $\text{PDF}_{\text{PDC}}(\Delta t^m_{\text{edge}})$ can be used to model the behavior of the PDC under known jitter conditions. This is particularly useful for system level modeling and can also serve as a tool for defining jitter constraints for the clock generation circuits.

Figure 3.13 shows the calculated σ_{PDC} at different amounts of clock jitter σ_{IN} and σ_{S} for a PDC with $N = 64$ and $M = 1$. The results show that σ_{PDC} has a symmetric behavior with respect to σ_{IN} and σ_{S} and is constant for small jitter values. The minimum standard deviation of

$$\min(\sigma_{\text{PDC}}) = \frac{1}{\sqrt{2}} \approx 0.707 \tag{3.25}$$

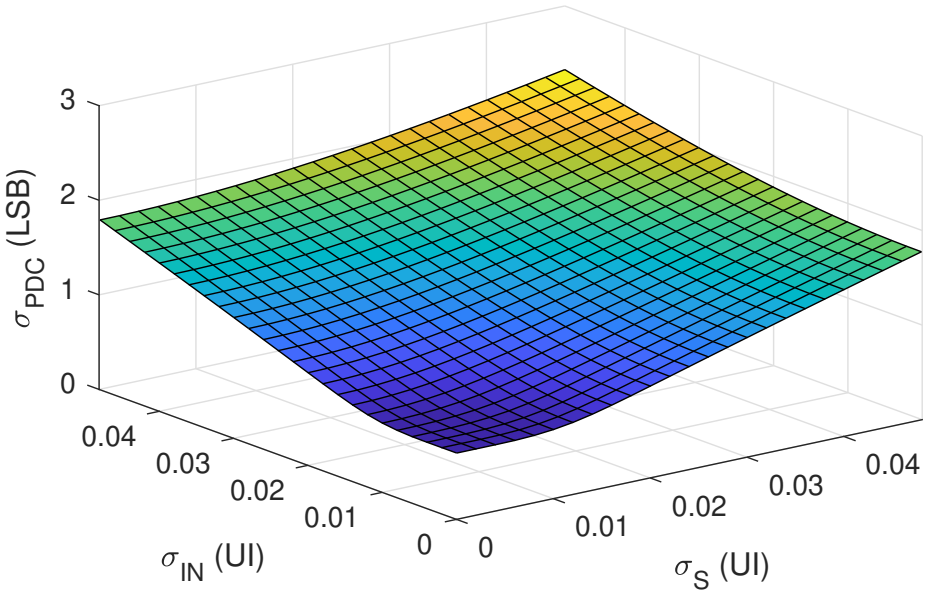


Figure 3.13: Standard deviation σ_{PDC} of the PDC measurement result for different RMS input clock jitter σ_{IN} and sample clock jitter σ_{S} with $N = 64$ and $M = 1$.

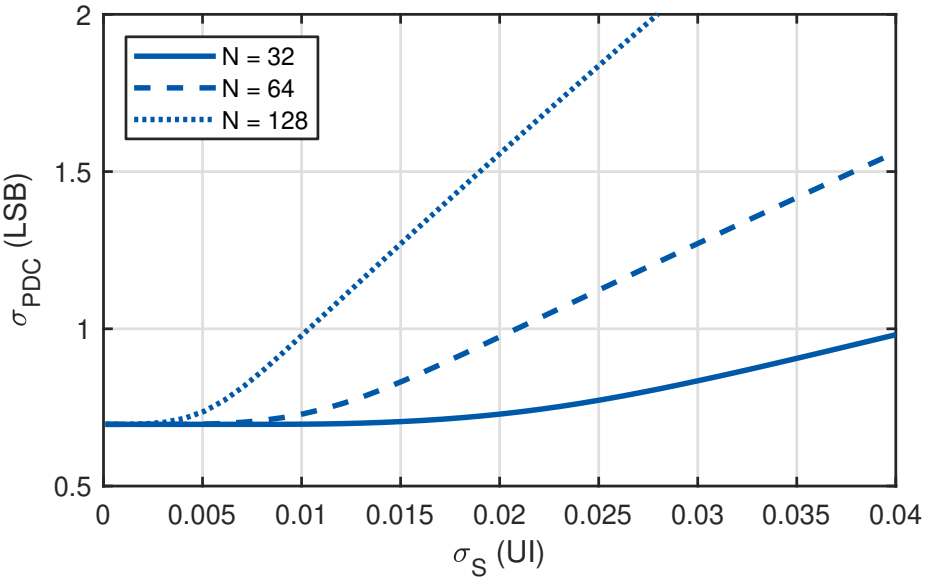


Figure 3.14: Standard deviation σ_{PDC} of the PDC measurement result for a constant RMS input clock jitter of $\sigma_{\mathrm{IN}} = 0.02\,\mathrm{UI}$ with $N = 32, N = 64, N = 128$ and $M = 1$.

stems from the case in which the start and stop transition times only toggle between two neighboring edge positions. The resulting PDF $P_{1,1}(\Delta t^n_{\mathrm{edge}})$ is then an even distribution between two values of $\Delta t^n_{\mathrm{edge}}$. The convolution of two such PDFs has the probabilities

$$\begin{pmatrix} \mathrm{PDF}_{\mathrm{PDC}}(\Delta T) \\ \mathrm{PDF}_{\mathrm{PDC}}(0) \\ \mathrm{PDF}_{\mathrm{PDC}}(-\Delta T) \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{2} \\ \frac{1}{4} \end{pmatrix} \tag{3.26}$$

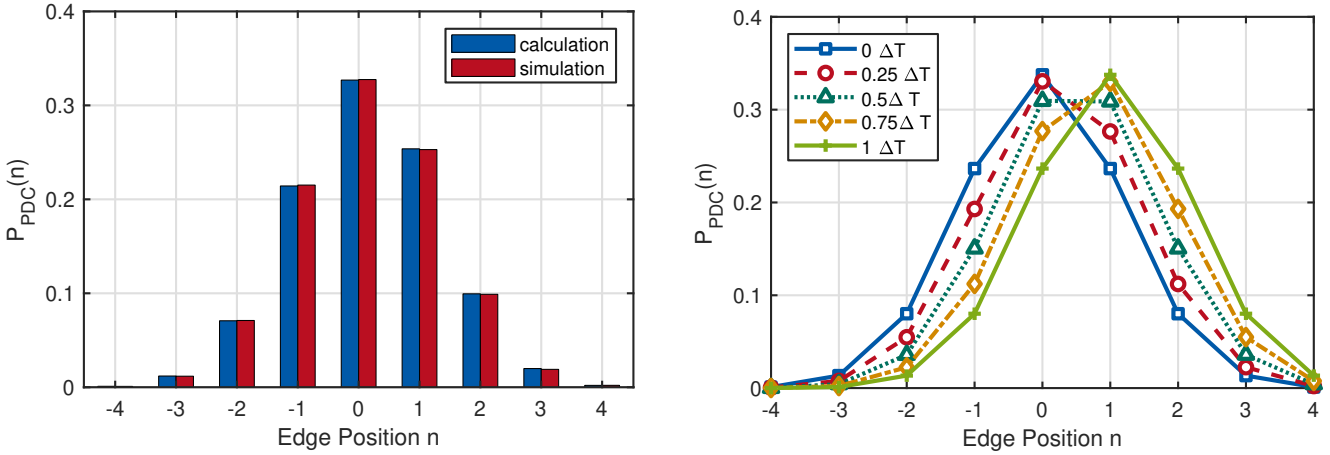
with a standard deviation of $\sigma_{\mathrm{PDC}} = \sqrt{\frac{1}{4} + \frac{1}{4}}$.

Outside of the constant region σ_{PDC} shows a linear dependence on either σ_{S} or σ_{IN} when one of the two is held constant. The transition between the constant and the linear region, as well as the slope of the linear region, depend on N as shown in Figure 3.14 for a constant input clock jitter of $\sigma_{\mathrm{IN}} = 0.02\,\mathrm{UI}$ and three different PDC resolutions $N = 32, N = 64$ and $N = 128$. An approximation for the transition between constant and linear part can be found by using the point at which the 3σ-width of the sample clock jitter is equal to the PDC step size ΔT. This way a specification for the sample clock jitter based on the constraint of a minimized standard deviation σ_{PDC} of the PDC can be defined as

$$\sigma_{\mathrm{S}} < \frac{1}{3 f_{\mathrm{IN}} N}. \tag{3.27}$$

Due to the linear relationship of σ_{PDC} and input clock jitter for sufficiently high values of σ_{IN} and σ_{S} the PDC can also serve as a jitter measurement circuit as long as one of the two clocks has a known and constant jitter. The calculation method for the PDC performance is verified using a behavioral simulation of the Verilog description of the PDC. Figure 3.15 (a) shows a comparison of the PDC output histogram $\mathrm{PDF}_{\mathrm{PDC}}(\Delta t^m_{\mathrm{edge}})$ for a clock jitter of $\sigma_{\mathrm{IN}} = \sigma_{\mathrm{S}} = 0.02\,\mathrm{UI}$ generated by the calculation method (equation (3.23)) and the behavioral simulation.

A positive side effect of the influence of clock jitter is that the PDC is able to resolve phase steps smaller than the nominal resolution ΔT by evaluating the mean of the output histogram. In Figure 3.15 (b) the histogram of the PDC output is plotted for an input phase difference shifted by increments of $\frac{\Delta T}{4}$. The mean of the histogram moves according to the input phase difference. Averaging multiple samples



(a) Verification against behavioral simulation.

(b) Input phase difference smaller than ΔT.

Figure 3.15: Histogram of the PDC output $\mathrm{PDF}_{\mathrm{PDC}}(\Delta t^{m}_{\mathrm{edge}})$ for a clock jitter of $\sigma_{\mathrm{IN}} = \sigma_{\mathrm{S}} = 0.02\,\mathrm{UI}$ generated by the calculation method (equation (3.23)) and the behavioral simulation of the PDC (a) and calculated for input phase increments of $\frac{\Delta T}{4}$ (b).

of the PDC output can therefore improve the resolution and is more efficient than increasing N since the averaging runs on the low frequency PDC output clock CK_{PDC}, while the PDC counter runs on the high-frequency clock CK_{S}.

3.3 Implementation

The PDC consisting of a conversion circuit and a measurement circuit is implemented using a fully digital standard cell based design flow. Implementation details and design considerations are presented in the following sections.

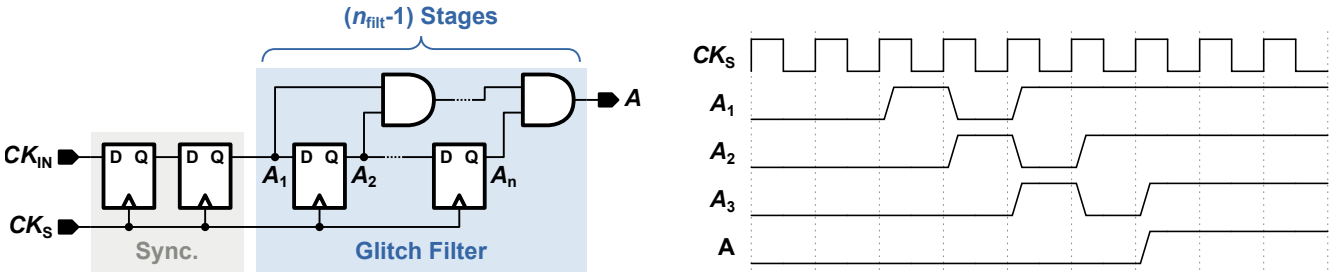
3.3.1 Conversion Circuit

The conversion circuit performs the asynchronous sampling of the input clock(s). Due to the asynchronous relationship between input clock and sample clock the setup and hold times of the sampling flip-flops will be violated regularly. Cascaded D-flip-flops are used to avoid metastability issues and ensure stable output signals. When the jitter on the input and sample clocks exceeds the area in which $\sigma_{PDC} = \frac{1}{\sqrt{2}}$ the maximum deviation of the clock edge positions is bigger than one LSB phase step ΔT. This can cause the sampled signals to generate glitches at the transitions where the signal toggles between one and zero for a few cycles before it settles to a new value. In order to remove these glitches and generate a clean transition for the control of the counter a glitch filter is added to the conversion stage. This glitch filter only generates a zero to one transition after n_{filt} consecutive ones of the sampled signal. The required size can be calculated using the peak-to-peak jitter of the input and sample clocks $TJ_{\mathrm{pk-pk,IN}}$ and $TJ_{\mathrm{pk-pk,S}}$ as

$$n_{\mathrm{filt}} = \left\lceil (TJ_{\mathrm{pk-pk,IN}} + TJ_{\mathrm{pk-pk,S}})\, N\, f_{\mathrm{IN}} \right\rceil. \tag{3.28}$$

Figure 3.16 (a) shows the schematic of a conversion circuit with double-flip-flop synchronizer and glitch filter. An example timing diagram of a glitch filter with $n_{\mathrm{filt}} = 3$ is given in Figure 3.16 (b).

Local variations of the setup and hold times of the sampling flip-flops can have an impact on the measurement result. The phase measurement can be affected by a mismatch between the sampler characteristics of the two clock signals to be measured. The influence also depends on the slope of the clocks and can be reduced using full custom clocked comparators with offset cancellation as the first sampling stage. In section 6.3.1 the impact in the application for the timing center calibration is analyzed



(a) Schematic. **(b)** Timing diagram.

Figure 3.16: Schematics of the conversion circuit with glitch filter (a) and timing diagram of the glitch filter for $n_{\text{filt}} = 3$ (b).

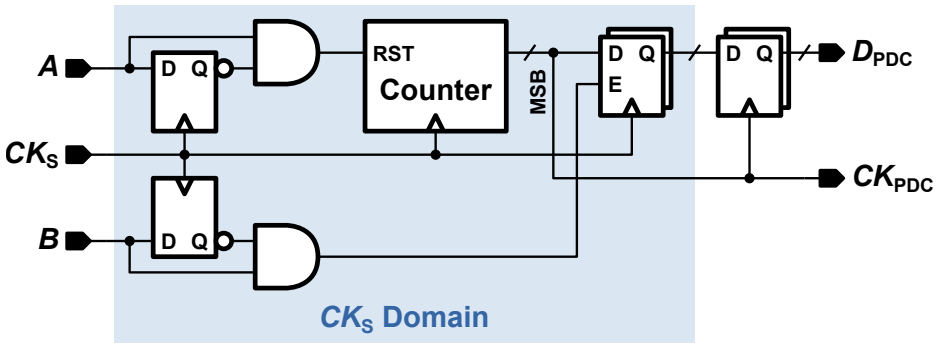
and a circuit for minimizing the phase measurement error is presented. For the duty-cycle measurement the sampler non-idealities can be compensated by measuring the inverted clock signal as well [65].

3.3.2 Measurement Circuit

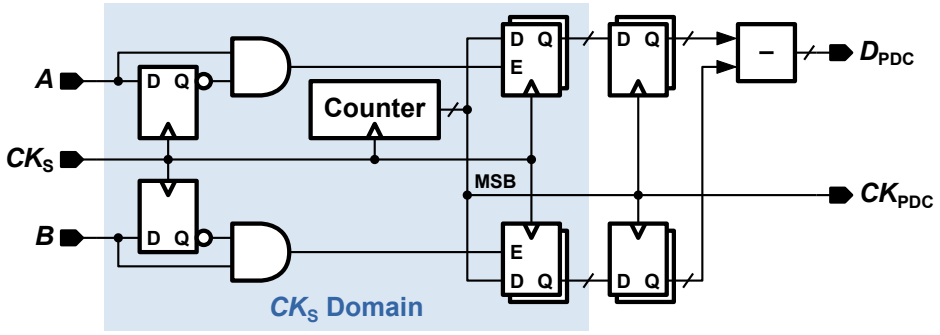
Two variants of the measurement circuit for the PDC are shown in Figure 3.17. Both designs use one-shot generators that create a single pulse at the rising edge of the sampled signal A or B for the control of the counter evaluation. The counter itself is continuously running with overflow and has a size of

$$n_{\text{cnt}} = \lceil \log_2(N) \rceil. \tag{3.29}$$

The implementation of the PDC is independent of the scaling factor M (cf. equation (3.3)) and therefore the sample clock frequency can be adjusted during operation without any modifications to the circuit. The



(a) Area optimized.



(b) Speed optimized.

Figure 3.17: Schematics of the measurement circuit optimized for area consumption (a) and for clock speed (b).

area optimized circuit in Figure 3.17 (a) resets the counter on the rising edge of A and stores the counter value to a register at the rising edge of B. The most significant bit (MSB) of the counter output is used as clock signal CK_{PDC} with $f_{\mathrm{PDC}} = \frac{f_{\mathrm{S}}}{N}$ for further processing of the PDC measurement results. The speed optimized circuit in Figure 3.17 (b) can run at higher sample clock frequencies f_{S} since the counter does not implement a reset, but is limited to resolutions that are a power of two ($N = 2^{n_{\mathrm{cnt}}}$). Therefore two registers are required to store the counter values at the rising edges of A and B that need to be subtracted after re-sampling in the CK_{PDC} domain. A disadvantage of the area optimized circuit is that CK_{PDC} can have a varying duty-cycle due to timing variations of the reset pulse caused by jitter on the clock signals. The speed optimized circuit always provides a stable CK_{PDC} with a duty-cycle of 50 % by using a free running counter. For this reason the speed optimized circuit is used for all PDC implementations in the scope of this work. In order to reduce the variance of the measurement results the PDC can be extend by additional filtering. Averaging of n_{avg} for example reduces the standard deviation of the PDC samples by a factor of $1/\sqrt{n_{\mathrm{avg}}}$.

3.3.3 Sample Clock Generation

In most high-speed I/O links the clock used for transmitting and receiving the data CK_{SYS} is generated from a low frequency reference clock CK_{REF} using a PLL. Since the sample clock CK_{S} for the PDC has a fixed frequency ratio to CK_{SYS} according to equation (3.3) it must be locked to the same reference clock. Figure 3.18 shows three possible PLL configurations for the generation of CK_{SYS} and CK_{S} with different implications on the PLLs and on the frequency of CK_{REF}. The configuration in Figure 3.18 (a) uses two integer PLLs where the PLL for CK_{SYS} implements the denominator of the frequency ratio ($MN + 1$) and the PLL for CK_{S} implements the numerator (N). A counter based feedback divider is required for the generation of CK_{SYS}, while a divider based on divide-by-2 circuits is sufficient for CK_{S} if N is a power of two. A drawback of this configuration is that the reference frequency is determined by the PDC parameters N and M. This can be conflicting with other system constraints such as other clock frequencies derived from the same reference clock and the availability of a crystal oscillator with the matching frequency $f_{\mathrm{REF}} = \frac{1}{MN+1} f_{\mathrm{SYS}}$. Also a dynamic adjustment of the PDC sample clock by changing the scaling factor M would require a change of f_{ref} which is not a practicable solution. The concept in Figure 3.18 (b) avoids these drawbacks by using a fractional PLL for the generation of CK_{S}. This way f_{REF} can be chosen independent of the PDC parameters and M can be adjusted only by making changes to the fractional PLL feedback ratio. A fractional PLL might not always be a desirable solution as it is more complex to implement and requires more chip area than an integer PLL. In order to avoid the use of a fractional PLL the configuration in Figure 3.18 (c) uses a clock divider with a division ratio of ($MN + 1$) to generate the reference clock for the sample clock PLL from the system clock. This way both PLLs can be integer PLLs with feedback ratios that are a power of two. The scaling of the sample clock is realized by changing the division ratio of the divider circuit. Another advantage is that the concept can also be used to generate CK_{S} in ICs that are directly supplied with the high-frequency clock CK_{SYS} like DRAM ICs for example.

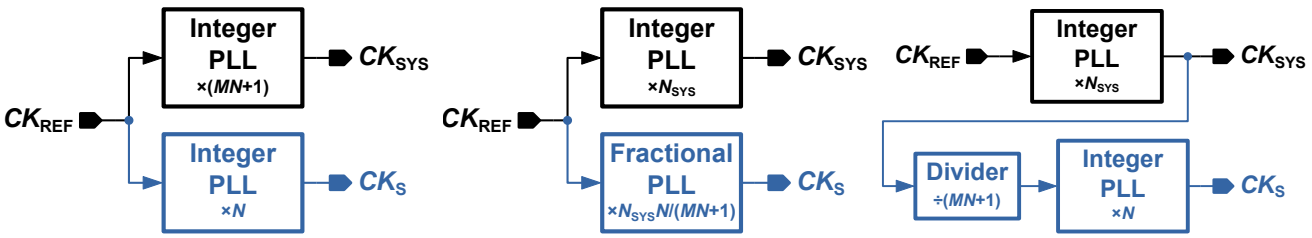


Figure 3.18: Block diagrams of possible sample clock generator configurations using only integer PLLs (a), a fractional PLL (b) and integer PLLs with a clock divider (c).

3.4 Measurement Results

The performance of the PDC is verified on a Xilinx Virtex-6 FPGA using the integrated PLLs for clock generation as illustrated by the block diagram in Figure 3.19. The design is implemented with a resolution of $N = 64$ and a glitch filter size of $n_{\text{filt}} = 4$. Table 3.2 lists the required FPGA resources of the PDC for the two measurement circuit options presented in section 3.3.2 in terms of slices, registers and lookup tables (LUT). The speed optimized circuit requires about 21 % more resources than the area optimized circuit. For the performance analysis a PI is used to generate a phase shifted version $CK_{\text{IN,B}}$ of the input clock $CK_{\text{IN,A}}$ and the PDC measures the phase difference between the two signals. Since only integer PLLs are available on the Virtex-6 FPGA the configuration in Figure 3.18 (a) is used for the clock generation with a scaling factor of $M = 1$. Three different input clock frequencies 650 MHz, 325 MHz and 162.5 MHz are tested by varying the reference clock frequency between 10 MHz, 5 MHz and 2.5 MHz. The sample clock frequency is then 640 MHz, 320 MHz and 160 MHz, respectively. The measured RMS jitter at $f_{\text{IN}} = 650\,\text{MHz}$ and the initial PI setting (0°) is $\sigma_{\text{IN}} = 0.052\,\text{UI}$ and $\sigma_{\text{S}} = 0.06\,\text{UI}$. Due to the contribution of the additional routing and off-chip driver these values are higher than the actual jitter present at the inputs of the PDC. The calculation method in section 3.2.2 therefore overestimates the standard deviation of the PDC output values with $\sigma_{\text{PDC}} = 2.56\,\text{LSB}$.

Figure 3.20 (a) shows the phase error of a PDC measurement with 10000 averaged samples compared to an oscilloscope measurement across the PI adjustment range. The maximum phase error increases with increasing clock frequency which is due to the increasing relative jitter $\frac{TJ}{T_{\text{IN}}}$ and the decreasing PDC

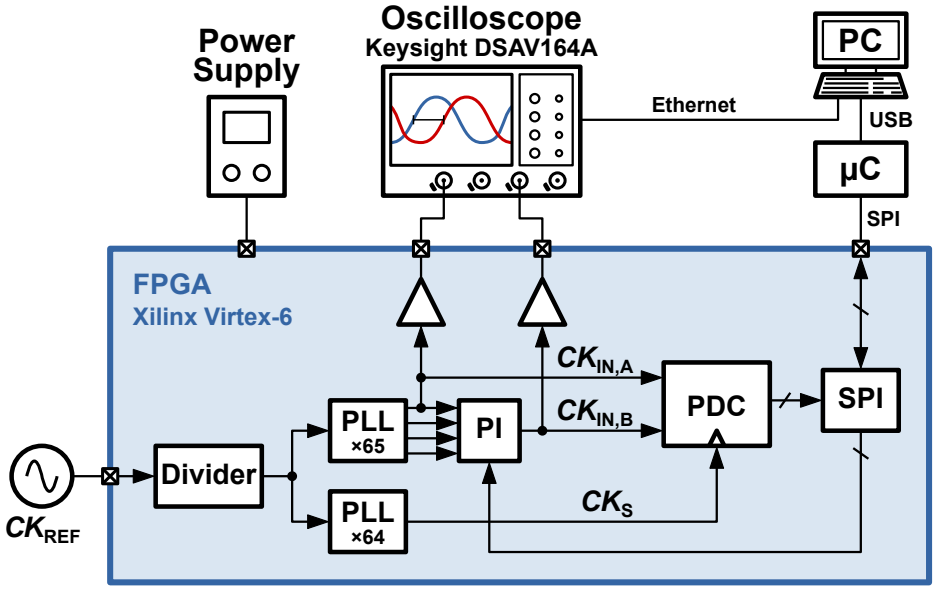
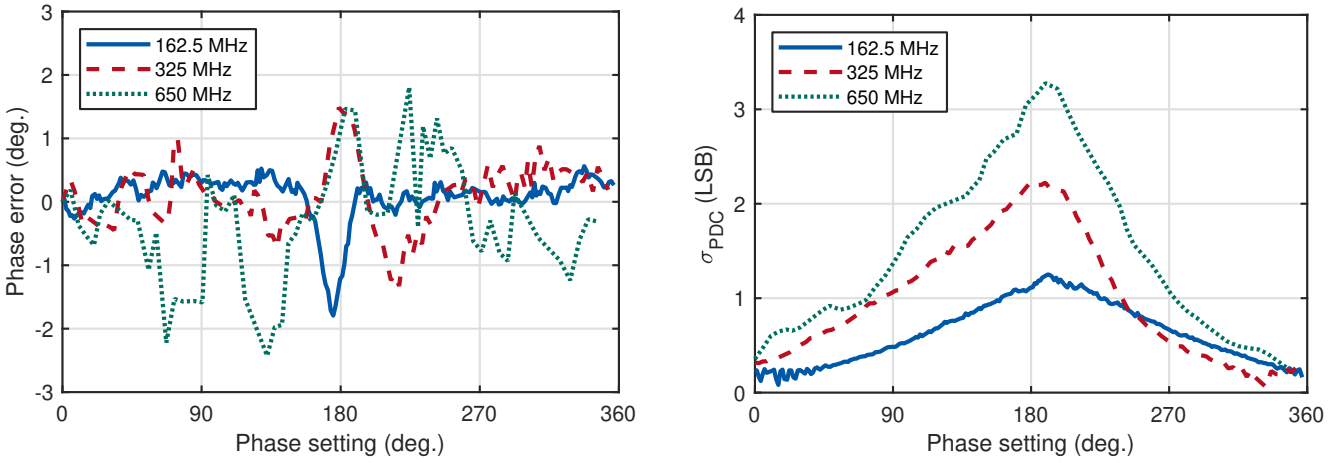


Figure 3.19: Block diagram of the measurement setup for the evaluation of the PDC performance on a Xilinx Virtex-6 FPGA.

Table 3.2: FPGA resources required for the PDC using the two different measurement circuit options shown in Figure 3.17, a resolution of $N = 64$ and a glitch filter size of $n_{\text{filt}} = 4$.

Configuration	Slices	Registers	LUTs
Area optimized	12	33	12
Speed optimized	14	39	16

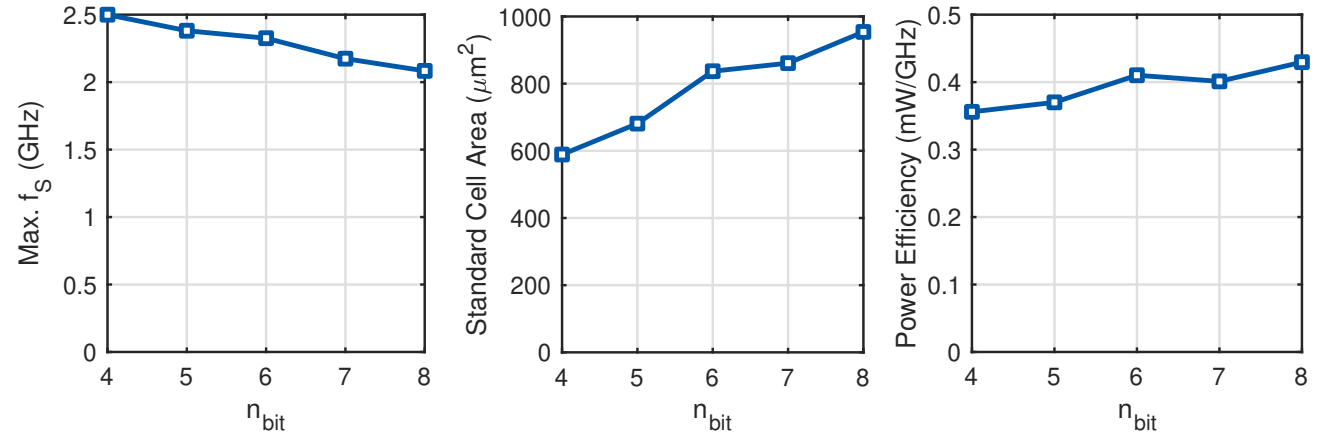


(a) Phase error. **(b)** Phase standard deviation.

Figure 3.20: Measured phase error (a) and standard deviation σ_{PDC} (b) of the PDC implemented on a Virtex-6 FPGA with $N = 64$ and $M = 1$ across the PI range for three different input clock frequencies f_{IN}.

resolution $\Delta T = \frac{T_{\mathrm{IN}}}{N}$. At around 180° the absolute phase error is strongly increased in all measurements. The maximum observed phase error at $f_{\mathrm{IN}} = 650\,\mathrm{MHz}$ is 2.43° (0.0135 UI). The standard deviation of the 10000 PDC samples in Figure 3.20 (b) at the initial PI setting of 0.36 LSB is much lower than the overestimated value based on the external jitter measurements. It is even below the minimum value of 0.707 LSB of the worst case estimation using the calculation method in section 3.2.2. All three measurements of σ_{PDC} show a significant increase towards 180° indicating that the jitter of the PI depends on the tap position of the eight stage voltage controlled oscillator.

The PDC is implemented in a 65 nm low-leakage CMOS process (UMC 65 LL) using a standard cell library provided by UMC for analysis of the required area and power consumption and the maximum possible sample clock frequency in an ASIC implementation. Due to the small design the provided wire load models overestimate the loading of the cells and therefore a manual tweaking of the design constraints for the synthesis is required to maximize the sample clock frequency. In Figure 3.21 a comparison of the maximum f_{S}, the standard cell area and the power efficiency (related to f_{S}) after



(a) Maximum f_{S}. **(b)** Cell area. **(c)** Power efficiency.

Figure 3.21: Maximum sample clock frequency f_{S} (a), standard cell area (b) and power efficiency (c) of the PDC synthesized for different resolutions $N = 2^{n_{\mathrm{bit}}}$ with $M = 1$ and $n_{\mathrm{filt}} = 4$.

Table 3.3: Performance of the PDC after place and route in the UMC 65 LL CMOS process for a resolution of $N = 64$ ($n_{\text{bit}} = 6$) and an input clock frequency of $f_{\text{IN}} = 2.5\,\text{GHz}$.

Process	65 nm		
Supply voltage	1.2 V		
Input clock f_{IN}	2.5 GHz		
Resolution N	64		
Core area	1100 μm^2		
Scaling Factor M	1	2	4
Sample clock f_{S}	2.46 GHz	1.24 GHz	0.62 GHz
Sample rate f_{PDC}	38.46 MS/s	19.38 MS/s	9.73 MS/s
Power	1.72 mW	0.69 mW	0.35 mW
Efficiency	0.7 mW/GHz	0.57 mW/GHz	0.56 mW/GHz

synthesis for five different PDC resolutions n_{bit} and a glitch filter size of $n_{\text{filt}} = 4$ is shown. The standard cell area does not represent the area of the PDC after place and route since the clock tree is not yet included and a certain overhead must be added for the place and route. All three properties show a nearly linear relationship to the number of bits n_{bit} of the PDC counter and a sample clock frequency of $f_{\text{S}} > 2\,\text{GHz}$ is feasible for a PDC resolution of up to 8 bit. The power efficiency of the design is between 0.36 mW/GHz and 0.43 mW/GHz. The main cost for increasing the resolution is a larger silicon area while the impact on the power consumption is less pronounced.

Table 3.3 lists the PDC performance after place and route for a resolution of $N = 64$, an input clock frequency of $f_{\text{IN}} = 2.5\,\text{GHz}$ and different scaling factors M. The design requires only 1100 μm^2 of active area and consumes up to 1.72 mW with a sample rate of up to 38.46 MS/s depending on the scaling factor M. Increasing M enables the PDC to be operated at far below 1 mW and also improves the power efficiency by 25 % for $M > 1$.

3.5 Conclusion

The proposed synthesized PDC enables precise phase and duty-cycle measurements of high-speed clocks with minimal area overhead and scalable resolution, sample rate and power consumption. Table 3.4 compares the properties of the PDC to selected state-of-the-art all-digital phase measurement circuits presented in section 3.1. The silicon area of the circuits is compared using the normalized area FoM

$$a_{\text{FoM}} = \frac{\frac{a}{L_{\text{min}}^2}}{\frac{a_{\text{PDC}}}{(65\,\text{nm})^2}} \tag{3.30}$$

with the design area a divided by the minimum gate length of the underlying CMOS process L_{min}, normalized to the area FoM of the PDC. It is noted that the area in [3] is only given as the standard cell area after synthesis and therefore the actual silicon area after place and route is larger by approximately 30-40 %. Using this assumption the PDC has the smallest active area in the comparison and provides the best flexibility. Sample rate, resolution and power consumption are scalable during operation using the scaling factor M and the optional averaging of the output values. Other asynchronous sampling methods only provide scalable sample rate and resolution by adjustment of the measurement duration, while TDC based methods are the least flexible with all properties being fixed. The term ”fixed” in this context means that the property cannot be modified after implementation but may depend on the measurement conditions (e.g. the input phase difference).

Table 3.4: Comparison of the proposed synthesized PDC and selected state-of-the-art all-digital phase measurement circuits in terms of flexibility during operation.

Design	Method	Area FoM	Sample rate	Resolution	Power
PDC	Async. sampling	1	Scalable	Scalable by averaging	Scalable
[2]	Async. sampling	3.77	Scalable	Scalable by measurement duration	Fixed
[3]	Async. sampling	0.77*	Scalable	Scalable by measurement duration	Fixed
[60]	TDC	1.19	Fixed	Fixed	Fixed
[61]	TDC	9.12	Fixed	Fixed	Fixed

* Standard cell area only

Overall the measurement results show the potential of the proposed PDC for efficient phase and duty-cycle measurements in high-speed parallel I/O links. The all-digital implementation together with the flexibility make it a promising candidate for many applications with fast implementation and good portability between technologies. The frequency of the clock signals to be measured is only limited by the performance of the first sampling stage, which can be optionally implemented as a full-custom clocked comparator.

4 Design of Delay-Locked Loops for Multiphase Clock Generation Using a Phase-to-Digital Converter

A major contributor to the overall power consumption of high-speed I/O links is the clock distribution. The phase alignment of parallel data links is frequently solved by the use of phase interpolators (PI) requiring a multiphase clock signal. Distributing a multiphase clock signal to the individual transceivers can be very power consuming and requires a careful layout design to maintain the phase relationship of the clock signals. Delay-locked loops (DLL) are widely used as local multiphase generators to avoid this problem and to keep the power consumption low. Therefore many efforts are put into the development of power efficient local multiphase DLLs with small area consumption. This chapter elaborates the design of multiphase DLLs based on the PDC presented in chapter 3. Two different delay cell architectures are analyzed and compared for their performance. Major parts of this chapter were published in [105] and [101] and the results presented in section 4.2.1 were published in [102].

4.1 Related Work

The general structure of a typical multiphase DLL consists of a delay line, a phase detector (PD) and a loop filter as shown in Figure 4.1. Most state-of-the-art implementations use a binary PD (see section 3.1.1) with a 360° feedback from the delay line. Some publications propose the use of a global DLL with a TDC based phase detector (see section 3.1.2) for coarse adjustment in combination with binary PDs for fine adjustment in the local DLLs [71, 72]. Since the typical PD architectures are already presented in section 3.1 the following sections focus on the delay line circuits and the loop filter architectures of multiphase DLLs found in the literature.

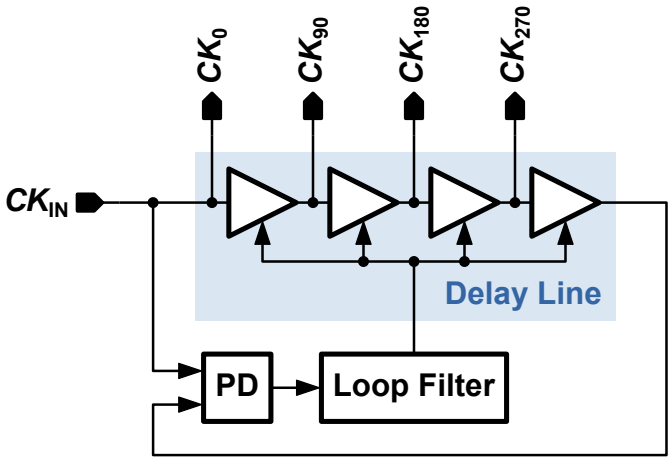


Figure 4.1: Block diagram of a typical multiphase DLL.

4.1.1 Delay Line Circuits

The delay line takes up a major part of the overall power consumption of a DLL. Several circuits have been proposed for delay cells in high-frequency delay lines with different advantages and disadvantages. An excellent overview of delay cell circuits for high-resolution delay lines can be found in [73]. This

section briefly presents the CMOS gate based delay cell circuits as they provide the best power and area efficiency, which is a major design goal of multiphase DLLs for high-speed parallel I/O links. Other delay cell types are based on CML buffers with active load [74, 75] or folded CMOS delay lines in combination with phase interpolators or fine delay elements (coarse/fine delay line) [71, 76].

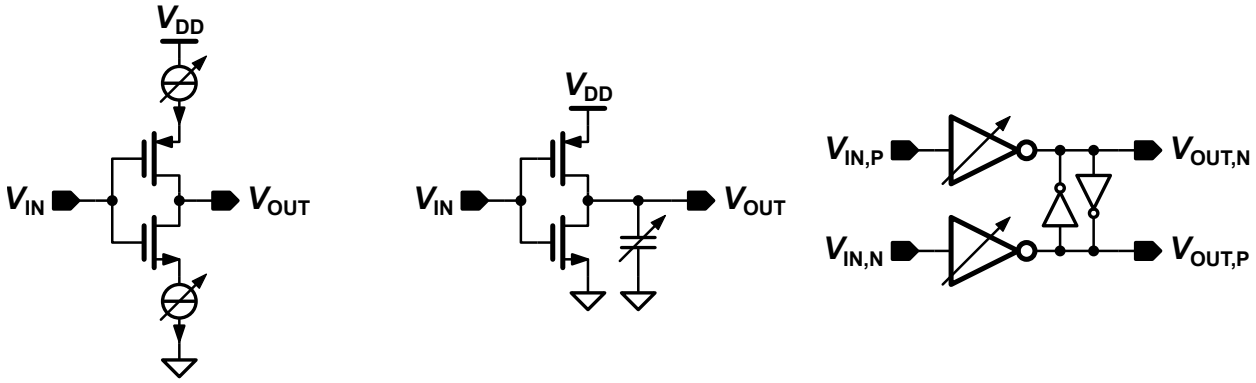
The general delay equation of a CMOS gate [77]

$$t_\mathrm{d} = \frac{2LC_\mathrm{L}V_\mathrm{DD}}{W\mu C_\mathrm{ox}(V_\mathrm{DD} - V_\mathrm{t})^\alpha} \tag{4.1}$$

depends on the transistor parameters gate length L, gate width W, carrier mobility μ, gate oxide capacitance C_ox, threshold voltage V_t and technology parameter α (used to express the carrier-velocity saturation effect), as well as the load capacitance C_L and the supply voltage V_DD. The simplified equation

$$t_\mathrm{d} \propto \frac{C_\mathrm{L}}{I_\mathrm{D}} V_\mathrm{DD} \tag{4.2}$$

based on [78] reflects the proportionality of t_d to the load capacitance C_L and the supply voltage V_DD and the inverse proportionality to the drain current I_D. This results in three possible control mechanisms for the design of adjustable delay cells based on CMOS inverters. The control of the drain current is the most common approach in multiphase DLLs by the use of current-starved inverter (CSI) delay cells [44, 79] for the delay line. In CSI delay cells the drain current of a CMOS inverter is limited by adjustable current sources at the NMOS and/or the PMOS transistor as shown in Figure 4.2 (a). Reducing the maximum drain current increases the transition time at the output of the CSI delay cell and results in an increased delay. The adjustable current sources are realized using NMOS or PMOS transistors with an adjustable gate voltage. The gate voltage is often generated using a voltage-to-current converter [80] or a current-DAC [79] with a minimum output current to avoid gate voltages below the threshold voltage of the transistors. Control of the load capacitance in the form of shunt-capacitor inverter (SCI) delay cells (see Figure 4.2 (b)) is also frequently found in multiphase DLL implementations [43, 81]. SCI based delay cells have the advantage of a linear delay control but require more area and power due to the additional load capacitance. Both, direct digital control through switchable capacitors and analog control through voltage controlled capacitors are possible [81]. According to [73] SCI delay cells also provide smaller delay resolution and better robustness against process and temperature variations than CSI delay cells. CMOS inverter based delay cells can also be used for differential delay lines using the pseudo-differential delay cell architecture in Figure 4.2 (c). Weak cross-coupled inverters are connected to the outputs of the two parallel adjustable CMOS delay cells and provide some inherent duty-cycle correction within the delay line. Control of the supply voltage is mostly used as a secondary delay adjustment to increase the



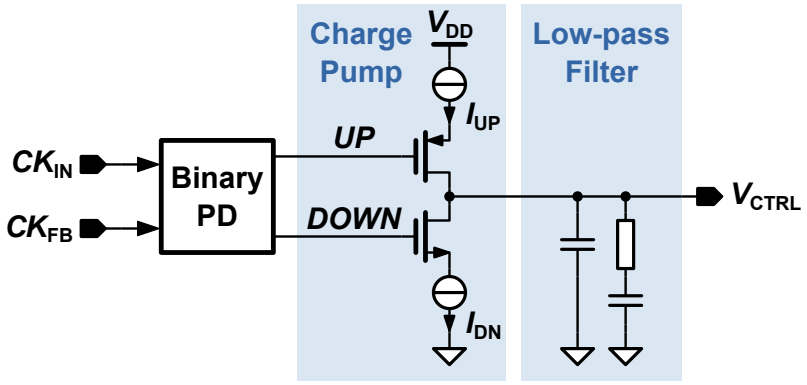
(a) CSI delay cell. **(b)** SCI delay cell. **(c)** Pseudo-differential delay cell.

Figure 4.2: Schematics of a CSI delay cell (a), an SCI delay cell (b) and a pseudo-differential delay cell (c).

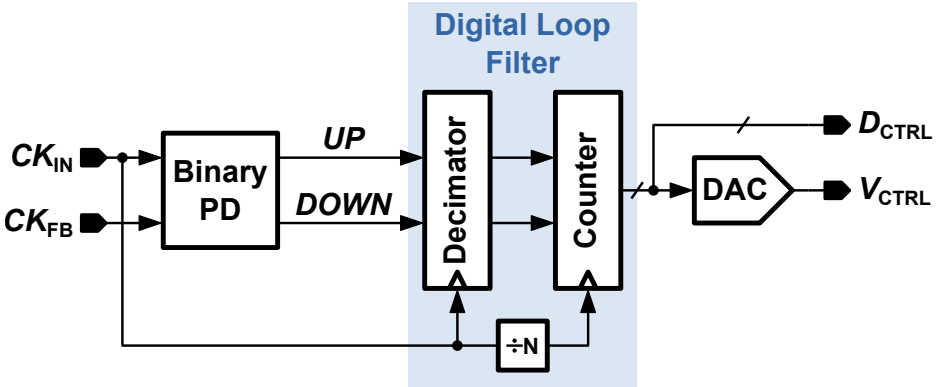
frequency range of multiphase DLLs [44, 43]. In most cases this is implemented as a static frequency band selection instead of a continuous feedback control as for the primary delay adjustment. Typical designs use low-dropout regulators (LDO) to control the supply voltage of the delay line. Another advantage of an LDO for the delay line is the suppression of supply noise and therefore a significant jitter reduction. But due to the large area requirement of the LDO itself only a few publications propose supply regulation for local multiphase DLLs.

4.1.2 Loop Filter Architectures

The classical approach for the loop filter of DLLs is to generate an analog control voltage V_{CTRL} from the phase detector signals UP and $DOWN$ using a charge pump and a low-pass filter (Figure 4.3 (a)). Drawbacks of an analog loop filter are the PVT sensitivity of the charge pump and the large area of the passive low-pass filter. Recent publications avoid these drawbacks by the use of a digital loop filter [43]. Figure 4.3 (b) shows the basic schematic that is common to all published digital loop filter implementations. The output signals of a binary PD are used to control a counter (sometimes called accumulator) and optionally a decimator or a demultiplexer in combination with a clock divider are employed to reduce the speed requirements of the counter in very high-frequency applications [44, 76]. When the feedback clock CK_{FB} lags the reference clock CK_{IN} the counter is incremented and otherwise it is decremented. The counter value can be either directly used as a control word for a digitally adjustable delay line or it can be converted to an analog control voltage V_{CTRL} using a DAC. In [81] both concepts are combined with a direct digital control for coarse delay adjustment and an analog control through a DAC for fine delay adjustment. In [76] a digital filter is added at the output of the counter to control



(a) Analog loop filter.



(b) Digital loop filter.

Figure 4.3: Schematics of a typical analog loop filter based on a charge pump and a passive low-pass filter (a) and a digital loop filter based on a decimator and a counter (b).

the loop bandwidth of the DLL. Another advantage of a digital loop filter is that the loop control can be disabled once the DLL is locked to remove clock jitter introduced by the dithering behavior of the digital control and to save power. Therefore a digital lock detection and an FSM are usually added to the loop filter [44, 81, 76].

4.2 Shunt-Capacitor Inverter Based Architecture

The intention of the SCI based multiphase DLL architecture is to minimize the amount of full-custom circuits and to follow a digital dominated design flow. Therefore the SCI based delay cells use a direct digital capacitance control and represent the only analog component in the DLL architecture. The proposed area optimized transmission gate based shunt-capacitors presented in section 4.2.1 offer very small capacitance increments and a low overall load capacitance which helps to minimize the power consumption of the SCI based DLL. A block diagram of the proposed DLL design with phase shifter is shown in Figure 4.4. The differential delay line consists of three pseudo-differential SCI delay stages. Since the PDC can measure any phase difference a 360° feedback is not required and therefore the delay line is shorter by one stage than conventional eight phase generators. Two dummy delay cells are used to match the driving at the first stage and the loading at the last stage in order to achieve the same delay characteristics for all delay cells. The 0° and the 135° outputs of the delay line are fed to a PDC for phase measurement. An additional measurement of the complementary 180° to 315° phase difference is performed to compensate sampler nonidealities as well as a possible duty-cycle error of the clock signals. The two alternating measurements are fed to a digital loop filter consisting of a subtractor to obtain the phase error to the ideal $\varphi_{\mathrm{DLL}} = 135°$ phase difference and a counter that accumulates the phase error. The output of the counter serves as a direct digital control word for the SCI delay cells. The digital loop filter is clocked by the output clock CK_{PDC} of the PDC.

