ON-CHIP JITTER MEASUREMENT AND MITIGATION TECHNIQUES FOR
CLOCK AND DATA RECOVERY CIRCUITS

by

Joshua Liang

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Electrical and Computer Engineering
University of Toronto

Copyright © 2017 by Joshua Liang
Abstract

On-Chip Jitter Measurement and Mitigation Techniques for Clock and Data Recovery Circuits

Joshua Liang

Doctor of Philosophy

Graduate Department of Electrical and Computer Engineering

University of Toronto

2017

This thesis describes three contributions in the area of on-chip jitter measurement and characterization, which can be used to help optimize the performance of wireline transceivers. Two on-chip jitter measurement techniques are developed and demonstrated, along with an adaptive loop gain CDR, which characterizes jitter on-chip to optimize its jitter tolerance.

In the first measurement technique, the absolute jitter of random data is measured on-chip by correlating the phase detector outputs of two 10Gb/s CDRs locked to the same data. This technique allows the jitter’s autocorrelation function to be estimated, from which the jitter’s RMS value and power spectral density are extracted without using any external reference clock. Correlating the phase detectors in the CDRs with a third phase detector, which measures the phase difference between the clocks recovered by the two CDRs, allows measurement of the absolute recovered clock jitter. A test chip fabricated in 65nm CMOS demonstrates that this scheme can measure RMS jitter with sub-picosecond accuracy.

To demonstrate the usefulness of on-chip jitter characterization in improving CDR performance, a loop gain adaptation strategy is proposed, which optimizes the jitter tolerance of a 28Gb/s PI-based CDR. The technique increases the CDR’s loop gain to suppress the most jitter while monitoring the autocorrelation function of the bang-bang PD output to prevent the CDR from becoming too underdamped. The proposed technique requires no prior knowledge of the CDR’s latency or jitter characteristics and can also be extended to operate in the presence of sinusoidal jitter. The concept is demonstrated in a test chip fabricated in 28nm CMOS.

Lastly, a second jitter measurement technique is proposed, which estimates the relative jitter between the input data and recovered clock of a 28Gb/s half-rate digital PI-based CDR
without using an eye monitor. Square wave jitter with a known amplitude is injected into the CDR, by adding a corresponding signal to the CDR’s PI code. By measuring the effect of the injected jitter on the autocorrelation function of the CDR’s bang-bang PD output, the RMS relative jitter is estimated with sub-picosecond accuracy. The scheme is demonstrated in the same 28nm test chip described above.
Acknowledgements

Completing a PhD thesis is indeed a journey, made possible by the support and advice of numerous people, and a healthy dose of God’s grace.

Firstly, I must thank my supervisor Prof. Ali Sheikholeslami for always pushing me to deepen my understanding of a topic, and to aim for better. This motivated me to dig deeper and to stretch myself and my ideas farther. Thanks also go to Hirotaka Tamura, for always making himself available to listen to our ideas and provide valuable comments. Thank you also to Masaya Kibune, Hisakatsu Yamaguchi and Nikola Nedovic from Fujitsu, who supported our tapeouts at various points. I would also like to thank those who served on committees for my proposal and thesis defense: Prof. Tony Chan Carusone, Sean Hum, Antonio Liscidini, Wai Tung Ng as well as Prof. Pavan Hanumolu, who served as external examiner. Their comments and feedback were valuable in improving this thesis.

I owe much to my peers throughout my studies. I learned a great deal from Ravi Shivnaraine, Cliff Ting and Sadegh Jalali who helped me get to grips with the world of wireline. The company of these gentlemen made the windowless Fuji room much more tolerable, even at the cost of often listening to the constant sound of Sadegh laughing while watching ten year old sitcoms. Special thanks also to Shayan Shahramian and Alireza Sharif-Bakhtiar who were always there to help. More recently, thanks to Wahid Rahman, Danny Yoo, and Masumi Shibata. Along with Alireza, we miraculously pulled off our first tapeout in 28nm and escaped with working chips! Thanks also to the many others in BA5000 who made days more entertaining and who could always commiserate and share in wondering when graduation would ever come.

I owe the deepest debt to my family, especially my parents who supported me throughout my PhD both materially and through their prayers and encouragement. They were quick to cast away doubts about whether this journey was worth it all and did whatever they could to ease my burdens. Last but certainly not least, I must thank my soon-to-be other half Joyce for her extreme patience, loving encouragement and prayers. Her lunch time visits brought light into some otherwise dreary days and quickly reminded me of life outside of school and the life to which I could look forward.
# Contents

1 Introduction  
   1.1 Motivation ........................................... 2  
   1.2 Thesis Outline ....................................... 3  

2 Background  
   2.1 Clock Recovery in Wireline Links ................. 6  
   2.2 Basic CDR Architectures ............................ 7  
   2.3 Jitter in CDRs ..................................... 18  
   2.4 Summary ........................................ 31  

3 On-chip Jitter Measurement for Multilane CDRs  
   3.1 Background ........................................ 33  
   3.2 Proposed Jitter Measurement Scheme ............. 38  
   3.3 Analysis of Jitter Measurement with Two CDRs .... 42  
   3.4 Implementation ..................................... 47  
   3.5 Measurement Results ................................. 52  
   3.6 Application Example: Jitter-Based Equalization  60  
   3.7 Summary ........................................ 63  