The phase shifter consists of a 8:2 multiplexer (MUX) and a CMOS PI based on the design in [4]. The schematics of the implemented phase shifter are shown in Figure 4.5. The 8:2 MUX selects two adjacent phases from the eight equally spaced clock phases generated by the DLL. The PI consists of a thermometer coded array of 16 2:1 MUXes and generates an intermediate clock phase by mixing the two signals with a variable ratio. In total the phase shifter achieves a resolution of 7 bit. A drawback of CMOS based phase shifters is the sensitivity of the intrinsic delay through the MUXes to voltage and temperature variations. In order to compensate these variations during operation an optional phase

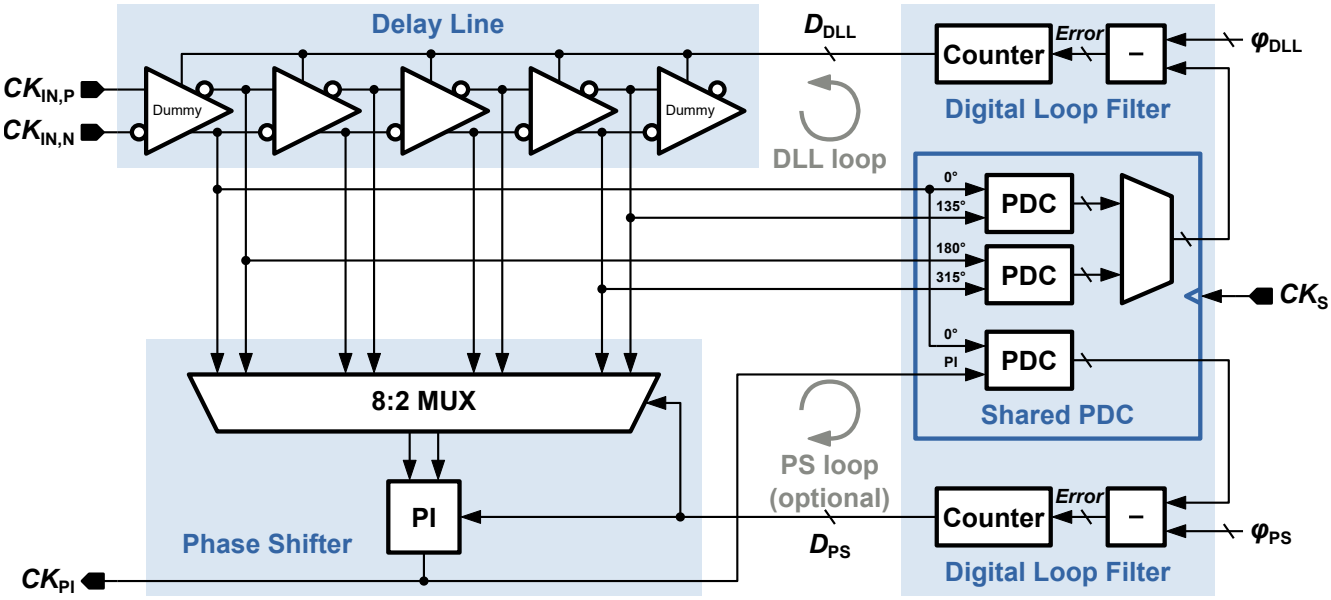


Figure 4.4: Block diagram of the proposed SCI based multiphase DLL with phase shifter.

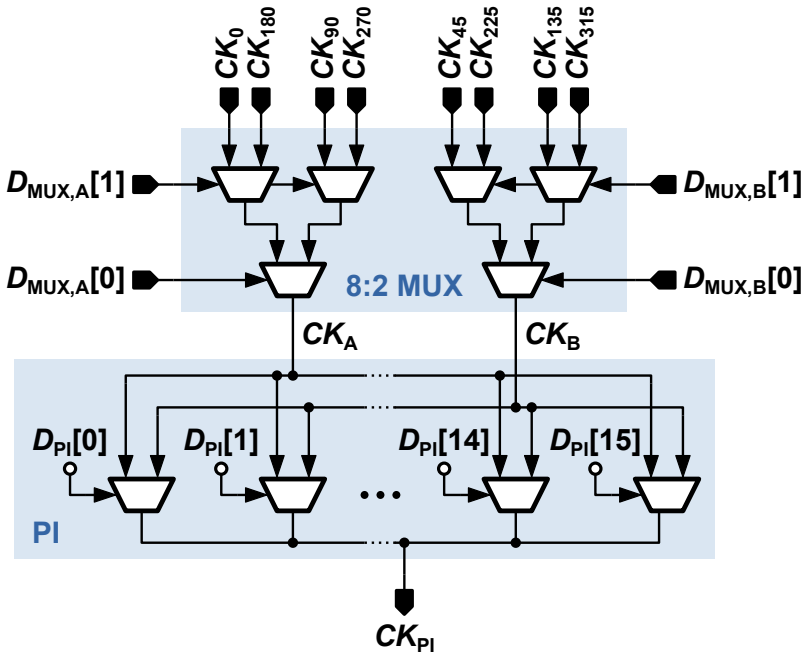
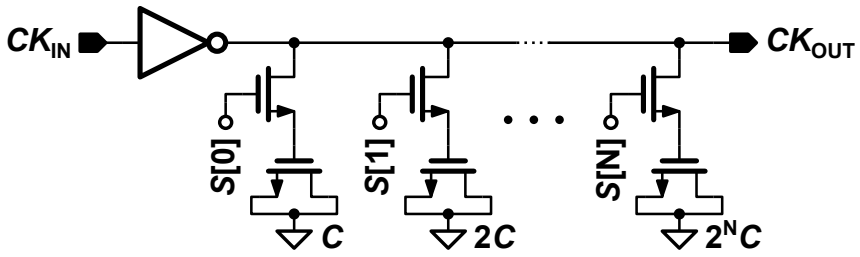


Figure 4.5: Schematic of the 7 bit CMOS phase shifter based on the design in [4].

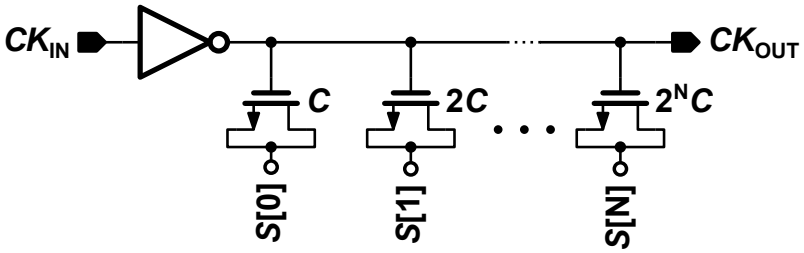
shifter control loop can be employed. Additionally, the control loop can compensate the nonlinearity of the phase interpolation by setting a target phase φ_{PS} for the phase shifter with respect to the 0° output of the DLL. The digital loop filter of the phase shifter control loop is implemented similarly to the DLL loop filter. A subtractor calculates the phase error to the target phase φ_{PS} and a counter accumulates the phase error to control the PI setting. All PDCs shown in Figure 4.4 are implemented with a resolution of $N = 64$ and share a single core counter to minimize the power consumption of the circuit. The clock signals CK_{IN} and CK_{S} are generated by external PLLs in the test setup and the required glitch filter size for the PDC at $f_{\mathrm{IN}} = 2.5\,\mathrm{GHz}$ was calculated to be $n_{\mathrm{filt}} = 2$. The PDCs are synthesized for a maximum sample clock frequency of $f_{\mathrm{S}} = 2.46\,\mathrm{GHz}$ allowing the design to be operated with a scaling factor down to $M = 1$.

4.2.1 Transmission Gate Based Shunt-Capacitor Inverter Delay Cells

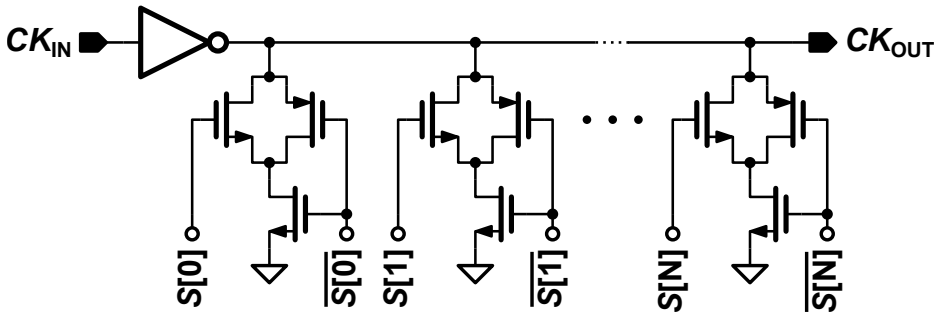
Different topologies for digitally controlled shunt-capacitor delay cells have been proposed in the literature. The topology in Figure 4.6 (a) consists of an NMOS or a transmission gate acting as a switch and a separate capacitor [81]. The binary scaled capacitances are usually implemented using metal-oxide-semiconductor (MOS) capacitors due to the better area efficiency compared to metal-insulator-metal (MIM) capacitors or metal-oxide-metal (MOM) capacitors despite their inferior process variation. A drawback of this approach is the significant contribution of the parasitic capacitance of the NMOS switch to the total load capacitance. Also, the switch transistors must have an increasing W/L ratio with an increasing load capacitance in order to provide a sufficiently low-ohmic path for the charging and discharging of the capacitors. Otherwise the delay adjustment becomes nonlinear since the larger capacitors cannot be fully charged or discharged within one clock period. The architecture in Figure 4.6 (b) uses MOS capacitors where the gate is connected to the delay cell output and the drain and source voltage can be switched between GND and V_{DD} [82]. This approach exploits the fact that the parasitic gate capacitances of a MOSFET depend on the operating region and therefore on the gate-source voltage. A drawback is that the switchable capacitance increment $\Delta C = C_{\mathrm{on}} - C_{\mathrm{off}}$ is smaller compared to the topology in Figure 4.6 (a). An effect that is observed when designing digitally controlled shunt-capacitors with separate switches is that the parasitic capacitance of the switch itself changes when the control signal is changed. The design in [83] makes use of this property by implementing only transmission gate switches without separate capacitors (Figure 4.6 (c)). The change in parasitic capacitance is used as the digitally controlled load



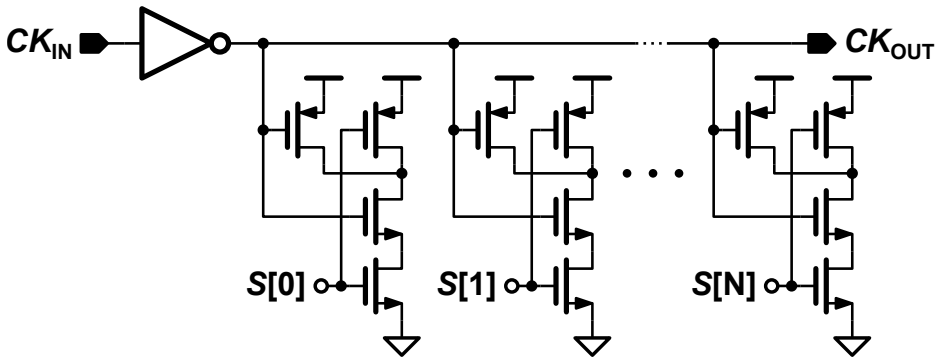
(a) Separate switch and capacitor [81].



(b) Voltage controlled capacitor [82].

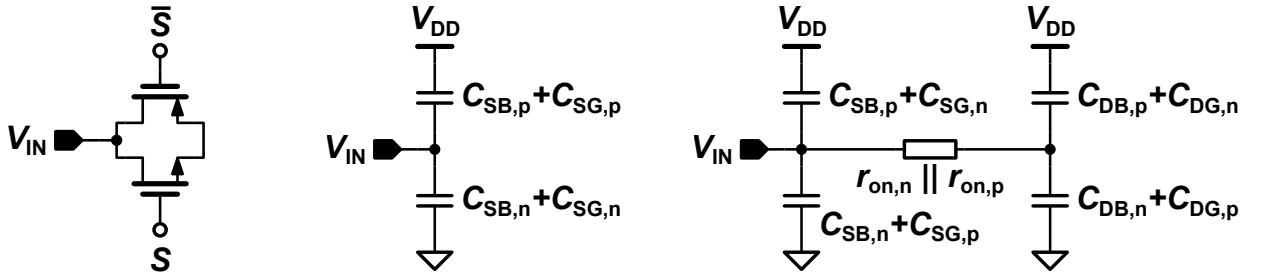


(c) Transmission gate with discharging transistor [83].



(d) NAND gate [84].

Figure 4.6: Schematics of SCI delay cells using digitally controlled capacitors based on a separate switch and capacitor (a), a voltage controlled capacitor (b), a transmission gate with discharging transistor (c) and a two-input NAND gate (d).



(a) Transmission gate. **(b)** Off state ($S = 0$). **(c)** On state ($S = 1$).

Figure 4.7: Schematic of a single side connected transmission gate for the use as a switchable capacitor (a) and equivalent schematics of the transmission gate for the on state (b) and the off state (c).

capacitance. An additional weak NMOS transistor discharges the open end of the transmission gate when it is turned off to avoid charge accumulation. The authors of [84] propose the use of two-input CMOS NAND gates as switchable load capacitors (Figure 4.6 (d)). The small change in parasitic capacitance at one of the NAND gate inputs when the other input is being switched between GND and V_{DD} is used to create small delay increments in the order of 1.55 ps. An advantage of this approach is that the design can be fully synthesized. In [85] the NAND gate approach is used to implement a digitally controlled oscillator for a synthesized PLL and a modified shunt-capacitor with an additional inverter between the main inverter and the NAND gate is introduced that provides an improved delay resolution of 0.4 ps. Both, the transmission gate based design in [83] and the NAND gate based design in [84, 85] use equally sized capacitors with a thermometer coded digital control making the designs quite area consuming.

In this section an improved version of the transmission gate based SCI delay cell is presented. Figure 4.7 shows the equivalent circuits of a transmission gate for the on state and the off state to illustrate how the parasitic capacitance seen at one port of the gate is changed. The discharging transistor proposed in [83] for the open node of the transmission gate is omitted to reduce the load capacitance. Neither simulations nor measurements have shown any negative impacts on the operation of the circuit by this action. When $S = 0$ the source-bulk capacitance $C_{SB,p}$ and the source-gate capacitance $C_{SG,p}$ of the PMOS transistor are connected between the input node and V_{DD} and the equivalent capacitances $C_{SB,n}$ and $C_{SG,n}$ of the NMOS transistor are connected between the input node and GND. When $S = 1$ the gate potentials of the NMOS and the PMOS transistor are inverted and the parallel on resistances $r_{on,n}||r_{on,p}$ connect the drain nodes with the parasitic capacitances $C_{DB,p}, C_{DG,n}, C_{DB,n}$ and $C_{DG,p}$ to the input node. For a symmetric swing of the clock signal and therefore a minimized duty-cycle distortion it is advantageous to have symmetrical capacitances to V_{DD} and to GND for both the on state and the off state. Therefore an identical sizing of the NMOS and the PMOS transistor is applied with a symmetric transistor layout in order to achieve equal parasitic capacitances at source and drain ($C_{DB} = C_{SB}$ and $C_{DG} = C_{SG}$). As a result the capacitance seen at the input node in the on state C_{on} is about twice the capacitance in the off state C_{off}. The parasitic capacitances as well as the on resistance of the transmission gate depend on the sizing of the transistors. Increasing the transistor width increases the parasitic capacitances and therefore the switchable capacitance $\Delta C = C_{on} - C_{off}$ and decreases the on resistance which is beneficial for the charging and discharging of the parasitic capacitances. The proposed transmission gate based SCI delay cell design makes use of this property by implementing binary scaled switchable capacitors instead of a thermometer coded array and targets a minimized load capacitance to improve the power efficiency of the delay cells. Figure 4.8 shows the simulated C_{on} and C_{off} of a transmission gate for different transistor gate widths $W_n = W_p$ and the three different threshold voltage options low V_t (LVT), regular V_t (RVT) and high V_t (HVT) of the UMC 65 LL process. For a minimized capacitance increment $C_{on} - C_{off}$ the minimum gate length $L_{min} = 60$ nm is chosen and will be used in the following for all transistors of the shunt-capacitor

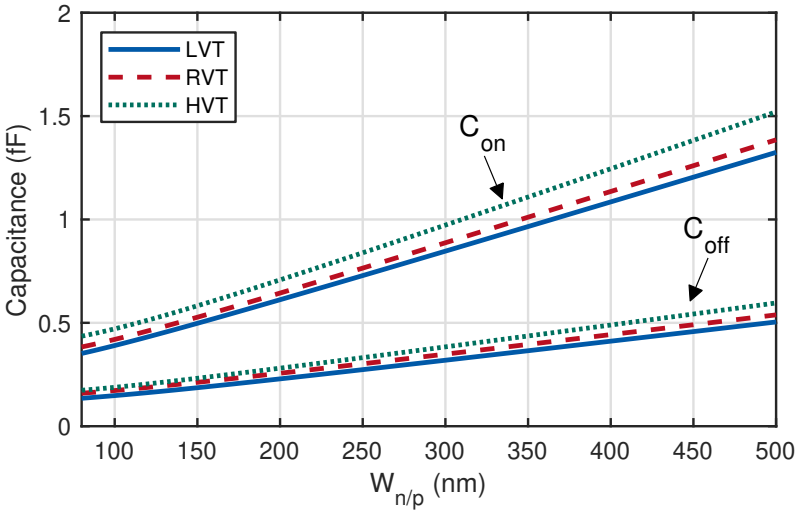


Figure 4.8: Simulated C_{on} and C_{off} of a transmission gate for different gate widths $W_n = W_p$, a gate length of $L_g = 60\,\text{nm}$ and three different threshold voltage options.

design. The capacitance values seen at the input of the transmission gate are obtained from a Z-parameter simulation at $f_{IN} = 2.5\,\text{GHz}$ using the equation

$$C_{in} = \frac{1}{2\,\pi\,f_{IN}\,\text{imag}(Z_{11})}. \tag{4.3}$$

All three V_t options show a linear dependency of C_{on} and C_{off} on the gate width except for the range $W_{n/p} < 120\,\text{nm}$. The nonlinear relationship for small gate widths can be explained by the different layout of the transistors in that range. The drain and source contacts have a minimum width constraint which is bigger than the minimum gate width of the transistors and therefore the size of the drain and source contacts is constant for $W_{n/p} < 120\,\text{nm}$ resulting in a lower slope of the simulated capacitance to gate width relationship. The smallest capacitances are obtained with the LVT transistors making them the preferred choice for the SCI delay cells.

The delay adjustment range of an SCI delay cell is defined by the minimum possible load capacitance $C_{L,min}$ and the maximum possible load capacitance $C_{L,max}$. For the application in a multiphase generator the strength of the inverter is adapted to the load capacitance for a nominal delay of $t_d = \frac{T_{in}}{8}$ ($\phi_d = 45°$) at a load capacitance setting of

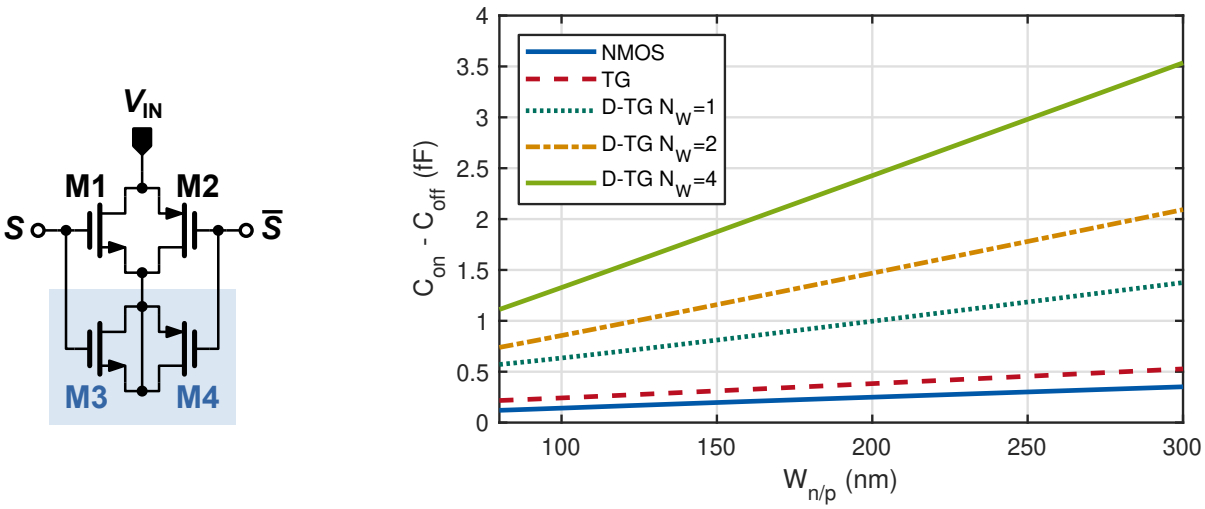
$$C_{L,mid} = C_{L,min} + \frac{C_{L,max} - C_{L,min}}{2}. \tag{4.4}$$

The DLL then has a symmetric delay margin around the nominal setting to cover PVT variations. The minimum load capacitance

$$C_{L,min} = C_o + C_l + C_g + \sum_{i=1}^{n_{bit}} C_{off,i} \tag{4.5}$$

of an SCI delay cell with a binary scaled shunt-capacitor array is composed of the output capacitance C_o of the inverter, the interconnect capacitance C_l, the gate capacitance C_g of the following delay stage and the off state capacitances $C_{off,i}$ of the n_{bit} transmission gate shunt-capacitors. For an ideal shunt-capacitor array with a uniform capacitance increment the maximum load capacitance is then

$$C_{L,max} = C_{L,min} + \sum_{i=1}^{n_{bit}} 2^i\,\Delta C_L. \tag{4.6}$$



(a) Double transmission gate.

(b) Simulated $C_{\mathrm{on}} - C_{\mathrm{off}}$.

Figure 4.9: Schematic of the double transmission gate topology (a) and post layout simulation of $C_{\mathrm{on}} - C_{\mathrm{off}}$ of an NMOS transistor, a transmission gate (TG) and a double transmission gate (D-TG) with three different width ratios N_{W}, $L_{\mathrm{g}} = 60\,\mathrm{nm}$ and a varying gate length $W_{\mathrm{n,p}}$ (b).

For a maximum PVT coverage the normalized capacitance range

$$c_{\mathrm{range}} = \frac{C_{\mathrm{L,max}} - C_{\mathrm{L,min}}}{C_{\mathrm{L,max}}} = \frac{\sum_{i=1}^{n_{\mathrm{bit}}} 2^i \, \Delta C_{\mathrm{L}}}{C_{\mathrm{o}} + C_{\mathrm{l}} + C_{\mathrm{g}} + \sum_{i=1}^{n_{\mathrm{bit}}} C_{\mathrm{off,i}}} \tag{4.7}$$

should be as large as possible. This requires switchable capacitors with a large ratio of incremental capacitance to off state capacitance

$$c_{\mathrm{ratio,i}} = \frac{2^i \, \Delta C_{\mathrm{L}}}{C_{\mathrm{off,i}}} = \frac{C_{\mathrm{on,i}} - C_{\mathrm{off,i}}}{C_{\mathrm{off,i}}}. \tag{4.8}$$

A transmission gate only has a capacitance ratio of $c_{\mathrm{ratio}} \approx 1$ as can be seen by the simulation results in Figure 4.8. In order to increase c_{range} and minimize $C_{\mathrm{L,min}}$ the double transmission gate topology in Figure 4.9 (a) is proposed. The additional transistors M3 and M4 increase the parasitic capacitance that is connected to the input node when $S = 1$ and therefore increases C_{on} without affecting C_{off}. The concept is similar to the separate switch and capacitor topology in [81] except that the capacitor is connected to the control signals S and $\overline{S}$ instead of a fixed potential. The advantage of the proposed topology is that the layout of the double transmission gate can be made extremely compact. The size of the transistors M3 and M4 can be chosen independent of the size of M1 and M2 to further increase c_{ratio}. The scaling factor N_{W} is used in the following to describe the gate width ratio of the transistors in the double transmission gate topology

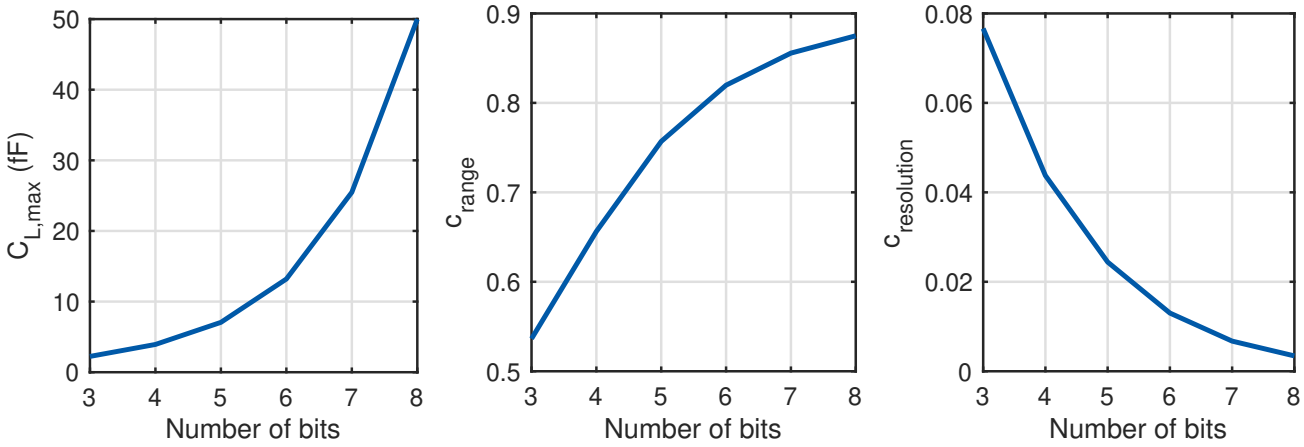
$$W_{\mathrm{M3/M4}} = N_{\mathrm{W}} \, W_{\mathrm{M1/M2}}. \tag{4.9}$$

The maximum possible increase of c_{ratio} using the scaling factor N_{W} is limited by the fact that the on resistance of the transmission gate transistors M1 and M2 remains the same and therefore limits the charging and discharging of the switchable capacitance. The delay resolution of the SCI delay cell is proportional to the normalized capacitance resolution

$$c_{\mathrm{resolution}} = \frac{\Delta C_{\mathrm{L}}}{C_{\mathrm{L,max}}} \tag{4.10}$$

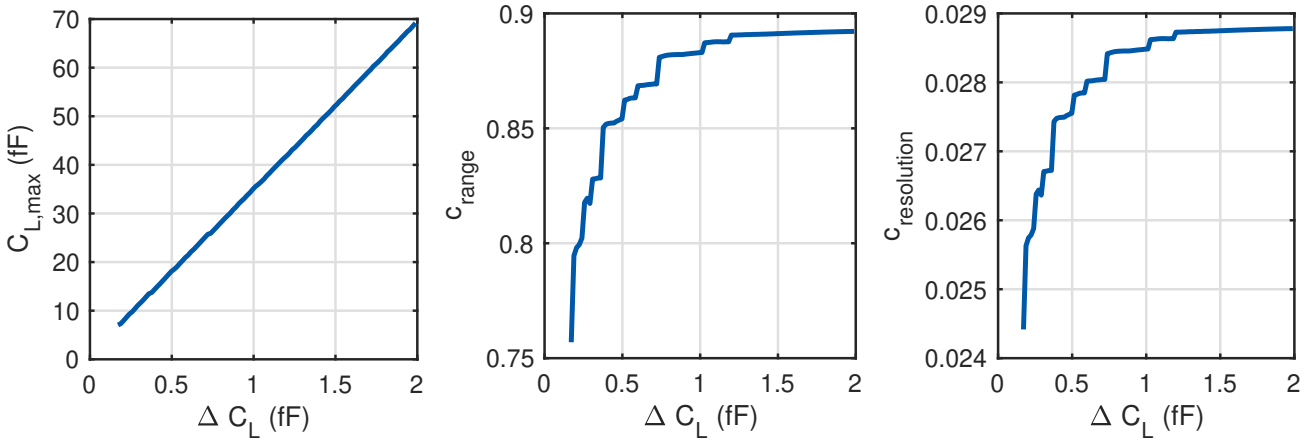
with the incremental load capacitance ΔC_L of the digitally adjustable shunt-capacitors. In order to achieve the smallest possible capacitance increment ΔC_L a third topology is introduced which simply consists of a single NMOS transistor with an open drain. This topology is solely used for the LSB capacitor and therefore the influence of the asymmetric capacitance distribution on the duty-cycle of the clock signal is negligible. Figure 4.9 (b) shows a simulation of the switchable capacitance $C_\mathrm{on} - C_\mathrm{off}$ of the proposed topologies with a varying gate width $W_\mathrm{n/p}$. For the double transmission gate topology $W_\mathrm{n/p}$ is the gate width of M1 and M2, while M3 and M4 have a gate width of $N_\mathrm{W}\, W_\mathrm{n/p}$. The results are obtained from a post layout simulation with extracted parasitics using the extraction tool Assura QRC in order to reflect the actual capacitance increment of the final shunt-capacitor array as accurate as possible. However, the capacitances of the final SCI delay cell will differ from the simulation results of the isolated switchable capacitors due to additional interconnect capacitances.

For the design of the shunt-capacitor array the simulation results in Figure 4.9 (b) are used as a look-up table to find the best topology for each bit and estimate the ideal gate width $W_\mathrm{n/p}$. First the



(a) Maximum load capacitance. **(b)** Normalized range. **(c)** Normalized resolution.

Figure 4.10: Maximum load capacitance $C_\mathrm{L,max}$ (a), normalized range c_range (b) and normalized resolution $c_\mathrm{resolution}$ (c) of a transmission gate shunt-capacitor array for different numbers of bits and $\Delta C_\mathrm{L} = 0.172\,\mathrm{fF}$ calculated by the MATLAB function in Appendix A.



(a) Maximum load capacitance. **(b)** Normalized range. **(c)** Normalized resolution.

Figure 4.11: Maximum load capacitance $C_\mathrm{L,max}$ (a), normalized range c_range (b) and normalized resolution $c_\mathrm{resolution}$ (c) of a 5 bit transmission gate shunt-capacitor array for different capacitance increments ΔC_L calculated by the MATLAB function in Appendix A.

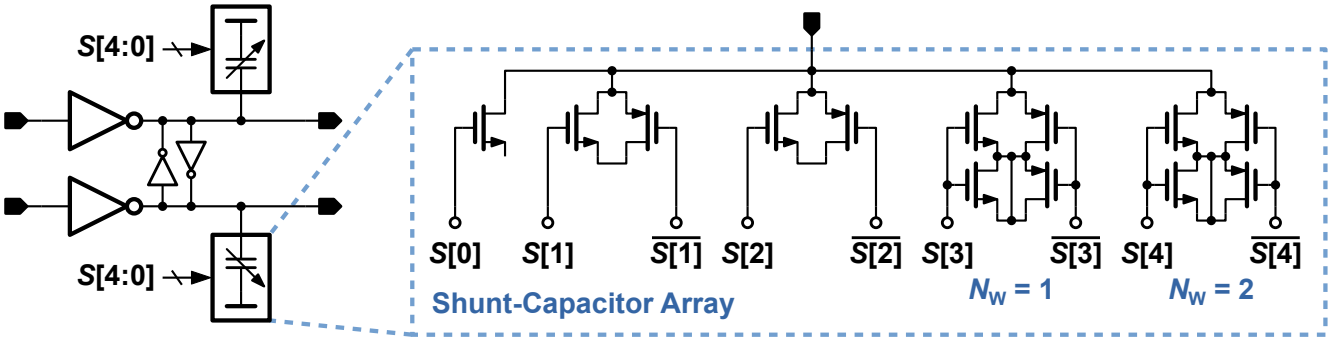


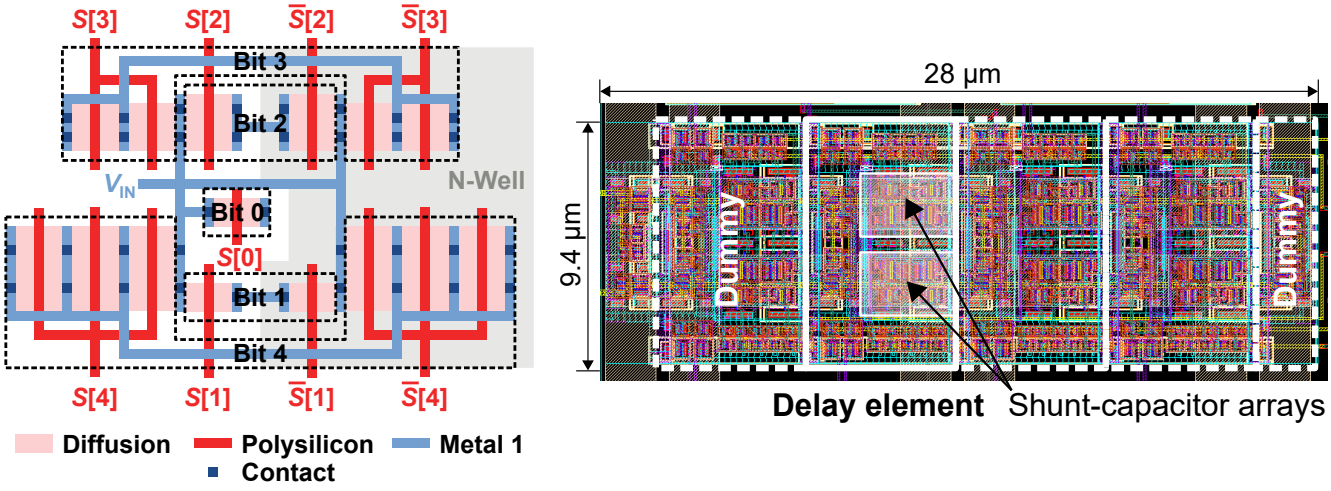
Figure 4.12: Schematic of the proposed SCI delay cell with an optimized 5 bit shunt-capacitor array based on three different shunt-capacitor topologies.

Table 4.1: Sizing results of the proposed 5 bit shunt-capacitor array.

Bit	0	1	2	3	4
Topology	NMOS	TG	TG	D-TG ($N_W = 1$)	D-TG ($N_W = 2$)
Initial $W_{n/p}$	150 nm	240 nm	560 nm	445 nm	560 nm
Final $W_{n/p}$	150 nm	150 nm	400 nm	350 nm	530 nm

desired capacitance increment ΔC_L is selected and the LSB topology is determined. The estimated gate width is obtained by linear interpolation of the simulation results. Due to the almost linear relationship between capacitance increment and gate width only a few simulations are necessary. Next, the remaining bits are dimensioned for their binary scaled capacitance increments $2^i \Delta C_L$. The best topology for each bit is the one with the highest c_{ratio} that can implement the required capacitance increment. A MATLAB script for the automatic dimensioning of a shunt-capacitor array based on a post layout simulation of the individual topologies is given in Appendix A. In Figure 4.10 the maximum load capacitance $C_{L,max}$, the normalized range c_{range} and the normalized resolution $c_{resolution}$ of a transmission gate shunt-capacitor array with different sizes from 3 bit to 8 bit and the minimum capacitance increment $\Delta C_L = 0.172$ fF are shown. The simulated $C_{L,max}$ does not include C_o, C_l and C_g in equation (4.6) as they are unknown at this point. Increasing the number of bits improves both the range and the resolution of the capacitor array, but the changes are getting smaller with higher numbers of bits, while the maximum load capacitance $C_{L,max}$ increases quadratically. Since $C_{L,max}$ is directly related to the power consumption a trade-off has to be made between resolution and power. The proposed SCI based multiphase DLL uses a 5 bit shunt-capacitor array for the SCI delay cells. The influence of the capacitance increment ΔC_L on $C_{L,max}$, c_{range} and $c_{resolution}$ is shown in Figure 4.11. The maximum load capacitance increases linear with ΔC_L as predicted by equation (4.6). The normalized range shows a noticeable improvement with increasing $\Delta C_L$for values below 1 fF, while the impact on the resolution is negligible. In order to achieve a sufficient range for the compensation of PVT variations the proposed shunt-capacitor array is implemented with a capacitance increment of $\Delta C_L = 0.245$ fF.

Figure 4.12 shows the schematic of the implemented pseudo-differential SCI delay cell with the optimized 5 bit shunt-capacitor array. The weak cross coupled inverters have a sizing ratio of $W_{main}/W_{cc} \approx 2.5$ to the main inverters. The selected topologies of the shunt-capacitors and the gate widths estimated by the proposed design procedure are listed in Table 4.1 together with final gate widths after manual tweaking to optimize the linearity of the delay cell. All transistor sizes were reduced during the manual tweaking due to the additional interconnect capacitances in the final layout. The ordering of the transistors in the final layout of the shunt-capacitor array is shown in Figure 4.13 (a). The transistors connected to



(a) Shunt-capacitor array. **(b)** SCI delay line.

Figure 4.13: Layouts of the optimized 5 bit shunt-capacitor array (a) and of the complete pseudo-differential SCI delay line (b).

the common node V_{IN} are placed as close as possible to minimize the minimum load capacitance $C_{L,min}$ of the array. The NMOS shunt-capacitor (Bit 0) and the two transmission gate shunt-capacitors (Bit 1 and Bit 2) are placed in the center with a short interconnect between the NMOS and the PMOS transistors in order to keep the capacitance increment of the lower bits as small as possible. The NMOS and PMOS sections of the double transmission gate shunt-capacitors (Bit 3 and Bit 4) are therefore placed further apart with a longer interconnect. The complete SCI based delay line with three main delay stages and two dummy delay cells has an active area of 263.2 µm^2 (see Figure 4.13 (b)). The terminating dummy delay cell has no shunt-capacitor arrays since only the input capacitance of the driving inverters is relevant for the matching.

A comparison of the delay resolution and range to previously published digitally controlled SCI delay cells is given in Table 4.2. The proposed design achieves the smallest incremental delay of 0.82 ps, which is an improvement of almost 2× over the NAND gate based design in [84]. The topology based on a separate switch and capacitor in [86] achieves a similar resolution as the NAND gate approach. Due to the fact that the NAND gate SCI delay cell was implemented in a 0.35 µm process it could probably realize smaller resolutions in smaller process nodes, but still requires significantly more area than the proposed transmission gate SCI delay cells. The delay range of the presented SCI delay cell is intentionally kept small in favor of an improved power efficiency since the operating frequency is constant at 2.5 GHz. Other implementations are designed to cover an extended frequency range and therefore provide a larger delay range.

Table 4.2: Performance comparison of the proposed SCI delay cell to selected state-of-the-art digitally controlled SCI delay cells.

Shunt capacitor type	Resolution	Range	Max. Frequency	Process
Transmission gate	0.82 ps	25.4 ps	2.5 GHz	65 nm
NAND gate [84]	1.55 ps	396.8 ps	200 MHz	0.35 µm
Switch + capacitor [86]	1.6 ps	78 ns	2.4 GHz	32 nm
Transmission gate + discharging [83]	80 ps	1280 ps	150 MHz	130 nm

4.2.2 Loop Filter Design and Stability Analysis

The digital loop filter of the SCI based DLL consists of a subtractor and a counter as shown in Figure 4.4. The proposed SCI delay cells offer a sufficiently small delay resolution for the intended application that allows to dispense the use of analog voltage controlled shunt-capacitors for fine delay adjustment as proposed in [81] for example. The direct digital control of the SCI delay cells does not require a DAC and a passive low-pass filter and therefore the area of the DLL is significantly reduced and the stability analysis is simplified. Due to the fast response time of the SCI delay cells compared to the PDC sample time T_{PDC} the loop filter can be designed solely in the digital domain. The digital loop filter operates on the PDC output clock with the frequency

$$f_{\mathrm{PDC}} = \frac{1}{T_{\mathrm{PDC}}} = \frac{n_{\mathrm{PDC}}}{NM+1} f_{\mathrm{IN}}, \tag{4.11}$$

based on equation (3.3) with $n_{\mathrm{PDC}} = 2$ for the DLL loop due to the alternating calculation of the 0°-to-135° and the 180°-to-315° phase difference. The optional phase shifter loop also uses a direct digital control for the CMOS phase shifter and is therefore designed in the same manner with $n_{\mathrm{PDC}} = 1$. The z-domain is chosen for the representation of the transfer functions with

$$z = e^{s\,T_{\mathrm{PDC}}} \tag{4.12}$$

due to the single clock domain of the digital loop filter and the negligible response times of the remaining components. The characteristics of the digitally controlled delay line (DCDL) and the CMOS phase shifter are both approximately linear and are therefore represented by constant gain factors K_{DCDL} and K_{PS}, respectively. The proposed digital loop filter has an integrating behavior with the transfer function

$$H_{\mathrm{DLF}}(z) = \frac{K_{\mathrm{DLF}}\, z}{z-1} \tag{4.13}$$

and the loop filter gain K_{DLF} that is able to handle the significant latency of the PDC measurement

$$H_{\mathrm{PDC}}(z) = z^{-2} \tag{4.14}$$

of two samples. This results in a closed-loop transfer function of the SCI based DLL of

$$H_{\mathrm{SCI}}(z) = \frac{K_{\mathrm{DCDL}}\, K_{\mathrm{DLF}}\, z^2}{z^2 - z + K_{\mathrm{DCDL}}\, K_{\mathrm{DLF}}}. \tag{4.15}$$

In the z-domain the transfer function is independent of the PDC scaling factor M and therefore the stability of the loop can be ensured using a fixed integral path gain K_{DLF}. This allows to dynamically adjust M during the operation of the DLL without affecting the loop stability. The bandwidth of the closed-loop transfer function is directly proportional to the sample clock frequency f_{S}. The calculated magnitude for the proposed SCI based DLL with $K_{\mathrm{DCDL}} = 0.0825$ and $K_{\mathrm{DLF}} = 2$ is shown in Figure 4.14 for $M = 1$, $M = 2$ and $M = 4$ to illustrate this relationship. The maximum closed-loop 3 dB bandwidth of the DLL is 17.7 MHz with $M = 1$ and $f_{\mathrm{IN}} = 2.5$ GHz.

The proposed calculation method in section 3.2.2 for the estimation of the influence of clock jitter on the PDC measurement results can be used to estimate the deterministic jitter of the SCI based DLL caused by DCDL control word dithering. Due to the direct digital control an adjustment of the DCDL by the control loop temporarily increases or decreases the clock period at the delay line outputs, where $\phi(n_{\mathrm{tap}}) = n_{\mathrm{tap}} \cdot 45°$ is the clock phase at the tap position n_{tap}. This temporary adjustment occurs at

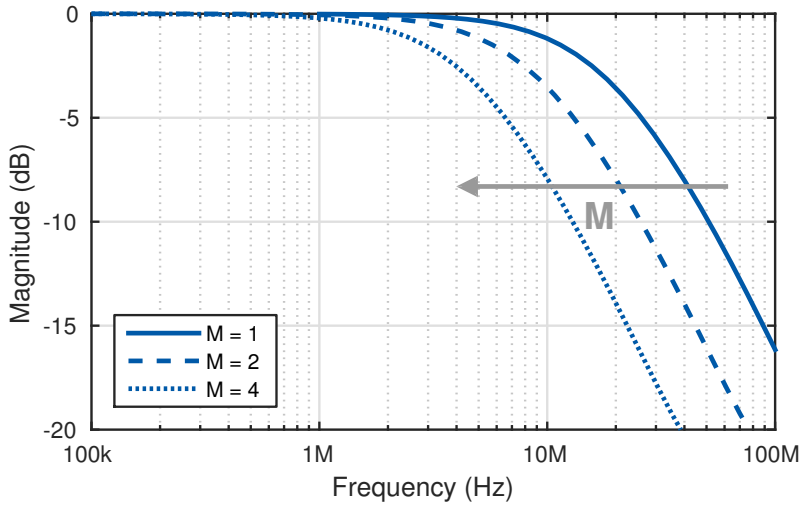


Figure 4.14: Calculated magnitude of the SCI based DLL closed-loop transfer function in equation (4.15) with $K_{\mathrm{DCDL}} = 0.0825$ and $K_{\mathrm{DLF}} = 2$ for three different scaling factors M and $f_{\mathrm{IN}} = 2.5\,\mathrm{GHz}$.

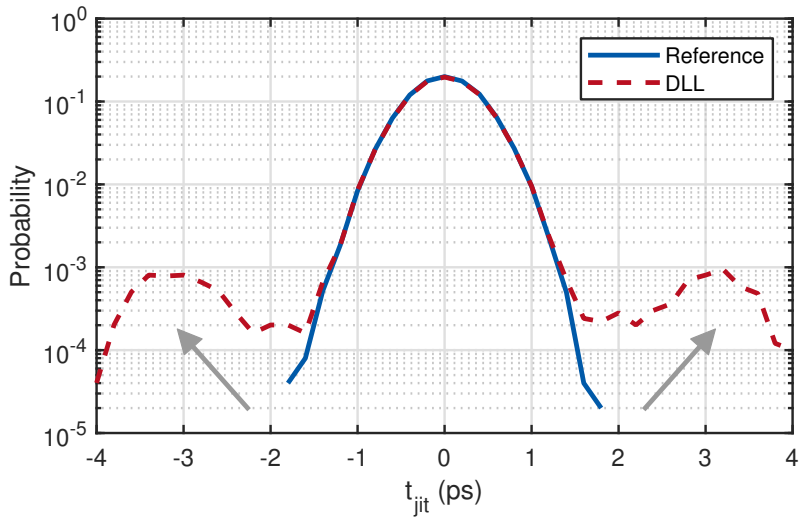


Figure 4.15: Simulated jitter histogram of the reference clock and the 90° output of the SCI based DLL in a behavioral simulation. The tails of the distribution are spread by the contribution of the deterministic jitter component.

most every $(NM + 1)/2$ clock edges due to the limited sample rate of the PDC. The frequency of the deterministic jitter is therefore bounded to

$$f_{\mathrm{DJ}} \leq \frac{f_{\mathrm{PDC}}}{2} \tag{4.16}$$

and leads to a spreading of the tails of the jitter histogram of the DLL, while the rest of the distribution remains nearly unaffected. Figure 4.15 illustrates this effect by a behavioral simulation of the jitter histogram at the 90° output of the SCI based DLL. The jitter amplitude depends on the tap position n_{tap}, the delay increment Δt_{SCI} of the SCI delay cells and the amplitude of the control word adjustment calculated by the digital loop filter. Assuming the DLL to be locked the PDC measurement deviates from its ideal value φ_{DLL} with the standard deviation σ_{PDC} of the PDC measurement results according to equation (3.24). The resulting low frequency deterministic jitter is therefore estimated to

$$DJ_{\mathrm{RMS}}(n_{\mathrm{tap}}) = n_{\mathrm{tap}}\,\sigma_{\mathrm{PDC}}\,\Delta t_{\mathrm{SCI}}. \tag{4.17}$$

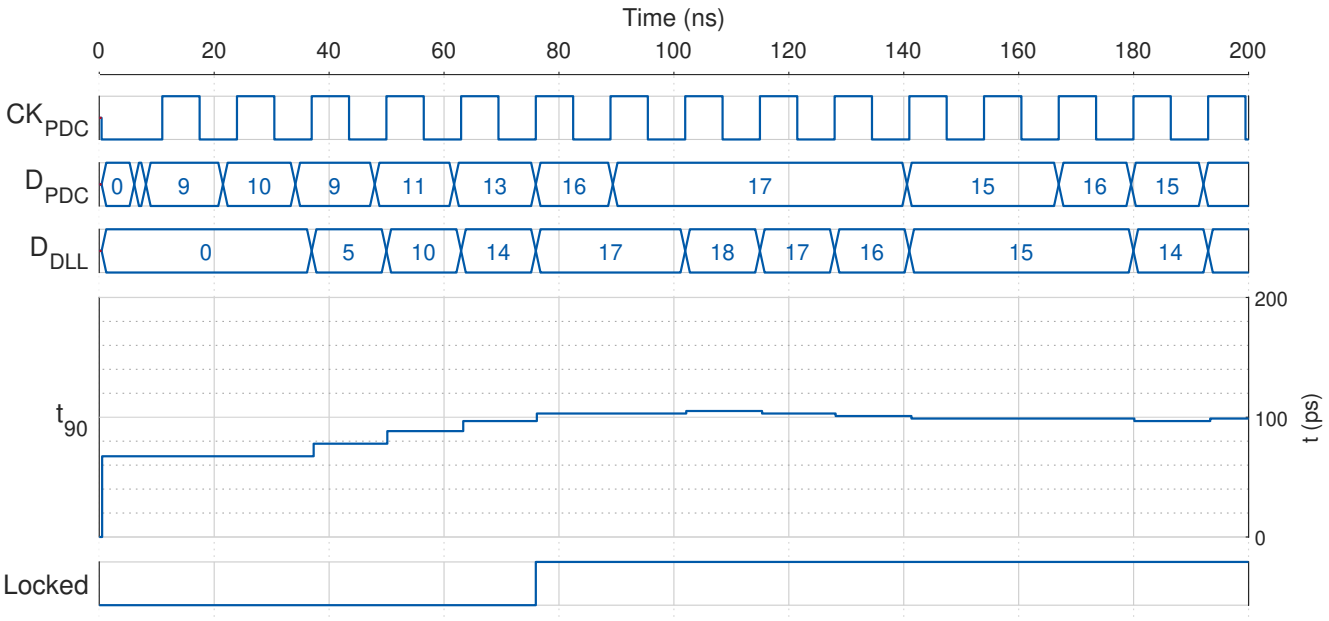


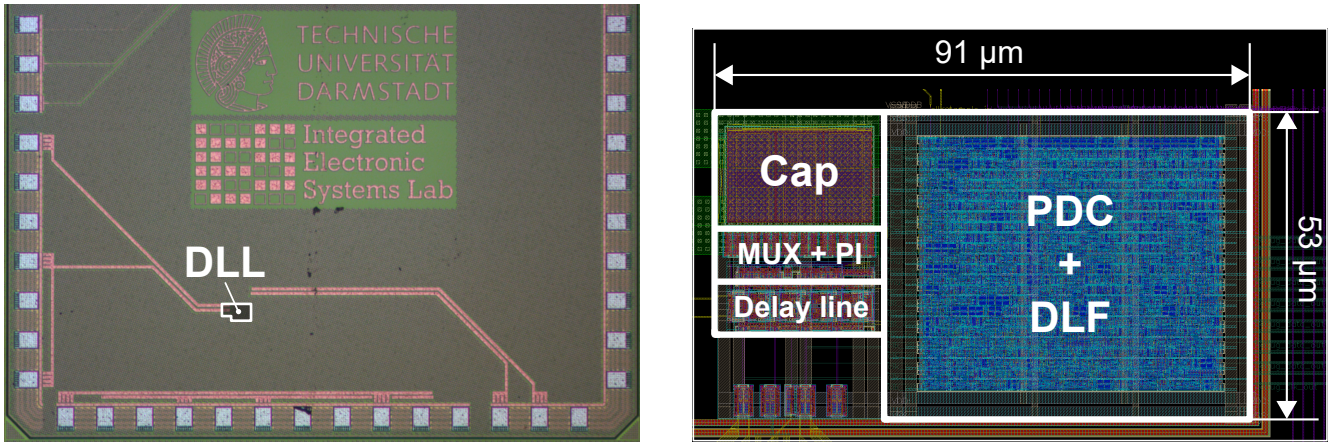
Figure 4.16: Behavioral simulation of the locking of the SCI based DLL with $f_{IN} = 2.5\,\text{GHz}$, $M = 1$ and $TJ_{\text{RMS}} = 400\,\text{fs}$. The DLL is locked within 6 PDC samples (76 ns).

Assuming the sample clock jitter specification in equation (3.27) to be fulfilled (minimized standard deviation of the PDC measurement results), the estimated worst case deterministic jitter at the 90° output ($n_{\text{tap}} = 3$) of the SCI based DLL is $DJ_{\text{RMS}}(3) = 1.74\,\text{ps}$ with a measured delay increment of $\Delta t_{\text{SCI}} = 0.82\,\text{ps}$ of the proposed SCI delay cells.

Since the SCI based DLL does not require any analog low-pass filtering and the response times of the SCI delay cells are small compared to the PDC sample time the complete system can be evaluated in a behavioral simulation using Verilog models for the delay line and the PI. These models implement a fixed minimum delay to account for the intrinsic delay through the buffer circuits and a digitally adjustable delay with increments of $\Delta t_{\text{SCI}} = 0.82\,\text{ps}$ for the delay cells and $\Delta t_{\text{PI}} = T_{\text{IN}}/(2^7 - 1)$ for the 7 bit PI. This allows a much faster simulation than using mixed signal simulators like AMS. Figure 4.16 shows the results of a behavioral simulation of the locking process with $f_{IN} = 2.5\,\text{GHz}$, $M = 1$ and $TJ_{\text{RMS}} = 400\,\text{fs}$ for both CK_{IN} and CK_{S}. The DLL lock indication is asserted when the PDC measurement result D_{PDC} lies within ±1 LSB of the ideal phase $\varphi_{\text{DLL}} = 16$. This is the case after 6 PDC samples (76 ns) for the given parameters. The spreading of the jitter histogram due to the low frequency deterministic jitter component leads to a simulated RMS total jitter of 520 fs at the 90° output of the DLL with $M = 1$. With $M = 4$ the RMS total jitter reduces to 460 fs due to the reduced DLL adjustment frequency and therefore a reduced influence of the deterministic jitter component. Therefore, a reduction of the DLL bandwidth is beneficial for the jitter performance. As a trade-off, the capability of the DLL to compensate timing drift due to low frequency supply noise or temperature variations is reduced by lowering the DLL bandwidth. Another drawback is that in order to fulfill equation (3.27) the jitter requirement for CK_{LO} in terms of UI is increased with decreased f_{LO}. This can be difficult to realize since the clock jitter of PLLs is typically related to the oscillator frequency. Generating a low frequency clock with a very small RMS jitter therefore requires a high frequency oscillator with a clock divider that consume significantly more power than a low frequency oscillator.

4.2.3 Measurement Results

The SCI based DLL and CMOS phase shifter are implemented in the UMC 65 LL CMOS process. A die photograph and the layout of the proposed design are shown in Figure 4.17. The total active area is



(a) Die photograph. **(b)** Layout.

Figure 4.17: Die photograph (a) and layout (b) of the SCI based DLL test chip in the UMC 65 LL technology.

0.0048 mm^2, of which 226 µm^2 are covered by the delay line and 3309 µm^2 by the PDC and the digital loop filter. The CMOS phase shifter consisting of a 8:2 MUX and a PI cover another 204 µm^2. The measurement setup with the test chip placed in a QFN56 package is shown in Figure 4.18. Both clock signals CK_{IN} and CK_{S} are generated externally from a 25 MHz reference clock by two PLLs of the type ADF4351 by Analog Devices. The Keysight DSAV164A oscilloscope is used to perform delay, jitter and phase noise measurements of the clock signals.

Since the PLL for the generation of CK_{S} is not integrated into the DLL ASIC its area is not included in the total design area of the DLL. However, in the target application of parallel high-speed I/O links multiple instances of the DLL will be used and therefore the sample clock PLL can be shared between these instances. An example implementation of a GDDR5 memory interface in [35] with 32 parallel I/O links requires 16 multiphase DLLs that are shared between neighboring transceiver slices. A matching PLL for the generation of CK_{S} with a sufficiently low RMS jitter of 2.8 ps can be found in [85]. The PLL is fully synthesized in a 65 nm CMOS process and therefore supports the aim of an all-digital implementation style. Using the proposed SCI based multiphase DLL the shared sample clock PLL increases the proportional area of each DLL instance by only 0.0004 mm^2. The impact on the proportional power efficiency is also negligible with only 54 µW/GHz.

A jitter measurement is performed on the 90° output of the DLL with a decomposition into random and periodic jitter using the EZJIT analysis software by Keysight. The measured RMS jitter for $M = 1$, $M = 2$ and $M = 4$ is shown in Figure 4.19 (a). The RMS periodic jitter shows a linear increase with the sample clock frequency f_{S} while the RMS random jitter remains almost constant at around 1.2 ps. This indicates that the main causes of periodic jitter are the kick-back noise from the asynchronously clocked samplers and the supply noise from the digital part. The latter can be efficiently suppressed by adding an LDO to regulate the supply of the delay line [44, 43]. A technique for the suppression of the sampler kick-back noise is applied in the CSI based DLL design presented in section 4.3. The power consumption of the DLL in Figure 4.19 (b) shows an almost linear dependency on the sample clock frequency f_{S} as expected due to the dominating digital part that is clocked by CK_{S}. The phase shifter output shows a slightly higher RMS random jitter of 1.44 ps caused by the additional contributions of the CMOS MUX and PI. The periodic jitter in this case not only depends on M but also on the phase shifter setting since the deterministic jitter of the DLL is depending on the tap position as estimated by equation (4.17). The typical RMS periodic jitter of 2.28 ps with $M = 4$ is similar to the result of the 90° DLL output.

The measured phase noise plot of the 90° output and the 2.5 GHz reference clock is shown in Figure 4.20. The influence of the scaling factor M on the DLL bandwidth is clearly visible in the range of 1 MHz to 100 MHz offset. With $M = 4$ the phase noise at 10 MHz offset is reduced by 10.5 dB compared

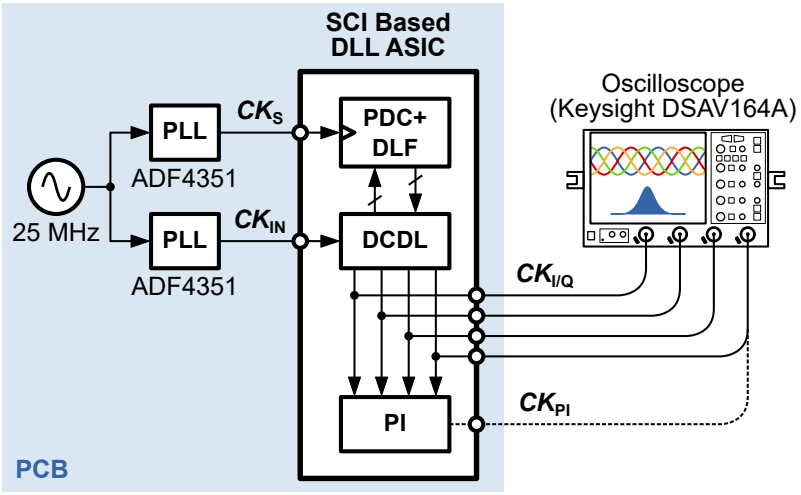
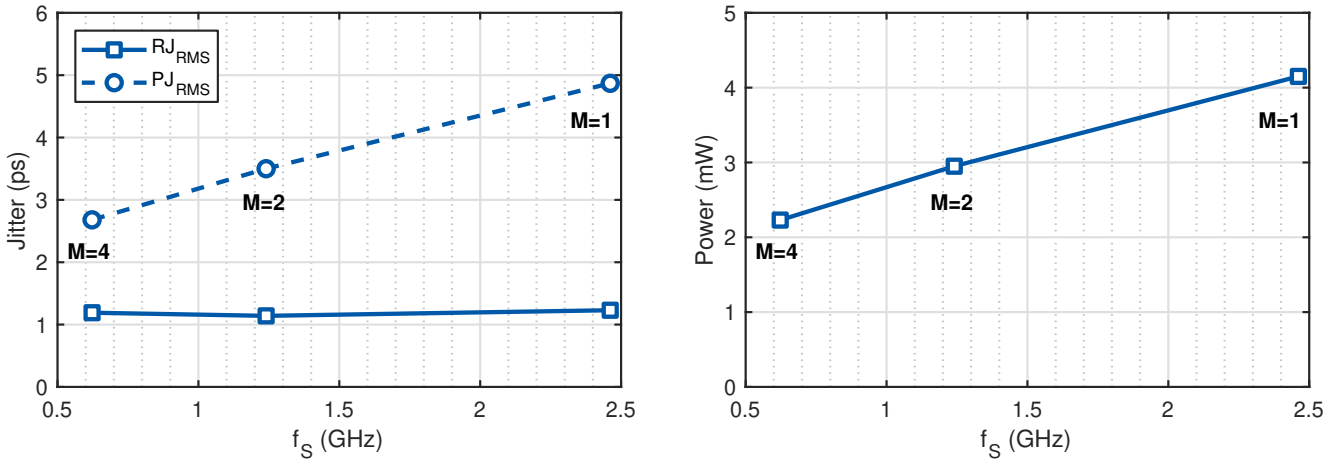


Figure 4.18: Block diagram of the measurement setup for the proposed SCI based DLL test chip.



(a) Random and periodic jitter.

(b) Power consumption.

Figure 4.19: Measured RMS random and periodic jitter (a) and power consumption (b) of the proposed SCI based multiphase DLL with $f_{\mathrm{IN}} = 2.5$ GHz and different scaling factors M.

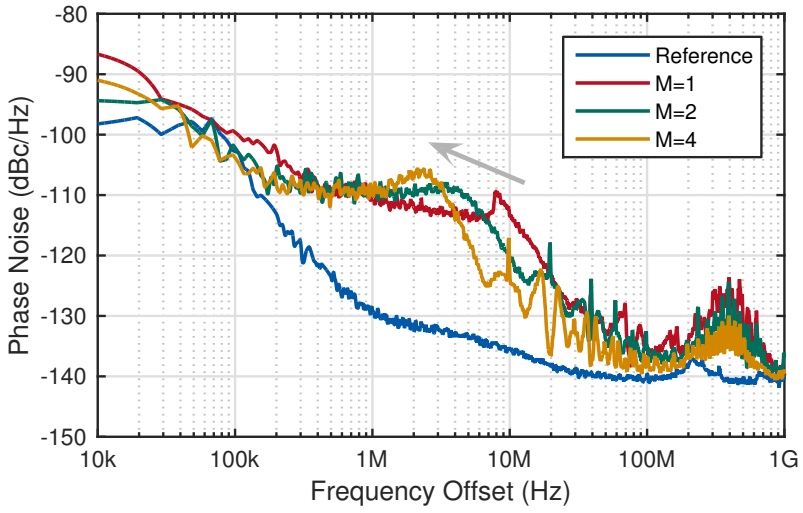


Figure 4.20: Measured phase noise of the 2.5 GHz reference clock and the 90° output of the proposed SCI based multiphase DLL for different values of M.

to the results with $M = 1$ in trade for an increase of 5.9 dB at around 3 MHz offset. The phase noise at 1 MHz offset remains unaffected at −110.7 dBc/Hz for all considered values of M. The integrated RMS jitter from 10 kHz to 100 MHz offset also improves from 1.085 ps with $M = 1$ to 862.4 fs with $M = 4$. Due to the significant amount of kick-back noise and supply noise it cannot be unambiguously determined to which amount the DLL bandwidth impacts the jitter performance. The results of the behavioral simulation in section 4.2.2 indicate that the impact of the bandwidth is small compared to the noise contributions.

4.3 Current-Starved Inverter Based Architecture

The CSI based multiphase DLL architecture in Figure 4.21 represents an alternative trade-off between jitter performance and area consumption. Instead of a digitally controlled SCI delay cell, a voltage controlled CSI delay cell is used that offers less parasitic capacitance and therefore a lower delay line power consumption. The CSI based DLL is intended for the use with a CML PI that requires only four instead of eight clock phases compared to a MUX based PI. Therefore, the delay line consists of only two main delay cells and two dummies for matching. Additionally, the use of a high resolution DAC for the generation of the control voltage improves the jitter performance by reducing the deterministic jitter caused by control word dithering. This in return consumes more power in the digital part and requires an analog low-pass filter that increases the area consumption of the DLL. Similar to the SCI based DLL a counter is used in the digital loop filter to accumulate the phase error and generate a digital control word for the delay line. A synthesized 11 bit ΔΣ modulator is used as a DAC in order to keep the design simple. The modulator is operated at half the sample clock frequency. The PDC is again implemented with a resolution of $N = 64$ and a glitch filter size of $n_{\mathrm{filt}} = 3$. The target frequency for the DLL is significantly increased to $f_{\mathrm{IN}} = 4$ GHz compared to the SCI based DLL. The sample clock frequency of the PDC with $N = 64$ is limited to 2.5 GHz in the given 65 nm CMOS process and therefore a scaling factor of $M \geq 2$ is required for the operation of the CSI based DLL. In order to reduce the periodic jitter caused by kick-back noise from the PDC samplers, the CSI based DLL features clocked comparators with additional input buffers for the sampling of the in-phase (CK_{I}) and quadrature (CK_{Q}) clock signals. The input buffers are designed to have a large slew rate in order to minimize delay mismatch between the buffers as well as the influence of the samplers input referred offset voltage. The clocked comparator is based on the alternative StrongARM latch topology in [30] with reduced kick-back noise (Figure 4.22 (a)) combined with a symmetric SR-latch [87] (Figure 4.22 (b)) to minimize the offset of the comparator. A drawback of the reduced kick-back noise topology is the larger dynamic offset (the input referred offset depends on the common mode of the input signals) of the sampler. Since the input signals are provided by CMOS inverters with a large slew rate the common mode voltage of the input signals is constant at $V_{\mathrm{DD}}/2$ and therefore the dynamic offset does not affect the measurement result.

Two different approaches to the design of the CSI delay cells are elaborated. In section 4.3.1 a direct control using the voltage generated by the ΔΣ modulator is presented. This approach offers a low power consumption and a small area but has a nonlinear delay characteristic. The design in section 4.3.2 proposes the use of a multi-bit ΔΣ modulator in combination with a current-DAC to generate a more linear delay characteristic. Additionally, a secondary control is integrated to increase the frequency range of the DLL.

4.3.1 Direct Voltage Controlled Current-Starved Inverter Delay Cells

Figure 4.23 (a) shows the schematic of the proposed pseudo-differential voltage controlled CSI delay cell. The main inverters are supplied by two voltage controlled current sources. One source is controlled by the low-pass filtered ΔΣ modulator output voltage V_{LPF} and the other applies a fixed bias current to define the maximum delay of the delay cell. The constant-g_{m} bias circuit in Figure 4.23 (b) is used to generate the bias voltage V_{bias}. Current mirrors provide equal currents for the PMOS and the NMOS transistors of the main inverters. Weak cross-coupled inverters with a sizing ratio of $W_{\mathrm{main}}/W_{\mathrm{cc}} = 3$ at the output of the delay cell generate some intrinsic duty-cycle correction and ensure a rail-to-rail voltage swing. Despite

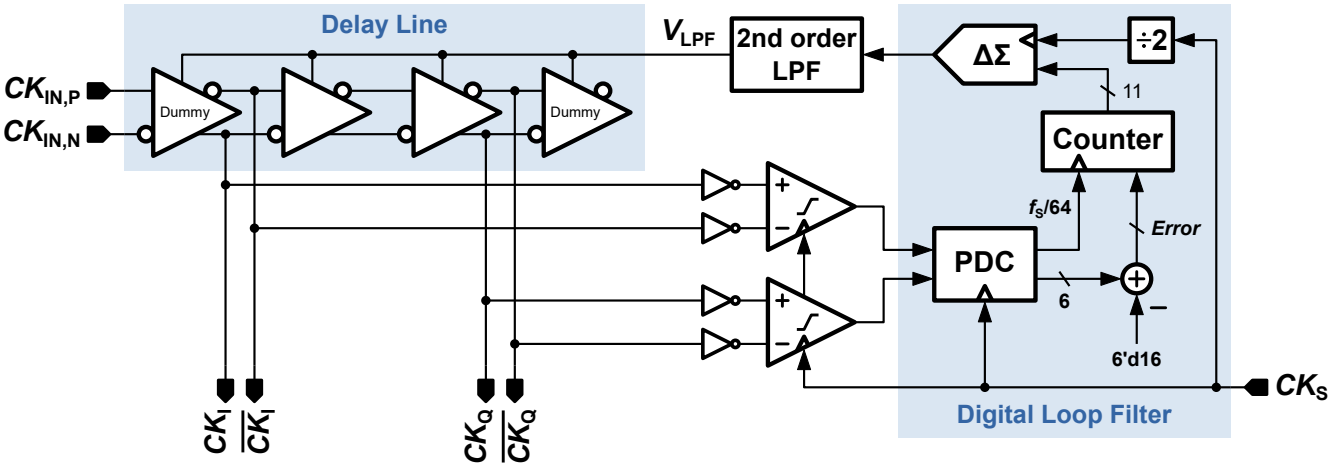
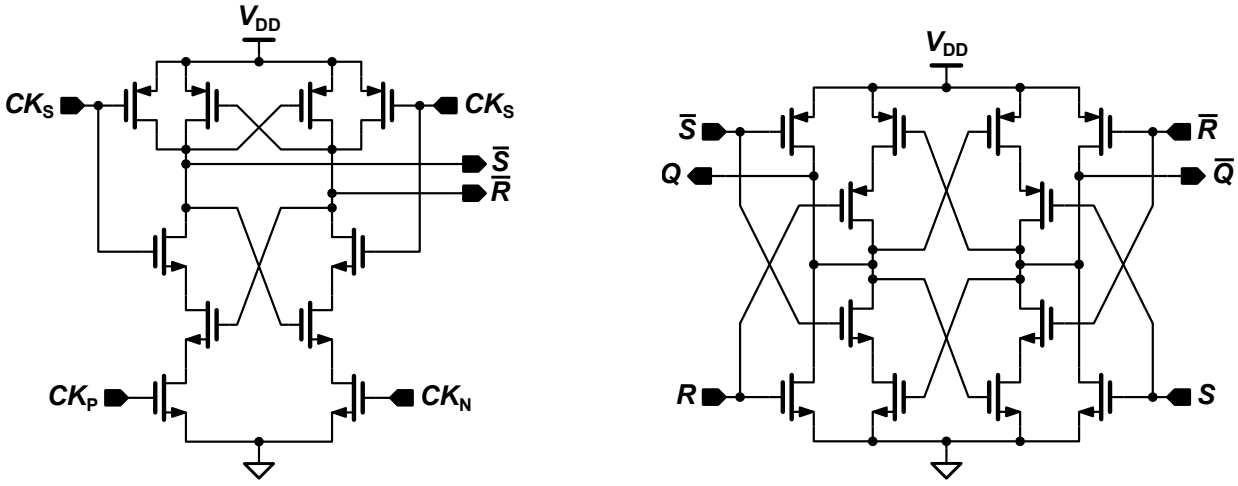
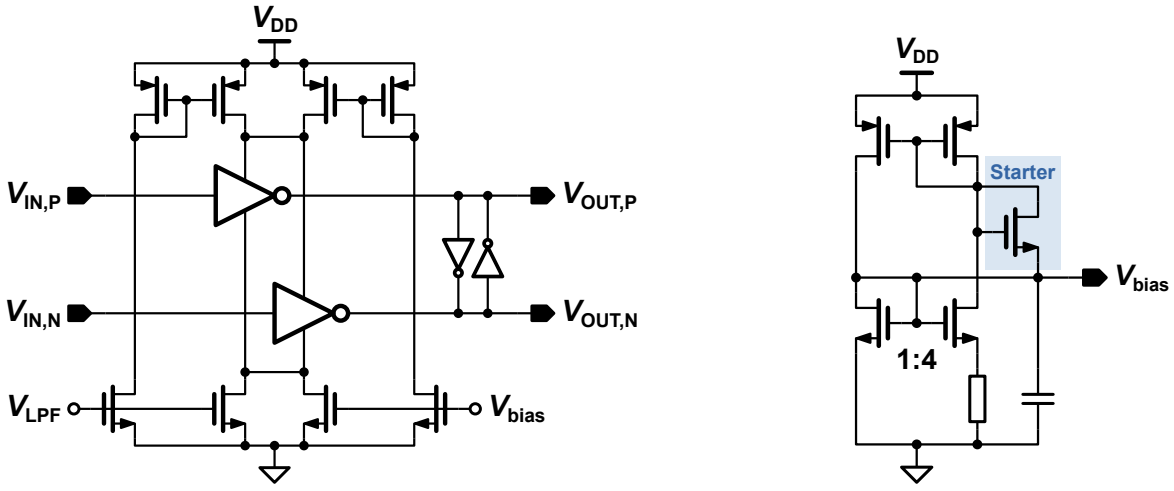


Figure 4.21: Block diagram of the proposed CSI based multiphase DLL.



(a) StrongARM latch. **(b)** SR-latch.

Figure 4.22: Schematics of the sampler for the CSI based DLL consisting of a StrongARM latch with reduced kick-back noise (a) and a symmetric SR-latch (b).



(a) CSI delay cell. **(b)** Constant-g_m bias circuit.

Figure 4.23: Schematics of the proposed voltage controlled CSI delay cell (a) and the constant-g_m bias circuit (b).

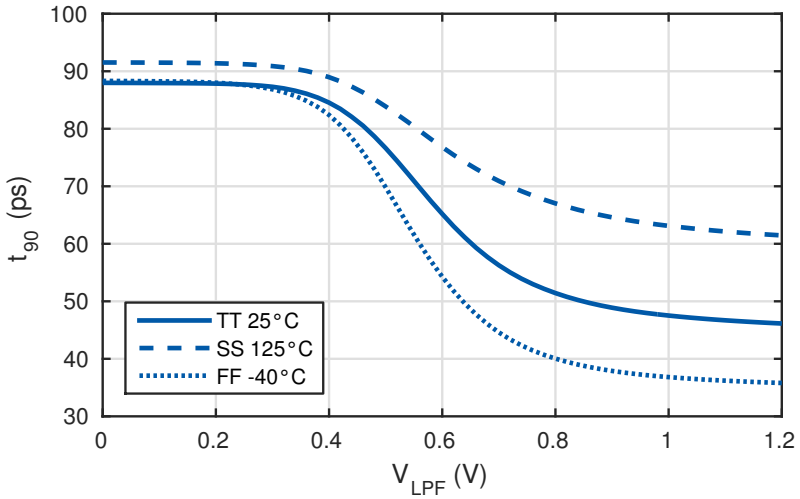


Figure 4.24: Simulated delay characteristics of the proposed voltage controlled CSI delay line at the 90° output and in the typical (TT), best (FF) and worst case (SS) corners.

the simplicity of the delay cell the transistor sizing procedure is more complex than for the SCI based delay cell in section 4.2.1. Both, the sizing of the inverters as well as the sizing of the current sources have a direct impact on the delay of the CSI delay cell. The load capacitance is composed of the gate capacitance of the main inverters, the input and output capacitances of the cross-coupled inverters and the parasitic layout capacitances and is therefore very sensitive to layout changes. For a structured design process the delay cell is separated into two parts. First the inverters are dimensioned using a fixed supply current and a layout is iteratively created to consider the interconnect parasitics. In the second step the current sources are designed to implement the required delay range using the extracted layout of the inverter section. Finally the layout of the current sources is created with iterative adjustments to take the interconnect parasitics into account. The complete four stage CSI delay line occupies an active area of 257 μm^2 and thus has a slightly bigger area than the five stage SCI delay line in section 4.2.1 with 226 μm^2. One reason is that the CSI delay line is designed for a higher frequency range of up to 4 GHz and therefore requires larger transistors to implement sufficiently small delays.

The simulated delay characteristics between CK_I and CK_Q of the CSI delay line in the typical, best and worst case corners are shown in Figure 4.24. The design covers a frequency range of 2.8 GHz to 4 GHz over all corners. Due to the direct voltage control of the current sources the delay cell has a nonlinear delay characteristic with a flat delay response for a control voltage V_LPF below the transistors threshold voltage. The maximum gain of the delay cell in the fastest corner is 166.5 ps/V. This value is used for the stability analysis of the DLL in section 4.3.3.

4.3.2 Linearized Current-Starved Inverter Delay Cells

The CSI delay cells for a more linear delay characteristic consist of three equally sized current sources for the PMOS and NMOS supply as shown in Figure 4.25. One current source is always enabled while the other two can be individually activated by the 2 bit frequency band selection $D_\mathrm{b}[1:0]$. Instead of implementing a fixed bias source for the definition of the maximum delay as in section 4.3.1 the minimum supply current is realized by a lower limit of the control voltage V_LPF. All CSI delay cells in the delay line share a single current mirror for the generation of the complementary PMOS bias voltage from the NMOS bias voltage V_LPF. The main inverters and the weak cross-coupled inverters have a sizing ratio of $W_\mathrm{main}/W_\mathrm{cc} = 3$.

For the generation of the control voltage V_LPF a $\Delta\Sigma$ modulator with a multi-bit output is used in combination with a current-DAC and a second order low pass filter. Figure 4.26 shows a comparison of the

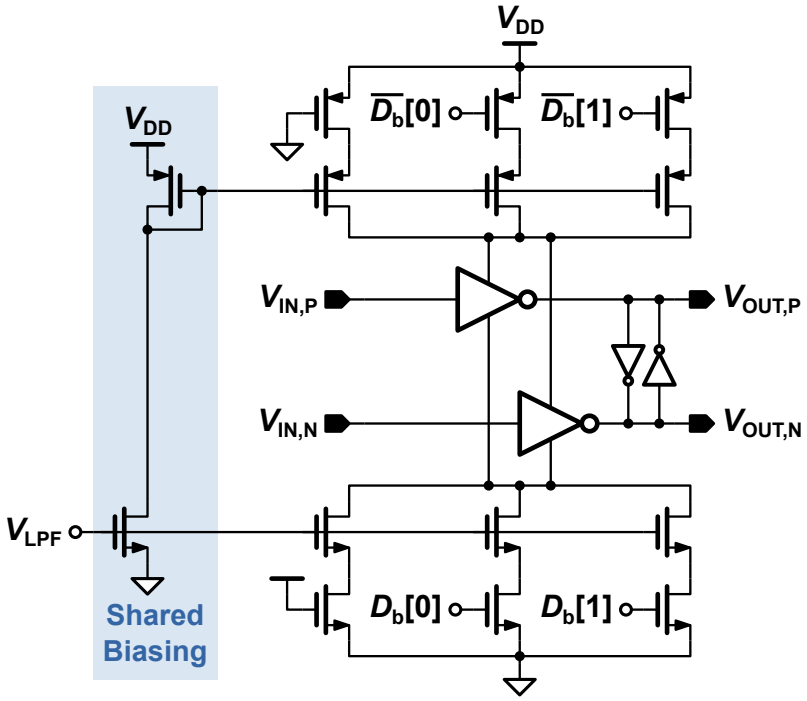
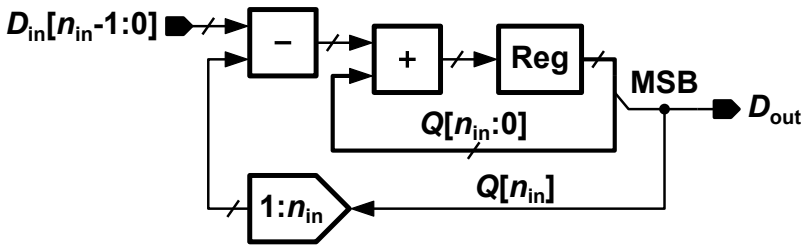
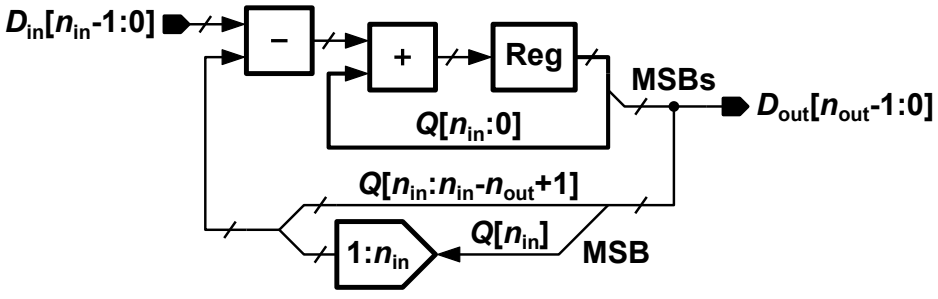


Figure 4.25: Schematic of the proposed current controlled CSI delay cell with frequency band selection.



(a) Single bit $\Delta\Sigma$ modulator.



(b) Multi-bit $\Delta\Sigma$ modulator.

Figure 4.26: Block diagrams of a conventional single bit $\Delta\Sigma$ modulator (a) and a multi-bit $\Delta\Sigma$ modulator (b).

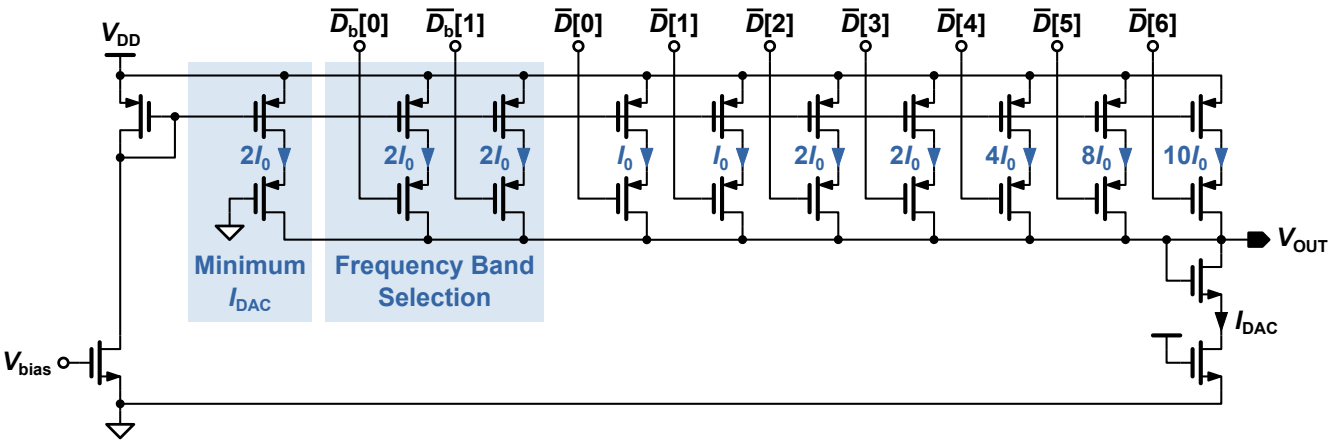


Figure 4.27: Schematic of the proposed 3 bit current-DAC with thermometer coded control for the generation of an approximately linear delay characteristic of the CSI delay line.

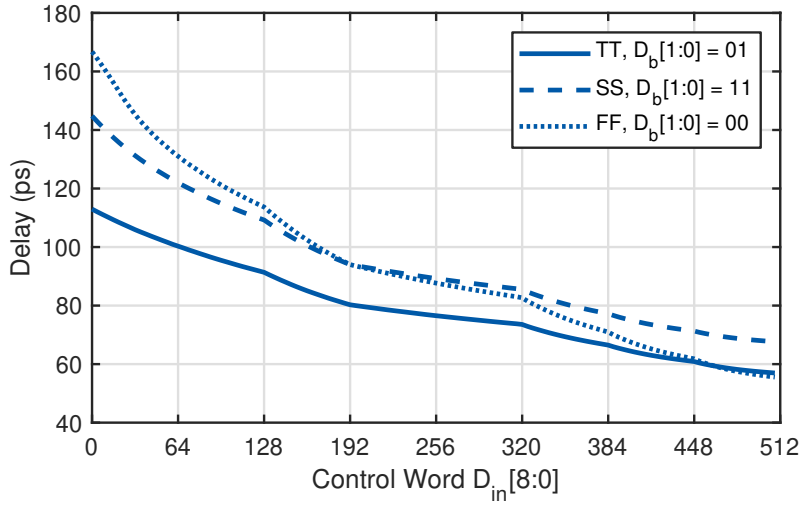


Figure 4.28: Simulated delay characteristics of the proposed linearized CSI delay line at the 90° output and in the typical, best and worst case process corners with the frequency band selection used to compensate the process variation.

block diagrams of a conventional single bit ΔΣ modulator and a multi-bit ΔΣ modulator. For the single bit output in Figure 4.26 (a) only the MSB of the $(n_{in}+1)$ bit wide register is used and a 1-to-n_{in} bit replication of the MSB serves as feedback for the modulator. The multi-bit architecture in Figure 4.26 (b) uses the n_{out} MSBs of the register as output signal and the feedback is the concatenation of the output signal as MSBs and a replication of the MSB for the remaining bits. The proposed DLL design uses an $n_{in} = 9$ bit ΔΣ modulator with a $n_{out} = 3$ bit output. The modulator output is fed to a binary-to-thermometer converter resulting in a 7 bit thermometer coded control signal for the current-DAC in Figure 4.27. The proposed current-DAC generates the bias voltage for the CSI delay cells using digitally controlled PMOS current sources and an NMOS current mirror. A constant current source ensures a minimum output current $I_{DAC,min} = 2\,I_0$ to define the maximum delay of the CSI delay cells. The current-DAC also implements part of the frequency band selection with two additional current sources controlled by $D_b[1:0]$. Since the delay of a CSI delay cell is inversely proportional to the supply current as shown in equation (4.2) the digitally controlled current sources for the 7 bit input signal $D[6:0]$ have a nonlinear scaling in order to linearize the delay characteristic. The reference current of the current-DAC is $I_0 \approx 10\,\mu A$ and the bias voltage V_{bias} is generated using the low-voltage bandgap current reference in Appendix B.

A simulation of the delay characteristics in the typical, best and worst case process corners with adapted frequency band settings to compensate the process variation is shown in Figure 4.28. The nonlinear scaling of the 3 bit current-DAC creates a set of $(2^3 + 1)$ equally spaced support points in the transfer curve for linearization. In between these support points the curve shows the typical nonlinear behavior of the CSI delay cells. The nonlinear scaling of the current-DAC requires iterative optimization to achieve similar characteristics in all frequency bands and for all PVT corners. Increasing the number of bits n_{out} for the current-DAC and therefore the number of support points would help to improve the linearity of the delay characteristic, but would also require more area and power and make the scaling of the current-DAC more difficult. The three delay characteristics in Figure 4.28 show a similar gain across the process corners and a frequency range of $1.4 - 3.3\,\text{GHz}$ for the DLL.

4.3.3 Loop Filter Design and Stability Analysis

Compared to the SCI based DLL in section 4.2.2 the digital loop filter of the CSI based DLL is extended by the $\Delta\Sigma$ modulator and the second order low-pass filter. Also, the delay characteristic of the delay line is nonlinear, which is handled by designing the loop filter with the maximum possible gain K_{DL} of the delay line. The transfer function of the modulator can be modeled as a delay

$$H_{\text{MOD}}(s) = K_{\text{MOD}}\, \mathrm{e}^{-2s\, T_{\text{MOD}}} \tag{4.18}$$

with the modulator clock period T_{MOD} and the gain $K_{\text{MOD}} = 1$. Since the modulator clock is much faster than the PDC sample rate

$$T_{\text{MOD}} = \frac{2}{f_{\text{S}}} \ll T_{\text{PDC}} = \frac{64}{f_{\text{S}}} \tag{4.19}$$

the transfer function of the modulator can be neglected. The dominating factor in the CSI based DLL design is the transfer function of the second order passive low-pass filter

$$H_{\text{LPF}}(s) = \frac{1}{(1 + R_{\text{LPF}}\, C_{\text{LPF}}\, s)^2} \tag{4.20}$$

with the resistance R_{LPF} and the capacitance C_{LPF}. Together with the PDC transfer function in the s-domain

$$H_{\text{PDC}}(s) = \mathrm{e}^{-2s\, T_{\text{PDC}}} \tag{4.21}$$

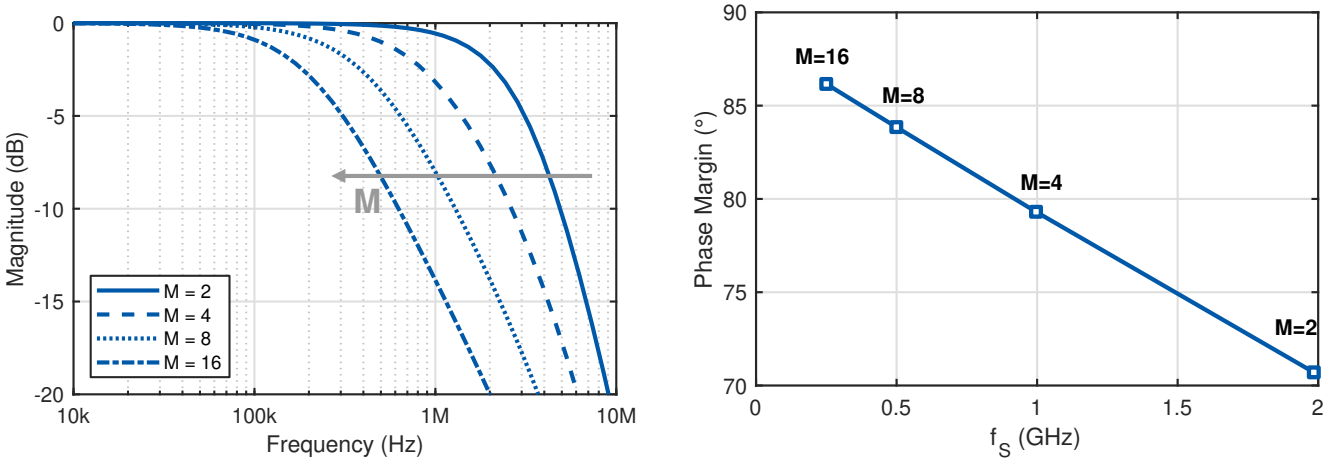
and the integrating loop filter transfer function

$$H_{\text{DLF}}(s) = \frac{K_{\text{i}}}{1 - \mathrm{e}^{-s\, T_{\text{PDC}}}} \tag{4.22}$$

the resulting closed loop transfer function of the CSI based DLL is

$$H_{\text{CSI}}(s) = \frac{K_{\text{I}}\, K_{\text{DL}}\, \mathrm{e}^{-2s\, T_{\text{PDC}}}}{(1 - \mathrm{e}^{-s\, T_{\text{PDC}}})(1 + R_{\text{LPF}}\, C_{\text{LPF}}\, s)^2\, \mathrm{e}^{-2s\, T_{\text{PDC}}} + K_{\text{I}}\, K_{\text{DL}}}. \tag{4.23}$$

Like with the SCI based DLL the bandwidth of the closed loop transfer function is proportional to f_{S}. Due to the second order low-pass filter the bandwidth is lower ($BW_{\text{CSI}} \approx 2.35\,\text{MHz}$ with $M = 2$ and $f_{\text{IN}} = 4\,\text{GHz}$) than the bandwidth of the SCI based DLL with direct digital delay control ($BW_{\text{SCI}} \approx 17.7\,\text{MHz}$ with $M = 1$ and $f_{\text{IN}} = 2.5\,\text{GHz}$). Figure 4.29 (a) shows the magnitude of the closed loop transfer function $H_{\text{CSI}}(s)$ of the direct voltage controlled CSI based DLL for different PDC scaling factors M. The relationship between M and the bandwidth is similar to the SCI based DLL in Figure 4.14, but the stability is no longer independent on M. Since the phase margin of the open loop transfer function increases with increasing M (see Figure 4.29 (b)) it is sufficient to ensure the stability of the DLL at the highest sample clock frequency



(a) Closed loop magnitude.

(b) Phase margin.

Figure 4.29: Simulated magnitude of the closed loop transfer function $H_{\mathrm{CSI}}(s)$ (a) and phase margin (b) of the direct voltage controlled CSI based DLL for different scaling factors M and $f_{\mathrm{IN}} = 4\,\mathrm{GHz}$.

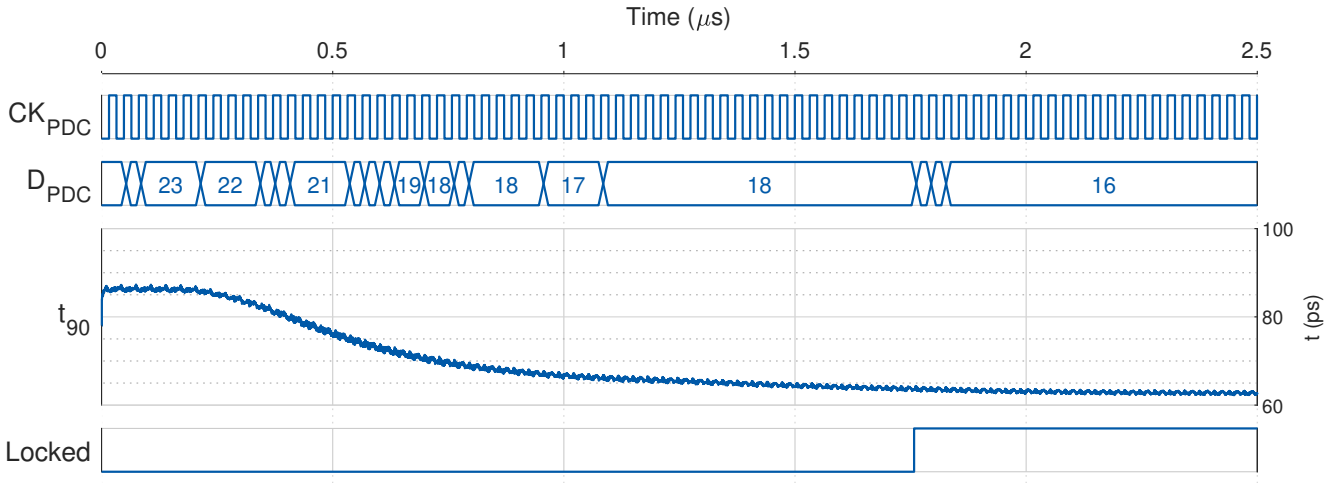


Figure 4.30: Mixed mode simulation of the locking of the DLL with direct voltage controlled CSI delay cells for $f_{\mathrm{IN}} = 4\,\mathrm{GHz}$, $M = 2$ and $TJ_{\mathrm{RMS}} = 400\,\mathrm{fs}$. The DLL is locked within 54 PDC samples (1.76 µs).