4 Adaptive Loop Gain CDR for Jitter Tolerance Optimization  
   4.1 Background ........................................ 66  
   4.2 Analysis of Jitter in PI-based CDR ............... 68  
   4.3 Finding Optimal Loop Gain .......................... 72
List of Tables

3.1 Limitations of existing jitter measurement techniques .................. 37
3.2 Comparison to Previous Jitter Measurement Circuits .................... 59
4.1 Table of Comparison .................................................. 98
# List of Figures

1.1 Forecast of total global monthly internet traffic [1] ............................................. 1
1.2 Simulated vs. measured jitter totals for a 25Gb/s transceiver [3] ................................. 3

2.1 Basic concept of a wireline link .................................................................................. 5
2.2 Concept of NRZ signalling .......................................................................................... 5
2.3 Clock recovery schemes in wireline transceivers: (a) global clock (b) source synchronous and (c) embedded clock ................................................................. 7
2.4 Definition of phase error $\phi_{ER}$ when recovered clock is (a) locked (b) late or (c) early ................................................................................................................................ 8
2.5 Hogge Linear PD ......................................................................................................... 9
2.6 Alexander Bang-Bang Linear PD ................................................................................ 10
2.7 Response of (a) linear and (b) bang-bang phase detectors ......................................... 10
2.8 Block diagram of conventional VCO-based CDR ......................................................... 11
2.9 Linear model of a conventional VCO-based CDR ....................................................... 12
2.10 Loop gain of VCO-based CDR .................................................................................. 12
2.11 $H_{\text{JTRAN}}(f)$ for a VCO-based CDR ....................................................................... 13
2.12 $H_{\text{JTOl}}(f)$ for a VCO-based CDR ............................................................................ 14
2.13 $H_{\text{JTOl}}^{-1}(f)$ for a VCO-based CDR .......................................................................... 15
2.14 Block diagram of a digital VCO-based CDR ............................................................. 15
2.15 Linear model of a digital VCO-based CDR ............................................................... 16
2.16 Basic (a) CML and (b) CMOS implementations of phase interpolators .................... 16
2.17 Concept of PI-based CDR ......................................................................................... 17
2.18 Linear model of PI-based CDR .................................................................................. 18
2.19 Concept of absolute clock jitter ........................................ 19
2.20 Concept of absolute data jitter ........................................ 19
2.21 Concept of relative jitter between data and clock jitter .......... 20
2.22 Concept of period jitter ................................................. 20
2.23 Transfer function between absolute and period jitter $S_N(f)/S(f)$ . . . . 21
2.24 Data jitter caused by transmit clock jitter ........................... 22
2.25 (a) Data pattern with ideal and lossy channel and (b) corresponding jitter caused by ISI .................................................. 23
2.26 Relationship between excess phase and jitter for a sine wave: (a) sine wave with excess phase $\phi(t)$ (b) excess phase $\phi(t)$ (c) corresponding clock waveform and (d) jitter $\psi_k$ .................................................. 24
2.27 Phase noise vs. $\omega$ (measured as offset from carrier frequency) according to Lee-sen’s equation .......................................... 25
2.28 (a) Interpolated PI output when I and Q are triangular waves (b) and corresponding constellation of I and Q weights ........................................ 26
2.29 (a) Interpolated PI output when I and Q are sinusoidal waveforms (b) and output phase delay vs. code for PIs using triangular and sinusoidal input waveforms . . 26
2.30 (a) Expected BB-PD output and (b) PDF of input jitter as a function of input jitter ............................................................... 28
2.31 (a) Actual vs. linearized PD output and corresponding (b) error at PD output . 29
2.32 (a) Functional and (b) linearized model of bang-bang PD ................... 29
2.33 Linear model of analog VCO-based CDR showing major sources of jitter . . . 29
2.34 Linear model of PI-based CDR showing major sources of jitter ............ 31
3.1 Conventional (a) TDC-based (b) self-referenced and (c) PD-based jitter measurement techniques ............................................. 34
3.2 Concept of time-to-digital conversion by (a) oversampling or (b) sampling delayed versions of clock .......................................... 35
3.3 (a) Delay-line based and (b) Vernier delay-line based time-to-digital converters . 35
3.4 (a) Implementation and (b) operating concept of on-chip eye monitor ....... 37
3.28 Mean of edge jitter PDF for different data patterns ............................................. 60
3.29 Measured RMS data jitter for different patterns using on-chip measurement ... 61
3.30 Measured RMS data and CDR1 clock jitter (using on-chip measurement) and
    jitter tolerance vs. CTLE pattern ................................................................. 62
3.31 Pattern-filtering based CTLE adaptation ...................................................... 62
3.32 Pattern jitter-based CTLE adaptation curve ................................................. 62
4.1 (a) Conventional bang-bang CDR and proposed adaptive loop gain CDR and the
    impact of adaptation on jitter tolerance when jitter is (b) too small or (c) too large 65
4.2 Existing concepts for CDRs with adaptive loop filters using (a) direct jitter mea-
    surement (b) Kalman filter theory (c) estimation of jitter bandwidth .............. 66
4.3 (a) Jitter profile and (b) PI-based CDR assumed in [40,41] ......................... 67
4.4 Use of LPF to estimate bandwidth of jitter ................................................. 68
4.5 Jitter model of PI-based CDR ................................................................. 70
4.6 $S_{ER}(f)$ found through simulation of nonlinear vs linear models, and direct cal-
    culation using (4.9) ................................................................. 72
4.7 Response of (a) $H^{-1}_{JTOL}(f)$ and (b) $H_{JTRAN}(f)$ as $K_G$ is increased .... 73
4.8 Calculated contributions to $\sigma_{ER}$ from each jitter source for Case I as CDR loop
    gain is varied ................................................................. 74
4.9 Calculated contributions to $\sigma_{ER}$ from each jitter source for Case II as CDR loop
    gain is varied ................................................................. 74
4.10 Jitter and phase margin vs. unity gain frequency for Case II ..................... 75
4.11 (a) Normalized impulse response and (b) ideal jitter tolerance of CDR for differ-
    ent phase margins ................................................................. 76
4.12 (a) Spectrum and (b) corresponding autocorrelation function of BB-PD for var-
    ious loop gain settings showing peaking caused by excessive loop gain ......... 77
4.13 Overview of proposed adaptive loop gain CDR ........................................ 77
4.14 $R(n)$ for different $K_G$ values measured using (a) raw and (b) lowpass filtered
    BB-PD output in the presence of high white random jitter and (c) corresponding
    values of $R(n_{peak})$ ................................................................. 79
4.15 Simplified linear model of PI-based CDR for estimating $n_{peak}$