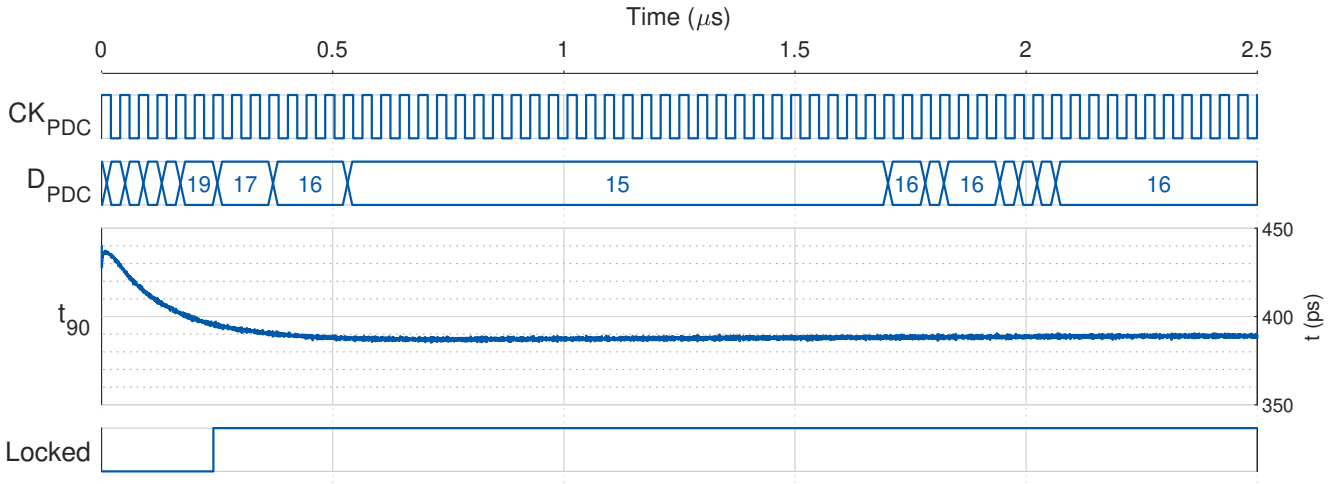


Figure 4.31: Mixed mode simulation of the locking of the DLL with linearized CSI delay cells for $f_{\mathrm{IN}} = 3.2\,\mathrm{GHz}$, $M = 2$ and $TJ_{\mathrm{RMS}} = 400\,\mathrm{fs}$. The DLL is locked within 6 PDC samples (0.24 µs).

f_S. The linearized CSI based DLL has a similar transfer function with a bandwidth of 2.46 MHz at $M = 2$ and $f_{IN} = 3.2$ GHz.

The mixed mode simulation of the locking procedure of the DLL with direct voltage controlled CSI delay cells in Figure 4.30 shows that the DLL is locked to ±1 LSB of the ideal phase $\varphi_{DLL} = 16$ within 54 PDC samples (1.76 µs) for $f_{IN} = 4$ GHz, $M = 2$ and $TJ_{RMS} = 400$ fs in the typical corner. Due to the nonlinear delay characteristic with a low gain over major parts of the delay range, as well as the lower DLL bandwidth, the locking time is significantly longer compared to the SCI based DLL with direct digitally controlled delay cells. The simulated RMS jitter of the 90° output of the CSI based DLL is 0.86 ps. The design shows significantly less dithering of the measurement result and therefore less deterministic jitter due to the high resolution DAC and the second order low-pass filter. As a result the jitter requirements for CK_S are relaxed since a violation of equation (3.27) has a much lower impact on the jitter performance of the CSI based DLL compared to the SCI based DLL.

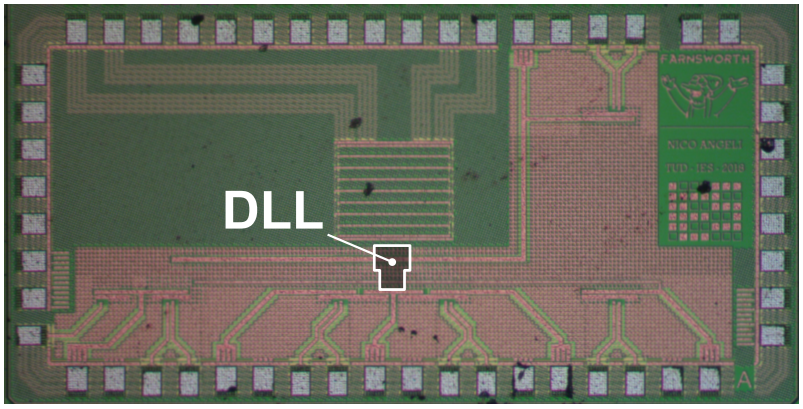
With the linearized CSI delay cells the locking time of the DLL is significantly improved to 6 PDC samples (0.24 µs) by the more equalized gain of the delay cells, as can be seen from the mixed mode simulation results in Figure 4.31. The simulated RMS jitter is also reduced to 0.58 ps as a result of the reduced control voltage ripple at the output of the 3 bit current-DAC.

4.3.4 Measurement Results

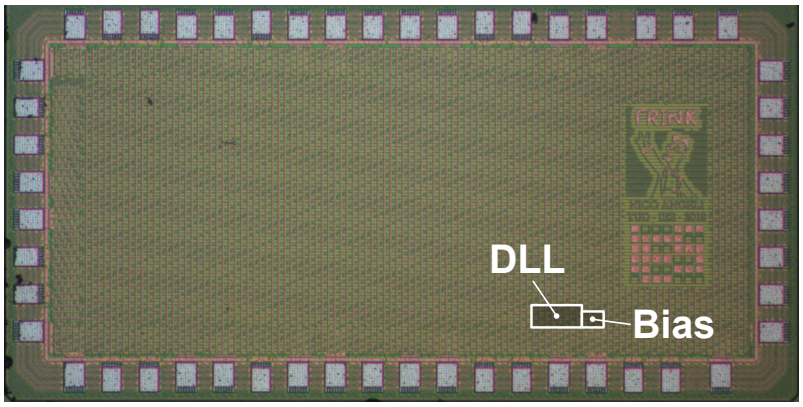
The die photographs of the two test chips in the UMC 65 LL CMOS process are shown in Figure 4.32. The DLL based on direct voltage controlled CSI delay cells in Figure 4.33 (a) has an active area of 0.0086 mm^2 including bias generation and supply decoupling. The size of the digital part is increased over the SCI based DLL due to the additional ΔΣ modulator. The linearized CSI delay cell based DLL in Figure 4.33 (b) occupies 0.0067 mm^2 and is smaller than the direct voltage controlled CSI based DLL due to a more optimized layout and a separate shared bias generation. The simulated power efficiency with $M = 4$ is 0.66 mW/GHz for the direct voltage controlled CSI based DLL and 0.62 mW/GHz for the linearized CSI based DLL.

The measurements of the direct voltage controlled CSI based DLL are performed at $f_{IN} = 3.6$ GHz. Although the DLL itself was designed for frequencies of up to 4 GHz the test chips could not reach this value due to an insufficient bandwidth of the clock input buffer. Above 3.7 GHz the AC-coupled CML2CMOS converter preceding the DLL oscillates because the amplitude of the CML clock signal is too low. Figure 4.35 (a) shows the measured RMS random and periodic jitter for three different PDC scaling factors M. The random jitter is approximately constant with $RJ_{RMS} = 0.85$ ps at $M = 8$ and is in good accordance with the simulated jitter of 0.86 ps. The periodic jitter is noticeably reduced compared to the SCI based DLL (see section 4.2.3) thanks to the voltage control with a high resolution DAC and a second order low-pass filter. The dependency on the sample clock frequency is also clearly reduced and now shows a slight increase with M instead of strong decrease. This is due to the fact that the clock for the ΔΣ modulator is derived from CK_S. Decreasing f_S leads to a lower frequency of the modulator output signal which is less attenuated by the fixed bandwidth low-pass filter. For $M = 8$ the periodic jitter is $PJ_{RMS} = 0.86$ ps. A noticeable improvement of the jitter performance is also visible in the phase noise plot of the 90° output in Figure 4.35 (b).

The phase noise at 1 MHz offset is significantly reduced by 8.7 dB to −119.4 dBc/Hz compared to the SCI based DLL. Also, a change of the PDC scaling factor M has less impact on the phase noise performance. An increase of M leads to a slight increase of the phase noise between 1 MHz and 10 MHz offset and a slight decrease between 100 kHz and 1 MHz offset. The reduced influence of the DLL bandwidth is due to the fixed bandwidth of the second order low-pass filter which effectively suppresses the dithering of the PDC measurement result due to jitter. The integrated RMS jitter from 10 kHz to 100 MHz is 693 fs with $M = 2$ and 771 fs with $M = 8$.

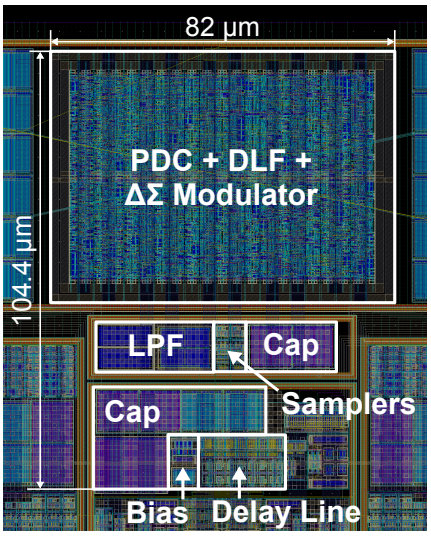


(a) Direct voltage controlled CSI based DLL.

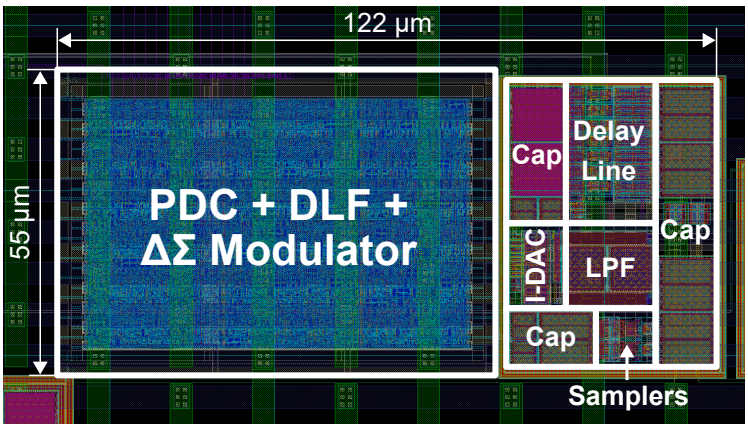


(b) Linearized CSI based DLL.

Figure 4.32: Die photographs of the direct voltage controlled CSI based DLL test chip (a) and the linearized CSI based DLL test chip (b) in the UMC 65 LL CMOS process.

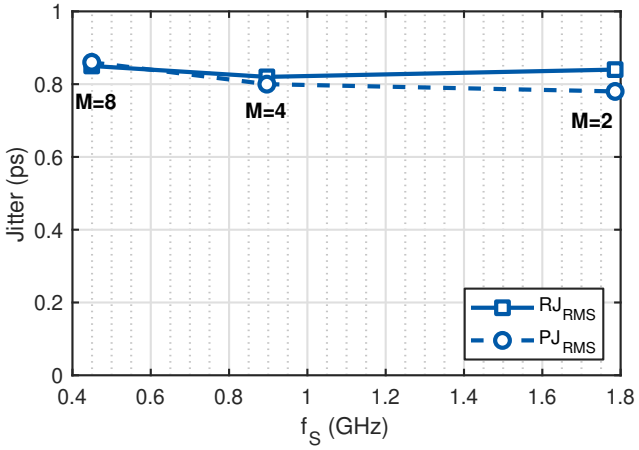


(a) Direct voltage controlled CSI based DLL.

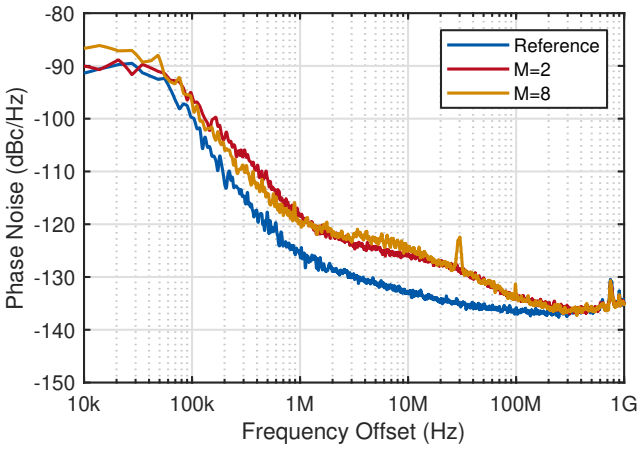


(b) Linearized CSI based DLL.

Figure 4.33: Layouts of the direct voltage controlled CSI based DLL (a) and the linearized CSI based DLL (b) in the UMC 65 LL CMOS process.

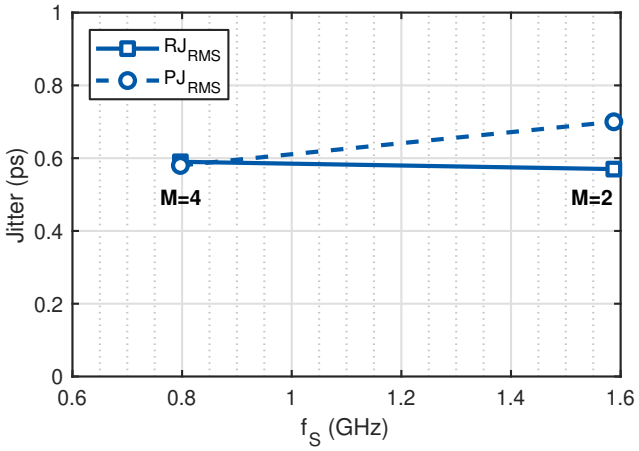


(a) Random and periodic jitter.

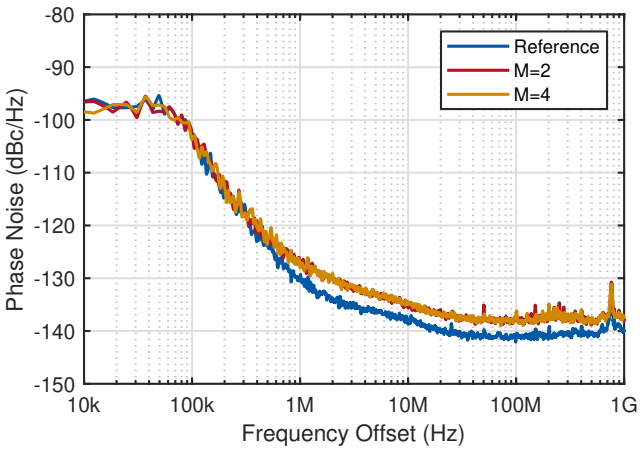


(b) Phase noise.

Figure 4.34: Measured RMS random and periodic jitter (a) and phase noise (b) at the 90° output of the proposed DLL with direct voltage controlled CSI delay cells for $f_{\mathrm{IN}} = 3.6\,\mathrm{GHz}$ and different PDC scaling factors M. The integrated RMS jitter (10 kHz to 100 MHz) is 693 fs with $M = 2$.



(a) Random and periodic jitter.



(b) Phase noise.

Figure 4.35: Measured RMS random and periodic jitter (a) and phase noise (b) at the 90° output of the proposed DLL with linearized CSI delay cells for $f_{\mathrm{IN}} = 3.2\,\mathrm{GHz}$ and different PDC scaling factors M. The integrated RMS jitter (10 kHz to 100 MHz) is 296 fs with $M = 4$.

The benefits of the linearized CSI delay cells can be clearly seen in the measured jitter performance in Figure 4.35 (a). The random jitter is further reduced to 0.57 ps and the periodic jitter is also lower than with the direct voltage controlled CSI delay cells. PJ_{RMS} now again shows a decrease with increasing M and is 0.58 ps for $M = 4$. The phase noise is reduced over the complete frequency range and has a value of $-126.3\,\mathrm{dBc/Hz}$ at 1 MHz offset. There is no visible difference between the phase noise plots at $M = 2$ and $M = 4$. The significant improvement over the direct voltage controlled CSI based design is assumed to be caused by the current-DAC which generates less ripple on the control voltage for the CSI delay cells than a single-bit modulated signal. In combination with the more uniform gain of the delay cells this leads to less jitter induced by the digital modulation and the control word dithering. The overall phase noise performance is close to the reference signal with an integrated RMS jitter (10 kHz to 100 MHz) of 296 fs at $M = 4$.

4.4 Conclusion

Table 4.3 summarizes the performance of the three proposed PDC based DLL designs using digitally controlled SCI delay cells (SCI), direct voltage controlled CSI delay cells (CSI) and linearized CSI delay cells (Lin. CSI) and compares them to selected state-of-the-art all-digital DLLs. A PDC scaling factor of $M = 4$ is chosen for all PDC based designs as a balanced operating condition for the comparison. The area does not include the PLL for the generation of the sample clock CK_S since in the target application of parallel high-speed interfaces CK_S is shared between all instances of the DLL, as well as other PDC based circuits. As described in section 4.2.3 the proportional area and power overhead for the CK_S PLL is negligible for typical interfaces. The DLL design using SCI delay cells has the smallest active area in the comparison thanks to the proposed area efficient transmission gate based shunt-capacitors and the direct digital control without passive low pass filter. The direct voltage controlled CSI based DLL is optimized for an alternative trade-off with a better power efficiency and an improved jitter performance in return for a larger active area. The third design using linearized CSI delay cells achieves improvements in all aspects over the direct voltage controlled CSI based DLL. Especially the jitter performance is significantly improved with a phase noise close to the reference clock and on a similar level as DLL designs with additional LDO [44, 43]. Additionally, the frequency range is extended by application of a secondary digital frequency band selection and the locking time is reduced to only 6 PDC samples.

The comparison of previously published DLL designs shows that there is always a trade-off between the area consumption and the jitter performance and power efficiency. This trade-off is illustrated in Figure 4.36 using the area FoM

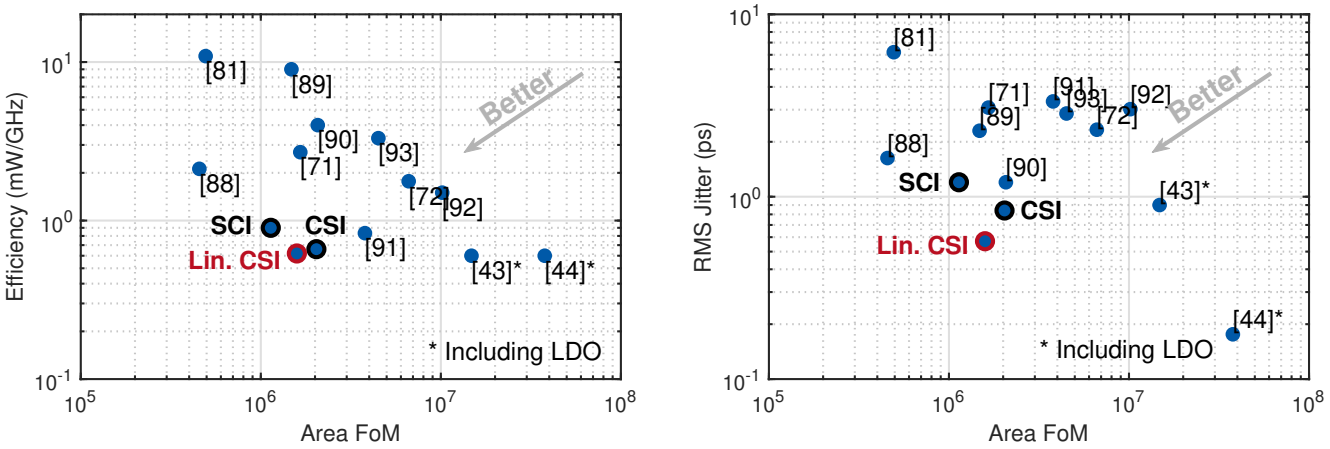
$$a_{\mathrm{FoM}} = \frac{a}{L_{\mathrm{min}}^2} \tag{4.24}$$

with the design area a and the minimum channel length L_{min} of the utilized CMOS process. The best values for power efficiency and RMS jitter are achieved by designs featuring an LDO for the delay line. In return these designs have the largest layout area in the comparison. The proposed PDC based DLL designs

Table 4.3: Performance comparison of the proposed PDC based DLL designs with selected state-of-the-art all-digital DLLs.

Design	Delay Cell Type	Process	Frequency Range (GHz)	Area (mm^2)	Efficiency (mW/GHz)	RMS Jitter (ps)	Phase noise @ 1 MHz (dBc/Hz)
This	SCI	65 nm	2.5	0.0048*†	0.9*	1.2	−110.7
Work	CSI	65 nm	2.8 − 4	0.0086*	0.66*	0.84	−119.4
(M = 4)	Lin. CSI	65 nm	1.4 − 3.3	0.0067*	0.62*	0.57	−126.3
[88]	coarse/fine	130 nm	1.5 − 3.3	0.0077	2.12	1.63	N/A
[71]	coarse/fine	65 nm	0.27 − 2.13	0.007	2.7	3.08	N/A
[72]	N/A	65 nm	0.11 − 2.5	0.028†	1.77	2.33†	−119
[44]	CSI	14 nm	2 − 7.5	0.0074‡	0.6	0.18	−126
[89]	coarse/fine	130 nm	0.4 − 0.8	0.025	9	2.3	N/A
[43]	SCI	130 nm	1.5	0.25‡	0.6	0.9	−129

* Excluding CK_S generation, † Including PI, ‡ Including LDO



(a) Random and periodic jitter.

(b) Phase noise.

Figure 4.36: Comparison of the area FoM and the power efficiency (a) and the area FoM and the RMS jitter (b) of the proposed PDC based DLL designs with digitally controlled SCI delay cells (SCI), direct voltage controlled CSI delay cells (CSI) and linearized CSI delay cells (Lin. CSI) to selected state-of-the-art all-digital DLLs.

offer a power efficiency and an RMS jitter close to these designs without the use of an LDO leading to a factor of ten smaller layout area. The linearized CSI based DLL achieves the best compromise of area, power and jitter among the proposed designs, as well as in the overall comparison.

5 Design of a Phase-to-Digital Converter Based Duty-Cycle Corrector

Double data rate (DDR) signaling is the most common way to transfer data in wire-linked I/O links at high data rates. In order to maximize the timing margin for the sampling of the data in the receiver the clock signals for the serialization and the deserialization require a duty-cycle of 50 %. Due to local PVT variations and mismatch a global duty-cycle correction is not sufficient and local duty-cycle correctors (DCC) in each transceiver slice need to be employed. Figure 5.1 shows the clocking architecture of a typical parallel high-speed I/O link with local clock adjustment circuits based on the example GDDR5 interface in [35]. For 32 parallel I/O links a total number of 48 DCCs is required. Thus, the DCCs are a substantial part of high-speed I/O links and it is desirable to minimize their area and power consumption. In this chapter the design of an all-digital DCC based on the PDC described in chapter 3 as a duty-cycle detector is presented. The results of this chapter were published in [104].

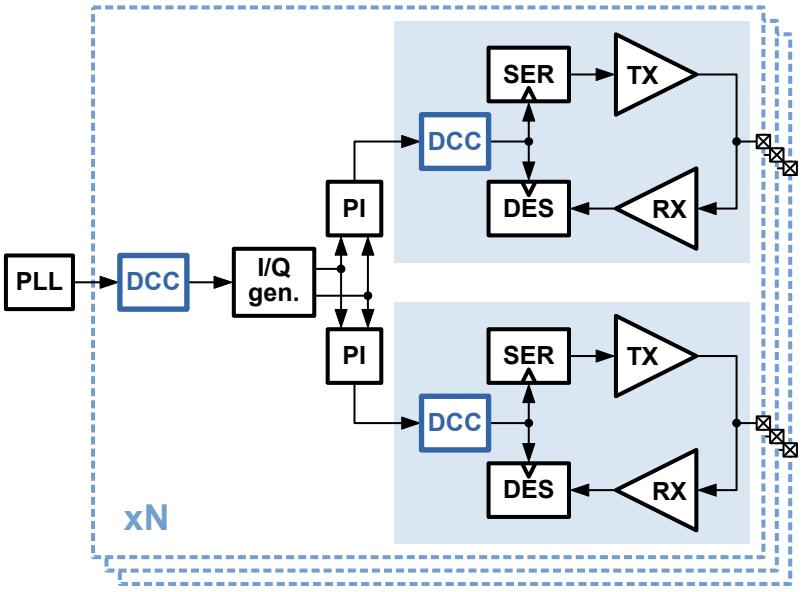
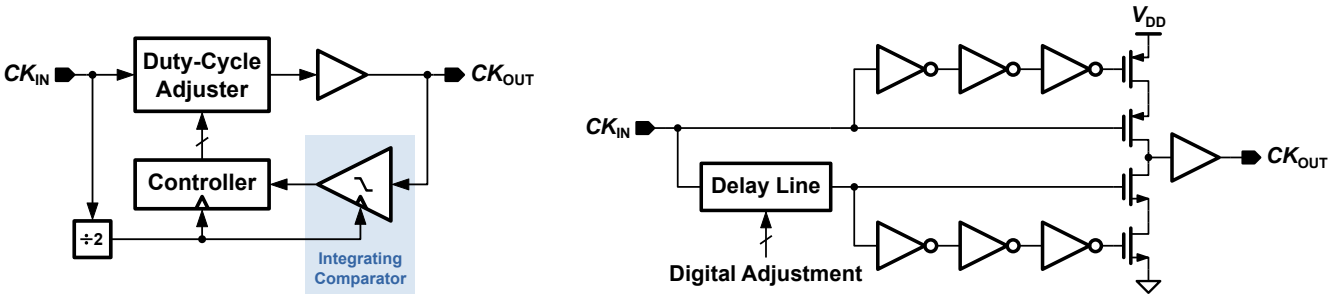


Figure 5.1: Clocking architecture of a typical parallel high-speed I/O link with local clock adjustment circuits.

5.1 Related Work

Most DCCs are implemented as a closed loop system consisting of a duty-cycle detector and a duty-cycle adjuster. Following the trend towards all-digital implementation more and more DCCs use a digital feedback. This requires a digital duty-cycle detector that is able to measure the duty-cycle of a high-frequency clock signal with a fine resolution. Many implementations rely on a circuit that integrates the high and low phases of the clock signal and generates a binary output signal indicating whether the duty-cycle is above or below 50 % [48, 94, 49]. This binary signal is then fed to a digital loop controller (e.g. an SAR circuit [48]) to adjust the duty-cycle of the clock signal. Figure 5.2 (a) shows the block diagram of a typical DCC based on an integrating comparator for duty-cycle detection. A major drawback of this approach is the PVT sensitivity of the full custom integrating comparator. The design must be optimized for low mismatch in the integrating stage and a minimized input referred offset of



(a) Complete duty-cycle corrector. **(b)** Duty-cycle adjuster.

Figure 5.2: Block diagram of a conventional DCC based on an integrating comparator and a digital loop controller (a) and simplified schematic of an edge combiner based duty-cycle adjuster (b).

the comparator. Alternative approaches for the digital duty-cycle detector are based on TDCs [52, 53], oscillators [50, 51] or asynchronous sampling [54] and are elaborated in section 3.1.

A common solution for the duty-cycle adjuster is to generate a 180° phase shifted representation of the input clock signal and use an edge combiner [94, 49, 48] to generate a 50 % duty-cycle clock signal from the 0° and the 180° signals as illustrated in Figure 5.2 (b). In [95] an SR-latch is used instead of an edge combiner to generate the output clock.

5.2 Implementation

The proposed all-digital DCC uses the PDC presented in chapter 3 as a linear digital duty-cycle detector as illustrated in the block diagram in Figure 5.3. Similar to the DLL design in section 4.3 a strongARM latch is used to sample the differential clock signal at the output of the duty-cycle adjuster. An FSM controls the duty-cycle correction and the complete digital part can be switched off after calibration using clock gating on CK_S. This way the power consumption during normal operation is kept low and the introduction of jitter by the asynchronous sampling is avoided.

The DCC consists of an AC-coupled CML2CMOS converter followed by a 5 bit digitally controlled CSI based duty-cycle adjuster (DCA). The CML2CMOS converter is based on inverters with resistive feedback to set the common mode voltage for the input signals. Due to the AC-coupling the design shows a lower frequency bound but therefore provides inherent duty-cycle correction as shown by the simulated

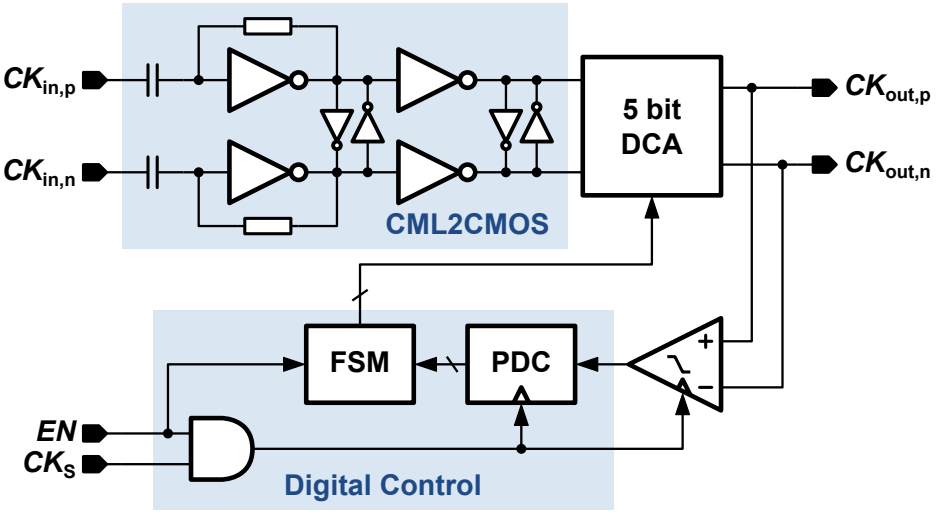


Figure 5.3: Block diagram of the proposed PDC based DCC with an AC-coupled CML2CMOS converter.

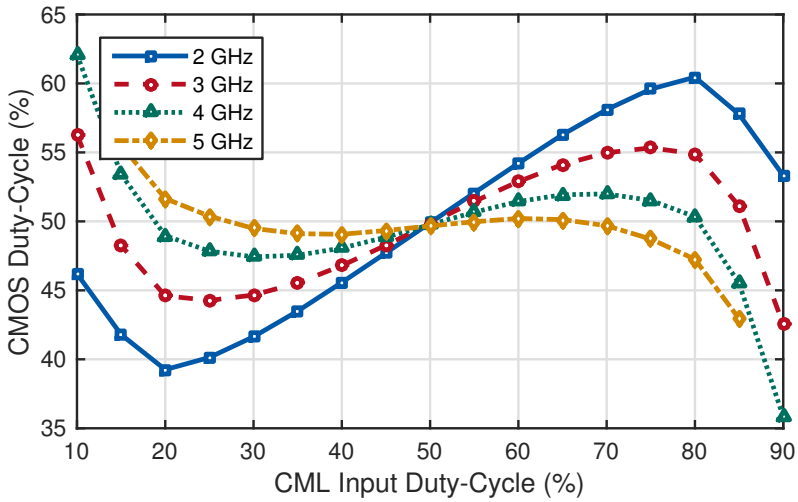


Figure 5.4: Simulated duty-cycle transfer curves of the AC-coupled CML2CMOS converter for different clock frequencies.

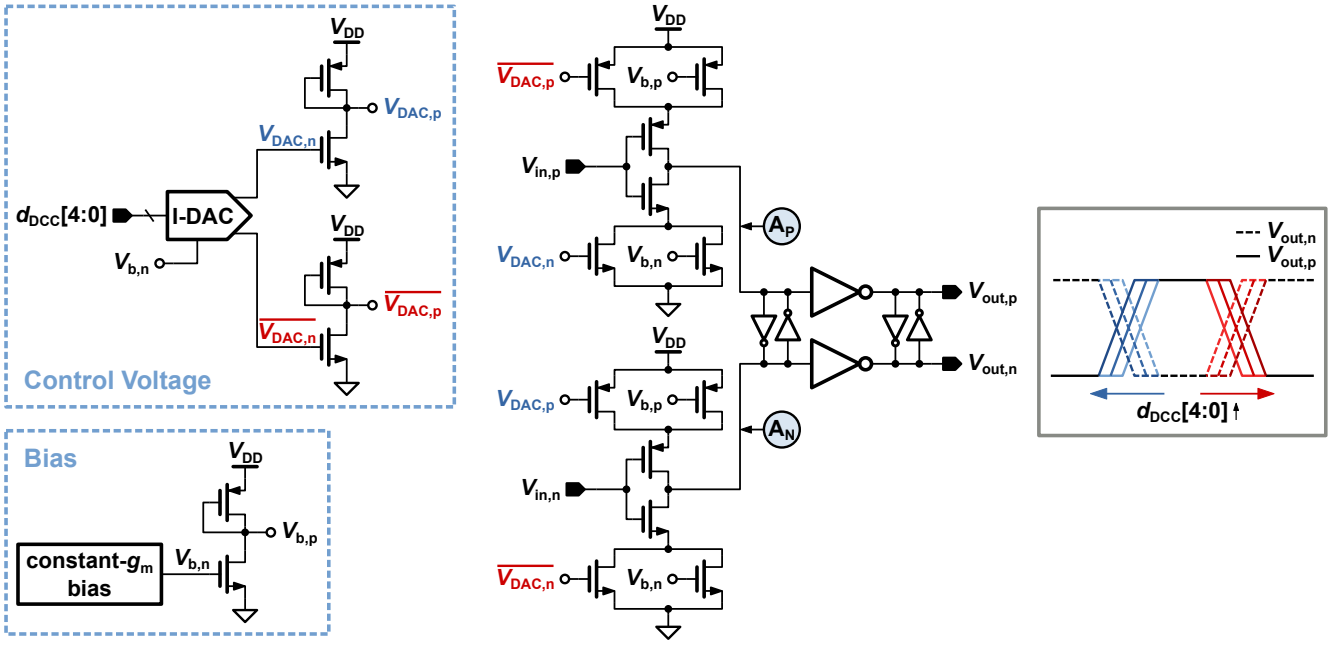
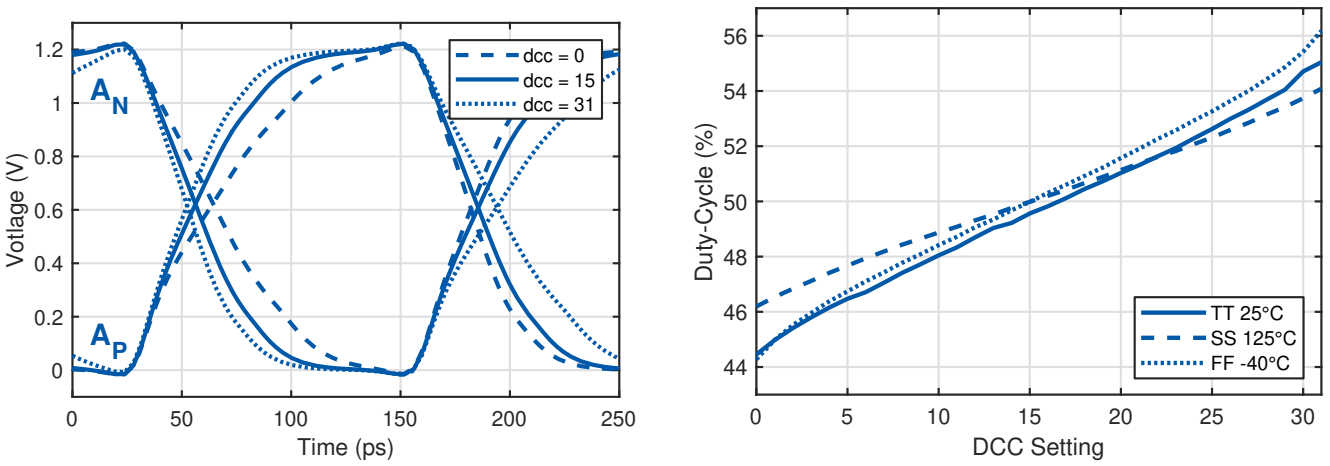


Figure 5.5: Schematic of the proposed 5 bit digitally controlled duty-cycle adjuster based on a pseudo-differential CSI delay cell.

duty-cycle transfer curves in Figure 5.4. This reduces the required range of the 5 bit duty-cycle adjuster to cover the desired input duty-cycle range of at least $40-60\,\%$. At 5 GHz the output duty-cycle error of the CML2CMOS converter is even below 1 % for an input duty-cycle range of $25-70\,\%$. An extension of the frequency range of such a CML2CMOS converter towards lower frequencies is proposed in [96] and enables the application in variable data rate systems with ultra low power modes. The schematic of the 5 bit duty-cycle adjuster in Figure 5.5 is based on a pseudo-differential CSI delay cell. But in contrast to the delay cells in section 4.3 the proposed duty-cycle adjuster individually controls the current of the four current sources for the PMOS and NMOS supplies of the differential delay cell. This way the delays of the rising and falling edges of both the positive and the negative signal of the differential clock can be adjusted independently. A fixed bias limits the maximum delay of each edge and a 5 bit current-steering DAC generates the complementary control voltages $V_{\mathrm{DAC,n}}$ and $\overline{V_{\mathrm{DAC,n}}}$. The control voltages for the PMOS current sources $V_{\mathrm{DAC,p}}$ and $\overline{V_{\mathrm{DAC,p}}}$ are generated using current mirrors. For increasing the duty-cycle of



(a) Waveforms at nodes A_P and A_N.

(b) Duty-cycle range.

Figure 5.6: Simulated waveforms of the proposed 5 bit duty-cycle adjuster in Figure 5.5 at the nodes A_P and A_N (a) and simulated output duty-cycle range for the typical, best and worst case corners (b) at $f_\mathrm{IN} = 4\,\mathrm{GHz}$ and an input duty-cycle of 50 %.

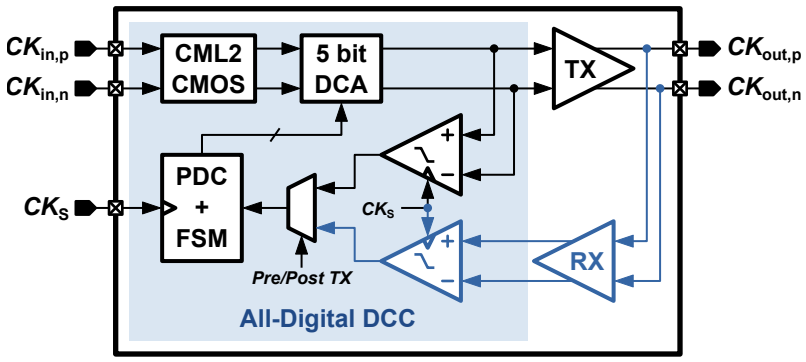


Figure 5.7: Block diagram of the DCC test chip with additional post TX measurement option.

the output signal the delay of the rising edge at node A_P is increased and the delay of the falling edge is decreased. At node A_N the opposite is the case, the delay of the rising edge is decreased and the delay of the falling edge is increased. The CSI delay cells have a nonlinear delay characteristic (see non-equidistant spacing of the zero-crossing points in Figure 5.6 (a)), but due to the complementary control of the rising and falling edges this nonlinearity cancels out resulting in a nearly linear duty-cycle characteristic of the complete duty-cycle adjuster (see Figure 5.6 (b)). The characteristics for the typical, best and worst case corners are very similar with a minimum adjustment range of $46 - 54\,\%$. In combination with the CML2CMOS converter the complete DCC is able to correct input duty-cycles in the range of $15 - 85\,\%$ for clock frequencies above 3 GHz and in the range of $40 - 60\,\%$ for clock frequencies between 2 GHz and 3 GHz.

The test chip for the DCC in the UMC 65 LL CMOS process features the option to feed back the duty-cycle after the off-chip driver instead of directly after the duty-cycle adjuster. This way the duty-cycle distortion of the off-chip driver can be included in the measurement and a reduced duty-cycle error at the pads of the chip is achieved. A typical receiver front-end with a CML preamplifier is used to feed the clock signal from the off-chip driver to a second sampling latch of the PDC. In bidirectional transceivers a receiver front-end is already existing and therefore the additional hardware cost is negligible. Figure 5.7 shows the architecture of the test chip with external clock generation.

Another effect that has an influence on the duty-cycle measurement is the input referred offset of the sampler. A standard deviation of the sampled duty-cycle of less than 0.4 % was observed in a Monte Carlo simulation of the strongARM latch with a worst case duty-cycle error of 1 %. The precision of the strongARM latch without offset calibration is therefore sufficient for the DCC to compete with state-of-the-art designs that reach a duty-cycle error of $\geq 1\,\%$. However, an offset calibration can further reduce the duty-cycle error and is applied in many high-speed transceiver designs anyway.

The influence of jitter on the measurement result is handled by averaging multiple samples of the PDC as described in chapter 3. The total duration of a duty-cycle measurement is then

$$t_\mathrm{m} = (n_\mathrm{p} + n_\mathrm{avg}) \frac{NM+1}{f_\mathrm{IN}}, \tag{5.1}$$

with the PDC parameters N and M, the number of samples for the averaging n_avg and the number of samples it takes to propagate the input duty-cycle to the output of the PDC $n_\mathrm{p} = 2$. The FSM starts with the duty-cycle adjuster in the center setting and depending on the measurement result the duty-cycle adjuster control word is incremented or decremented until the desired duty-cycle of 50 % is reached. The maximum amount of iterations is therefore 16 for a worst case duty-cycle error at the input of the DCC.

5.3 Measurement Results

Figure 5.8 shows a die photograph of the DCC test chip. The digital part (PDC and FSM) occupies 3200 µm^2 and the duty-cycle adjuster consisting of CML2CMOS converter, 5 bit CSI delay cell, current-DAC and sampling latch occupies 330 µm^2 of active area. The placement of the digital part is flexible and allows to place the duty-cycle adjuster close to the other clocking circuits keeping the paths for the high-speed clock as short as possible. The power consumption of the analog components is 1.26 mW at

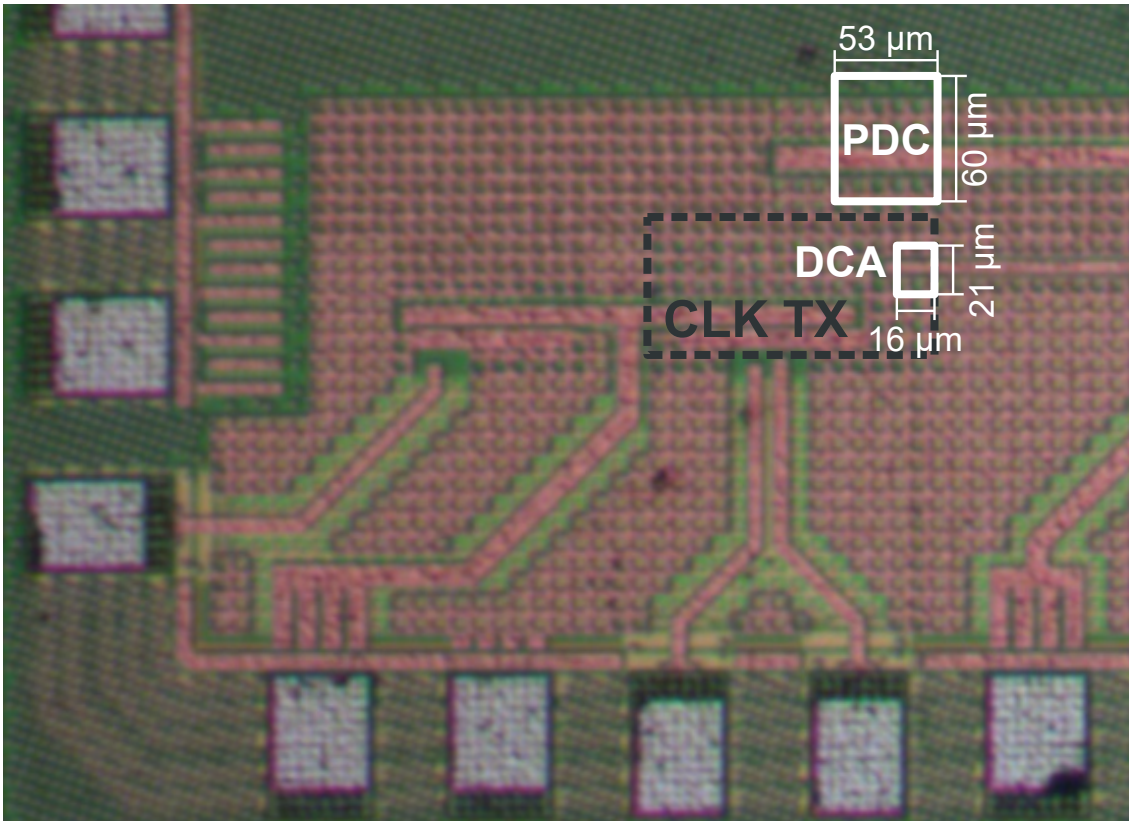
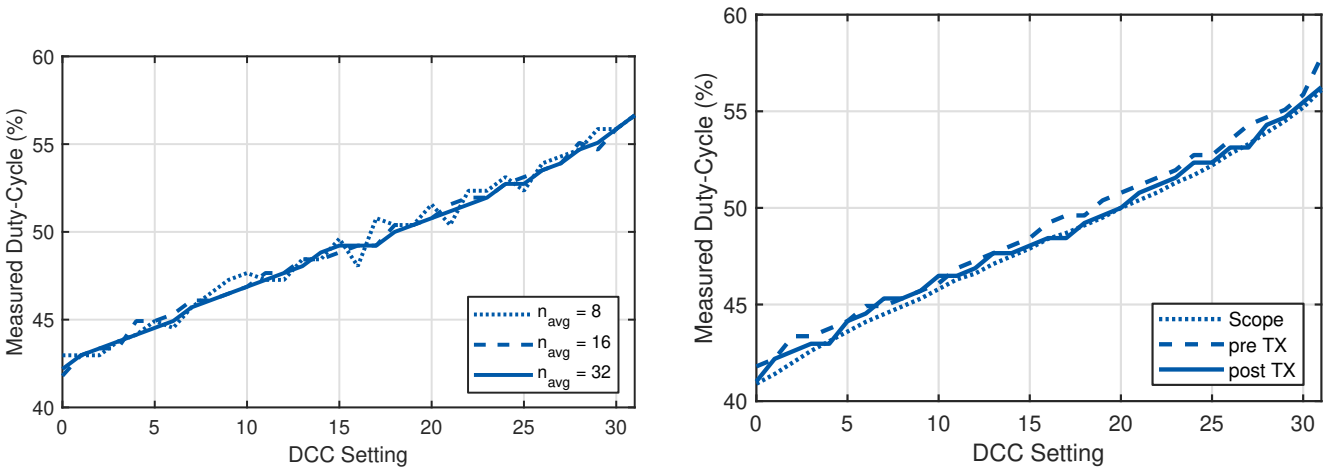


Figure 5.8: Die photograph of the DCC test chip.