4.16 (a) Concept behind adaptation of $n_{peak}$ and (b) feedback loop used to identify $n_{peak}/2$

4.17 Autocorrelation of (a) single-tone SJ, (b) two-tone SJ with equal amplitudes and $f_1 \approx f_2$ and (c) two-tone SJ with equal amplitudes and $f_1 >> f_2$

4.18 Block diagram of adaptive loop gain CDR

4.19 CTLE with active feedback and inverter-based second stage

4.20 Phase interpolator

4.21 Block diagram of digital loop filter

4.22 Digital implementation of $R(n)$ measurement

4.23 Die photo with area and power breakdowns. Total power is measured while percentage breakdown is based on simulation

4.24 Wire-bond between chip and PCB

4.25 Test setup

4.26 (a) Recovered quarter-rate data and (b) half-rate clock with CDR locked to 28Gb/s PRBS31 data

4.27 (a) Adaptation of $n_{peak}$ while $K_G$ set to maximum and (b) $R(n)$ plotted for same condition showing that $n_{peak}$ adapts to the correct value.

4.28 (a) Phase noise profiles of Ref Ck, (b) jitter tolerance for adapted, min and max $K_G$ (c) minimum jitter tolerance measured between 10-100MHz after adaptation and (d) recovered clock jitter vs. $K_G$ for three test cases.

4.29 Measured $K_G$ adaptation curves for all three test cases

4.30 Measured $R(n)$ with and without lowpass filtering of PD output

4.31 Maximum SJ the CDR can tolerate; comparing max over all $K_G$ settings, to basic adaptation ($K_G$ adapted and fixed before applying SJ) and dynamic adaptation ($K_G$ dynamically adapts to input jitter).

5.1 (a) Conventional eye monitor-based vs (b) proposed autocorrelation-based jitter measurement
5.2  (a) CDR model showing estimation of \( R(n) \) with injected jitter (b) \( R(n) \) for white jitter, injected square wave jitter and combination of both ........................................ 102
5.3  Overview of Half-Rate CDR from chapter 3 showing added jitter injection function ................................................................................................................................. 104
5.4  (a) Linear model and (b) simplified linear model of BB-PD followed by majority voting ........................................................................................................................................ 105
5.5  Normalized frequency response of majority voting ...................................................................................................................................................... 105
5.6  \( R(n) \) measured with and without jitter injected. .............................................................................................................................................................. 106
5.7  Estimated relative CK vs. Data jitter at CDR input as CDR loop gain setting is swept for two test cases. ........................................................................................................ 107
5.8  Estimated relative CK vs. Data jitter as (a) data jitter is swept and (b) as the CTLE code is swept, with corresponding jitter tolerance at 100MHz. .......... 108
5.9  Jitter tolerance (BER<10^{-12}, PRBS31) with and without jitter injection enabled .......................................................... 109
A.1  (a) Linear model and (b) actual nonlinear model of two bang-bang PD outputs being correlated .................................................................................................................. 116
A.2  \( \sqrt{E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})]/(K_{P1}K_{P2})} \) as a function of \( \sigma_{\psi_A} \) using numerical integration. \( \psi_B \) and \( \psi_C \) are Gaussian with \( \sigma_{\psi_B}=\sigma_{\psi_C}=1 \) .................................................................................................................. 117
A.3  Contour plot of \( f_{\psi_{ER1}\psi_{ER2}}(\psi_{ER1},\psi_{ER2}) \) for Gaussian \( \psi_B \) and \( \psi_C \) with \( \sigma_{\psi_B}=\sigma_{\psi_C}=1 \).
(a) \( \psi_A = 0 \), (b) Gaussian \( \psi_A \) with \( \sigma_{\psi_A} = 1 \), (c) Uniformly distributed \( \psi_A \) with \( \sigma_{\psi_A} = 1 \), (d) Sinusoidal \( \psi_A \) with \( \sigma_{\psi_A} = 1 \). .................................................................................................................. 118
A.4  \( \sqrt{E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})]/(K_{P1}K_{P2})} \) as a function of \( \sigma_{\psi_A} \) using Matlab simulation ........................................ 119
List of Abbreviations

$\alpha_T$  Transition Density

$\mathcal{F}\{\cdot\}$  Fourier Transform

$\sigma$  Standard Deviation

$E[\cdot]$  Expected Value

ADC  Analog-to-Digital Converter

BB-PD  Bang-Bang Phase Detector

BER  Bit Error Rate

BERT  Bit Error Rate Tester

BW  Bandwidth

CDF  Cumulative Distribution Function

CDR  Clock and Data Recovery

CP  Charge Pump

CPU  Central Processing Unit

CTLE  Continuous Time Linear Equalizer

D2S  Differential to Single-Ended

DCD  Duty Cycle Distortion
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCO</td>
<td>Digitally Controlled Oscillator</td>
</tr>
<tr>
<td>DeMUX</td>
<td>Demultiplexer</td>
</tr>
<tr>
<td>DFF</td>
<td>D-Flip Flop</td>
</tr>
<tr>
<td>DJ</td>
<td>Deterministic Jitter</td>
</tr>
<tr>
<td>DLL</td>
<td>Delay-Locked Loop</td>
</tr>
<tr>
<td>DMUX</td>
<td>Demultiplexer</td>
</tr>
<tr>
<td>EB</td>
<td>Exabyte ((10^{18} \text{ Bytes}))</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out</td>
</tr>
<tr>
<td>FM</td>
<td>Frequency Modulation</td>
</tr>
<tr>
<td>GB</td>
<td>Gigabyte</td>
</tr>
<tr>
<td>Gb/s</td>
<td>Gigabits per second</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>ISI</td>
<td>Intersymbol Interference</td>
</tr>
<tr>
<td>JTOL</td>
<td>Jitter Tolerance</td>
</tr>
<tr>
<td>JTRAN</td>
<td>Jitter Transfer</td>
</tr>
<tr>
<td>LF</td>
<td>Loop Filter</td>
</tr>
<tr>
<td>LMS</td>
<td>Least Mean Squares</td>
</tr>
<tr>
<td>LPF</td>
<td>Lowpass Filter</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Bit</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>MV</td>
<td>Majority Voting</td>
</tr>
<tr>
<td>NRZ</td>
<td>Non-Return-to-Zero</td>
</tr>
<tr>
<td>PAM</td>
<td>Pulse Amplitude Modulation</td>
</tr>
<tr>
<td>PD</td>
<td>Phase Detector</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PI</td>
<td>Phase Interpolator</td>
</tr>
<tr>
<td>PLL</td>
<td>Phase-Locked Loop</td>
</tr>
<tr>
<td>PM</td>
<td>Phase Modulation</td>
</tr>
<tr>
<td>ppm</td>
<td>Parts Per Million</td>
</tr>
<tr>
<td>PRBS</td>
<td>Pseudorandom Binary Sequence</td>
</tr>
<tr>
<td>PSD</td>
<td>Power Spectral Density</td>
</tr>
<tr>
<td>PVT</td>
<td>Process, Voltage, and Temperature</td>
</tr>
<tr>
<td>RJ</td>
<td>Random Jitter</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Squared</td>
</tr>
<tr>
<td>Rx</td>
<td>Receiver</td>
</tr>
<tr>
<td>SJ</td>
<td>Sinusoidal Jitter</td>
</tr>
<tr>
<td>SSC</td>
<td>Spread-Spectrum Clocking</td>
</tr>
<tr>
<td>TDC</td>
<td>Time-to-Digital Converter</td>
</tr>
<tr>
<td>Tx</td>
<td>Transmitter</td>
</tr>
<tr>
<td>UI</td>
<td>Unit Interval</td>
</tr>
<tr>
<td>$\text{UI}_{PP}$</td>
<td>Unit Interval (Peak-to-Peak)</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>VCO</td>
<td>Voltage Controlled Oscillator</td>
</tr>
<tr>
<td>WSS</td>
<td>Wide-Sense Stationary</td>
</tr>
<tr>
<td>XOR</td>
<td>Exclusive Or</td>
</tr>
<tr>
<td>ZOH</td>
<td>Zero-Order Hold</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Global internet traffic continues to increase, and is expected to grow by nearly three times between 2015 and 2020 [1], fuelled by new applications such as the internet of things (IoT) and growth in bandwidth-intensive services such as streaming video, which already accounted for 70% of all internet traffic in 2015 [1]. This trend is shown in Fig. 1.1, which plots the monthly worldwide internet traffic forecast by Cisco [1].