(a) Averaging.

(b) Pre and post TX measurement.

Figure 5.9: Measured duty-cycle across the duty-cycle adjuster range using the integrated PDC with different amounts of averaging (a) and using the pre and post TX measurement options compared to an oscilloscope measurement (b) with $M = 8$ and $f_{\mathrm{IN}} = 3.6\,\mathrm{GHz}$.

$f_{\mathrm{IN}} = 4\,\mathrm{GHz}$, while the power consumption of the digital part depends on the PDC scaling factor M and is only active during the calibration. With $M = 8$ the digital part consumes 0.655 mW at a sample clock frequency of $f_{\mathrm{S}} = 499\,\mathrm{MHz}$.

Since the DCC is on the same test chip as the DLL in section 4.3.1 it suffers from the same problem of an insufficient bandwidth of the clock input buffer. The following measurements are therefore performed at $f_{\mathrm{IN}} = 3.6\,\mathrm{GHz}$. In simulations the design is able to operate at up to 5 GHz. In Figure 5.9 (a) the duty-cycle across the adjustment range measured by the PDC is plotted for three different averaging depths n_{avg}. A significant improvement from $n_{\mathrm{avg}} = 8$ to $n_{\mathrm{avg}} = 16$ is visible, while $n_{\mathrm{avg}} = 32$ shows only a small difference to the $n_{\mathrm{avg}} = 16$ measurement. In Figure 5.9 (b) duty-cycle measurements before and after the off-chip driver with $n_{\mathrm{avg}} = 32$ are compared to an external oscilloscope measurement. The post TX measurement clearly shows a better matching with the oscilloscope measurement than the pre TX measurement with a maximum deviation of 0.8 %. The pre TX measurement has a constant offset to the oscilloscope measurement due to the duty-cycle distortion of the off-chip driver. After calibration using the FSM the duty-cycle error at the output of the chip is 0.9 % with the pre TX measurement and < 0.1 % with the post TX measurement. The measured total jitter at the output is $TJ_{\mathrm{RMS}} = 1.87\,\mathrm{ps}$ and $TJ_{\mathrm{pk-pk}} = 12.97\,\mathrm{ps}$ (10k samples).

5.4 Conclusion

The performance of the proposed PDC based DCC is summarized in Table 5.1 and compared to selected state-of-the-art designs. The area FoM in equation (4.24) is used to compare the design area of the DCCs. The area and power of the PLL for the generation of CK_{S} are not included for the same reasons as in the comparison of the DLL designs in chapter 4. The proposed DCC achieves the smallest design area and the best power efficiency (0.48 mW/GHz with $M = 8$) in the comparison. Also, a slight improvement over the state-of-the-art designs is achieved for the RMS and peak-to-peak jitter, as well as the maximum duty-cycle error. The duty-cycle correction range is wide for frequencies between 2 GHz and 3 GHz ($15 - 85$ %), but relatively narrow for frequencies above 3 GHz ($40 - 60$ %). However, in the target application the expected duty-cycle error (e.g. at the output of the phase interpolator) is within $45 - 55$ %. The proposed DCC is intended as a supplement to one of the DLL designs presented in chapter 4 and therefore the overall system efficiency can profit from the fact that the CK_{S} generation can be shared between all DLL and DCC instances.

Table 5.1: Performance comparison of the proposed PDC based DCC with selected state-of-the-art DCCs.

Design	Frequency Range (GHz)	Area FoM	Efficiency (mW/GHz)	RMS Jitter (ps)	Peak-to-Peak Jitter (ps)	Correction Range	Max. Duty-Cycle Error
This	$2-5$	0.876^{*}	0.48^{*}	$1.87^{\dagger}$	$12.97^{\dagger}$	$\geq 40-60\,\%$	$0.8\,\%$
[97]	$0.12-2$	6.149	2.09	N/A	12.53	$20-80\,\%$	$2\,\%$
[49]	$0.35-1$	3.491	5.6	2.9	20.5	$10-90\,\%$	$1.4\,\%$
[94]	$0.12-2$	4.497	N/A	2.7	14.0	$20-80\,\%$	$0.9\,\%$
[98]	$0.262-1.02$	2.367	6.5	3.23	23.64	$14-86\,\%$	$1.4\,\%$
[48]	$0.3125-1$	2.84	3.2	2.05	15.5	$40-60\,\%$	$1\,\%$
[52]	$0.001-1.3$	1.759	3.7	1.95	13.2	$30-70\,\%$	N/A

* Excluding generation of CK_{S}, † Measured at $f_{\mathrm{IN}} = 3.6\,\mathrm{GHz}$

Part III

Phase-to-Digital Converter Based Fast Link Training of Asymmetric Bidirectional Source-Synchronous I/O Links

6 Timing Center Calibration by Receiver Phase Measurement

Clock centering is one of the key calibration steps in high-speed I/O links. The receiver clock phase is adjusted to sample the data signal exactly in the middle of the data eye. This way the robustness against timing variations of the data or the clock signal is maximized by providing a sufficient timing margin. In parallel high-speed I/O links the clock centering can also be used to compensate trace length differences within the data bundle to simplify the PCB layout. The clock centering is usually performed at the startup of the system and is repeated in regular intervals to compensate timing drift due to voltage or temperature variations in the system. Since the data bus is not available during the clock centering it is desirable to minimize the duration of the calibration procedure. The proposed method based on receiver phase measurement reduces the calibration time significantly compared to the state-of-the-art passing window approach. Another advantage of the proposed method is that nonlinearities of the phase generation circuits can be compensated by additional calibration steps which don't require any signals on the data bus and therefore consume less power than the initial calibration. This results in a minimized timing center error providing a reliable timing margin under all operating conditions. The concept is evaluated using a 6.4 Gb/s/pin bidirectional asymmetric source-synchronous I/O link based on a prototype transceiver ASIC implemented in the UMC 65 LL CMOS process. The prototype system uses differential signaling for the data links in order to avoid the need for a voltage threshold calibration and to focus only on the timing center calibration. Major parts of this chapter were published in [107].

6.1 Related work

In section 2.5.4 the passing window approach for calibration of the timing center of high-speed I/O links is presented. A sweep of the receiver clock phase across one unit interval is performed and the BER at each clock phase position is determined to find the borders between the pass region (BER below the calibration limit) and the fail regions (BER above the calibration limit). This requires the transmission of a known data pattern with $1/BER_{\mathrm{min}}$ bits (typ. 10^3 to 10^4) and a comparison of the received data with the transmitted data pattern. The total duration of the passing window calibration is then

$$t_{\mathrm{pw}} = n_{\mathrm{PI}} \frac{1}{BER_{\mathrm{min}}} \frac{1}{f_{\mathrm{bit}}} \tag{6.1}$$

with the number of phase interpolator steps n_{PI} within one UI (e.g. 64) and the data rate f_{bit}. Due to jitter on the data and clock signals the borders between the pass and the fail regions vary between consecutive measurements. Therefore the conventional timing center calibration performs multiple repeated phase sweeps and computes the average to improve the precision of the calibration [40]. The calibrated timing center is defined as the center of the passing window. Due to the nonlinearity of the phase interpolation circuits the actual sampling clock phase can be offset from the calibrated timing center.

An alternative approach to the passing window method presented in [40] is to use a clock pattern instead of a data pattern. Again, the receiver sweeps the clock phase across one UI, but instead of measuring the BER the receiver only tracks whether a one or a zero is sampled at the current phase position. Based on the average of multiple phase sweeps the boundary between the one and the zero regions is determined which reflects the position of the rising or falling edge of the clock signal in relation to the internal clock. This position is called the *fuzz mean* and the receiver clock phase is adjusted to be 90° offset from the fuzz mean. Similar to the passing window method the fuzz mean approach is prone to

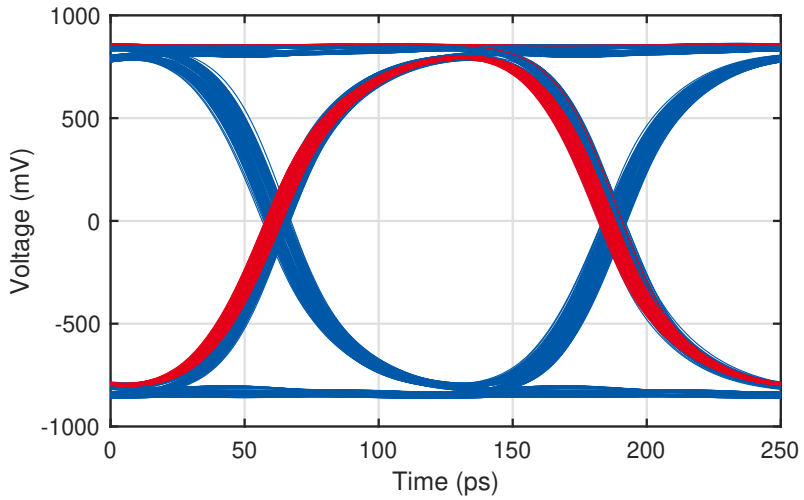


Figure 6.1: Simulated eye diagrams at the receiver of an 8 Gb/s differential I/O link with a PRBS7 data pattern (blue) and a clock pattern (red).

phase interpolator nonlinearity resulting in a clock centering error. Simulation results in [40] show that the calibrated timing center using a clock pattern can be slightly offset from the calibrated timing center using a pseudo random data pattern, but compared to the phase interpolator nonlinearity this error is small. The fuzz mean approach reduces the calibration time significantly since the clock pattern can be much shorter than the pseudo random data pattern as no BER measurement is performed. Figure 6.1 shows a comparison between the simulated eye diagrams at the receiver of an 8 Gb/s differential I/O link using a PRBS7 data pattern and a clock pattern. The difference between the calculated eye centers using the two different patterns is only 1.5 ps (0.012 UI). Other effects that influence the calibration with the fuzz mean approach are duty-cycle distortion and input referred offset of the sampler. These effects are canceled out by performing the edge detection for the rising and the falling edge of the clock pattern and by measuring an inverted clock pattern as well. This in return increases the calibration time, but the method is still faster than the passing window calibration.

In [99] the use of an adaptive step size is proposed to reduce the calibration time of the passing window method by reducing the number of data pattern transfers between the bus nodes. The calibration time in [99] is approximated by

$$t_{\mathrm{pw,adaptive}} \approx \frac{n_{\mathrm{PI}}}{K} \frac{1}{BER_{\mathrm{min}}} \frac{1}{f_{\mathrm{bit}}} \tag{6.2}$$

with the maximum gain K of the adaptive step size algorithm. A value of $K = 2$ is proposed by the authors and therefore the calibration time can be reduced at most by a factor of two.

6.2 Concept of PDC Based Timing Center Calibration

The proposed timing center calibration method uses a clock pattern on the data links similar to the fuzz mean method. But instead of sweeping the receiver clock phase the proposed method performs a single phase measurement using the PDC presented in chapter 3. This way the phase shift between the data link and the receiver clock is determined without repeated measurements at different phase settings. The precision is improved by averaging multiple samples of the PDC measurement result and duty-cycle distortion as well as sampler offset are compensated by measuring the phase on the rising and on the falling edges of the clock signals. The alternating measurement of the rising and falling edges happens inside the PDC and does not increase the measurement time compared to a measurement on only the rising or falling edges. Figure 6.2 compares the flow charts of the conventional timing center calibration method and the proposed PDC based receiver phase measurement approach. Both flow charts

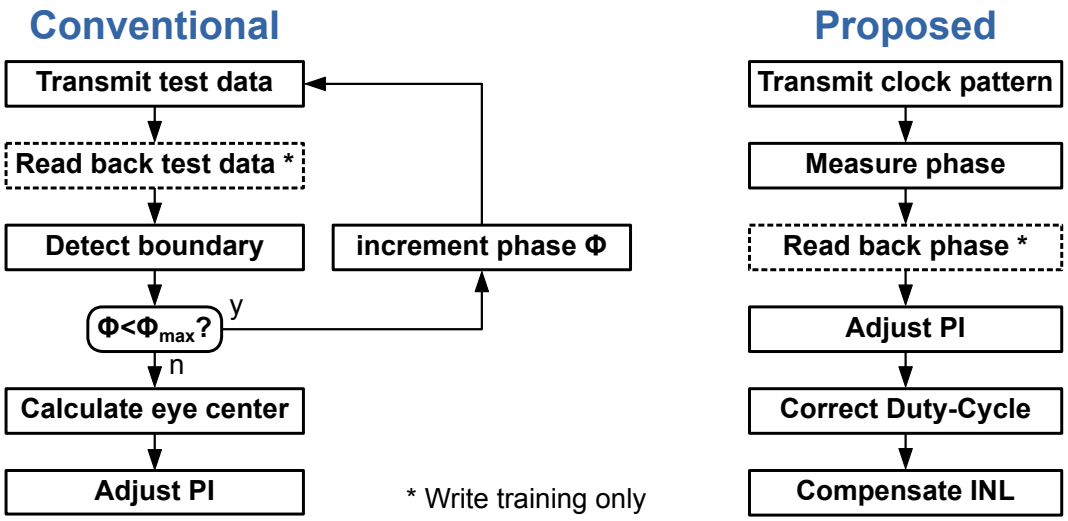


Figure 6.2: Flowchart comparison of the conventional timing center calibration methods and the proposed approach based on receiver phase measurement using a PDC.

describe the link training of an asymmetric bidirectional source-synchronous I/O link (see section 2.2). As the typical application of such I/O links is the interface between a processor and a memory the terms *read training* and *write training* will be used in the following to distinguish between the link training procedures for the two directions in which the interface can be operated. The *read* operation transfers data from the memory to the controller device and the *write* operation transfers data from the controller to the memory device. Since the clock adjustment happens only in the controller device the write training has an additional step of transmitting data back to the controller device that is not required for the read training. Sections 6.2.2 and 6.2.3 go deeper into the differences between the read and the write training procedures. The conventional passing window or fuzz mean methods sweep over the receiver clock phase ϕ and transmit a set of test data (PRBS or clock pattern) in every iteration of the loop. After the detection of the boundaries the eye center is calculated and the PI is adjusted accordingly. In the proposed PDC based clock centering the I/O link first transmits a clock pattern and the receiver measures the phase difference to the internal clock using the PDC. After the initial phase measurement the I/O link is set to the idle state and the PI is adjusted to 90° offset from the received clock pattern. The duty-cycle variation and the nonlinearity of the PI are compensated in two optional correction steps which perform measurements on the internal clock signal.

6.2.1 Differential Transceiver Architecture

Figure 6.3 shows a block diagram of the differential asymmetric bidirectional source-synchronous I/O link for the evaluation of the proposed PDC based clock centering method. The system consists of a controller device and a memory device with a unidirectional data clock CK, two bidirectional data links (DQ) and a unidirectional source-synchronous 4 bit command bus with the separate clock CK_{CMD}. The controller device provides clock adjustment circuits (PI and DCC) for each individual DQ link while the memory device has no clock adjustment circuits. A PDC in each DQ slice is used to measure the phase shift between the received clock pattern and the internal clock. A single transceiver ASIC was implemented that can be configured to act as either the controller or the memory device. A detailed description of the configuration options is given in section 6.3.1. The command bus operates on a lower frequency than the data links and is used to configure the memory device for read or write operation and to initiate the link training procedures similar to the GDDR5/5X/6 standard. PRBS generators and checkers are connected to each DQ slice to generate data patterns and to measure the BER for evaluation of the link performance. The PDC sample clock CK_{S} for the controller device is generated using a PLL locked to the same reference as the DQ clock. For the memory device there are two options to provide CK_{S}. Either the sample clock

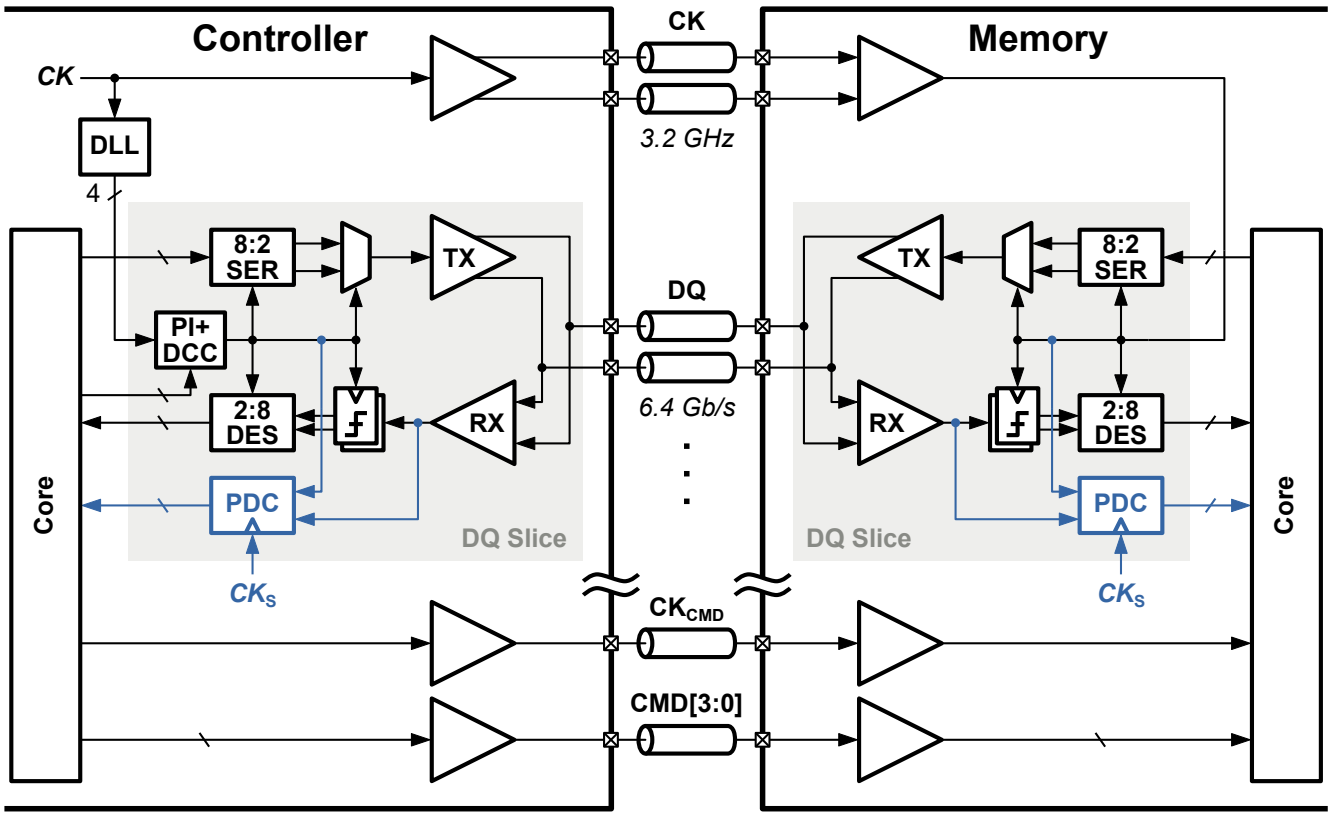


Figure 6.3: Block diagram of the proposed differential asymmetric bidirectional source-synchronous I/O link with PDC based timing center calibration.

is forwarded from the controller, which requires two additional pins for the differential clock signal, or CK_S is generated from CK using an additional PLL in the memory device. Since CK_S is only required in the memory device during the write training the forwarding or the PLL can be switched off during read training and normal operation. Today, most high-performance DRAM ICs include a PLL for jitter reduction on CK, so the feasibility of implementing a PLL in a DRAM process is already proven. Since only one PLL is needed to generate CK_S for all transceivers in the DRAM IC the cost is small compared to the cost of additional pins for clock forwarding.

6.2.2 Read Training

The read training procedure of the proposed PDC based timing center calibration method is illustrated in Figure 6.4. The memory device sends a constant clock pattern on all DQ links and the controller device measures the phase difference ϕ_{RX} to the internal read clock. The PI in each DQ slice is then adjusted to the ideal 90° offset from the received clock pattern. The main source for timing center errors in the conventional calibration methods is the integral nonlinearity (INL) of the PI. The proposed PDC based link training overcomes this problem by an additional INL compensation step. The PDC measures the clock phase of the internal clock again after the initial adjustment and readjusts the PI in order to reach the desired phase shift. In the case of read training the controller can either perform another measurement between the received clock pattern and the internal clock or it can measure the phase of the internal clock with respect to a reference clock before and after the initial adjustment. The later has the advantage that the DQ link is idle during the INL compensation which reduces the power consumption of the system. With a phase ϕ_0 of the internal clock at the initial PI setting d_0 the corrected PI setting d_1 is calculated as

$$d_{n+1} = d_n + \frac{\phi_{ideal} - \phi_n}{360°} d_{max} \tag{6.3}$$

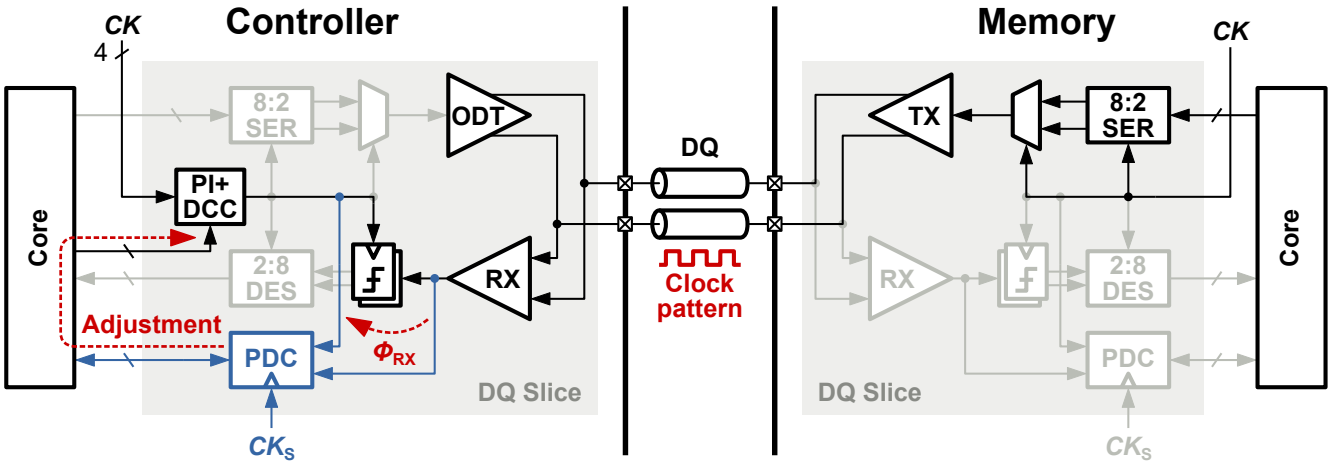


Figure 6.4: Illustration of the read training procedure in the proposed PDC based timing center calibration method.

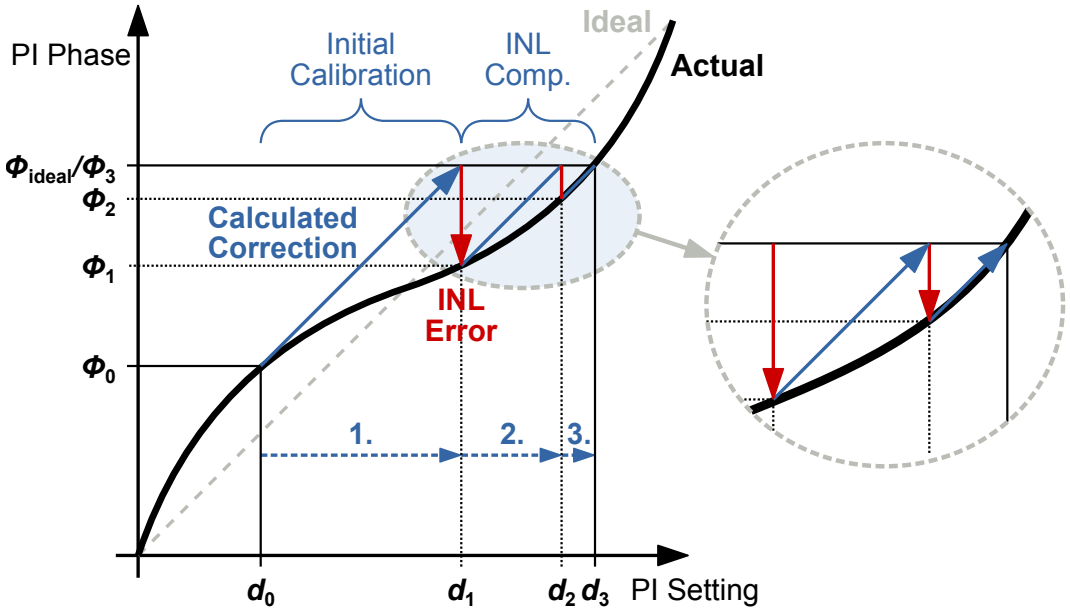


Figure 6.5: Illustration of the phase interpolator nonlinearity compensation in the proposed PDC based timing center calibration method.

with the target phase

$$\phi_{\text{ideal}} = \phi_0 + (90^\circ - \phi_{\text{RX}}) \tag{6.4}$$

assuming an ideal PI with a linear characteristic and a maximum setting d_{max}. The second measurement after the initial adjustment yields the actual phase ϕ_1 with an INL error

$$\epsilon_{\text{inl}} = \phi_{\text{ideal}} - \phi_{\text{n}} \tag{6.5}$$

to the desired phase ϕ_{ideal}. The INL compensation repeats the calibration step in equation (6.3) until the INL error ϵ_{inl} is sufficiently small. Figure 6.5 illustrates the INL compensation for a worst case example with a large INL error after the initial adjustment. PIs typically show a duty-cycle variation across the PI range which can also be compensated by the proposed PDC based timing center calibration method. The prototype transceiver ASIC uses the DCC presented in chapter 5 after each PI to correct the duty-cycle. One effect that needs to be considered in the calibration procedure is that the DCC shifts the phase of the

output clock when the duty-cycle is adjusted. Once the initial INL error is compensated the duty-cycle variation caused by the following iterations is small and can be neglected. Therefore the duty-cycle correction happens before the INL compensation (see the flow chart in Figure 6.2).

The link training duration can be calculated using the duration t_{m} of a single PDC phase or duty-cycle measurement in equation (5.1) based on the PDC parameters M and N, the clock frequency f_{CK} and the number of averaged samples n_{avg}. With $M = 2$ and $n_{\mathrm{avg}} = 16$ the duration of a single measurement is $t_{\mathrm{m}} = 766\,\mathrm{ns}$ at $f_{\mathrm{CK}} = 3.2\,\mathrm{GHz}$. The total link training time

$$t_{\mathrm{tr}} = (1 + n_{\mathrm{inl}} + n_{\mathrm{dcc}})\, t_{\mathrm{m}} \tag{6.6}$$

is composed of the measurement for the initial phase adjustment, the number of INL compensation steps n_{inl} and the number of duty-cycle correction steps n_{dcc}. The number of iterations for the INL compensation depends on the INL error and is limited by the FSM to $0 \leq n_{\mathrm{inl}} \leq 2$. The number of iterations for the duty-cycle correction depends on the duty-cycle error. The DCC in the prototype ASIC (see chapter 5) has a resolution of 0.32 % and the measured worst case duty-cycle error of the PI is 2.5 %. The duty-cycle correction is therefore completed within $1 \leq n_{\mathrm{dcc}} \leq 8$ iterations in any case. The worst case training duration according to equation (6.6) is then $t_{\mathrm{tr,max}} = 8.426\,\mu\mathrm{s}$.

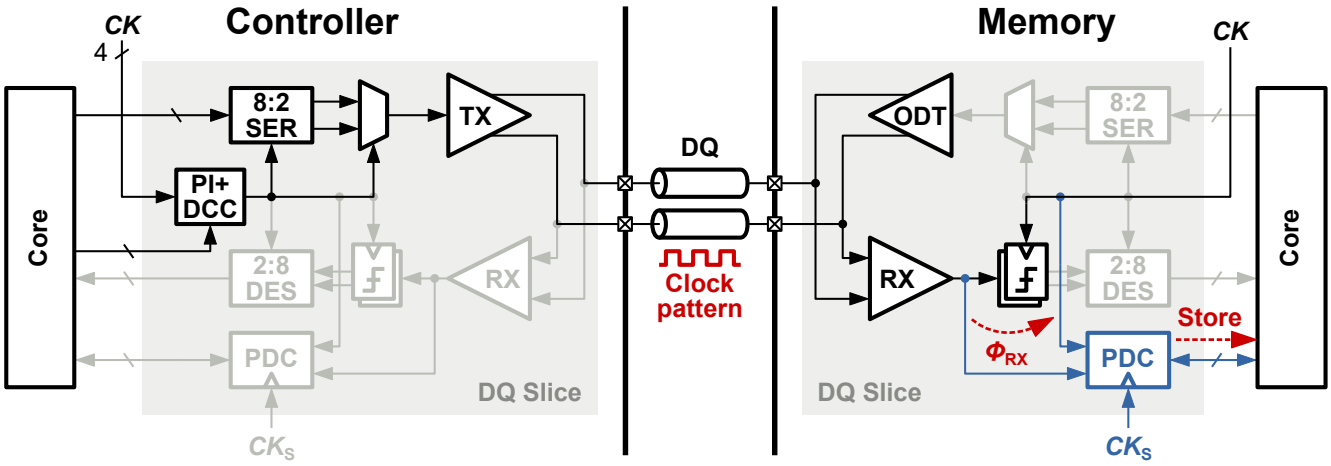
6.2.3 Write Training

The write training procedure of the proposed PDC based timing center calibration is divided into two steps. First the controller device sends a clock pattern on all DQ links to the memory device, where the PDC in each transceiver slice measures the phase difference ϕ_{RX} to the internal clock and the individual measurement results (8 bit) are stored (see Figure 6.6 (a)). Simultaneously, the controller performs a phase measurement between the local write clocks and a reference clock to obtain the initial phase ϕ_0 for the INL compensation. In the second step the memory device sends the measurement results of the PDCs back to the controller device (see Figure 6.6 (b)). Since the read training is performed in advance of the write training the timing center for the read operation is already calibrated and a reliable data transmission is possible. The controller adjusts the PIs in each transceiver slice to the ideal write phase using the calibration step in equation (6.3). Subsequently, the duty-cycle correction and the INL compensation are performed in the same way as in the read training.

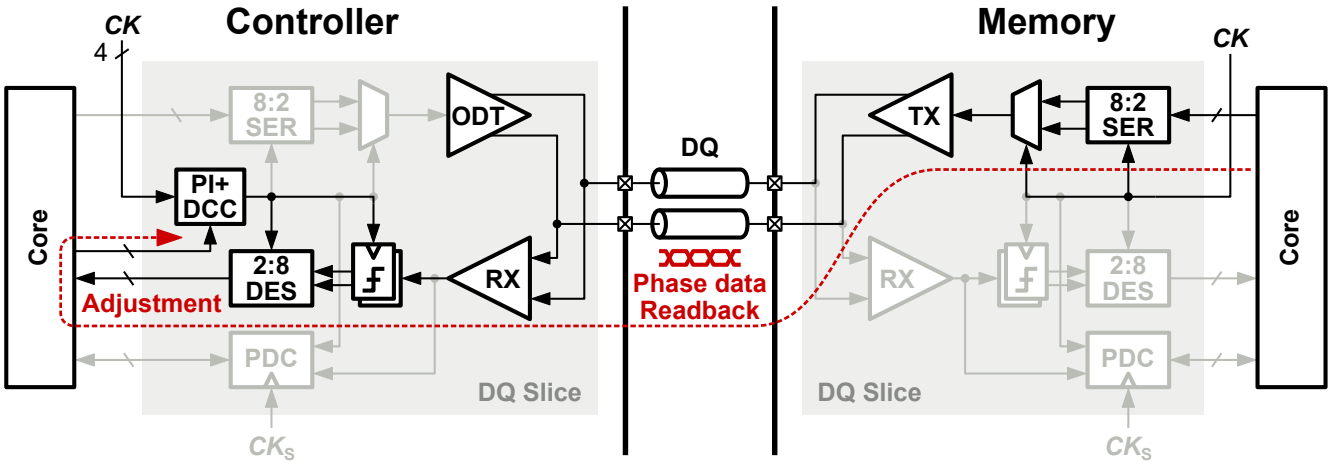
The duration of the write training procedure in the proposed PDC based timing center calibration is only slightly longer than the duration of the read training. The time for the readback of the PDC measurement results of the individual data links must be considered in addition to the link training duration in equation (6.6). Since only one byte (8 bit) of information must be transferred this additional time is small compared to the duration of the PDC measurements. In the conventional timing center calibration methods the complete data or clock pattern that is send to the memory device must be read back to the controller device at every phase position in order to evaluate the received data and detect the timing center. Therefore the proposed method shows an even more significant improvement of the write training duration compared to the read training duration.

6.3 Implementation

The following sections present implementation details of key circuits in the prototype ASIC for the proposed PDC based timing center calibration that are relevant for understanding the calibration concept and the functionality of the prototype interface. Circuits that are not presented in detail as they don't advance the state-of-the-art are the impedance calibration, which is based on the concept explained in section 2.5.1, the PRBS generators based on linear feedback shift registers and the digital control logic including the link training FSM and the SPI interface. The prototype ASIC is implemented in the UMC 65 LL CMOS process and a supply voltage of 1.2 V is used for all high-speed core and I/O circuits to avoid the need for high-speed level-shifters in the transceivers.



(a) Measurement.



(b) Adjustment.

Figure 6.6: Illustration of the write training procedure in the proposed PDC based timing center calibration method divided into measurement (a) and adjustment (b).

6.3.1 Clocking Architecture and Receiver Front-End

Figure 6.7 shows the schematic of the clocking architecture and the receiver front-end in the prototype ASIC. The shared clock generation consists of the DCC presented in chapter 5 and the direct voltage controlled CSI based DLL presented in section 4.3. The four phase clock generated by the DLL is used by both transceiver slices for phase interpolation. The schematic of the CML PI is given in Figure 6.8. Four NMOS differential pairs are connected to the load resistors R_L with the four different clock phases generated by the DLL $CK_{I,P}, CK_{I,N}, CK_{Q,P}$ and $CK_{Q,N}$ as differential input signals. The two MSBs of the digital control signal $d_{PI}[6:5]$ are used to select two adjacent clock phases for the interpolation. The five LSBs $d_{PI}[4:0]$ are passed to a current-steering DAC that controls the bias current ratio between the two branches of the PI and sets the mixing ratio of the adjacent clock phases for the interpolation. A low total bias current is used in the CML PI in order to keep the power consumption low, which in return results in a significant nonlinearity. Another instance of the DCC is connected to the output of the PI for the compensation of the duty-cycle variation across the PI range. The ASIC can be configured to act as a memory device by bypassing the DLL and the local PI and DCC. A typical half-rate receiver front-end based on the strongARM latches in Figure 4.22 (a) and a CML preamplifier is implemented. The PDC in each transceiver slice has three clock inputs, the reference clock for the INL compensation (input clock of the DLL), the local clock CK_{DQ} for the receiver front-end and the received signal RX after the preamplifier

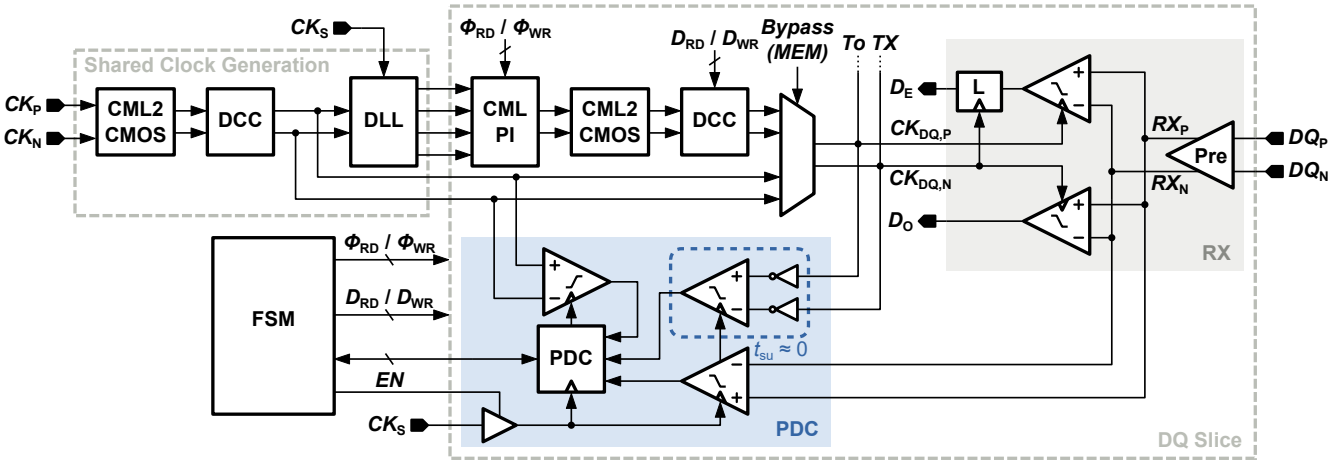


Figure 6.7: Schematic of the clocking architecture and the receiver front-end in the prototype ASIC for the proposed PDC based timing center calibration.

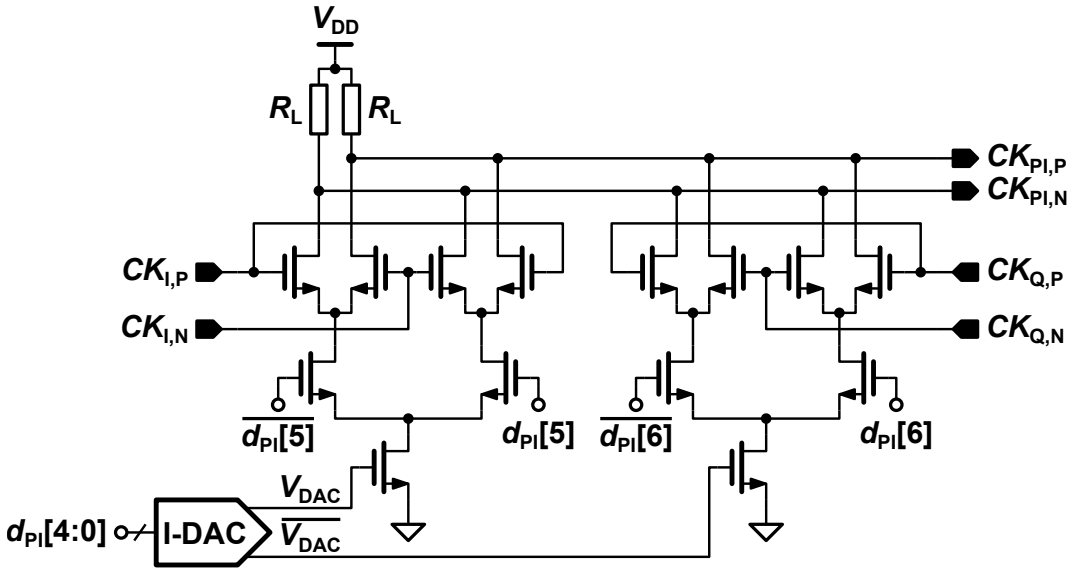


Figure 6.8: Schematic of the low-power CML PI in the prototype ASIC for the proposed PDC based timing center calibration.

of the receiver front-end. All three inputs use replicas of the strongARM latches of the receiver front-end as samplers. The resolution of the PDC is $N = 64$ and the maximum sample clock frequency is 2 GHz resulting in a scaling factor of $M \geq 2$ for a maximum input clock frequency of 4 GHz. After calibration the PDC is shut down by clock gating of CK_S. An FSM controls the PDC and adjusts the settings of the local PI and DCC. The transmitter operates on the same clock signal as the receiver (CK_{DQ}) and therefore the FSM stores separate PI and DCC settings for the read and the write operation (ϕ_{RD}, D_{RD} and ϕ_{WR}, D_{WR}).

An important aspect in the timing center calibration using the proposed PDC based phase measurement between a received clock pattern and the local receiver clock is that the setup time of the strongARM latches in the receiver needs to be considered in the calculation of the ideal timing center. The ideal phase shift between the local receiver clock and the received clock pattern is therefore

$$\phi_{RX,ideal} = 90° + \frac{t_{su}}{T_{CK}} \cdot 360° \tag{6.7}$$

with the setup time t_{su} of the strongARM latches and the period T_{CK} of the half-rate clock CK. The conventional timing center calibration methods elaborated in section 6.1 automatically take the setup

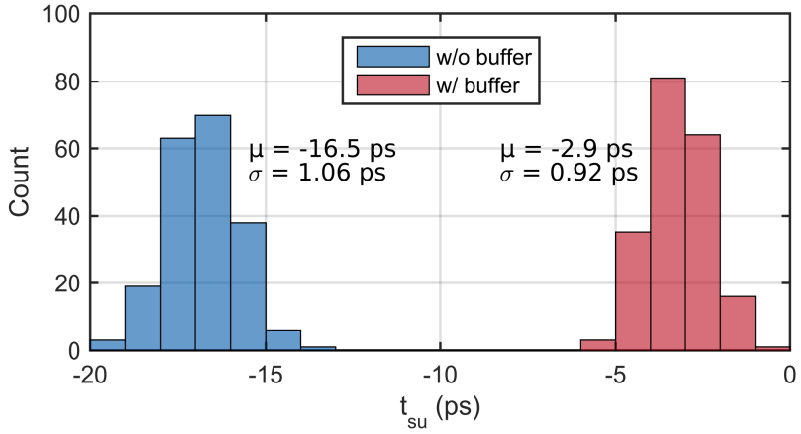


Figure 6.9: Histograms of the setup time t_{su} from Monte Carlo simulations of the strongARM latch in Figure 4.22 (a) with and without additional input buffers.

time into account by using the sampled data signals D_{E} and D_{O} for the phase detection of the received signal. The proposed method uses separate samplers in the PDC for CK_{DQ} and RX to measure the phase difference. With identical samplers for both clock signals the setup time of the samplers would cancel out in the measurement and the PDC would return the real phase difference. An adjustment of the receiver clock phase to $\phi_{\mathrm{RX,ideal}}$ according to equation (6.7) would therefore require knowledge of the exact value of t_{su}. Due to PVT variations and dynamic offset this value is not constant and must be measured in some way to minimize the timing center error. Instead of directly measuring t_{su} the proposed circuit in Figure 6.7 uses a replica of the receiver samplers for the sampling of RX and a modified sampler with additional buffers to compensate the setup time for CK_{DQ}. Assuming the delay of the buffers to match the setup time of the sampler ($t_{\mathrm{buf}} = t_{\mathrm{su}}$), the phase difference measured by the PDC

$$\phi_{\mathrm{RX,pdc}} = \phi_{\mathrm{RX}} - \frac{t_{\mathrm{su}}}{T_{\mathrm{CK}}} \cdot 360° + \frac{t_{\mathrm{su}} - t_{\mathrm{buf}}}{T_{\mathrm{CK}}} \cdot 360° \tag{6.8}$$

includes the setup time of one replica sampler in addition to the real phase difference ϕ_{RX}. Inserting the ideal phase difference $\phi_{\mathrm{RX,ideal}}$ from equation (6.7) as ϕ_{RX} into equation (6.8) shows that the ideal timing center is reached when the PDC measurement result equals 90°, independent of t_{su}. This calculation is based on the assumptions that the replica sampler has the exact same setup time as the receiver samplers and that the buffers for the CK_{DQ} sampler fully compensate its setup time. In the real circuit implementation multiple non-idealities cause deviations from these assumptions and lead to a timing center error. A careful circuit design and layout placement of the PDC samplers can reduce these deviations and minimize the timing center error. By placing the replica sampler for the RX signal close to the receiver samplers the impact of PVT variations on the setup time of the receiver samplers is tracked by the replica sampler. A perfect compensation of the setup time of the sampler for the CK_{DQ} signal is hard to achieve. On the one hand the buffers should provide a sufficient slew rate to the inputs of the sampler in order to minimize the influence of the input referred offset. This on the other hand limits the maximum delay of the buffers that can be used for the compensation. In the proposed circuit a small remaining average setup time is tolerated in favor of a small standard deviation. Figure 6.9 shows the histograms of t_{su} from Monte Carlo simulations of the strongARM latch with and without input buffers. The average setup time of −16.5 ps is reduced to −2.9 ps with the input buffers and is considered in the calculation of the calibrated timing center so that the remaining timing center error is defined by the standard deviation of the sampler setup time of 0.92 ps. Another source for a timing center error is a possible difference of the input referred offset between the receiver and PDC samplers due to uncorrelated mismatch. But such a difference can also exist between the two receiver samplers and therefore also affects the conventional timing center calibration methods. An offset calibration as discussed in section 2.5.2 can eliminate this error source and is frequently employed in commercial transceiver designs.

6.3.2 Deserializer with Self-Synchronizing Clock Divider

The data rate of high-speed transceivers in typical applications like memory interfaces or periphery interconnects is much higher than the operating frequency of the processors or memory cores that generate or process the data. Serializers and deserializers are used to convert the data between a low frequency parallel data stream for the internal processing and a high-frequency serial data stream for the data transfer. A synchronization of the corresponding serializers and deserializers of the I/O links is required prior to the data transmission in order to receive the bits of the data words in the same order as they are transmitted. The conventional method for the synchronization is to transmit a known data pattern and shift the sequence of the clock division in the deserializer until the data pattern is received correctly. Figure 6.10 shows the schematic of the proposed deserializer with a self-synchronizing clock divider. When the synchronization is enabled by the EN signal and the special data word "00110000" is transmitted the received even and odd data signals D_E and D_O generate a reset pulse using a NAND gate. Latches re-time the reset pulse for the two clock dividers consisting of a flip-flop and a NOR gate. A simulation of the self-synchronization process is shown in Figure 6.11. In the simulation the special data word "00110000" is transmitted three times in a row and the synchronization is enabled with the second data word. The first word is received in an incorrect order due to the arbitrary deserialization sequence at the startup of the circuit. With the second data word the two clock dividers are both forced to skip half a clock cycle by the reset pulse generated from D_E and D_O. After the synchronization of the clock dividers the deserialization sequence is aligned to the serial data stream and the third data word is received correctly.