Figure 1.1: Forecast of total global monthly internet traffic [1]

Such explosive growth is made possible by the expansion of network infrastructure and data centres which provide exabytes (EB=10^9 GB) of content to users every month. Before making
it to the user, data must pass through multiple wireline transceivers which transmit serial data between devices over a variety of channels. At the larger scale, the channel could be the long copper cables between servers and switches in large data centres. At the smaller scale, the channel could consist of short traces on a silicon interposer or package substrate, connecting multiple chips integrated in a single package.

Increasing bandwidth requirements has caused the data rates of these transceivers to nearly double every 4 years [2]. Increasing channel losses and slowing progress in device scaling are among the challenges of further improving data rates. Another difficulty comes from timing jitter. As data rates increase, the symbol period decreases, leaving less margin for jitter. The topic of this thesis is jitter in clock and data recovery (CDR) circuits, and how it can be monitored on-chip, with the goal of mitigating its impact.

1.1 Motivation

Jitter arises from many sources. While some sources such as oscillator phase noise are relatively well understood, jitter can also arise from power supply noise, crosstalk or other coupling effects, which can be more difficult to accurately model and simulate. Even well-studied jitter sources can vary over process, voltage, and temperature (PVT), making circuit performance unpredictable. This presents a major challenge to circuit designers.

Given certain specifications, designers must typically perform some form of jitter budgeting, setting the maximum jitter that each circuit block can generate. Unfortunately, the measurement results can often differ from what was simulated as illustrated in Fig. 1.2, which compares the simulated and measured jitter totals for a 25Gb/s transceiver [3]. As seen in the figure, several jitter sources increased significantly in measurements compared to simulations, decreasing the available timing margin. In fact some sources of jitter were not anticipated at all. Furthermore, since not all jitter sources could be measured in [3], some could only be estimated from other data.

On-chip jitter measurement can help bridge the gap between simulations and measurements, helping to diagnose performance issues, and assisting designers by providing a better understanding of jitter’s impact on existing designs, and thereby helping to improve future designs. It
can also be used to improve system reliability by monitoring for device failures or performance degradation caused by aging effects. Perhaps even more attractive, is the potential to use on-chip measurement in helping reduce jitter’s adverse effects, by adapting circuit parameters on-chip.

This thesis describes two schemes for monitoring jitter in CDRs on-chip. To demonstrate how jitter characterization can be used to improve performance, a digital CDR is also developed whose loop gain automatically adapts to optimize its jitter tolerance.

1.2 Thesis Outline

The remainder of this thesis is organized as follows.

- Chapter 2 provides general background, reviewing basic clock and data recovery techniques and jitter concepts.

- Chapter 3 provides background on existing jitter measurement approaches before describing a jitter measurement technique for multilane voltage controlled oscillator (VCO)-based CDRs. The technique is capable of measuring the RMS value and power spectral density (PSD) of the absolute data and clock jitter of a CDR, without relying on any clean external reference clock. Separately measuring clock and data jitter provides insight into CDR operation, allowing jitter from the data and clock paths to be separately characterized. Measurement results from a test chip fabricated in 65nm CMOS are presented to validate the concept.
Chapter 1. Introduction

- Chapter 4 gives a brief overview of existing CDR loop gain adaptation approaches before presenting a digital CDR whose loop gain automatically adapts to optimize jitter tolerance. The proposed scheme ensures that jitter tolerance remains optimal as the CDR is subjected to different jitter profiles, and does not rely on any prior knowledge of the jitter’s characteristics. Results from a test chip, implemented in 28nm CMOS are presented.

- Chapter 5 introduces a method for estimating the RMS value of the relative jitter between the data and clock in a phase interpolator (PI)-based digital CDR without using any dedicated eye monitor, or other jitter measurement circuits. By providing observability, the proposed technique can be used to help minimize the relative jitter of the CDR. Measurement results from the 28nm test chip demonstrate the effectiveness of the jitter measurement approach.

- Chapter 6 summarizes the results and contributions of the thesis as well as identifying possible areas for future research.
Chapter 2

Background

This chapter introduces the basic concepts of clock recovery and jitter in wireline links. A typical wireline link consists of a transmitter (Tx) sending a non return-to-zero (NRZ) signal over some channel to a receiver (Rx) as shown in Fig. 2.1. NRZ signalling consists of two-level (binary) pulse amplitude modulated (PAM) signalling, where each bit is sent as a pulse as shown in Fig. 2.2. The duration of each bit ($T$) is the inverse of the data rate and is also referred to as the unit interval (UI).

![Figure 2.1: Basic concept of a wireline link](image)

![Figure 2.2: Concept of NRZ signalling](image)

The channel can range from hundreds of metres of copper cables to millimetre-long traces within a single chip. At high data rates, the frequency response of the channel is typically insufficient to allow data transmission without introducing intersymbol interference (ISI) into
the transmitted signal. The Tx and Rx therefore may also include equalizer circuits, which compensate for ISI. Equalization is a topic of much research but is not central to this thesis. Instead, this thesis focuses on clock recovery, which ensures that the Rx clock $CK_{RX}$ samples the incoming data in the optimum position.

Section 2.1 of this chapter provides an overview of clock distribution and recovery concepts. Section 2.2 then describes two classes of clock and data recovery circuits (CDR) used in this thesis: voltage controlled oscillator (VCO)-based and phase interpolator (PI)-based CDRs. Lastly, Section 2.3 reviews jitter terminology and the major sources of jitter in CDRs.

2.1 Clock Recovery in Wireline Links

2.1.1 Clocking Architectures

The clock recovery method used depends on how clocks are distributed to the transmitters and receivers in a system. Clock distribution methods fall into three broad categories: global clock, source synchronous and embedded clock [4] as illustrated in Fig. 2.3.

In the global clock scheme shown in Fig. 2.3(a), the same clock is distributed to both the Rx and Tx. Although seemingly simple, the global clock scheme can be difficult to implement as the Tx and Rx are often separated by large distances. More common is the source synchronous scheme shown in Fig. 2.3(b), where the Tx clock is transmitted along with the data using another channel. In theory, the clock and data experience the same delay and can be sent over long distances.

In practice however, some phase mismatch will always exist between the Tx and Rx clocks in global clock and source synchronous systems. Some form of clock recovery will therefore be required to correct the phase mismatch between the receiver’s input data and clock. These schemes are referred to as mesochronous [5], since the Rx only needs to recover the phase of the clock and not its frequency, leading to comparatively simple clock recovery circuits.

This thesis deals instead with embedded clock systems as shown in Fig. 2.3(c) (also referred to as plesiochronous [5]), where the Rx clock is recovered from the received data itself. This means that both the phase and frequency of the Tx clock must be recovered, which is accomplished using a CDR.
2.2 Basic CDR Architectures

In a conventional full-rate CDR, the received data is sampled with a clock whose period is equal 1UI. The data is ideally sampled in the centre of the UI by the rising edge of the clock, generated by the CDR. When this happens, the CDR is said to be locked. The phase difference in between the data and clock is denoted as $\phi_{ER}$ and is defined as zero when the CDR is locked as shown in Fig. 2.4. When the clock samples the data after the desired sample point, $\phi_{ER}$ is positive and we say that the clock is late. When the clock samples the data before the desired point, we say that the clock is early. $\phi_{ER}$ can have units of radians, seconds or UI.

If a full-rate clock is not available, multiple phases of a lower-frequency clock can also be
Figure 2.4: Definition of phase error $\phi_{ER}$ when recovered clock is (a) locked (b) late or (c) early used to sample the data at the required rate. In this thesis, we use half-rate CDRs, which sample the data on both the rising and falling edge of a clock whose period is equal to 2UI.

To achieve and maintain lock, CDRs use a phase detector (PD) to detect the phase error between the incoming data and the recovered clock, which is generated by a VCO or PI. The PD output is then filtered and used to control the VCO or PI in a feedback loop, forcing the phase of the recovered clock to align to that of the incoming data.

### 2.2.1 Linear and Bang-Bang Phase Detectors

Two types of PDs are commonly used: linear and bang-bang. As its name suggests, the output of a linear PD is linearly proportional to the phase error seen at its input. An example of a linear PD is the Hogge PD [6] shown in Fig. 2.5, which contains two flip-flops clocked by a full-rate clock and has two outputs, $ERR$ and $REF$. The bottom XOR gate generates a positive $REF$ pulse half a bit period wide each time the input data transitions. The second XOR gate, produces an $ERR$ pulse whose width is proportional to $\phi_{ER}$. The PD output is taken as the average of $ERR - REF$. When the rising edge of the clock is aligned to the centre of the data, the widths of the $REF$ and $ERR$ pulses are equal and the PD output is zero.
Linear PDs have several disadvantages. Firstly, the clock-to-Q delay of the flip-flops can cause the PD to output zero when $\phi_{ER}$ is not exactly zero, causing the CDR to lock with some residual phase offset. Secondly, at higher data rates, the input data transitions are not sharp, degrading the performance of the XOR gate, which may also have insufficient bandwidth to output the required short pulses. For these reasons, bang-bang PDs are more popular.

Bang-bang PDs (BB-PD) only output the sign of the phase error, and not its magnitude. An example is the Alexander PD [7] shown in Fig. 2.6, which samples the data on both the rising and falling edges of the full-rate clock to produce both data samples ($D_n$) near the centre of the UI, and edge samples ($E_n$) near the edge of the UI.

When locked, the edge samples should fall exactly on the transitions in the data. By comparing the edge samples to the adjacent data samples, the sign of $\phi_{ER}$ can be determined as shown in Fig. 2.6(b). Accordingly, the PD outputs a 1UI-wide LATE or EARLY pulse for each data transition detected.

By only operating on the sampled data, the bandwidth requirements of the XOR gate are relaxed. The clock-to-Q delays of the flip-flops are also cancelled out. By only determining the sign of $\phi_{ER}$, the bang-bang PD provides less information than the linear PD as shown in Fig.
2.7, but since its output is also essentially digital, it is suitable for digital filtering as will be discussed later in this chapter. For this reason, we use bang-bang PDs throughout this thesis, using digital processing to characterize their outputs and extract information on the jitter of CDRs. Although it is nonlinear, the bang-bang PD can be linearized, as will be discussed in section 2.3.2.