The main advantage of the proposed deserializer is that the synchronization process does not require any data evaluation on the receiver side. Only the EN signal needs to be enabled for one cycle of the divided clock $CK/4$ during the transmission of the special data word and the deserializer automatically aligns its clock dividers with the serial data stream. The synchronization itself takes no longer than one cycle of $CK/4$ depending on the initial sequence of the clock dividers. The critical path for the maximum operating frequency of the circuit is from the half-rate receiver (RX) through the NAND gate to the first latch of the self-synchronizing divider. The delay limit

$$t_{\mathrm{RX,CK\to Q}} + t_{\mathrm{NAND,d}} \leq T_{\mathrm{CK}} - t_{\mathrm{L,su}} \tag{6.9}$$

is based on the clock to output delay of the samplers in the half-rate receiver $t_{\mathrm{RX,CK\to Q}}$, the delay through the NAND gate $t_{\mathrm{NAND,d}}$ and the setup time of the latch $t_{\mathrm{L,su}}$. The prototype design can be operated at frequencies of up to 4 GHz. Since the delay of the critical path is dominated by $t_{\mathrm{RX,CK\to Q}}$ because of

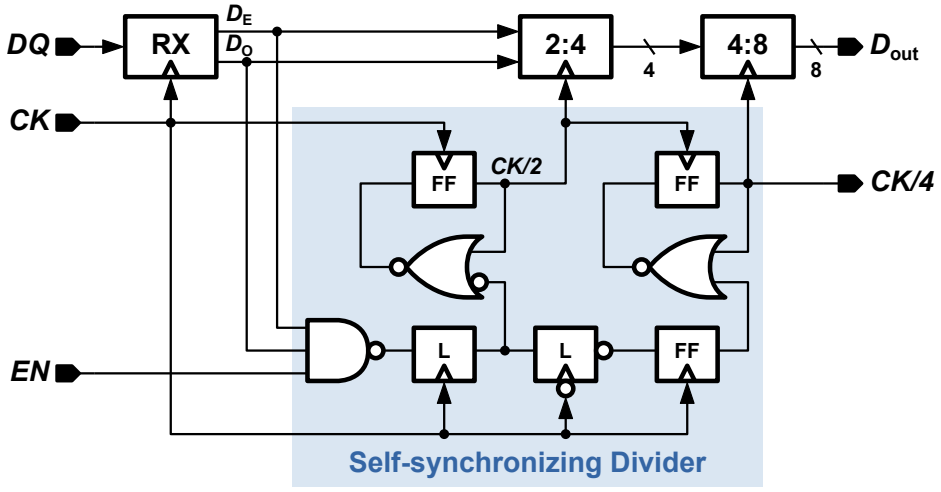


Figure 6.10: Schematic of the proposed deserialzer circuit with self-synchronizing clock divider.

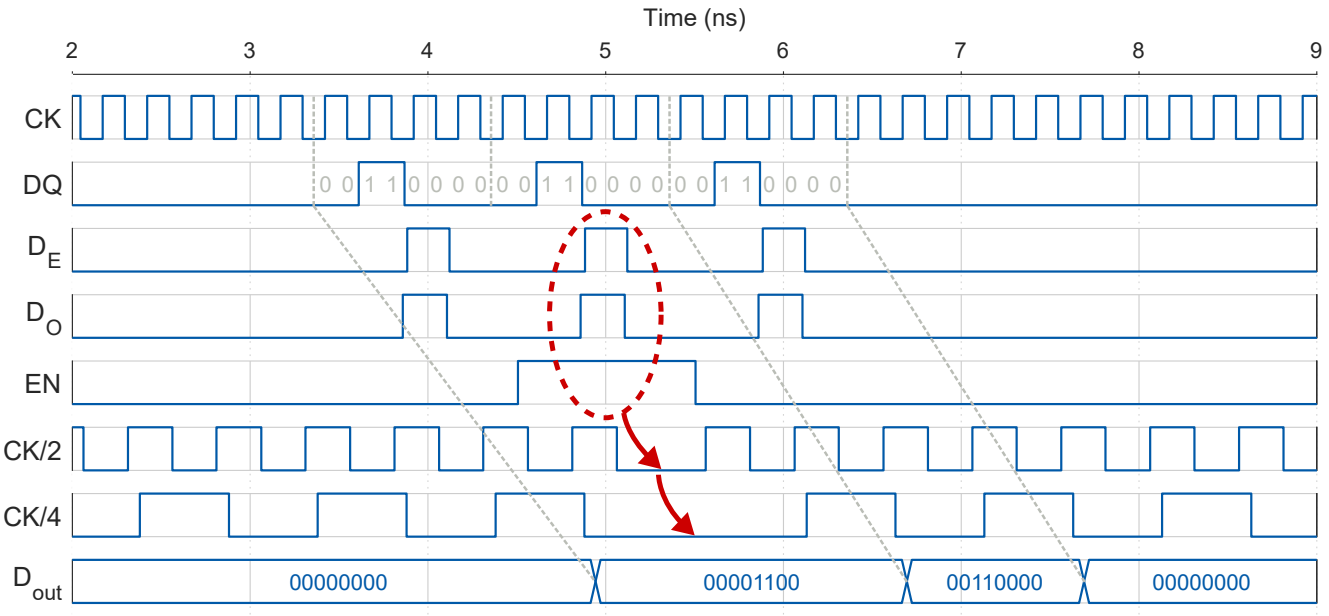


Figure 6.11: Simulation of the synchronization process of the proposed deserializer with self-synchronizing clock divider.

the SR-latches (see Figure 4.22 (b)) that evaluate the outputs of the strongARM latches in the half-rate receiver a re-timing of D_E and D_O using CMOS latches can help to increase the maximum operating frequency.

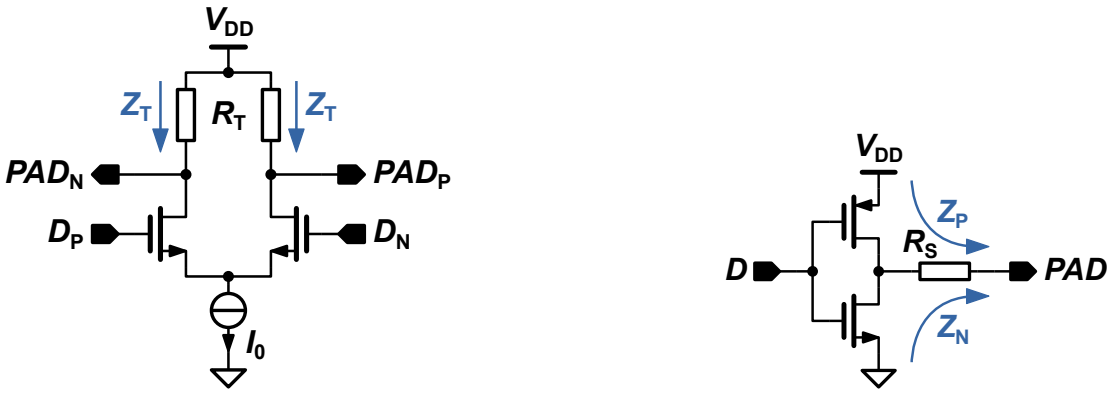
6.3.3 Transmitter

The most common transmitter types for source-synchronous high-speed I/O links are current-mode logic (CML) and source-series terminated (SST) off-chip drivers (OCD) illustrated in Figure 6.12. CML drivers are preferably used for differential I/O links and consist of a current source, a differential pair and two load resistors R_T that serve as termination. One advantage of CML drivers is that the output voltage swing can be adjusted by the source current I_0 independent of the termination impedance Z_T. A disadvantage is that CML drivers draw a constant current of I_0 even if the signal is static. SST drivers consist of a CMOS inverter and a series resistor R_S. Ideally the transistors have an on-resistance of $R_{on} = 0\,\Omega$ to represent an ideal voltage source, so that the output impedance is solely defined by R_S. Integrated MOSFETs with a sufficiently small R_{on} would require a large layout area and create a significant parasitic capacitance. Therefore the output impedance is split into a part covered by R_S and a part covered by R_{on} of the PMOS transistor when a one is transmitted (Z_P) or R_{on} of the NMOS transistor when a zero is transmitted (Z_N). Since R_{on} of a MOSFET depends on the drain-source voltage V_{DS} the resulting output impedance is nonlinear. A typical impedance ratio for SST drivers is

$$\frac{R_S}{R_{on}} \approx 3 \tag{6.10}$$

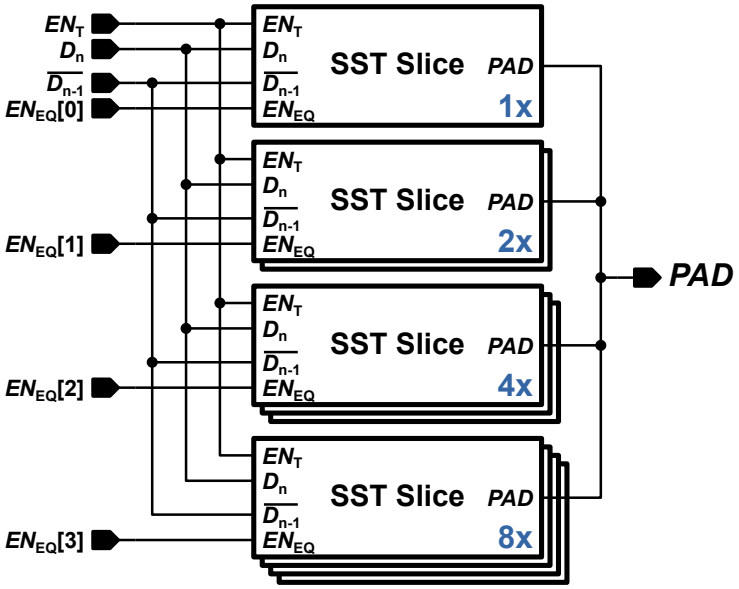
with the intention to keep the contribution of the nonlinear R_{on} as small as possible. The SST driver draws no static current from the supply, but during switching a high dynamic current occurs when both transistors are conducting. This generates a significant amount of supply noise that needs to be decoupled from sensitive analog circuits. The basic SST driver circuit in Figure 6.12 (b) is single-ended. For differential I/O links two instances of the single-ended SST driver are used as a pseudo-differential OCD.

All high-frequency transmitters in the prototype ASIC are implemented as SST drivers because the circuits are re-used for the single-ended I/O links of the design presented in chapter 7. The proposed

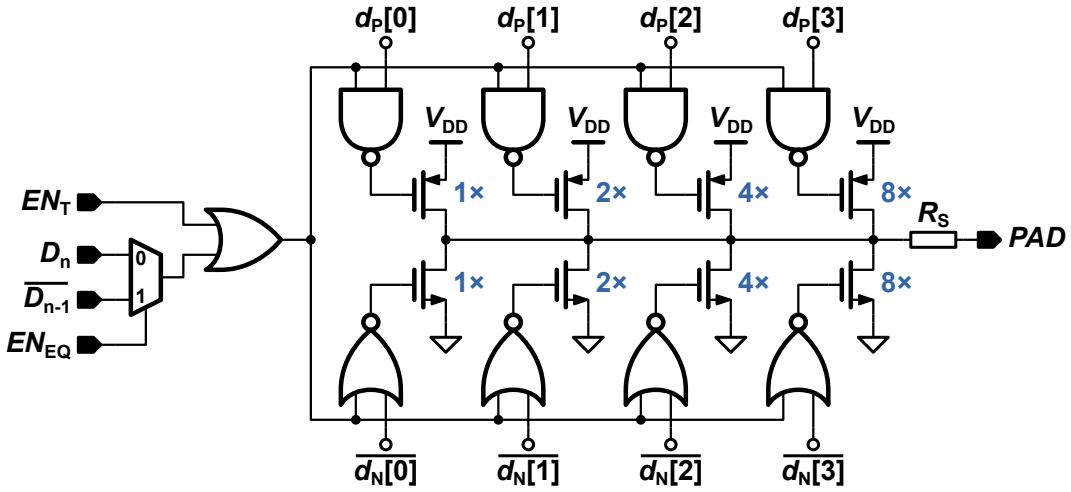


(a) CML off-chip driver. **(b)** SST off-chip driver.

Figure 6.12: Schematics of a CML off-chip driver (a) and an SST off-chip driver (b).



(a) SST driver.



(b) SST driver slice.

Figure 6.13: Schematics of the proposed SST driver (a) composed of 15 SST driver slices (b).

SST driver circuit offers an independent control of the driver impedance (Z_P and Z_N) and the 2-tap FFE. A description of the working principle of an FFE is given in section 2.5.3. The driver is composed of 15 equal slices as illustrated in Figure 6.13 (a), divided into four binary scaled groups for the weighting of the precursor compensation in the 2-tap FFE. Each slice has an impedance of $Z_{slice} = 15\,Z_T$ and the termination impedance $Z_T = 50\,\Omega$ is chosen for compatibility with the measurement equipment. The schematic of an SST slice in Figure 6.13 (b) consists of four binary scaled CMOS drivers for the impedance calibration and a single series resistor R_S with a resistance ratio of $R_S/R_{on} = 2$. The signal EN_T puts all SST slices into termination mode in which a static one is driven to the pad creating a 50 Ω termination to V_{DD}. The impedance calibration is implemented using a replica SST driver and an external precision resistor similar to the method described in section 2.5.1. First Z_P is calibrated against the external resistor connected to GND, then Z_N is calibrated against the previously calibrated Z_P of the SST driver connected to V_{DD}.

6.4 Measurement Results

A die photograph of the prototype ASIC implemented in the UMC 65 LL CMOS process is shown in Figure 6.14. The two differential transceivers with PDCs and a shared DLL for multiphase clock generation occupy an area of 0.053 mm^2. The area of a single PDC is 0.004 mm^2. The clock transmitter (CK TX) has a separate PDC for duty-cycle correction of the forwarded half-rate clock CK in the source-synchronous interface. Figure 6.15 shows the test PCB with two transceiver ASICs in QFN48 packages configured as a controller and a memory device. The trace lengths of the two differential I/O links DQ 0 and DQ 1 are 86 mm and 60 mm, respectively. The PLLs (ADF4351 by Analog Devices) for the generation of CK and CK_S are also mounted on the PCB and are locked to the same 25 MHz reference oscillator. The transceivers are operated at $f_{CK} = 3.2$ GHz and $f_S = 1.588$ GHz (the PDC parameters are $N = 64$ and $M = 2$). The measured RMS jitter is 790 fs for CK and 990 fs for CK_S. A microcontroller is used to multiplex the SPI interfaces of the two transceiver ASICs and connect them to a computer running MATLAB for control and data evaluation. Eye diagram, phase and jitter measurements are performed by probing the differential I/O links near the pins of the QFN48 packages using an oscilloscope (DSAV164A by Keysight).

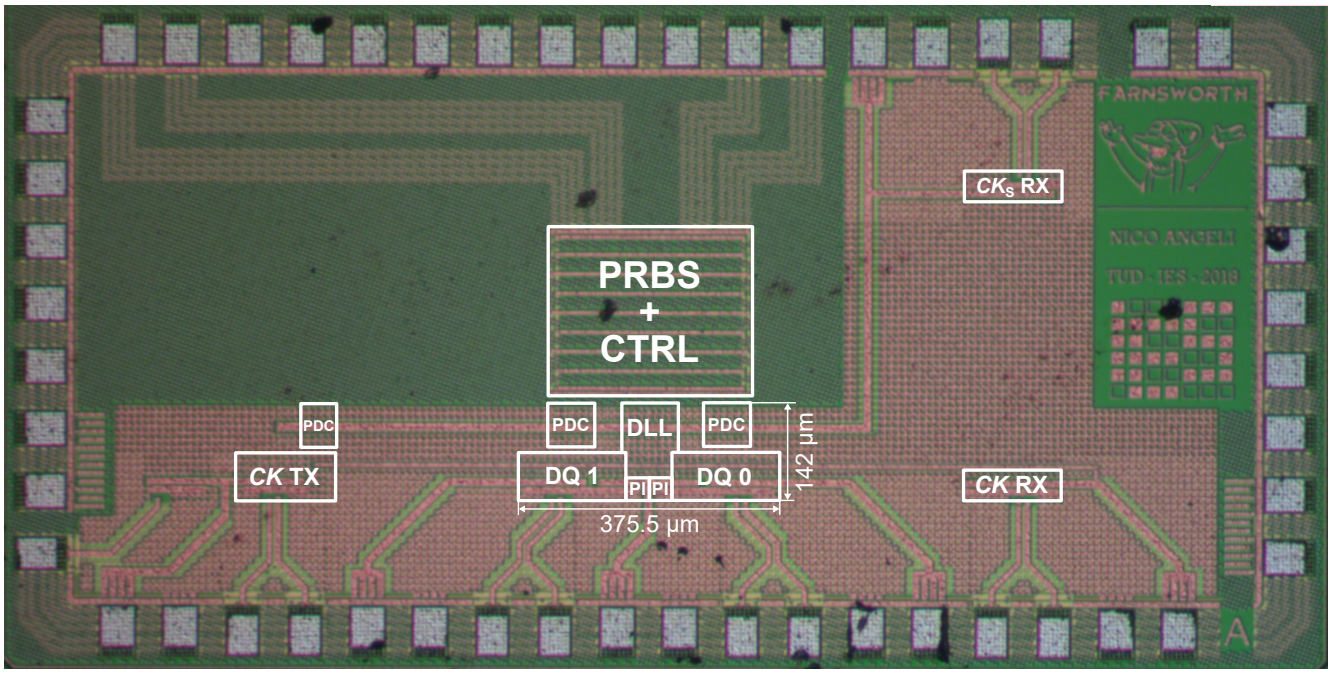


Figure 6.14: Die photograph of the prototype ASIC for the PDC based timing center calibration in the UMC 65 LL CMOS process.

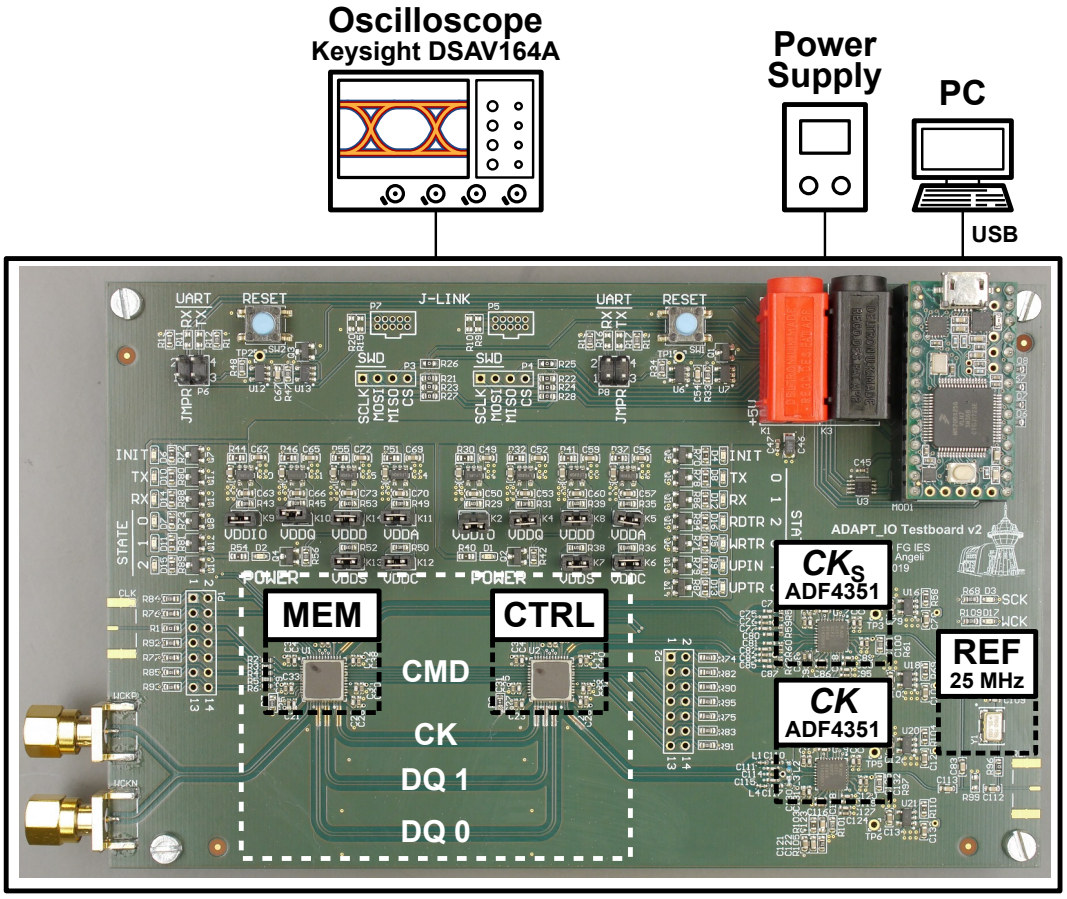
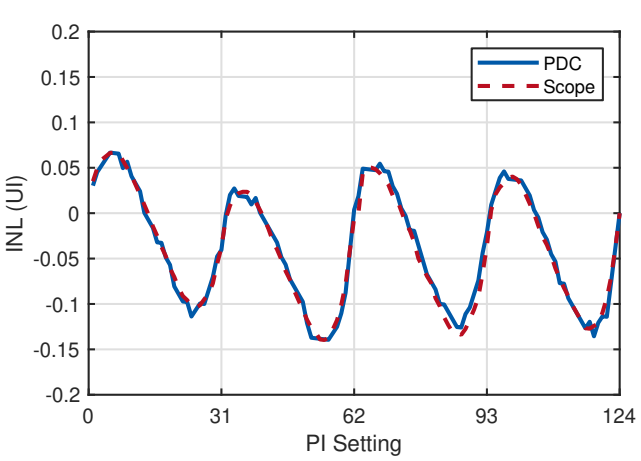


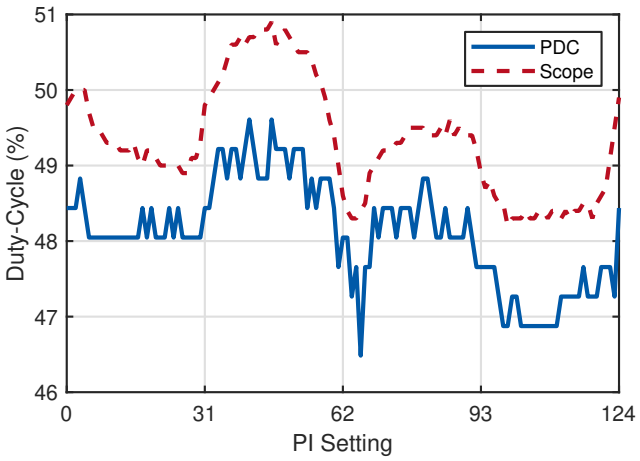
Figure 6.15: PCB and measurement setup for the prototype ASIC of the PDC based timing center calibration in Figure 6.14.

In order to confirm the capability of the PDC based timing center calibration to compensate the nonlinearity of the PI a measurement of the PI output phase with respect to the reference clock at the input of the DLL using the integrated PDC is performed and compared to an external oscilloscope measurement. Figure 6.16 (a) shows the INL of the PI output phase across the PI range. The PDC measurement is in good accordance with the oscilloscope measurement and shows that the low-power PI can create a worst case INL error in the timing center calibration of 0.19 UI. The excellent matching of the PDC measurement and the oscilloscope measurement proves that the proposed INL compensation method is able to adjust the PI to the desired phase shift with high precision. In Figure 6.16 (b) a duty-cycle measurement across the PI range is shown. The oscilloscope measurement has a constant offset of $\approx 1\,\%$ because the PDC measurement is performed on the internal clock CK_{DQ} before the transmitter while the oscilloscope measurement is performed on the I/O signal DQ which includes the duty-cycle distortion contributed by the SST driver. However, the PDC measurement correctly captures the duty-cycle variation across the PI range and therefore a compensation is possible. The worst case duty-cycle difference between two PI settings is 2.5 % limiting the number of correction steps required in the calibration procedure (see section 6.2.2).

Figure 6.17 shows the eye diagrams of the prototype I/O link (DQ 0) for the read and the write operation using a PRBS31 pattern. The 2-tap FFEs in both directions have been adjusted to compensate ISI as good as possible. The eye diagram for the read direction shows an increased jitter compared to the write direction. This is a result of the asymmetric bidirectional interface architecture with a unidirectional clock. The transceiver clock in the memory device is first transmitted over the channel which adds significant jitter to the clock signal by the transmitter and receiver circuits. This is the reason why high-performance memory chips like GDDR6 include a PLL for jitter cleaning of the forwarded clock.



(a) INL.



(b) Duty-cycle.

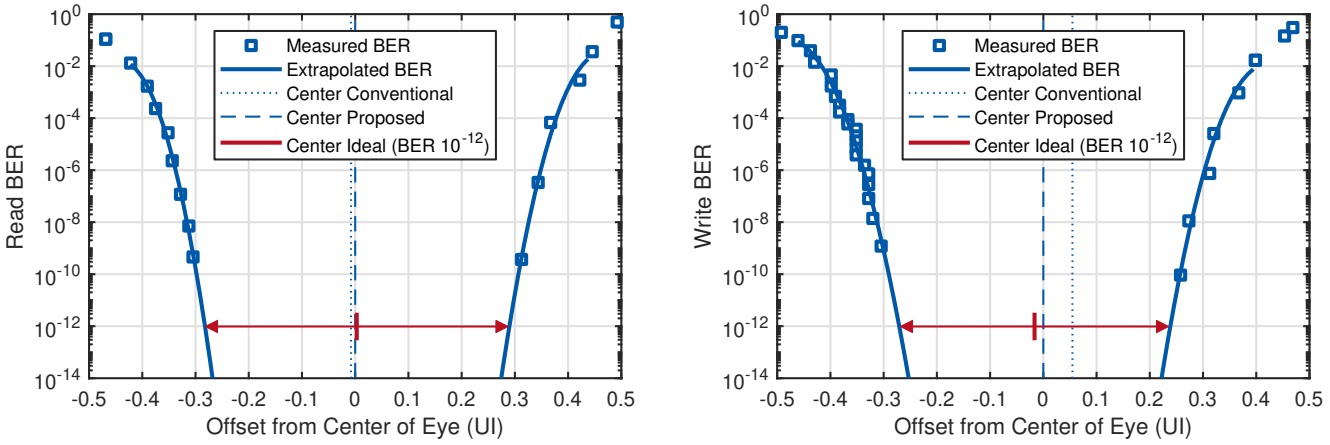
Figure 6.16: Measured INL (a) and duty-cycle (b) of the PI using the integrated PDC of the prototype ASIC for the PDC based timing center calibration and an external oscilloscope.

(a) Read operation.

(b) Write operation.

Figure 6.17: Eye diagrams of the read (a) and the write (b) operation of DQ 0 in the prototype I/O link at 6.4 Gb/s with active feed-forward equalization, each measured at the receiving side near the pins of the QFN48 package.

The timing center calibration is evaluated by performing BER measurements with a PRBS31 pattern at every PI position to create a bathtub plot. A total number of 10^{10} bits is transmitted at every PI position resulting in a minimum BER of 10^{-10} to be measured. Additionally, the phase of CK_{DQ} is measured using the PDC at each PI position for a correct placement of the BER measurement values on the time axis of the bathtub plot. This increases the precision of the bathtub plot compared to other publications which use the PI setting to estimate the clock phase ignoring the nonlinearity of the PI. A model is fitted to the measured BER data using the MATLAB function in Appendix C that extrapolates the data to a BER of 10^{-14} in order to keep the measurement time short. The calibrated timing center of the proposed PDC based calibration method is compared to the ideal timing center at $BER = 10^{-12}$ using the extrapolated model fit. Additionally, the timing center calibrated by the conventional passing window method is calculated based on the PI settings at which the fitted bathtub curve crosses $BER = 10^{-4}$ and the captured nonlinear PI characteristic for the determination of the phase at the calculated center PI setting. The measured and fitted bathtub curves for the read and the write operation are shown in Figure 6.18 annotated with the calibrated timing centers using the proposed method and the conventional passing window approach, as well as the ideal timing center. In both cases the calibrated timing center of the proposed PDC based method is closer to the ideal timing center than the conventional passing



(a) Read operation.

(b) Write operation.

Figure 6.18: Bathtub plots with extrapolated BER data using the MATLAB function in Appendix C for the read (a) and the write (b) operation of the prototype interface for the PDC based timing center calibration. The calibrated timing center is compared with the ideal timing center at $BER = 10^{-12}$ and the timing center calibrated by the conventional passing window method.

Table 6.1: Timing center error and link training duration of the proposed PDC based timing center calibration with and without INL compensation and the conventional passing window method at $BER = 10^{-4}$ for DQ 0 of the prototype interface.

Method	Read Training Duration	Read Timing Center Error	Write Training Duration	Write Timing Center Error
PDC Based + INL Comp.	3.1 µs	0.003 UI	3.3 µs	−0.017 UI
PDC Based	0.77 µs	−0.125 UI	0.97 µs	−0.047 UI
Passing Window	97.9 µs	−0.008 UI	193.8 µs	0.055 UI

window approach. The remaining timing center errors of 0.003 UI and −0.017 UI for the read and the write training, respectively are within the simulated maximum deviation of the sampler setup time from its average (see Figure 6.9) of less than 0.02 UI. An offset calibration of the PDC samplers could further reduce the timing center error by eliminating the setup time variation due to the input referred offset. Table 6.1 summarizes the training durations and the timing center errors of the proposed PDC based timing center calibration and the conventional passing window approach. The results without the INL compensation are also included to show the significant impact on the timing center error. They also show that the PDC based method and the passing window approach are affected differently by the INL of the PI. The passing window method calculates the center of two PI positions while the proposed PDC based approach calculates the 90° shift from a single PI position similar to the fuzz mean method. The training duration of the PDC based method is significantly shorter than with the conventional passing window method, even with the INL compensation. Also the durations of the read and the write training are nearly the same in the proposed design since there is no need to send the complete training data back to the controller during the write training.

Another advantage of the proposed PDC based timing center calibration is the reduced power consumption during the link training since no access to the data FIFOs is required and the transmission of the clock pattern can be switched off after the initial phase adjustment. The total power consumption of

Table 6.2: Power comparison of the proposed bidirectional asymmetric source-synchronous I/O link and similar state-of-the-art designs during the link training.

Method	Process	Signaling	Data rate (Gb/s/pin)	Read Efficiency (mW/Gb/s)	Write Efficiency (mW/Gb/s)
This Work	65 nm	Differential	6.4	1.38	3.05
[99]	65 nm	Single-Ended	4.266	1.86	5.68
[32]	40 nm	Differential	16	5.3	5.3
[35]	40 nm	Single-Ended	6.4	12.7	12.7
[34]	65 nm	Differential	16	13	13

the controller device in the prototype I/O link is 8.8 mW for the read training and 19.5 mW for the write training with $f_{CK} = 3.2$ GHz of which 2.27 mW are consumed by the PDC ($N = 64$ and $M = 2$). After the link training the PDC is switched off by clock gating so the power consumption during normal operation is even lower. Table 6.2 compares the power efficiency of the prototype transceiver ASIC with similar state-of-the-art controller implementations for source-synchronous I/O links. Since the state-of-the-art designs do not require dedicated circuitry for the link training their power consumption during the link training is the same as during normal operation. Even though the PDC consumes additional power during the link training the overall efficiency of the proposed design is still better than in the conventional implementations.

6.5 Conclusion

A timing center calibration method for source-synchronous I/O links is proposed that significantly advances the state-of-the-art in terms of link training duration and timing center error. Instead of performing repeated BER measurements at different clock phase positions across one UI to detect the timing center, the proposed method makes use of the PDC presented in chapter 3 to measure the phase difference between a received clock pattern and the internal clock. Subsequently the phase difference is adjusted to 90° for an ideal sampling in the center of the eye. Careful design of the PDC samplers ensures that the setup time of the data samplers is considered in the phase measurement in order to minimize the timing center error. An INL compensation that does not require an active signal transmission on the I/O link eliminates the timing center error due to the nonlinearity of the phase interpolation which is not possible with the conventional timing center calibration methods. Additionally, the duty-cycle variation across the PI range can also be compensated by the PDC based calibration method. Table 6.3 compares the link training duration and the residual timing center error of the proposed PDC based timing center calibration with previously published link training methods. The number of bits

$$n_{tr} = 2 f_{CK} t_{tr} \tag{6.11}$$

transmitted during the full link training (read and write) based on the training duration t_{tr} and the half-rate clock frequency f_{CK} is used to compare the training methods independent of the data rate. The values of n_{tr} in the table are normalized to the proposed PDC based approach. The adaptive 3-step method in [99] with adaptive gain which also performs a voltage threshold calibration but does not implement averaging takes 2.74 times longer than the PDC based approach. The improved fuzz mean method [40] which already significantly advances the conventional passing window method is 39.1 times slower than the proposed timing center calibration. The residual timing center error of the previously published methods is mainly defined by the INL of the PI and is approximately five times larger than with the PDC

Table 6.3: Performance comparison of the proposed PDC based timing center calibration and the conventional methods.

Method	Link Speed (Gb/s/pin)	Training Duration (µs)	$\frac{n_{tr}}{n_{tr,PDC}}$	Timing Center Error (UI)
PDC Based	6.4	3.1/3.3 (Read/Write)	1	< 0.02
Adaptive 3-Step [99]	4.266	11/15.3 (Read/Write)	2.74	N/A
Impr. Fuzz Mean [40]	3.2	500	39.1	0.11
Passing Window [40]	3.2	40000	3125	0.09

based approach. The implementation of an offset calibration for the PDC samplers can even further improve the timing center error without increasing the link training duration. The design presented in chapter 7 implements such an offset calibration for the PDC samplers, as well as for the receiver samplers and integrates a calibration algorithm into the PDC.

7 Calibration of the Receiver Voltage Threshold by Duty-Cycle Measurement

Differential I/O links have the advantage that the voltage threshold for the sampling of the received data is always $V_{\mathrm{th}} = 0\,\mathrm{V}$. In single-ended I/O links the ideal voltage threshold can vary due to impedance mismatch and channel attenuation. Usually, a voltage threshold calibration is performed in addition to the timing center calibration in order to sample the data at the ideal eye center. The conventional approach to the voltage threshold calibration is similar to the passing window method for the timing center calibration. The voltage threshold is varied over the input voltage range and the BER is measured at every step to determine a passing window and find the vertical center of the data eye. The proposed PDC based voltage threshold calibration reduces the calibration time by performing duty-cycle measurements instead of BER measurements. The idea is that the ideal threshold voltage is reached when the duty-cycle of a received clock pattern equals 50 %.

7.1 Related Work

The conventional method for the voltage threshold calibration in combination with a timing center calibration of a single-ended high-speed I/O link is to perform a full two dimensional eye scan as illustrated in Figure 7.1. A grid of sample points across the data eye is defined and the BER is measured at each sample point. The resulting two dimensional BER plot reflects the eye diagram and is used to find the ideal sampling point for the receiver in the center of the data eye. The drawback of this method is that a large number of BER measurements must be performed which results in a long link training duration. The two dimensional eye scan is also useful to analyze the eye diagram directly at the receiver, which is normally inaccessible for measurement equipment. In differential I/O links the sampler offset voltage can be used as the second dimension instead of the voltage threshold in order to analyze the eye diagram.

In [99] an adaptive 3-step calibration scheme is proposed based on the passing window method. In the first step the timing center at a nominal voltage threshold is determined using the conventional passing window method described in section 2.5.4. In the second step the ideal voltage threshold is determined at the previously detected timing center using the same concept (see section 2.5.5). Finally, another timing center calibration is performed at the detected voltage threshold to optimize the sampling position. Using the 3-step calibration scheme the training duration is significantly reduced, because the BER measurements are only performed on a fraction of the full two dimensional sample point grid. The

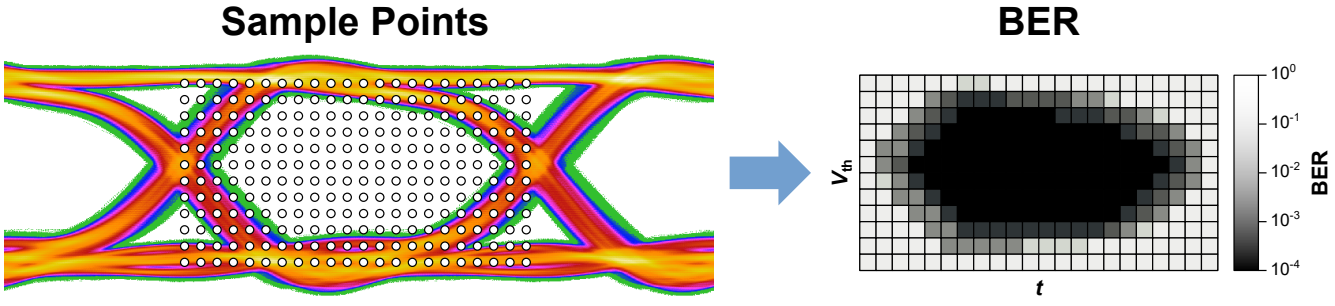


Figure 7.1: Illustration of the full two dimensional eye scan method for the combined timing center and voltage threshold calibration of a single-ended high-speed I/O link based on BER measurements on a grid of sample points across the data eye.

algorithm in [99] also uses an adaptive step size to further decrease the calibration time of the two dimensional eye detection to

$$t_{3\text{–step}} \approx \frac{1}{K}\left(2\,n_{\mathrm{PI}} + n_{\mathrm{th}}\right)\frac{1}{BER_{\mathrm{min}}}\frac{1}{f_{\mathrm{bit}}}, \tag{7.1}$$

with the number of PI steps n_{PI}, the number of voltage threshold steps n_{th}, the maximum gain K, the minimum BER to be measured BER_{min} and the data rate f_{bit}. One limitation of the adaptive 3-step method is that the nominal V_{th} for the initial timing center calibration must be located within the data eye with a sufficient voltage margin.

7.2 Concept of PDC based Voltage Threshold Calibration

As shown in section 6.1 a clock signal can be used for the timing center calibration instead of a PRBS data pattern with a negligible timing center error. The same clock signal shall be used in the proposed PDC based voltage threshold calibration. In high-speed I/O links with matched impedances a clock signal with the maximum possible frequency has a nearly sinusoidal form. When the voltage threshold of the receiver is exactly at the vertical eye center then the duty-cycle of the received signal is exactly 50 %. Due to the limited slew rate of the signal a deviation of V_{th} from the vertical eye center leads to an increase or a decrease of the duty-cycle as illustrated in Figure 7.2. The idea of the proposed voltage threshold calibration method is to measure the duty-cycle of the received clock pattern using the PDC and to adjust V_{th} until the duty-cycle is equal to 50 %. The same PDC configuration as in the PDC based timing center calibration in chapter 6 can be used for the duty-cycle measurement without modification, therefore no additional circuitry is required. The main advantage over the conventional passing window method is that the PDC based voltage threshold calibration does not need to perform measurements at every V_{th} level across the eye diagram. Instead the measurements start with a nominal V_{th} and iteratively increase or decrease V_{th} depending on the initial measurement until the measured duty-cycle is as close to 50 % as possible. Therefore the total number of measurements is reduced by at least a factor of two depending on the duty-cycle error. Similar to the PDC based timing center calibration the proposed approach does not require to send back the complete test pattern to the controller device in the write training procedure. Instead only the duty-cycle information (8 bit) is sent back and therefore the write training duration is again nearly the same as the read training duration.

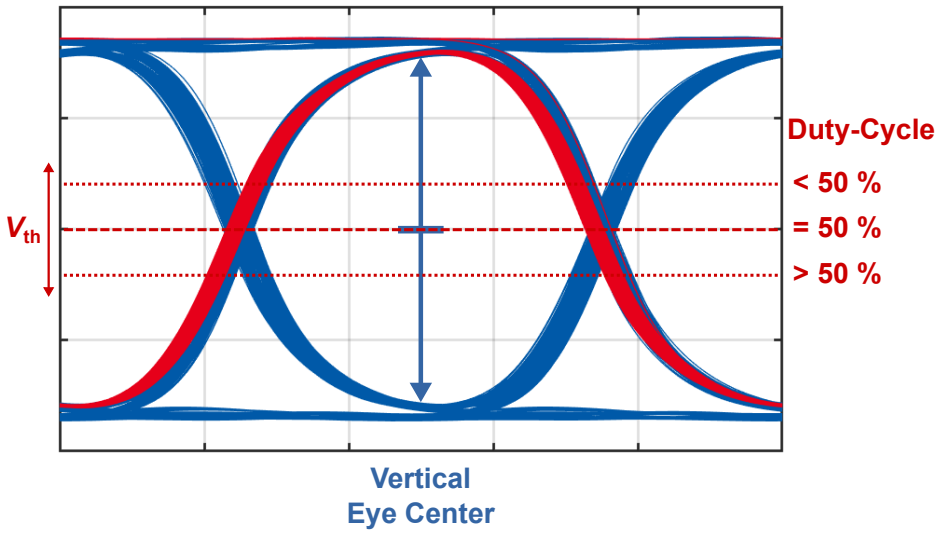


Figure 7.2: Illustration of the proposed PDC based voltage threshold calibration based on the duty-cycle measurement of a received clock pattern.

Another advantage of the proposed PDC based voltage threshold calibration in combination with the PDC based timing center calibration presented in chapter 6 is that only two calibration steps are required for a full two dimensional eye center detection. The voltage threshold calibration based on the duty-cycle measurement does not require a previous timing center calibration. Therefore the proposed two dimensional eye center detection first performs the voltage threshold calibration and then the timing center calibration as described in sections 6.2.2 and 6.2.3 for the read and the write training, respectively. The total read or write training duration

$$t_{\mathrm{tr,2D}} = n_{\mathrm{v}}\, t_{\mathrm{m}} + t_{\mathrm{tr}} \tag{7.2}$$

is composed of the number of iterations for the voltage threshold calibration $1 \leq n_{\mathrm{v}} \leq n_{\mathrm{th}}/2$, the duration of a single PDC measurement t_{m} according to equation (5.1) and the duration of the timing center calibration t_{tr} according to equation (6.6).

7.3 Single-Ended Transceiver Architecture

Figure 7.3 shows a block diagram of the proposed single-ended asymmetric bidirectional source-synchronous I/O link. A shared DLL generates a four phase clock signal for the local PIs in each DQ Slice. Both, the controller and the memory device have a shared V_{th} generator that generates the threshold voltage for the single-ended receiver front-end. Each DQ slice implements two separate PIs for the read and the write clock in order to simplify the synchronization of the deserializers since the clock phases of the read and the write clock do not change between read and write operations. The memory device does not implement any clock adjustment circuits in the DQ slice. The PDC is integrated similar to the design presented in chapter 6 and a detailed description of the PDC and receiver front-end architecture is given in section 7.4.1. The sample clock CK_{S} is generated from the half rate clock CK using an integer clock divider and a PLL in both devices to avoid the need for additional pins to forward CK_{S} from the controller to the memory device. A 2 bit command interface configures the memory device

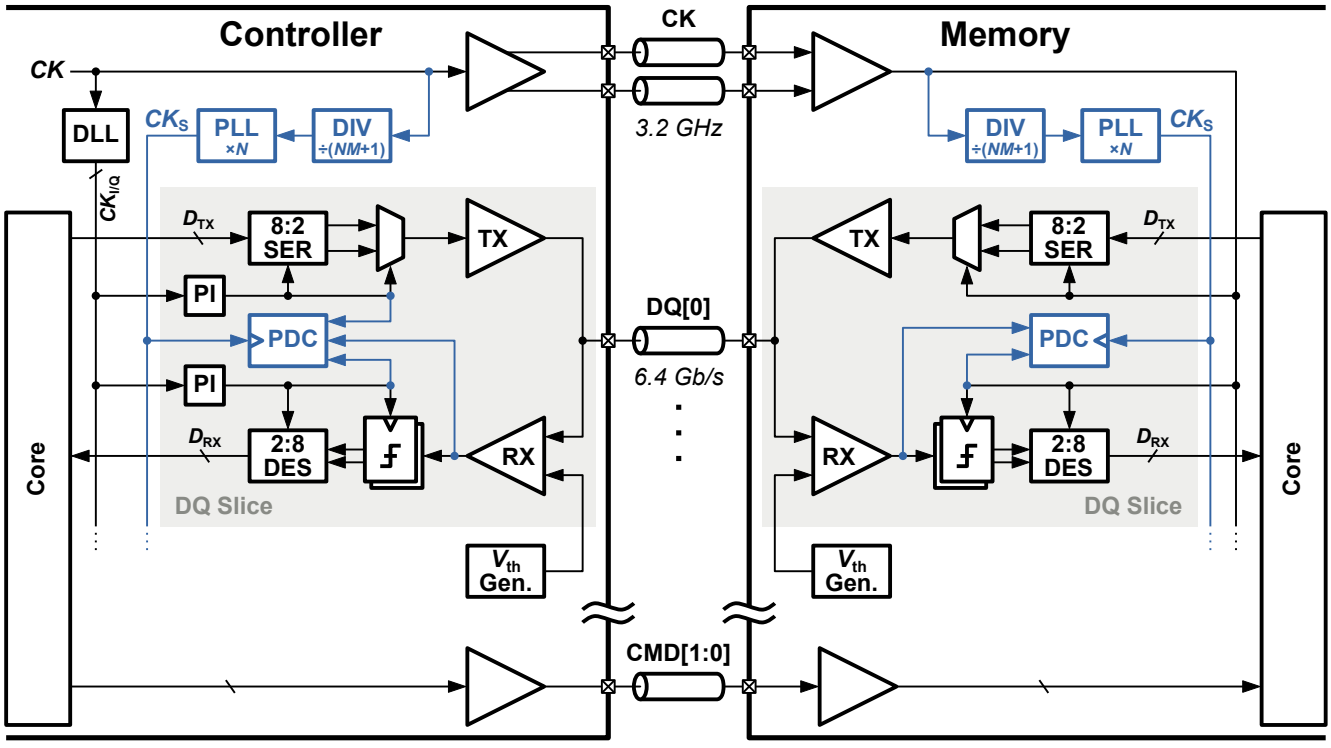


Figure 7.3: Block diagram of the proposed single-ended asymmetric bidirectional source-synchronous I/O link with PDC based voltage threshold and timing center calibration.

for read and write operation and controls the read and write training when the memory device is set to the calibration mode.

7.4 Implementation

The prototype ASIC for the PDC based voltage threshold calibration is implemented in the UMC 65 LL CMOS process and offers one single-ended transceiver slice. Separate implementations of the controller and the memory device are placed on the same ASIC rather than integrating a bypass option for the clock adjustment circuits as in the design presented in chapter 6 to configure the ASIC as a controller or a memory device. The SST drivers described in section 6.3.3 are re-used in this design for the high-frequency OCDs with the same impedance calibration circuit. The PIs are also re-used (see Figure 6.8), but the local DCC after the PI is omitted. The linearized CSI based DLL presented in section 4.3.2 is used in the controller device as the shared four phase generator. Global DCCs are placed at the input of the DLL and the clock transmitter of the controller device and at the output of the clock receiver in the memory device. In addition to the PRBS generators and checkers the core logic also offers FIFOs for the transmit data (D_{TX}) and the receive data (D_{RX}) that enable the implementation of the conventional link training methods on the prototype interface. Section 7.4.1 describes the relevant changes in the receiver front-end over the differential transceiver architecture presented in chapter 6. The prototype ASIC also includes a multi-modulus clock divider described in section 7.4.2 that generates a reference clock for the PDC sample clock PLL. Two external ADF4351 PLL ICs are used for the generation of CK and CK_S in the prototype interface.

7.4.1 Receiver Front-End

Figure 7.4 shows a schematic of the receiver front-end in the prototype ASIC with the local PDC for the voltage threshold and timing center calibration. The half rate receiver (RX) consists of two samplers based on the strongARM latch in Figure 2.18 (a) [30]. The offset calibration circuit in [32] based on current injections at the differential output of the strongARM latch is added to cancel the input referred offset voltage of the samplers. The basic offset cancellation scheme in section 2.5.2 is used to calibrate the receiver samplers. The PDC sampler for the received signal RX after the preamplifier is a replica of the

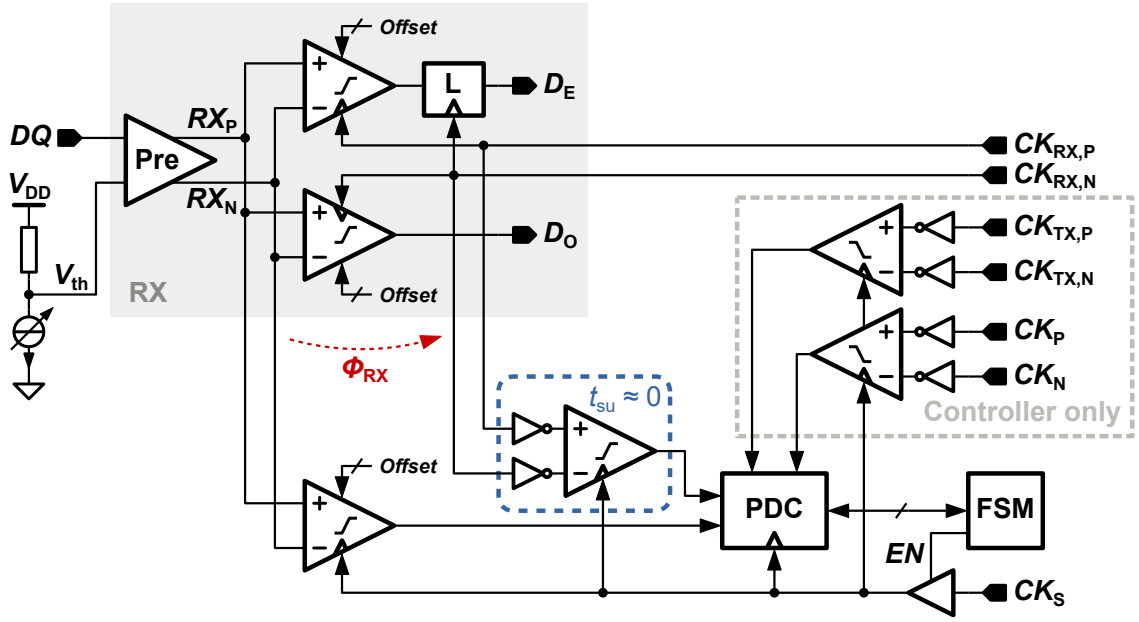


Figure 7.4: Schematic of the receiver front-end in the prototype ASIC for the proposed PDC based voltage threshold calibration.

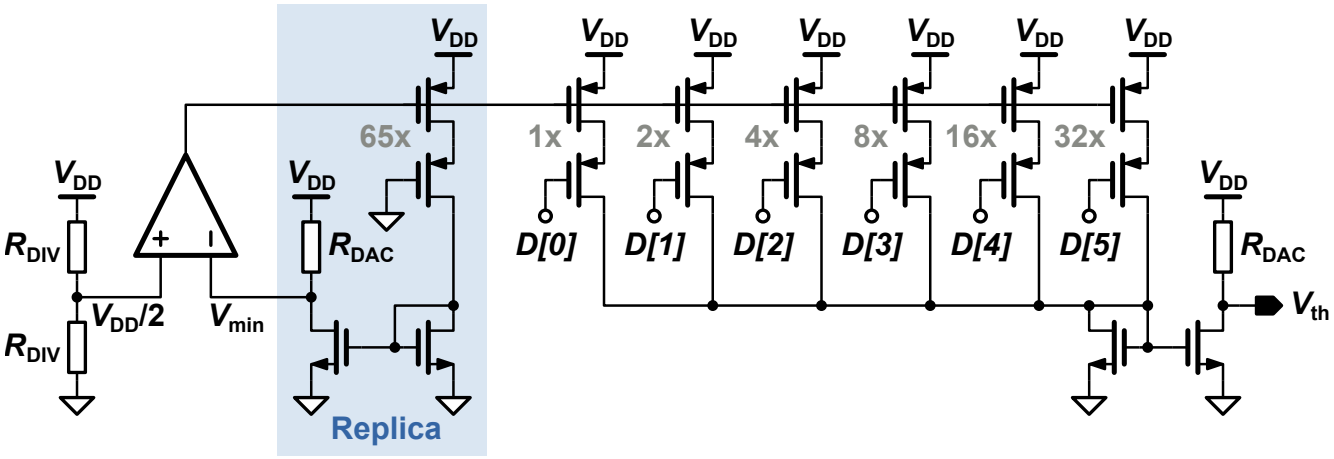


Figure 7.5: Schematic of the V_{th} generator based on a 6 bit current-DAC with replica biasing.

receiver samplers with the same current injection based offset calibration circuit. An additional counter in the PDC is used to count the number of ones detected during the offset calibration. The other PDC samplers are implemented without offset calibration, but with additional input buffers to compensate the setup time of the samplers as described in section 6.3.1.

The threshold voltage V_{th} for the single-ended receiver is generated by a DAC and fed to the negative input of the differential preamplifier. The schematic of the proposed V_{th} generator is shown in Figure 7.5. A 6 bit current-DAC generates a bias current for the load resistor R_{DAC} which acts as a linear current-to-voltage converter at the output. A replica biasing scheme is used to control the reference current in the current-DAC in order to achieve an output voltage range of $V_{\text{DD}}/2 \leq V_{\text{th}} \leq V_{\text{DD}}$. This range is equal to the maximum possible voltage swing of the transmitted signal DQ because of the use of an SST driver with a matched termination to V_{DD}. The replica circuit of the current-DAC and the current-to-voltage converter generates the lowest possible output voltage V_{min} and an operational amplifier compares V_{min} to the reference voltage $V_{\text{DD}}/2$ generated by a resistive voltage divider composed of two matched resistors R_{DIV}. The output of the operational amplifier is the bias voltage for the PMOS current sources in the DAC. An interdigitated layout of the transistors of the current-DAC and the replica circuit, as well as of the resistor pairs R_{DAC} and R_{DIV} minimizes a variation of the output voltage range due to process mismatch. The proposed V_{th} generator consumes 800 µW and has a resolution of 9.5 mV.

7.4.2 Multi-Modulus Clock Divider

The prototype ASIC for the PDC based voltage threshold calibration includes a multi-modulus clock divider to generate the reference clock for the PDC sample clock PLL from the transceiver half-rate clock CK. This way both, the PLL for CK and the PLL for CK_{S} can be implemented as integer PLLs and the system reference clock frequency f_{ref} can be chosen independent of the PDC parameters as explained in section 3.3.3. Figure 7.6 shows a schematic of the proposed multi-modulus clock divider. The two inputs SW_1 and SW_2 control the division ratio of the second clock divider stage consisting of a counter. The output of the counter serves as the modulus control signal MC that switches between a division by 4 and a division by 5 in the 4/5 prescaler. Using this configuration the proposed clock divider generates division ratios of the form $(2^n + 1)$. For the generation of the CK_{S} PLL reference clock a division ratio of $MN + 1$ is required. Since the PDC parameters M and N are preferably chosen as powers of two (see section 3.3.2) the required division ratio is also of the form $(2^n + 1)$. The two control signals SW_1 and SW_2 are used to control the value of M in the clock division ratio for a fixed value of N. The logic table in Table 7.1 lists the possible division ratios and the corresponding values of M for a PDC resolution of $N = 64$. The division of the multi-modulus clock divider into a prescaler and a counter helps to maximize the input frequency range and to minimize the power consumption since only the prescaler is clocked by

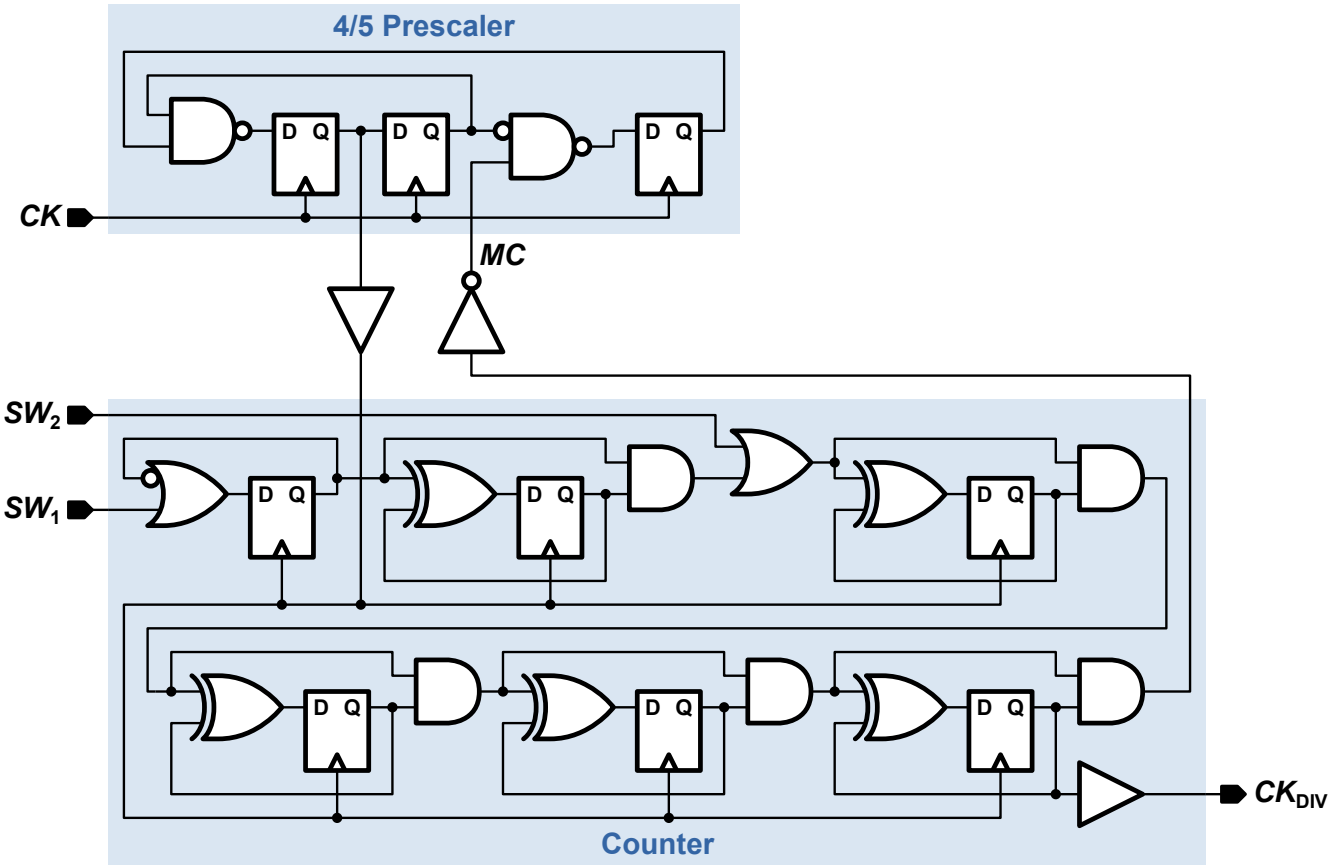


Figure 7.6: Schematic of the multi-modulus clock divider that generates the reference clock for the PDC sample clock PLL.

Table 7.1: Logic table of the proposed multi-modulus clock divider for the generation of the reference clock for the PDC sample clock PLL with $N = 64$.

SW_1	SW_2	Division Ratio	PDC Scaling Factor
0	0	257	$M = 4$
0	1	65	$M = 1$
1	0	129	$M = 2$
1	1	65	$M = 1$

the high-frequency half-rate clock CK and the counter is clocked by the lower frequency output clock of the prescaler. The maximum power consumption of the circuit in the typical corner is 223 µW with an input clock frequency of $f_{CK} = 3.2\,\text{GHz}$.

7.5 Measurement Results

In Figure 7.7 a die photograph of the prototype ASIC for the PDC based voltage threshold calibration is shown. The controller and the memory device are separated in the IC and only the impedance calibration circuit (IMP), the multi-modulus clock divider (CK_S DIV) and the receiver for the PDC sample clock (CK_S RX) are shared between the two devices. The transceiver circuits in the controller device occupy an area of $0.039\,\text{mm}^2$ including the shared DLL and V_{th} generator. In the memory device the transceiver size

is only 0.033 mm^2 since no DLL is required. The multi-modulus clock divider for the CK_S PLL reference clock has an active area of 1100 µm^2. Two instances of the transceiver ASIC in QFN48 packages are mounted on separate PCBs for the prototype interface. One device uses the memory part as a clock receiver for the externally generated half-rate clock CK and connects to the data link with the controller part. The other device connects the memory part of the ASIC to the data link. The forwarded CK, the data link DQ and the 2 bit command bus (CMD) are connected via SMA connectors between the two PCBs as shown in Figure 7.8. Both PCBs include a PLL for the generation of CK_S from the reference clock generated by the multi-modulus clock divider. A microcontroller connects the SPI interfaces of the two prototype ASICs to a computer and configures the PLLs for CK and CK_S. Eye diagram measurements are performed by probing the DQ signal close to the pins of the QFN48 packages. The voltage threshold V_{th} of the single-ended receiver is routed to a probe pad on the PCB next to the DQ pin as the reference voltage for the differential probe.

Two issues in the prototype ASIC affect the functionality and limit the possibilities to verify the effectiveness of the proposed PDC based voltage threshold calibration. One issue is that the PDC in the transceiver slice of the memory part does not run any measurements. Therefore the PDC based write training of the I/O link cannot be performed. Instead only the read training is analyzed in the following. The other issue is a bug in the deserializers. For a reason that is unknown to the date of publication the deserialzers only receive a byte correctly when the previous byte transmitted is “11111111”. As a result BER measurements using PRBS data patterns are not possible. As a workaround the BER measurements are realized by an alternating transmission of “11111111” and a byte of the PRBS pattern using the data FIFOs of the transceivers. This results in a very long measurement duration for a full eye scan with 4096 different sampling points and the transmitted data is not a true PRBS pattern anymore, especially it is not DC free. Also the deserializers don’t work properly at the target data rate of 6.4 Gb/s. Therefore all measurements in the following are performed at 4.4 Gb/s instead.

In order to analyze the effect of the voltage threshold level on the duty-cycle of a received clock pattern repeated duty-cycle measurements using the integrated PDC in the controller transceiver are performed at different V_{th} levels. The results are shown in Figure 7.9. The PDC is able to detect a

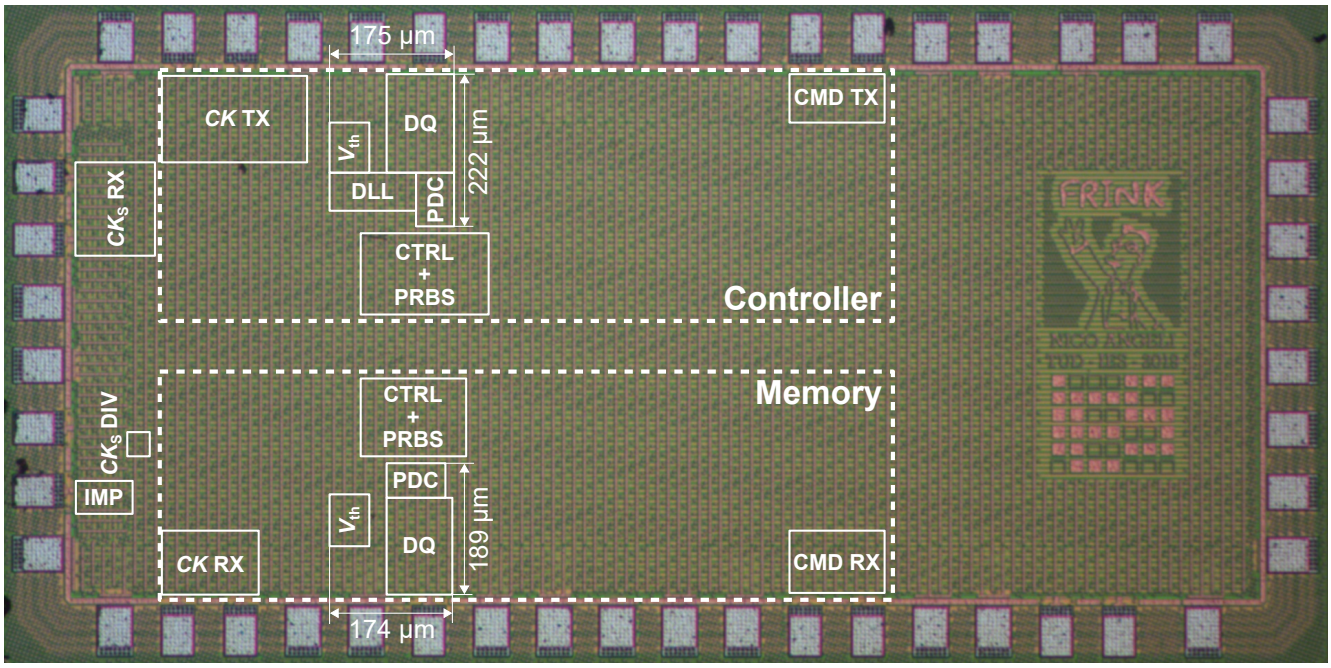


Figure 7.7: Die photograph of the prototype ASIC for the PDC based voltage threshold calibration implemented in the UMC 65 LL CMOS process.

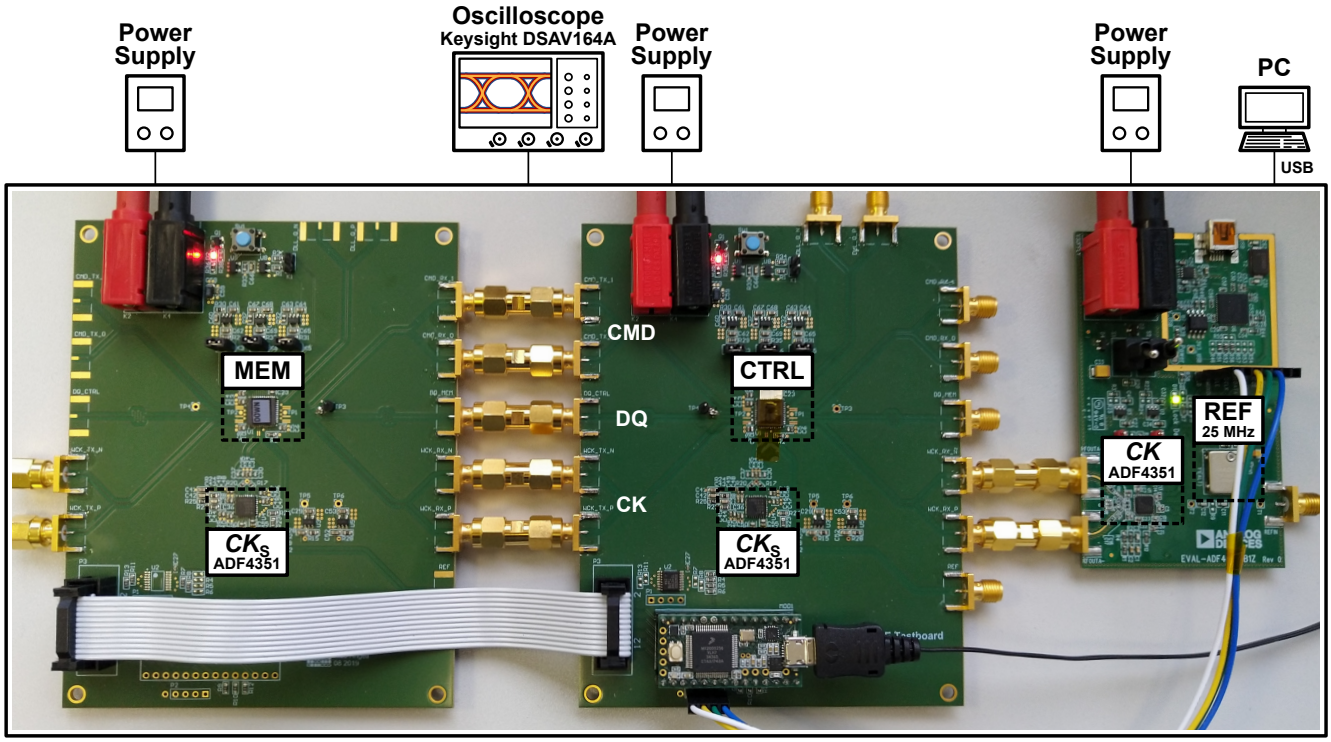


Figure 7.8: PCB and measurement setup for the prototype ASIC of the PDC based voltage threshold calibration in Figure 7.7.

duty-cycle range of 10 % to 90 % of the transmitted clock pattern for V_{th} between 640 mV and 1060 mV. Outside of this range the measurement results do not correlate with V_{th}. The curvature of the PDC measurement reflects a property of the I/O link that is also visible in the eye diagrams of the read and the write operation in Figure 7.10. Due to a duty-cycle distortion of the *DQ* signal the signal crossings in the eye diagram are not at the vertical eye center and therefore the ideal V_{th} cannot be found at the voltage threshold level that leads to a duty-cycle of 50 % (instead the measurement with a duty-cycle of 50 % reflects the V_{th} level with the maximum timing margin). But the PDC duty-cycle measurement in Figure 7.9 also reflects the vertical eye opening defined by the minimum and the maximum of the measurement results. This way the vertical eye center can be calculated using the V_{th} levels with the lowest and the highest duty-cycle similar to the conventional passing window method. The calibrated V_{th} using this method is very close to the ideal V_{th} determined by a BER measurement at $BER = 10^{-3}$ and differs only by one PI step (9.5 mV). The calibrated V_{th} at 50 % duty-cycle is significantly offset from the ideal V_{th} due to the duty-cycle distortion on the *DQ* signal. The duration of the PDC based voltage threshold calibration

$$t_{\mathrm{vt}} = n_{\mathrm{vt}}\, t_{\mathrm{m}} \tag{7.3}$$

depends on the number of voltage threshold levels n_{vt} to be measured and the duration of a single PDC measurement t_{m} according to equation (5.1). For the calibration method that finds V_{th} at a duty-cycle of 50 % the number of measurements is limited to $1 \leq n_{\mathrm{vt}} \leq n_{\mathrm{DAC}}/2$ and for the method that determines the center between the borders of the voltage margin $n_{\mathrm{vt}} = n_{\mathrm{DAC}}$, with the number of V_{th} DAC steps n_{DAC}.

Figure 7.11 shows a full eye scan for the read operation of the prototype I/O link at 4.4 Gb/s. The eye center determined by the proposed PDC based timing center and voltage threshold calibration is marked, as well as the ideal eye center based on the BER measurements. The data eye at the receiver is obviously degraded compared to the eye diagram measured at the package pins. This is probably caused by significant supply noise in the system due to the single-ended transceiver architecture. However, the PDC based timing center calibration is able to calibrate a timing center which is equal to the ideal

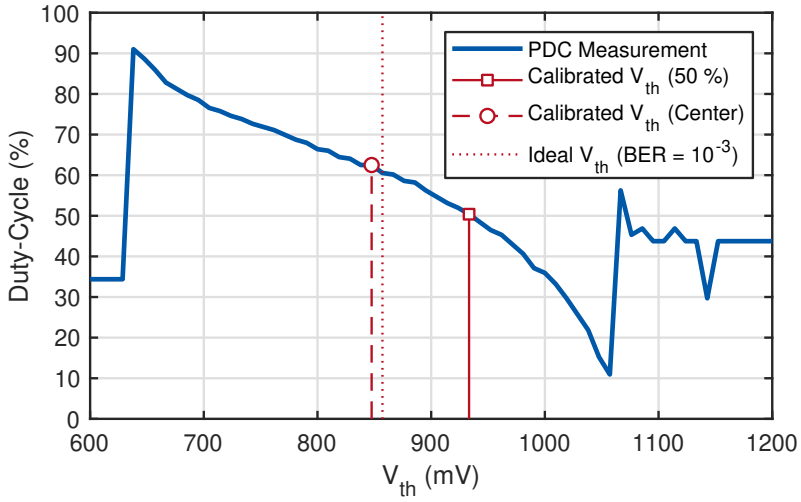
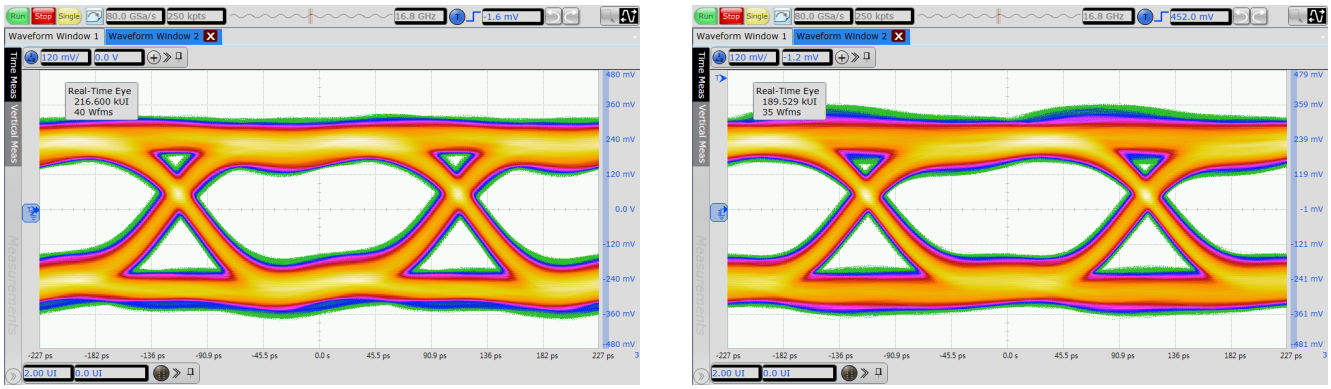


Figure 7.9: PDC measurement of the duty-cycle of a received clock pattern at different threshold voltage (V_{th}) levels for the read operation of the prototype I/O link.



(a) Read operation.

(b) Write operation.

Figure 7.10: Eye diagrams of the read (a) and the write (b) operation of the prototype I/O link at 4.4 Gb/s with active feed-forward equalization, each measured at the receiving side near the pins of the QFN48 package.

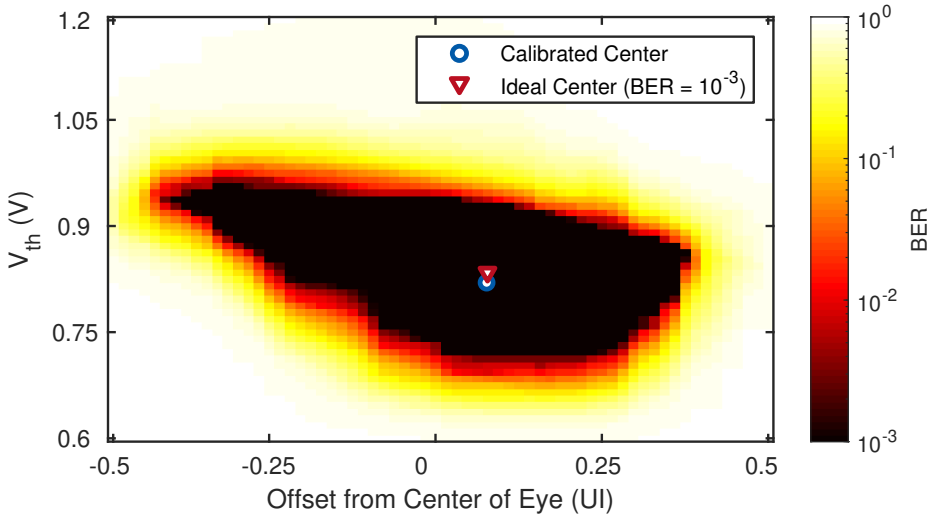


Figure 7.11: Full eye scan for the read operation of the prototype single-ended I/O link at 4.4 Gb/s annotated with the calibrated eye center, as well as the ideal eye center at $BER = 10^{-4}$.

timing center estimated from the BER measurements. The voltage threshold determined by the PDC based calibration scheme is slightly lower than the ideal voltage threshold (see Figure 7.9), but still offers sufficient margin to the borders of the data eye.

7.6 Conclusion

In this chapter a voltage threshold calibration for single-ended high-speed I/O links is proposed based on the PDC presented in chapter 3. The PDC is used to measure the duty-cycle of a received clock pattern at different V_{th} levels. As long as there is no duty-cycle distortion on the transmitted signal *DQ* the ideal V_{th} is found when the measured duty-cycle is equal to 50 %. In the presence of duty-cycle distortion a duty-cycle measurement across the full V_{th} range can be performed to find the minimum and the maximum of the voltage margin. The ideal V_{th} is then calculated as the center between the minimum and the maximum similar to the conventional passing window approach (see section 2.5.5). The proposed voltage threshold calibration method is combined with the PDC based timing center calibration presented in chapter 6 to provide a full two dimensional eye center detection. Both calibration schemes share the same PDC in each transceiver slice to minimize the area overhead. Since the voltage threshold calibration based on a duty-cycle measurement does not require a prior timing center calibration the complete link training requires only two steps. First the voltage threshold calibration is performed, then the timing center calibration. Previously published two dimensional eye detection methods require either three calibration steps (timing center, voltage threshold, timing center [99]) or a full two dimensional eye scan. This way the link training duration is significantly reduced. Both, the calibrated timing center and the calibrated voltage threshold are close to the ideal eye center determined by BER measurements. Due to the lack of suitable measurement results in the literature a comparison of the calibration precision with state-of-the-art designs is not possible at the date of publication of this thesis. Instead only the link training duration is compared in Table 7.2 using the number of bits transmitted during the training n_{tr} from equation (6.11) calculated from the sum of the durations of the voltage threshold calibration t_{vt} and the timing center calibration t_{tr} (equation 6.6) required in the prototype I/O link. The proposed two dimensional eye center detection based on the voltage threshold calibration using the 50 % method is about 60 % faster than the adaptive 3-step method in [99]. It is noted that the adaptive 3-step method in [99] is implemented without averaging and is therefore quite fast, but more prone to timing jitter and voltage noise. Implementing the proposed PDC based calirbation without averaging reduces the training duration by a factor of 16 making both methods faster than the adaptive 3-step algorithm. Using the center method for the voltage threshold calibration the link training duration increases by almost a

Table 7.2: Performance comparison of the proposed PDC based two dimensional eye center detection and the conventional methods.

Method	Link Speed (Gb/s/pin)	Training Duration (µs)	$\frac{\mathbf{n}_{\mathrm{tr}}}{\mathbf{n}_{\mathrm{tr,PDC}}}$
PDC Based (50 % Method)	4.4	7.9/8.1 (Read/Write)	1
PDC Based (Center Method)	4.4	75.8/76.1 (Read/Write)	9.49
Adaptive 3-Step [99]	4.266	11/15.3 (Read/Write)	1.59
Full eye scan ($BER = 10^{-3}$)	4.4	931/1862 (Read/Write)	174.56

factor of 10 compared to the 50 % method, but therefore the calibration can better handle I/O links with duty-cycle distortion. Compared to a full eye scan the proposed method is faster by at least a factor of 18 with a negligible difference of the calibrated eye center.

8 Conclusion

This thesis presents an all-digital clock calibration approach for high-speed I/O links based on a fully synthesized phase-to-digital converter (PDC). The proposed PDC is able to perform in-situ measurements of the phase and the duty-cycle of high-frequency clock signals with high precision. The scalable resolution and sample rate allow the circuit to be adapted to different requirements. Three ASICs have been developed in a 65 nm CMOS process to investigate four different applications of the PDC in the calibration sequence of an asymmetric bidirectional source-synchronous I/O link. In each of the applications the proposed PDC based methods show significant improvements of the power consumption, the calibration precision and the calibration time over the state-of-the-art and offer a fast and simple circuit design due to the all-digital implementation. High-speed I/O links benefit from these improvements as the reduced power consumption enables higher data rates in mobile applications, the better calibration precision improves the link reliability and the reduced calibration time increases the link availability. The main contributions of the work are summarized in the following section. Finally, an outlook on other possible applications of the PDC is given.

8.1 Summary

High-performance source-synchronous I/O links require multiple different circuit calibrations in order to achieve high data rates and to be robust against voltage and temperature variations during operation. One important effect caused by PVT variations is timing drift due to the PVT dependence of the circuit delay. The proposed PDC is ideally suited to track the timing drift precisely and to enable a simple digital compensation. Four different calibration scenarios have been investigated that are essential for a proper data transmission at high data rates in modern chip-to-chip interfaces:

- **Phase detection in multiphase delay-locked loops (DLL)**
  Multiphase DLLs are frequently used in parallel high-speed I/O links for local phase adjustment. Conventional phase detectors are limited to a 360° feedback from the delay line and require a careful full-custom circuit design. The proposed PDC enables the DLL to use any phase as a feedback for the control loop resulting in a shorter delay line and a reduced power consumption. Also the all-digital implementation of the PDC simplifies the circuit design. In combination with the proposed transmission gate based shunt-capacitors an extremely compact DLL with the delay line being the only full-custom element was developed. A worst case model of the PDC allows to estimate the jitter of the DLL and to define constraints for the jitter of the input clock of the DLL and the sample clock of the PDC. The loop filter for the shunt-capacitor based DLL can be designed solely in the digital domain avoiding time consuming mixed mode simulation for the analysis of the loop stability. An alternative approach using current-starved inverters for the delay line was implemented with excellent power efficiency and jitter performance in return for a slightly increased area compared to the shunt-capacitor based DLL.
- **Duty-cycle correction**
  Most high-speed I/O links use at least a double-data rate sampling scheme that requires a duty-cycle of exactly 50 % for the clock signals in the serializers and deserializers. The capability of the PDC to measure the duty-cycle of a high-frequency clock with high-precision is used in the design of an all-digital duty-cycle corrector. The duty-cycle adjustment circuit is designed especially for high-frequency applications and provides a wide duty-cycle adjustment range over a large frequency range with an almost linear duty-cycle characteristic.

- **Timing center calibration**
  An essential requirement for a reliable data transmission is that the data is being sampled in the timing center of the data eye. This way a maximum margin for variations of the delay in the clock and data paths is provided. Outside of the timing margin the bit error rate increases rapidly and a re-calibration of the timing center becomes necessary. Since the data link is unavailable during the timing center calibration it is desirable to minimize the calibration time. The proposed method performs a phase measurement between a transmitted clock pattern and the internal clock using the proposed PDC. The internal clock phase is then adjusted to be 90° offset from the received clock pattern in order to sample the data at the center of the eye. Additional measurements with the PDC enable the compensation of the nonlinearity and the duty-cycle variation of the phase interpolators that are used to shift the phase of the receiver clock. Compared to the conventional passing window method the proposed approach takes significantly less time to calibrate the timing center and reduces the timing center error thanks to the compensation of the phase interpolator nonlinearity. Also, the power consumption during the calibration is reduced since no access to the data FIFOs is required.
- **Voltage threshold calibration**
  In single-ended I/O links the data signal is compared to a voltage threshold in order to distinguish between a zero and a one in the data stream. The ideal voltage threshold is affected by mismatch in the transmitter and receiver circuits, as well as the characteristics of the transmission channel. Therefore a calibration of the voltage threshold is performed to maximize the voltage margin. The conventional approach for a combined timing center and voltage threshold calibration is to perform a full two dimensional eye scan covering all possible combinations of clock phase and voltage threshold. A calibration method based on a duty-cycle measurement of a received clock pattern is proposed which can find the ideal voltage threshold by scanning only the voltage axis. The PDC based timing center calibration can be performed afterwards without affecting the ideal voltage threshold.

An overview of the presented applications of the PDC and the key improvements over the state-of-the-art is shown in Figure 8.1. The use of the proposed PDC has led to a simpler circuit design, a reduced power consumption, an improved precision and a reduced calibration time. The all-digital PDC scales well with technology and can be placed anywhere on the ASIC to perform local phase and duty-cycle measurements.

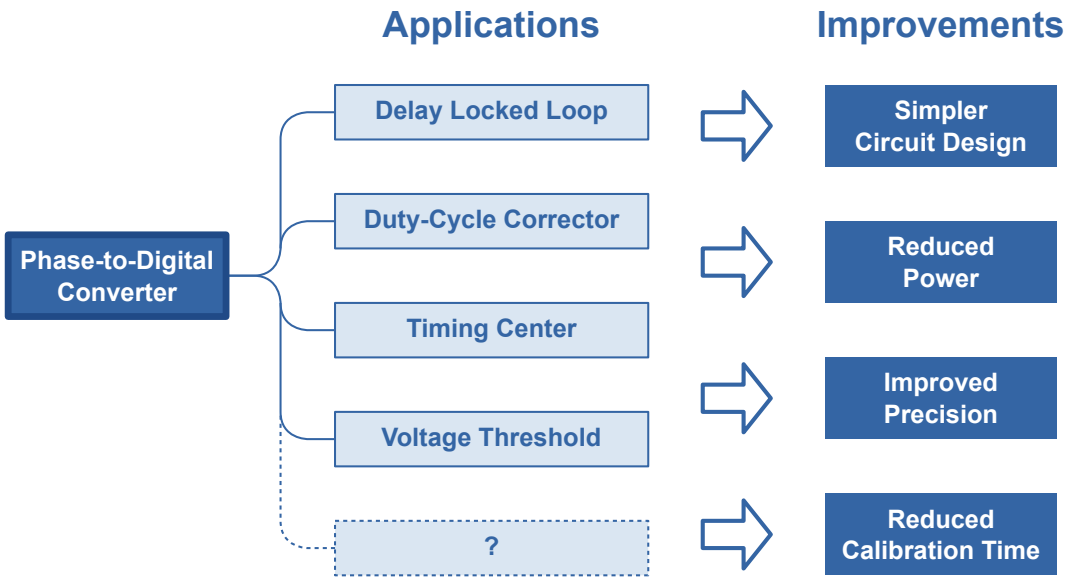


Figure 8.1: Key improvements to the calibration sequence of a typical source-synchronous high-speed parallel I/O link by the PDC based calibration techniques presented in this thesis.

The required sample clock can be shared between multiple instances of the PDC and its distribution is uncritical due to the low frequency (using a high PDC scaling factor) and the relaxed jitter requirements. Therefore, the efficiency of the PDC improves with an increasing number of instances within the ASIC.

8.2 Outlook

The proposed PDC offers a simple and universal way to measure the phase and the duty-cycle of clock signals within an ASIC. The small all-digital circuit can be placed nearly anywhere on the chip to measure the properties of signals that are inaccessible to external measurement equipment. Therefore, the PDC can be useful for debugging purposes of prototype ASICs or for performance analysis of integrated circuits without the need for sensitive analog I/Os. Apart from phase and duty-cycle the PDC can also be used to estimate the jitter of clock signals as long as the jitter of the sample clock is known.

The presented PDC based calibration techniques are not limited to the application in source-synchronous I/O link and can be applied in other high-speed interfaces as well. It could also be interesting to investigate the use of the PDC for the equalizer training. Due to the frequency dependent channel characteristics different frequencies experience a different attenuation and a different delay. It could be evaluated whether a measurement of the clock phase at different frequencies is sufficient to estimate the channel transfer function and to adapt the equalizer taps without the need for lengthy BER measurements.

Part IV
Appendix

A MATLAB Function for the Dimensioning of a Transmission Gate Shunt-Capacitor Array

The MATLAB function in Code A.1 determines the ideal topologies and the estimated gate widths of an n_bit transmission gate shunt-capacitor array with a capacitance increment dc_min. The required simulation data cap_data is obtained from a post layout simulation of the individual capacitor topologies as described in section 4.2.1 and is a three dimensional array with the structure illustrated in Figure A.1. The topology dimension is ordered with increasing c_{ratio}. The script also estimates the worst case DNL and INL of the shunt-capacitor array using the function in Code A.2.

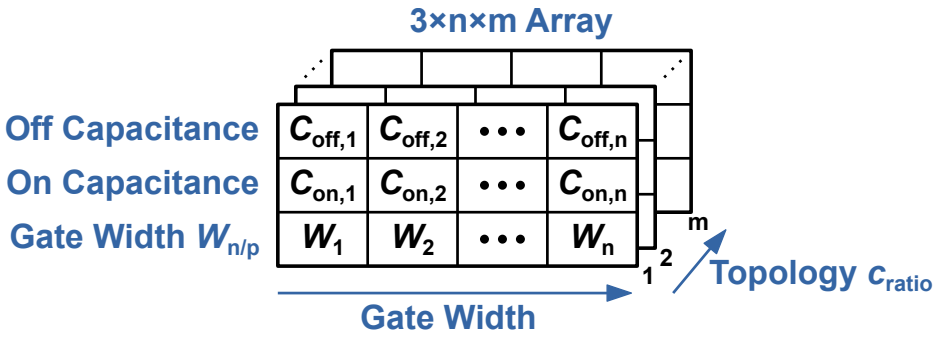


Figure A.1: Structure of the $3 \times n \times m$ simulation data array cap_data for the MATLAB function tg_cap_layout_sizing.

```
function [width, topo, c_min, c_max, dc_avg, resolution, range, c_out, dnl_max, inl_max] =
    tg_cap_layout_sizing(cap_data, dc_min, n_bit)

% Determine optimum topology and transistor width:
dc = dc_min;
for bit = 1:n_bit
    topology = 1;
    % Find the best topology for the current bit and determine optimum transistor width:
    while ((topology <= size(cap_data,3)) && ((cap_data(2,1,topology) - cap_data(1,1,topology)) <= dc)) %
        check if capacitance step can be realized with selected topology
        width(bit) = round(interp1((cap_data(2,:,topology) - cap_data(1,:,topology)), cap_data(3,:,
            topology), dc, 'linear', 'extrap')*2, -1)/2;   % Find ideal width and round to nearest
            multiple of 5 nm
        topo(bit) = topology;
        c_off(bit) = interp1(cap_data(3,:,topology), cap_data(1,:,topology), width(bit), 'linear', '
            extrap');    % get off capacitance
        c_on(bit) = interp1(cap_data(3,:,topology), cap_data(2,:,topology), width(bit), 'linear', 'extrap
            ');    % get on capacitance
        topology = topology + 1;
    end
    dc = 2*dc;  % next bit
end

% Determine range and resolution:
c_min = sum(c_off); % minimum capacitance
c_max = sum(c_on);  % maximum capacitance
dc_avg = (c_max-c_min)/(2^n_bit-1); % average capacitance step
resolution = dc_avg/c_max;  % step size normalized to maximum capacitance
range = (c_max-c_min)/c_max;    % range normalized to maximum capacitance

% Calculate final capacitance values:
for i = 0:(2^n_bit-1)
    num = i;
    dc_bit = 0;
```

```
    for j = n_bit:-1:1
        dc_bit = dc_bit +((num/2^(j-1)) >= 1)*(c_on(j) - c_off(j));
        num = mod(num, 2^(j-1));
    end
    c_out(i+1) = c_min + dc_bit;
end

% Calculate DNL and INL:
[dnl, inl] = dnl_inl([0:(2^n_bit-1); c_out]', 0);
dnl_max = max(abs(dnl));
inl_max = max(abs(inl));

end
```

Code A.1: MATLAB function for dimensioning a transmission gate shunt-capacitor array.

```
% dnl_inl
% This function calculates differential (DNL) and integral (INL) non-linearity of the
% given data in terms of LSB and shows a plot of them. The output plot can be disabled.

% Inputs:
%   data: Nx2 matrix, column 1: x-values, column 2: y-values
%   plot_disable: can be anything, disables output plot
%
% Outputs:
%   dnl: N-1x1 vector of DNL values
%   inl: N-1x1 vector of INL values

function [dnl, inl] = dnl_inl(data, plot_disable)

% calculate DNL:
dnl = (data(2:end,2)-data(1:end-1,2))./((data(end,2)-data(1,2))/(size(data,1)-1))-1;

% calculate INL:
inl = (data(2:end,2)-data(1,2))./((data(end,2)-data(1,2))/(size(data,1)-1))-(1:(size(data,1)-1))';

% check if plot output is enabled:
if (nargin == 1)

    % generate x-values:
    steps = 1:(size(data,1)-1);

    % plot dnl and inl:
    h = plot(steps, dnl, steps, inl);
    % apply styles:
    figure_style();
    line_plot_style(h, 0);
    % edit axes:
    xlim([steps(1) steps(end)]);
    xlabel('Step (LSB)');
    ylabel('DNL/INL (LSB)');
    % legend and title:
    title('DNL and INL');
    legend('DNL', 'INL');
end

end
```

Code A.2: MATLAB function for the calculation of the DNL and INL of an $N \times 2$ array.

B Low-Voltage Bandgap Current Reference

The low-voltage bandgap current reference in Figure B.1 consists of a PTAT voltage reference circuit using an 8:1 ratio for the diode connected PNP transistors and a resistance of $R_1 = 16.7\,\mathrm{k\Omega}$. An operational amplifier is used to amplify the voltage difference between the two branches and generates the voltage V_{PTAT}. In order to avoid startup failures when $V_{\mathrm{PTAT}} = V_{\mathrm{DD}}$ a reset circuit pulls V_{PTAT} down to ground during the global system reset and ensures that a current flows through the two branches of the PTAT voltage reference circuit. A cascoded current mirror with $R_2 = 39.9\,\mathrm{k\Omega}$ is used to improve the power supply rejection ratio of the circuit and the output current mirror creates the NMOS bias voltage V_{bias} from the reference current. The layout of the low-voltage bandgap current reference in the UMC 65 LL CMOS process is shown in Figure B.2. A Monte Carlo analysis is performed for the current in the output branch of the bandgap current reference and the resulting histogram is shown in Figure B.3. An average current of 207 µA with a standard deviation of 18.5 µA is observed in the simulation considering both, process and mismatch variations.