Having measured $\phi_{ER}$ with some form of PD, the CDR must then generate a clock such that $\phi_{ER} = 0$. The first CDR topology we discuss is the VCO-based CDR.
2.2.2 VCO-Based CDR

In a conventional VCO-based CDR, the phase detector output is converted to a current using a charge pump (CP) and integrated using a loop filter (LF) as shown in Fig. 2.8. The filtered output then drives the input of a voltage controlled oscillator, whose frequency is controlled by its input voltage.

![Block diagram of conventional VCO-based CDR](image)

Figure 2.8: Block diagram of conventional VCO-based CDR

A linear phase domain model of the VCO-based CDR is shown in Fig. 2.9. The phase detector measures $\phi_{ER}$ and multiplies it by its gain $K_{PD}$. The charge pump is modelled as a gain while the loop filter response is given by $LF(s)$. Since frequency is the derivative of phase, the VCO, whose frequency depends on its input voltage acts as an integrator in the phase domain, with some gain $K_{VCO}$.

If the charge pump current only drove a single capacitor $C_S$, the feedback loop would contain two integrations, making it unstable. The series resistor $R_S$ in the loop filter provides a stabilizing zero, while a parallel capacitor $C_P$ is added to suppress ripple on the loop filter output voltage $V_{CTRL}$ [8]. In the CDR shown in Fig. 2.8, $LF(s)$ is given by:

$$LF(s) = \frac{1 + sR_SC_S}{s(C_S + C_P)\left(1 + s\left(\frac{C_PC_S}{C_P + C_S}R_S\right)\right)} \quad (2.1)$$

The loop gain of the CDR $LG(s)$ is given by

$$LG(s) = \frac{K_{PD}I_CP_K_{VCO}(1 + sR_SC_S)}{s(C_S + C_P)\left(1 + s\left(\frac{C_SC_P}{C_S + C_P}R_S\right)\right)} \quad (2.2)$$
Chapter 2. Background

Figure 2.9: Linear model of a conventional VCO-based CDR

$LG(s)$ contains two poles at DC, a zero caused by $R_S$, and a third pole caused by $C_P$ as can be seen from the Bode plot in Fig. 2.10. The zero frequency is $f_Z = (2\pi R_S C_S)^{-1}$, and frequency of the third pole is $f_P = \left(2\pi R_S \left(\frac{C_S C_P}{C_S + C_P}\right)\right)^{-1}$.

Figure 2.10: Loop gain of VCO-based CDR

2.2.3 CDR Transfer Characteristics

Several characteristics of the CDR are of interest. Firstly, jitter transfer ($JTRAN$) describes how phase error at the CDR input propagates to the CDR output. We analyze it using $H_{JTRAN}(f)$, the transfer function from $\phi_{DAT}$ to $\phi_{CK}$. As given by the expression below, $H_{JTRAN}(f)$ has a lowpass response as shown in Fig. 2.11. This is because at low frequencies, the CDR drives $\phi_{ER}$ to zero by forcing $\phi_{CK}$ to track $\phi_{DAT}$.
Chapter 2. Background

\[
H_{\text{JTRAN}}(f) = \frac{\phi_{CK}(f)}{\phi_{\text{DAT}}(f)} = \frac{LG(f)}{1 + LG(f)}
\]  

(2.3)

Figure 2.11: \(H_{\text{JTRAN}}(f)\) for a VCO-based CDR

As will be discussed in section 2.3, jitter causes \(\phi_{\text{DAT}}\) to vary over time. Jitter tolerance (JTOL) measures how much \(\phi_{\text{DAT}}\) can vary before bit errors occur in the CDR. It is found by assuming that \(\phi_{\text{DAT}}\) is a sinusoid at frequency \(f\), and finding the largest amplitude \(\phi_{\text{DAT}}\) can have before bit errors occur, as a function of \(f\). In the absence of jitter, errors occur if \(\phi_{\text{ER}}\) exceeds \(\pm 0.5\text{UI}\) (1UI_{PP}), meaning that the data will be sampled outside of the bit period. To find jitter tolerance, we first find the transfer function between \(\phi_{\text{ER}}\) and \(\phi_{\text{DAT}}\), which we define as \(H_{\text{JTOL}}(f)\).

\[
H_{\text{JTOL}}(f) = \frac{\phi_{\text{DAT}}(f)}{\phi_{\text{ER}}(f)} = 1 + LG(f)
\]  

(2.4)

\(H_{\text{JTOL}}(f)\) is plotted in Fig. 2.12 and represents the relationship between \(\phi_{\text{DAT}}\) and \(\phi_{\text{ER}}\). At low frequencies, the CDR tracks \(\phi_{\text{DAT}}\) so that \(\phi_{\text{DAT}}\) becomes large compared to \(\phi_{\text{ER}}\). At high frequencies, the CDR no longer tracks \(\phi_{\text{DAT}}\) and \(\phi_{\text{DAT}}/\phi_{\text{ER}}\) becomes 1. Jitter tolerance (JTOL) is then found as the amplitude of \(\phi_{\text{DAT}}\) corresponding to the case when \(\phi_{\text{ER}} = 1\text{UI}_{\text{PP}}\).

\[
JTOL(f) = H_{\text{JTOL}}(f) \times 1\text{UI}_{\text{PP}}
\]  

(2.5)
Any jitter that is present reduces the timing margin of the CDR and lowers jitter tolerance. In this thesis, we rely on jitter tolerance as the primary metric to assess CDR performance.