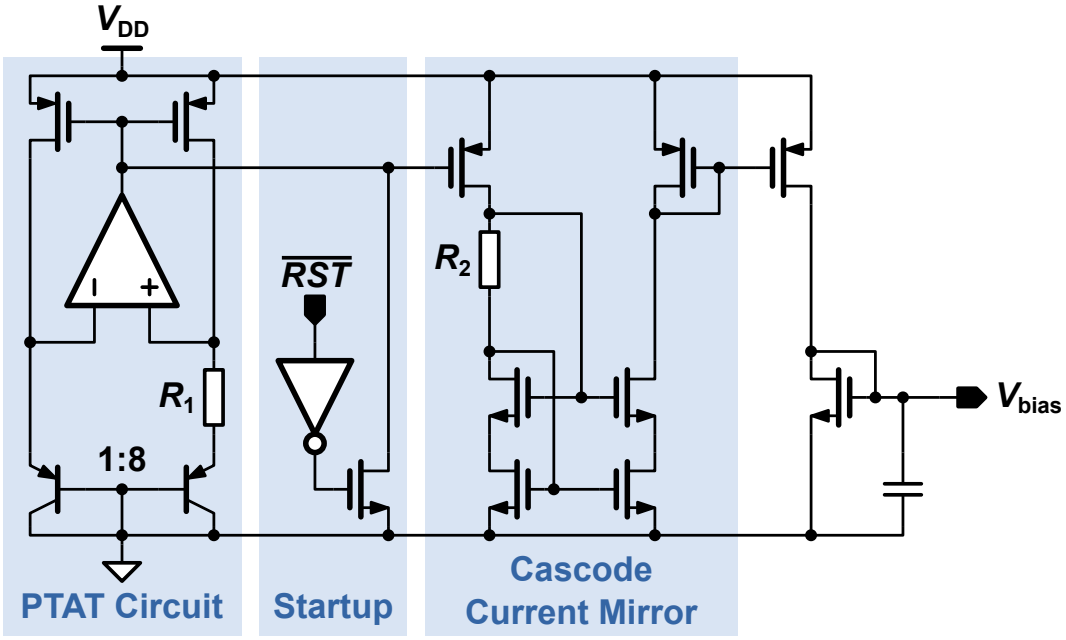


Figure B.1: Schematic of the low-voltage bandgap current reference based on a PTAT voltage reference and a cascoded current mirror.

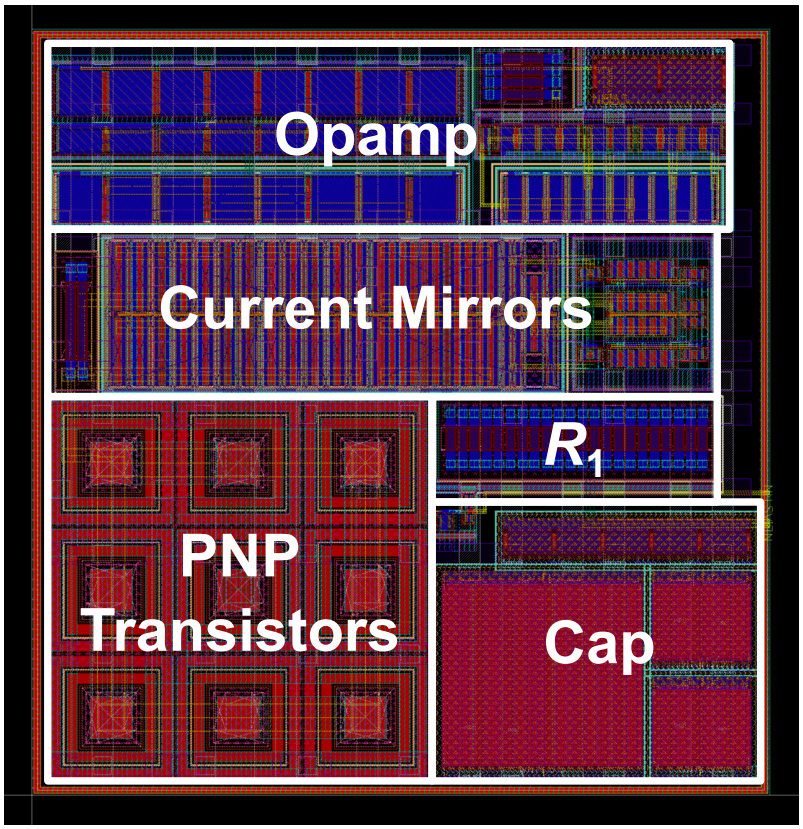


Figure B.2: Layout of the low-voltage bandgap current reference in the UMC 65 LL CMOS process.

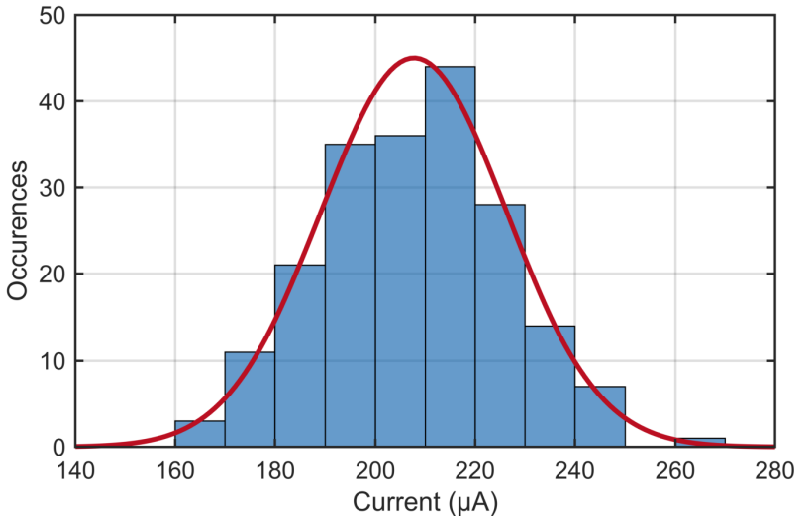


Figure B.3: Simulated current in the output branch of the low-voltage bandgap current reference using Monte Carlo analysis.

C MATLAB Function for Fitting a Bathtub Curve

In [100] a modeling technique for the bathtub curve of high-speed I/O links is described based on the assumption that near the center of the eye the BER is dominated by random jitter and at the borders of the unit interval the BER is dominated by deterministic jitter. The intention of fitting a model function to the measured BER data of a bathtub plot is to extrapolate the measured BER data to the target BER and determine the timing margin at that point. Therefore only the part dominated by random jitter is of interest for the model fit. The model equation

$$BER_{\mathrm{RJ}}(t) = N_{\mathrm{L}} \operatorname{erfc}\left(\frac{t - T_{\mathrm{L}}^{\mathrm{DJ}}}{\sqrt{2}\,\sigma_{\mathrm{L}}}\right) + N_{\mathrm{R}} \operatorname{erfc}\left(\frac{T_{\mathrm{R}}^{\mathrm{DJ}} - t}{\sqrt{2}\,\sigma_{\mathrm{R}}}\right) \tag{C.1}$$

in [100] fits two gaussian tails to the left and the right parts of the bathtub plot. For the fitting the transitions between the random jitter dominated parts and the deterministic jitter dominated parts at the left and the right side are estimated and used as the model parameters $(T_{\mathrm{L}}^{\mathrm{DJ}}, N_{\mathrm{L}})$ and $(T_{\mathrm{R}}^{\mathrm{DJ}}, N_{\mathrm{R}})$. The remaining parameters σ_{L} and σ_{R} are free fitting parameters to adapt the width of the gaussian tails to the measured BER values.

The MATLAB function in code C.1 fits the model equation (C.1) to the BER data measured at different sampling times t. Only measurement point below a limit of ber_limit $= 0.1$ are considered for the model fit and the nearest measurement points to ber_limit on the left and the right side of the bathtub plot are used as the model parameters $(T_{\mathrm{L}}^{\mathrm{DJ}}, N_{\mathrm{L}})$ and $(T_{\mathrm{R}}^{\mathrm{DJ}}, N_{\mathrm{R}})$. The function extrapolates the model to a BER of ber_min and calculates the ideal timing center at a BER of ber_center. Optionally, the function creates a plot with the measured BER data and the model fit.

```
function [x_left, y_left, x_right, y_right, center] = bathtub_fit(phase, ber, ber_min, ber_center,
    plot_enable)

ber_limit = 1e-1;

% Centering:
phase = phase - phase(64);

% Find BER values < 1e-1 for left side:
j = 1;
for i = 1:floor(size(ber,2)/2)
    if ((ber(i) < ber_limit) && (ber(i) > 0))
        X_L(j) = phase(i);
        Y_L(j) = ber(i);
        j = j + 1;
    end
end
T_L = X_L(1);

% Fit BER model:
res_L = fminsearch(@(vars) devsum_L(vars, T_L, X_L, Y_L), [1, 0.03]);
N_L = res_L(1);
sigma_final_L = res_L(2);

% Find BER values < 1e-1 for right side:
j = 1;
for i = ceil(size(ber,2)/2):size(ber,2)
    if ((ber(i) < ber_limit) && (ber(i) > 0))
        X_R(j) = phase(i);
        Y_R(j) = ber(i);
        j = j + 1;
    end
end
T_R = X_R(end);
```

```
34
35 % Fit BER model:
36 res_R = fminsearch(@(vars) devsum_R(vars, T_R, X_R, Y_R), [1, 0.03]);
37 N_R = res_R(1);
38 sigma_final_R = res_R(2);
39
40 % Create model plot vectors:
41 xl_max = erfcinv(ber_min/N_L)*(sqrt(2)*sigma_final_L)+T_L;
42 xr_min = -erfcinv(ber_min/N_R)*(sqrt(2)*sigma_final_R)+T_R;
43 x_left = X_L(1):0.01:xl_max+0.01;
44 x_right = xr_min-0.01:0.01:X_R(end);
45 y_left = N_L.*erfc((x_left-T_L)./(sqrt(2).*sigma_final_L));
46 y_right = N_R.*erfc((T_R-x_right)./(sqrt(2).*sigma_final_R));
47
48 % calculate ideal center @ ber_center
49 cl = interp1(y_left, x_left, ber_center);
50 cr = interp1(y_right, x_right, ber_center);
51 center = (cl+cr)/2;
52
53 if (nargin > 4)
54     if (plot_enable == 1)
55
56         % Plot measurement and model fit:
57         figure;
58         h = semilogy(phase, ber, x_left, y_left, x_right, y_right);
59         ylim([1e-20 1]);
60         xlim([-0.5 0.5]);
61         figure_style([50 50 500 300]);
62         line_plot_style(h(1), 6, 0, [1 1 1]);
63         line_plot_style(h(2), 0, 2, [2 1 1]);
64         line_plot_style(h(3), 0, 2, [2 1 1]);
65         set(gca, 'XTick', -0.5:0.1:0.5);
66         xlabel('Offset from Center of Eye (UI)');
67         ylabel('BER');
68         legend('Measurement', 'Fit', 'Location', 'North');
69     end
70 end
71
72 end
73
74 % Error function for left side:
75 function d = devsum_L(vars, T_L, X_L, Y_L)
76
77 d = sum((log10(Y_L) - log10(vars(1).*erfc((X_L-T_L)./(sqrt(2).*vars(2))))).^2./abs(X_L));
78
79 end
80
81 % Error function for right side:
82 function d = devsum_R(vars, T_R, X_R, Y_R)
83
84 d = sum((log10(Y_R) - log10(vars(1).*erfc((T_R-X_R)./(sqrt(2).*vars(2))))).^2./abs(X_R));
85
86 end
```

Code C.1: MATLAB function for fitting a bathtub curve to a set of BER measurements.

Bibliography

[1] K. Rupp, "Microprocessor trend data," [Accessed: 2019/11/19]. [Online]. Available: https://github.com/karlrupp/microprocessor-trend-data

[2] M. Mansuri, B. Casper, and F. O'Mahony, "An on-die all-digital delay measurement circuit with 250fs accuracy," *Symp. VLSI Circuits*, 2012. DOI: 10.1109/VLSIC.2012.6243808

[3] R. Z. Bhatti, M. Denneau, and J. Draper, "Phase measurement and adjustment of digital signals using random sampling technique," *IEEE Int. Symp. Circuits and Syst. (ISCAS)*, 2006. DOI: 10.1109/ISCAS.2006.1693477

[4] S. Joshi, J. T.-S. Liao, Y. Fan, S. Hyvonen, M. Nagarajan, J. Rizk, H.-J. Lee, and I. Young, "A 12-Gb/s transceiver in 32-nm bulk CMOS," *Symp. VLSI Circuits, Dig. Tech. Papers*, 2009.

[5] M. Sigalov, D. Regev, E. Kabatsky, and R. Shavit, "Quad flat non-lead package characterization and circuit modeling," *PIERS Proceedings*, pp. 642–646, Aug. 2009.

[6] *Jitter Analysis: The dual-Dirac Model, RJ/DJ, and Q-Scale (White Paper)*, Agilent Technologies, Inc. Std., Dec. 2004, [Accessed: 2019/11/11]. [Online]. Available: https://www.keysight.com/upload/cmc_upload/All/dualdirac1.pdf

[7] G. Moore, "Gordon moore: The man whose name means progress, the visionary engineer reflects on 50 years of moore's law," *IEEE Spectrum: Special Report*, vol. 50, 2017.

[8] "Samsung electronics starts production of EUV-based 7nm LPP process," SAMSUNG, Tech. Rep., 2018, [Accessed: 2019/11/19]. [Online]. Available: https://news.samsung.com/global/samsung-electronics-starts-production-of-euv-based-7nm-lpp-process

[9] J. Kim, A. Balankutty, R. K. Dokania, A. Elshazly, H. S. Kim, S. Kundu, D. Shi, S. Weaver, K. Yu, and F. O'Mahony, "A 112 Gb/s PAM-4 56 gb/s NRZ reconfigurable transmitter with three-tap FFE in 10-nm FinFET," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 29–42, Jan. 2019. DOI: 10.1109/jssc.2018.2874040

[10] *HyperTransport™ Technology I/O Link*, ADVANCED MICRO DEVICES, INC. Std., Jul. 2001, [Accessed: 2019/10/05]. [Online]. Available: https://docs.wixstatic.com/ugd/071cb6_5fe1f8d4c3b343799a9604568ca47c6d.pdf

[11] *An Introduction to the Intel ©QuickPath Interconnect*, Intel Corporation Std., Jan. 2009. [Online]. Available: https://www.intel.com/content/dam/doc/white-paper/quick-path-interconnect-introduction-paper.pdf

[12] *JESD212C GDDR5 SGRAM*, JEDEC Solid State Technology Association Std., Dec. 2009, [Accessed: 2019/10/05]. [Online]. Available: https://www.jedec.org/standards-documents/docs/jesd212c

[13] *JESD250B GDDR6 SGRAM*, JEDEC Solid State Technology Association Std., Nov. 2018, [Accessed: 2019/10/05]. [Online]. Available: https://www.jedec.org/standards-documents/docs/jesd250b

[14] *JESD235A High Bandwidth Memory (HBM) DRAM*, JEDEC Solid State Technology Association Std., Nov. 2018, [Accessed: 2019/10/05]. [Online]. Available: https://www.jedec.org/standards-documents/docs/jesd235a

[15] M. ta Hsieh and G. E. Sobelman, "Architectures for multi-gigabit wire-linked clock and data recovery," *IEEE Circuits Syst. Mag.*, vol. 8, no. 4, pp. 45–57, 2008. DOI: 10.1109/MCAS.2008.930152

[16] A. X. Widmer and P. A. Franaszek, "A DC-balanced, partitioned-block, 8b/10b transmission code," *IBM Journal of Research and Development*, vol. 27, no. 5, pp. 440–451, Sep. 1983. DOI: 10.1147/rd.275.0440

[17] *USB 3.2 Specification*, USB Implementers Forum, Inc. Std., Sep. 2017, [Accessed: 2019/10/06]. [Online]. Available: https://usb.org/document-library/usb-32-specification-released-september-22-2017-and-ecns

[18] *Thunderbolt™ Update*, Intel Corp. Std., [Accessed: 2019/10/06]. [Online]. Available: https://thunderbolttechnology.net/sites/default/files/tbt3presentation.pdf

[19] *NVIDIA NVLink*, NVIDIA Corporation Std., [Accessed: 2019/10/06]. [Online]. Available: https://www.nvidia.com/de-de/data-center/nvlink/

[20] *PCI Express Base Specification Revision 5.0, Version 1.0*, PCI-SIG Std., May 2019, [Accessed: 2019/10/06]. [Online]. Available: https://pcisig.com/specifications

[21] *AN-1205 Electrical Performance of Packages, Application Report*, Texas Instruments, May 2004, [Accessed: 2019/10/06]. [Online]. Available: http://www.ti.com/lit/an/snoa405a/snoa405a.pdf

[22] E. O. Hammerstad, "Equations for microstrip circuit design," in *5th European Microwave Conference*, Oct. 1975. DOI: 10.1109/euma.1975.332206

[23] M. P. Li, J. Wilstrup, R. Jessen, and D. Petrich, "A new method for jitter decomposition through its distribution tail fitting," in *International Test Conference 1999. Proceedings (IEEE Cat. No.99CH37034)*, 1999. DOI: 10.1109/test.1999.805809

[24] Y. Duan, "Accurate jitter decomposition in high-speed links," phdthesis, Iowa State University, 2017. [Online]. Available: https://lib.dr.iastate.edu/etd/16276

[25] A. Blankman, "Understanding SDAIII jitter calculation methods," Teledyne Lecroy, Tech. Rep., Sep. 2012, [Accessed: 2019/11/11]. [Online]. Available: http://cdn.teledynelecroy.com/files/whitepapers/understanding_sdaiii_jitter_calculation_methods.pdf

[26] R. Pulikkoonattu, "Oscillator phase noise and sampling clock jitter," ST Microelektrinics, Tech. Rep., Jun. 2007, [Accessed: 2019/11/12]. [Online]. Available: https://documents.epfl.ch/users/p/pu/pulikkoo/private/report_pn_jitter_oscillator_ratna.pdf

[27] H. Song, *VLSI High-Speed I/O Circuits*. Xlibris, 2009. ISBN 978-1-4415-5987-6

[28] P.-J. Peng, Y.-T. Chen, C.-H. Chen, S.-T. Lai, H.-E. Huang, H.-H. Lu, and T.-C. Yu, "A 50-gb/s quarter-rate voltage-mode transmitter with three-tap FFE in 40-nm CMOS," in *IEEE European Solid-State Circuits Conf. (ESSCIRC)*, Sep. 2018. DOI: 10.1109/esscirc.2018.8494286

[29] M. Kossel, C. Menolfi, J. Weiss, P. Buchmann, G. von Bueren, L. Rodoni, T. Morf, T. Toifl, and M. Schmatz, "A t-coil-enhanced 8.5 gb/s high-swing SST transmitter in 65 nm bulk CMOS with $<$ -16 dB return loss over 10 GHz bandwidth," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2905–2920, Dec. 2008. DOI: 10.1109/jssc.2008.2006230

[30] B. Razavi, "The strongARM latch," *IEEE Solid-State Circuits Mag.*, vol. 7, no. 2, pp. 12–17, Jun. 2015. DOI: 10.1109/MSSC.2015.2418155

[31] S. Palermo, A. Emami-Neyestanak, and M. Horowitz, "A 90 nm CMOS 16 Gb/s transceiver for optical interconnects," *IEEE J. Solid-State Circuits*, vol. 43, no. 5, pp. 1235–1246, may 2008. DOI: 10.1109/jssc.2008.920330

[32] K. Kaviani, T. Wu, J. Wei, A. Amirkhany, J. Shen, T. J. Chin, C. Thakkar, W. T. Beyene, N. Chan, C. Chen, B. R. Chuang, D. Dressler, V. P. Gadde, M. Hekmat, E. Ho, C. Huang, P. Le, Mahabaleshwara, C. Madden, N. K. Mishra, L. Raghavan, K. Saito, R. Schmitt, D. Secker, X. Shi, S. Fazeel, G. S. Srinivas, S. Zhang, C. Tran, A. Vaidyanath, K. Vyas, M. Jain, K.-Y. K. Chang, and X. Yuan, "A

tri-modal 20-Gbps/link differential/DDR3/GDDR5 memory interface," *IEEE J. Solid-State Circuits*, vol. 47, no. 4, pp. 926–937, apr 2012. DOI: 10.1109/jssc.2012.2185370

[33] M. Yoshioka, K. Ishikawa, T. Takayama, and S. Tsukamoto, "A 10-b 50-MS/s 820-μW SAR ADC with on-chip digital calibration," *IEEE Trans. Biomedical Circuits and Syst.*, vol. 4, no. 6, pp. 410–416, dec 2010. DOI: 10.1109/tbcas.2010.2081362

[34] H. Lee, K.-Y. K. Chang, J.-H. Chun, T. Wu, Y. Frans, B. Leibowitz, N. Nguyen, T. J. Chin, K. Kaviani, J. Shen, X. Shi, W. T. Beyene, S. Li, R. Navid, M. Aleksic, F. S. Lee, F. Quan, J. Zerbe, R. Perego, and F. Assaderaghi, "A 16 gb/s/link, 64 gb/s bidirectional asymmetric memory interface," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1235–1247, Apr. 2009. DOI: 10.1109/JSSC.2009.2014199

[35] A. Amirkhany, J. Wei, N. K. Mishra, J. Shen, W. T. Beyene, C. Chen, T. J. Chin, D. Dressler, C. Huang, V. P. Gadde, M. Hekmat, K. Kaviani, H. Lan, P. Le, Mahabaleshwara, C. Madden, S. Mukherjee, L. Raghavan, K. Saito, D. Secker, A. Sendhil, R. Schmitt, S. Fazeel, G. S. Srinivas, T. Wu, C. Tran, A. Vaidyanath, K. Vyas, L. Yang, M. Jain, K.-Y. K. Chang, and X. Yuan, "A 12.8-Gb/s/link tri-modal single-ended memory interface," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 4, pp. 911–925, Apr. 2012. DOI: 10.1109/jssc.2012.2185369

[36] T. Musah, J. Jaussi, G. Balamurugan, S. Hyvonen, T.-C. Hsueh, G. Keskin, S. Shekhar, J. Kennedy, S. Sen, R. Inti, M. Mansuri, M. Leddige, B. Horine, C. Roberts, R. Mooney, and B. Casper, "A 4-32 Gb/s bidirectional link with 3-tap FFE/6-tap DFE and collaborative CDR in 22 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 12, pp. 3079–3090, Dec. 2014. DOI: 10.1109/jssc.2014.2348556

[37] S. Gondi and B. Razavi, "Equalization and clock and data recovery techniques for 10-Gb/s CMOS serial-link receivers," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 9, pp. 1999–2011, Sep. 2007. DOI: 10.1109/jssc.2007.903076

[38] J. Zerbe, C. Werner, V. Stojanovic, F. Chen, J. Wei, G. Tsang, D. Kim, W. Stonecypher, A. Ho, T. Thrush, R. Kollipara, M. Horowitz, and K. Donnelly, "Equalization and clock recovery for a 2.5-10-Gb/s 2-PAM/4-PAM backplane transceiver cell," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2121–2130, Dec. 2003. DOI: 10.1109/jssc.2003.818572

[39] Y. Fang, "PHY link design and optimization for high-speed low-power communication systems," phdthesis, TU Darmstadt, 2015. [Online]. Available: https://tuprints.ulb.tu-darmstadt.de/id/eprint/4437

[40] D. Oh, A. Vaidyanath, C. Madden, Y. Frans, and W. Kim, "Optimizing the timing center for high-speed parallel buses," in *IEEE 62nd Electron. Components and Technol. Conf. (ECTC)*, may 2012. DOI: 10.1109/ectc.2012.6248823

[41] T. sheng Ku and A. R. Shaikh, "Data sampling clock edge placement training for high speed GPU-memory interface," patent US 7,259,606 B2, Aug. 21, 2007.

[42] D.-S. Kim, H. Song, T. Kim, S. Kim, and D.-K. Jeong, "A 0.3–1.4 GHz all-digital fractional-N PLL with adaptive loop gain controller," *IEEE J. Solid-State Circuits*, vol. 45, no. 11, pp. 2300–2311, Oct. 2010. DOI: 10.1109/JSSC.2010.2064050

[43] A. Elshazly, R. Inti, B. Young, and P. K. Hanumolu, "Clock multiplication techniques using digital multiplying delay-locked loops," *IEEE J. Solid-State Circuits*, vol. 48, no. 6, pp. 1416–1428, Jun. 2013. DOI: 10.1109/JSSC.2013.2254552

[44] A. Elshazly, A. Balankutty, Y.-Y. Huang, K. Yu, and F. O'Mahony, "A 2GHz-to-7.5GHz quadrature clock-generator using digital delay locked loops for multi-standard I/Os in 14nm CMOS," *Symp. VLSI Circuits*, 2014. DOI: 10.1109/VLSIC.2014.6858375

[45] S. Henzler, *Time-to-Digital Converters*. Springer Science + Buisness Media B.V., 2010. ISBN 978-90-481-8627-3

[46] M. Zanuso, S. Levantino, C. Samori, and A. Lacaita, "A 3mhz-bw 3.6ghz digital fractional-n pll with sub-gate-delay tdc, phase-interpolation divider, and digital mismatch cancellation," *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2010.

[47] Y. Jang, S. Bae, and H. Park, "Cmos digital duty cycle correction circuit for multi-phase clock," *Electron. Lett.*, vol. 39, no. 19, pp. 1383–1384, Sep. 2003. DOI: 10.1049/el:20030908

[48] Y.-J. Min, C.-H. Jeong, K.-Y. Kim, W. H. Choi, J.-P. Son, C. Kim, and S.-W. Kim, "A 0.31–1 GHz fast-corrected duty-cycle corrector with successive approximation register for DDR DRAM applications," *IEEE Trans. VLSI Syst.*, vol. 20, no. 8, pp. 1524–1528, Aug. 2012. DOI: 10.1109/TVLSI.2011.2158011

[49] C.-H. Jeong, A. Abdullah, Y.-J. Min, I.-C. Hwang, and S.-W. Kim, "All-digital duty-cycle corrector with a wide duty correction range for DRAM applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 1, pp. 363–367, Jan. 2016. DOI: 10.1109/tvlsi.2015.2394486

[50] K. Agarwal and R. Montoye, "A duty-cycle correction circuit for high-frequency clocks," *Symp. VLSI Circuits*, 2006. DOI: 10.1109/VLSIC.2006.1705332

[51] J.-H. Chae, H. Ko, J. Park, and S. Kim, "A quadrature clock corrector for DRAM interfaces, with a duty-cycle and quadrature phase detector based on a relaxation oscillator," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 4, pp. 978–982, Apr. 2019. DOI: 10.1109/tvlsi.2018.2883730

[52] K.-H. Cheng, C.-W. Su, and K.-F. Chang, "A high linearity, fast-locking pulsewidth control loop with digitally programmable duty cycle correction for wide range operation," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 399–413, Jan. 2008. DOI: 10.1109/JSSC.2007.914286

[53] S.-K. Kao and S.-I. Liu, "All-digital fast-locked synchronous duty-cycle corrector," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 12, pp. 1363–1367, Dec. 2006. DOI: 10.1109/tcsii.2006.885396

[54] R. Bhatti, M. Denneau, and J. Draper, "Duty cycle measurement and correction using a random sampling technique," in *Midwest Symposium on Circuits and Systems (MWSCAS)*, 2005. DOI: 10.1109/mwscas.2005.1594283

[55] Q. A. Khan, G. Siddhartha, D. Tripathi, S. K. Wadhwa, and K. Misri, "Techniques for on-chip process voltage and temperature detection and compensation," *Symp. VLSI Circuits*, 2006. DOI: 10.1109/VLSID.2006.155

[56] F. Lin, R. A. Royer, B. Johnson, and B. Keeth, "A wide-range mixed-mode DLL for a combination 512 Mb 2.0 Gb/s/pin GDDR3 and 2.5 Gb/s/pin GDDR4 SDRAM," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 631–641, Feb. 2008. DOI: 10.1109/JSSC.2007.916623

[57] H. Menninga, C. Favi, M. W. Fishburn, and E. Charbon, "A multi-channel, 10ps resolution, FPGA-based TDC with 300MS/s throughput for open-source PET applications," in *IEEE Nuclear Science Symposium Conference Record*, Oct. 2011. DOI: 10.1109/nssmic.2011.6154362

[58] J. Yu, F. F. Dai, and R. C. Jaeger, "A 12-bit vernier ring time-to-digital converter in 0.13 μm CMOS technology," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 830–842, Apr. 2010. DOI: 10.1109/jssc.2010.2040306

[59] H. Wang and F. F. Dai, "A 14-bit, 1-ps resolution, two-step ring and 2D vernier TDC in 130nm CMOS technology," in *IEEE European Solid State Circuits Conference*. IEEE, sep 2017. DOI: 10.1109/esscirc.2017.8094546

[60] Y. Park and D. D. Wentzloff, "A cyclic vernier time-to-digital converter synthesized from a 65nm CMOS standard library," in *IEEE International Symposium on Circuits and Systems*, May 2010. DOI: 10.1109/iscas.2010.5537815

[61] M. Z. Straayer and M. H. Perrott, "A multi-path gated ring oscillator TDC with first-order noise shaping," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1089–1098, Apr. 2009. DOI: 10.1109/jssc.2009.2014709

[62] K. Kim, W. Yu, and S. Cho, "A 9 bit, 1.12 ps resolution 2.5 b/stage pipelined time-to-digital converter in 65 nm CMOS using time-register," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 1007–1016, Apr. 2014. DOI: 10.1109/jssc.2013.2297412

[63] T. Iizuka, T. Koga, T. Nakura, and K. Asada, "A fine-resolution pulse-shrinking time-to-digital converter with completion detection utilizing built-in offset pulse," in *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Nov. 2016. DOI: 10.1109/asscc.2016.7844198

[64] W. Yu, K. Kim, and S. Cho, "A 0.22 ps rms integrated noise 15 MHz bandwidth fourth-order $\delta\sigma$ time-to-digital converter using time-domain error-feedback filter," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 5, pp. 1251–1262, May 2015. DOI: 10.1109/JSSC.2015.2399673

[65] T.-C. Hsueh, G. Balamurugan, J. Jaussi, S. Hyvonen, J. Kennedy, G. Keskin, T. Musah, S. Shekhar, R. Inti, S. Sen, M. Mansuri, C. Roberts, and B. Casper, "A 25.6Gb/s differential and DDR4/GDDR5 dual-mode transmitter with digital clock calibration in 22nm CMOS," *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2014. DOI: 10.1109/ISSCC.2014.6757506

[66] A. Hidvégi, P. Geßler, H. Kay, V. Petrosyan, G. Petrosyan, L. Petrosyan, A. Aghababyan, C. Stechmann, K. Rehlich, and C. Bohm, "Fpga based phase detector for high-speed clocks with pico-seconds resolution," *IEEE Nuclear Sci. Symp. Medical Imaging Conf.*, 2013. DOI: 10.1109/NSSMIC.2013.6829780

[67] I. Shake, H. Takara, and S. Kawanishi, "Simple measurement of eye diagram and BER using high-speed asynchronous sampling," *Journal of Lightwave Technology*, vol. 22, no. 5, pp. 1296–1302, May 2004. DOI: 10.1109/jlt.2004.827669

[68] B. Dehlaghi, S. Magierowski, and L. Belostotski, "A 12.5-Gb/s on-chip oscilloscope to measure eye diagrams and jitter histograms of high-speed signals," *IEEE Trans. VLSI Syst.*, vol. 22, no. 5, pp. 1127–1137, May 2014. DOI: 10.1109/TVLSI.2013.2265895

[69] P.-H. Lee, H.-Y. Lee, H.-B. Lee, and Y.-C. Jang, "An on-chip monitoring circuit for signal-integrity analysis of 8-Gb/s chip-to-chip interfaces with source-synchronous clock," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1386–1396, Apr. 2017. DOI: 10.1109/tvlsi.2016.2639289

[70] *JESD65B Definition of Skew Specifications for Standard Logic Devices*, JEDEC Solid State Technology Association Std., Sep. 2003, [Accessed: 2019/11/26]. [Online]. Available: https://www.jedec.org/sites/default/files/docs/jesd65b.pdf

[71] J.-H. Chae, M. Kim, H. Ko, Y. Jeong, J. Park, G.-M. Hong, D.-K. Jeong, and S. Kim, "266-2133 MHz phase shifter using all-digital delay-locked loop and triangular-modulated phase interpolator for LPDDR4X interface," *Electron. Lett.*, vol. 53, no. 12, pp. 766–768, Jun. 2017. DOI: 10.1049/el.2017.1291

[72] J.-H. Chae, M. Kim, G.-M. Hong, J. Park, H. Ko, W.-Y. Shin, H. Chi, D.-K. Jeong, and S. Kim, "0.11-2.5 GHz all-digital DLL for mobile memory interface with phase sampling window adaptation to reduce jitter accumulation," *J. Semiconductor Technol. and Sci.*, vol. 17, no. 3, pp. 411–424, Jun. 2017. DOI: 10.5573/JSTS.2017.17.3.411

[73] B. I. Abdulrazzaq, I. Abdul Halin, S. Kawahito, R. M. Sidek, S. Shafie, and N. A. M. Yunus, "A review on high-resolution CMOS delay lines: towards sub-picosecond jitter performance," *SpringerPlus*, vol. 5, no. 1, p. 434, Apr. 2016. DOI: 10.1186/s40064-016-2090-z

[74] J. Maneatis, "Low-jitter process-independent DLL and PLL based on self-biased techniques," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1723–1732, Nov. 1996. DOI: 10.1109/jssc.1996.542317

[75] C.-N. Chuang and S.-I. Liu, "A 0.5-5 GHz wide-range multiphase DLL with a calibrated charge pump," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 11, pp. 939–943, Nov. 2007. DOI: 10.1109/tcsii.2007.904155

[76] S. Chen, H. Li, and P. Y. Chiang, "A robust energy/area-efficient forwarded-clock receiver with all-digital clock and data recovery in 28-nm CMOS for high-density interconnects," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 2, pp. 578–586, Feb. 2016. DOI: 10.1109/tvlsi.2015.2409987

[77] J. Segura, C. Hawkins, and J. Soden, "Failure mechanisms and testing in nanometer technologies," in *Gizopoulos / Advances in ElectronicTesting*. Springer US, 2006, pp. 43–75.

[78] M. Mansour and N. Shanbhag, "Simplified current and delay models for deep submicron CMOS digital circuits," in *IEEE International Symposium on Circuits and Systems*, 2002. DOI: 10.1109/iscas.2002.1010652

[79] M. Maymandi-Nejad and M. Sachdev, "A digitally programmable delay element: design and analysis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 5, pp. 871–878, Oct. 2003. DOI: 10.1109/tvlsi.2003.810787

[80] H. Pekau, A. Yousif, and J. Haslett, "A CMOS integrated linear voltage-to-pulse-delay-time converter for time based analog-to-digital converters," in *IEEE International Symposium on Circuits and Systems*, 2006. DOI: 10.1109/iscas.2006.1693099

[81] K.-I. Oh, L.-S. Kim, K.-I. Park, Y.-H. Jun, and K. Kim, "Low-jitter multi-phase digital DLL with closest edge selection scheme for DDR memory interface," *Electron. Lett.*, vol. 44, no. 19, pp. 1121–1123, Sep. 2008. DOI: 10.1049/el:20081833

[82] J.-H. Chae, G.-M. Hong, J. Park, M. Kim, H. Ko, W.-Y. Shin, H. Chi, D.-K. Jeong, and S. Kim, "A 1.74mW/GHz 0.11-2.5GHz fast-locking, jitter-reducing, 180° phase-shift digital DLL with a window phase detector for LPDDR4 memory controllers," *IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, 2015. DOI: 10.1109/ASSCC.2015.7387434

[83] P. Raha, S. Randall, R. Jennings, B. Helmick, A. Amerasekera, and B. Haroun, "A robust digital delay line architecture in a 0.13μm CMOS technology node for reduced design and process sensitivities," *Int. Symp. Quality Electron. Design (ISQED)*, 2002.

[84] P.-L. Chen, C.-C. Chung, and C.-Y. Lee, "A portable digitally controlled oscillator using novel varactors," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 5, pp. 233–237, Jul. 2005. DOI: 10.1109/TCSII.2005.846307

[85] W. Deng, D. Yang, T. Ueno, T. Siriburanon, S. Kondo, K. Okada, and A. Matsuzawa, "A fully synthesizable all-digital PLL with interpolative phase coupled oscillator, current-output DAC, and fine-resolution digital varactor using gated edge injection technique," *IEEE J. Solid-State Circuits*, vol. 50, no. 1, pp. 68–80, Jan. 2015. DOI: 10.1109/JSSC.2014.2348311

[86] A. Ravi, P. Madoglio, H. Xu, K. Chandrashekar, M. Verhelst, S. Pellerano, L. Cuellar, M. Aguirre-Hernandez, M. Sajadieh, J. E. Zarate-Roldan, O. Bochobza-Degani, H. Lakdawala, and Y. Palaskas, "A 2.4 GHz 20-40 MHz channel WLAN digital outphasing transmitter utilizing a delay-based

wideband phase modulator in 32-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 47, no. 12, pp. 3184–3196, Dec. 2012. DOI: 10.1109/JSSC.2012.2216671

[87] B. Nikolic, V. Oklobdzija, V. Stojanovic, W. Jia, J. K.-S. Chiu, and M. M.-T. Leung, "Improved sense-amplifier-based flip-flop: design and measurements," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 6, pp. 876–884, Jun. 2000. DOI: 10.1109/4.845191

[88] E. Bayram, A. F. Aref, M. Saeed, and R. Negra, "1.5–3.3 ghz, 0.0077 mm^2 , 7 mw all-digital delay-locked loop with dead-zone free phase detector in 0.13 μm $cmos$," *IEEE Trans. Circuits Syst. I*, vol. 65, no. 1, pp. 39–50, Jan. 2018. DOI: 10.1109/TCSI.2017.2715899

[89] K. Ryu, D.-H. Jung, and S.-O. Jung, "Process-variation-calibrated multiphase delay locked loop with a loop-embedded duty cycle corrector," *IEEE Trans. Circuits Syst. II*, vol. 61, no. 1, pp. 1–5, Jan. 2014. DOI: 10.1109/TCSII.2013.2291052

[90] M. Jalalifar and G.-S. Byun, "A low-power low-jitter DLL with a differential closed-loop duty cycle corrector," *Analog Integrated Circuits and Signal Process.*, vol. 93, no. 1, pp. 149–155, Oct. 2017. DOI: 10.1007/s10470-017-0984-5

[91] K. Kim, S. Son, S. Ryu, H. Yeo, Y. Choi, and J. Kim, "A 1.3-mW, 1.6-GHz digital delay-locked loop with two-cycle locking time and dither-free tracking," *Symp. VLSI Circuits*, 2013.

[92] S.-Y. Kim, X. Jin, J.-H. Chun, and K.-W. Kwon, "A digital DLL with 4-cycle lock time and 1/4 NAND-delay accuracy," *IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, 2015. DOI: 10.1109/ASSCC.2015.7387472

[93] J. Kim and S. Han, "A fast-locking all-digital multiplying DLL for fractional-ratio dynamic frequency scaling," *IEEE Trans. Circuits Syst. II*, vol. 65, no. 3, pp. 276–280, Mar. 2018. DOI: 10.1109/TCSII.2017.2688369

[94] J.-H. Lim, J.-H. Bae, J. Jang, H.-K. Jung, H. Lee, Y. Kim, B. Kim, J.-Y. Sim, and H.-J. Park, "A delay locked loop with a feedback edge combiner of duty-cycle corrector with a 20%–80% input duty cycle for SDRAMs," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 2, pp. 141–145, Feb. 2016. DOI: 10.1109/tcsii.2015.2468911

[95] Y.-M. Wang and J.-S. Wang, "An all-digital 50% duty-cycle corrector," *IEEE Int. Symp. Circuits and Syst. (ISCAS)*, 2004. DOI: 10.1109/ISCAS.2004.1329424

[96] H.-Y. Joo, S.-J. Bae, Y.-S. Sohn, Y.-S. Kim, K.-S. Ha, M.-S. Ahn, Y.-J. Kim, Y.-J. Kim, Y.-J. Kim, J.-H. Kim, W.-J. Choi, C.-H. Shin, S. H. Kim, B.-C. Kim, S.-B. Ko, K.-I. Park, S.-J. Jang, and G.-Y. Jin, "A 20nm 9Gb/s/pin 8Gb GDDR5 DRAM with an NBTI monitor, jitter reduction techniques and improved power distribution," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan. 2016. DOI: 10.1109/isscc.2016.7418033

[97] K.-T. Kang, S.-Y. Kim, S. J. Kim, D. Lee, S.-S. Yoo, and K.-Y. Lee, "A 0.33–1 GHz open-loop duty cycle corrector with digital falling edge modulator," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 12, pp. 1949–1953, Dec. 2018. DOI: 10.1109/tcsii.2018.2811412

[98] C.-C. Chung, D. Sheng, and S.-E. Shen, "High-resolution all-digital duty-cycle corrector in 65-nm CMOS technology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 5, pp. 1096–1105, May 2014. DOI: 10.1109/tvlsi.2013.2260186

[99] M. Kim, J.-H. Chae, S. Choi, G.-M. Hong, H. Ko, D.-K. Jeong, and S. Kim, "A 4266 Mb/s/pin LPDDR4 interface with an asynchronous feedback CTLE and an adaptive 3-step eye detection algorithm for memory controller," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 12, pp. 1894–1898, Dec. 2018. DOI: 10.1109/tcsii.2018.2819430

[100] I. Agilent Technologies, "Jitter analysis techniques for high data rates (application note 1432)," Tech. Rep., 2003.

List of Own Publications

[101] N. Angeli, O. Bachmann, and K. Hofmann, "A 2.5 GHz all-digital multiphase DLL and phase shifter in 65 nm CMOS using a scalable phase-to-digital converter," in *IEEE Int. Symp. Circuits and Syst. (ISCAS)*, May 2019. DOI: 10.1109/iscas.2019.8702157

[102] N. Angeli and K. Hofmann, "A low-power and area-efficient digitally controlled shunt-capacitor delay element for high-resolution delay lines," in *IEEE Int. Conf. Electron., Circuits and Syst. (ICECS)*, Dec. 2018. DOI: 10.1109/icecs.2018.8617948

[103] ——, "A scalable fully synthesized phase-to-digital converter for phase and duty-cycle measurement of high-speed clocks," in *IEEE Int. Symp. Circuits and Syst. (ISCAS)*, 2018. DOI: 10.1109/iscas.2018.8351118

[104] ——, "An all-digital duty-cycle corrector for parallel high-speed I/O links," in *IEEE Nordic Circuits and Syst. Conf. (NorCAS)*, 2019. DOI: 10.1109/NORCHIP.2019.8906953

[105] ——, "Low-power all-digital multiphase DLL design using a scalable phase-to-digital converter," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 4, pp. 1158–1168, Apr. 2020. DOI: 10.1109/tcsi.2019.2945086

[106] ——, "Schaltungsanordnung und Verfahren zum Ermitteln eines Versatzes zwischen zwei Signalflanken," patent DE 10 2017 109 192 A1, 2017.

[107] ——, "Fast digital clock calibration of a differential 6.4 Gb/s/pin bidirectional asymmetric memory interface," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2020. DOI: 10.1109/TCSII.2020.2964908

Supervised Theses

[108] P. D. Bhat, “Design of ALU for coarse grained reconfigurable computing,” Master Seminar, TU Darmstadt, 2015.

[109] R. Koch, “Analyse digital einstellbarer Delaylines zur Phasenanpassung und Implementierung in einer 65 nm CMOS Technologie,” Projektseminar, TU Darmstadt, 2016.

[110] N. Schmid and T. Leipe, “Entwicklung eines Kopfhörerverstärkers,” Projektseminar, TU Darmstadt, 2016.

[111] A. Prabhakaran, “Schematic level design of a digitally controlled phase interpolator in 65 nm CMOS technology,” Master Seminar, TU Darmstadt, 2016.

[112] R. Koch, “Design und Implementierung einer digitalen coarse-fine Delayline mit Regelschleife für die Phasenanpassung bei High-speed I/Os,” Bachelor Thesis, TU Darmstadt, 2017.

[113] N. Schmid, “High speed PCB design,” Proseminar, TU Darmstadt, 2017.

[114] A. Burkhardt, “Implementation of a phase-to-digital converter in an FPGA and performance analysis with respect to jitter,” Master Thesis, TU Darmstadt, 2017.

[115] R. M. Mertel, “Comparison and evaluation of different duty cycle correction architectures,” Proseminar, TU Darmstadt, 2017.

[116] N. Eberlein, “Implementation of an SPI interface between a test chip in a 65 nm CMOS technology and an FPGA,” Master Thesis, TU Darmstadt, 2017.

[117] R. M. Mertel, “Design of a digitally controlled duty cycle corrector,” Projektseminar, TU Darmstadt, 2017.

[118] K. Saary, “Receiver front-end design for mulit-gigabit transceiver,” Proseminar, TU Darmstadt, 2017.

[119] S. Himer, “Design und Implementierung eines 8 Gb/s MOS-only Source-Series Terminated Transmitters in einem 65 nm Prozess,” Master Thesis, TU Darmstadt, 2017.

[120] M. Wesp, “Overview of common time-to-digital converter architectures,” Proseminar, TU Darmstadt, 2018.

[121] ——, “Methods for calibration and PVT compensation in time-to-digital converters,” Projektseminar, TU Darmstadt, 2018.

[122] ——, “Design und Implementierung eines Replica-Based Biased CML Verstärkers zur Minimierung der Setup Time Variation eines differentiellen Highspeed Receivers,” Bachelor Thesis, TU Darmstadt, 2018.

[123] M. Reuter, “Microprocessor based adaptive interface training of a source-synchronous i/o link,” Master Thesis, TU Darmstadt, 2018.

[124] H. Korn, “Methoden zur Offset-Kalibrierung von getakteten Komparatoren,” Proseminar, TU Darmstadt, 2019.

[125] ——, “Offsetkompensation eines getakteten Komparators mittels schaltbarer Kondensatoren basierend auf Transmission Gates,” Projektseminar, TU Darmstadt, 2019.

Curriculum Vitae

Personal Data

Name	Nico Angeli
Date of birth	26. June 1989
Place of birth	Würzburg, Germany

Professional Experience

01/2015 — Present	Research and Teaching Staff *Integrated Electronics System Lab* Technische Universität Darmstadt, Germany
07/2015 — 12/2015	Intern *Display Engineering* Apple Inc., Cupertino, USA
01/2014 — 12/2014	External Graduate Research Assistant *Group Research and Advanced Engineering Displays* Daimler AG, Sindelfingen, Germany
2009 — 2013	Student Research Assistant *Institute of Microelectronics* Universität des Saarlandes, Saarbrücken, Germany

Education

01/2015 — Present	PhD, Electrical Engineering *TU Darmstadt, Integrated Electronic Systems Lab*, Darmstadt, Germany
10/2011 — 12/2013	Master of Science, Electrical Engineering (honored degree) *Universität des Saarlandes*, Saarbrücken, Germany
10/2008 — 09/2011	Bachelor of Science, Mechatronics *Universität des Saarlandes*, Saarbrücken, Germany
03/2008	Abitur *Hofenfels Gymnasium Zweibrücken*, Germany

Awards

05/2019	"Student Best Paper Award" at the IEEE International Symposium on Circuits and Systems (ISCAS) 2019
03/2014	"Best Degree in the Subject of Electrical Engineering in the Year 2013" by the Association for Electrical, Electronic and Information Technologies (VDE)