Of note is the relationship between \( H_{JTOL}(f) \) and \( H_{JTOL}(f) \). We would like \( H_{JTOL}(f) \) to have a wide bandwidth, allowing the CDR to track the input data phase over a wide bandwidth. However, increasing the corner frequency of \( H_{JTOL}(f) \) \( (f_{3dB,CL}) \) also increases the bandwidth of \( H_{JTOL}(f) \), which has the same corner frequency. This means that phase error will also be transferred from the input data to the recovered clock over a wider bandwidth. This may not be desirable since any phase deviations present in the data will also corrupt the phase of the recovered clock, which could be used to drive other circuits. To break the coupling between \( H_{JTOL}(f) \) and \( H_{JTOL}(f) \), alternative topologies such as the D/PLL can be used \[9\].

![Figure 2.12: \( H_{JTOL}(f) \) for a VCO-based CDR](image)

Lastly, we examine the transfer function from \( \phi_{DAT} \) to \( \phi_{ER} \), which is the inverse of \( H_{JTOL}(f) \).

\[
H_{JTOL}^{-1}(f) = \frac{\phi_{ER}(f)}{\phi_{DAT}(f)} = \frac{1}{1 + LG(f)}
\] (2.6)

At lower frequencies, \( \phi_{CK} \) tracks \( \phi_{DAT} \) so that \( \phi_{DAT} \) does not propagate to \( \phi_{ER} \). This gives \( H_{JTOL}^{-1}(f) \) a highpass characteristic as seen in Fig. 2.13. We will see in section 2.3.3 that \( H_{JTOL}^{-1}(f) \) is also the transfer function from any jitter in the VCO to the output of the CDR.

### 2.2.4 Digital VCO-Based CDR

To avoid the large area and the process, voltage, and temperature (PVT)-sensitivity of passive loop filter components and analog charge pump circuitry, the charge pump and loop filter can be
Chapter 2. Background

Figure 2.13: $H_{JTOL}^{-1}(f)$ for a VCO-based CDR

replaced with a digital loop filter, which controls a digitally controlled oscillator (DCO). Unlike the analog CDR, which can use both linear and bang-bang PDs, digital CDRs rely on digital output of the bang-bang PD. The resulting DCO-based digital CDR is shown in Fig. 2.14. In the digital loop filter, the main loop filter capacitor $C_S$ is replaced by a digital integrator with gain $K_I$ while the zero created by the resistor $R_S$ is replaced by a proportional path gain $K_P$. Since there is no concept of voltage ripple in the digital domain, $C_P$ is not required.

Figure 2.14: Block diagram of a digital VCO-based CDR

The linear model for the digital CDR is shown in Fig. 2.15, where particular attention must be paid to the conversion between continuous and discrete time domains. In the context of VCO-based CDRs, we have treated $\phi$ as a continuous time variable for simplicity, since VCOs and RC circuits are more easily described in continuous time. In reality, phase detectors operate based on discrete samples, providing an output at each transition in the data. In the next section, $\phi$ will be more accurately described in discrete time in the context of PI-based CDRs. In section 2.3, we will also replace $\phi$ with jitter, which is a discrete time variable.
2.2.5 PI-Based CDR

Instead of generating the recovered clock using a VCO, PI-based CDRs align the phase of a reference clock to that of the received data using a phase interpolator. PI-based CDRs are common in multilane systems where a single reference clock can be shared among several CDRs.

Phase interpolators mix clocks with different phases to produce a clock with an adjustable delay. Two simple phase interpolator circuits are shown in Fig. 2.16. The circuits combine $CK_1$ and $CK_2$ with different weights by adjusting the current or drive strength of each signal. If the delay between $CK_1$ and $CK_2$ is $\Delta T$, then the phase of the output clock can be adjusted by up to $\Delta T$. We make use of this in chapter 3, using a phase interpolator to implement a variable delay. By interpolating between multiple different clock phases, phase interpolators can achieve the full $2\pi$ phase rotation of the clock, needed in PI-based CDRs. This will be described in more detail in section 2.3.2.

A typical PI-based CDR is shown in Fig. 2.17. Similar to the digital VCO-based CDR, a
bang-bang PD is used to drive a digital loop filter. Since the PI only adds and subtracts phase, a digital phase accumulator must be added to mimic the phase accumulation property of a VCO. Without the phase accumulator, the CDR would contain only one integrator, reducing the system from second to first-order.

While a first-order system can work, it cannot track both input phase steps and ramps. Ramps must be tracked when spread-spectrum clocking (SSC), which modulates the clock frequency to reduce electromagnetic emissions, is used, or whenever frequency offset is present between the Rx and Tx reference clocks. Some frequency offset will typically be present in embedded clock systems since the Rx and Tx often rely on different reference clocks, provided for example by different crystal oscillators, each having some mismatch.

![Figure 2.17: Concept of PI-based CDR](image)

The linear model for the PI-based CDR is shown in Fig. 2.18. The model is in discrete time, owing to the digital nature of the CDR. The loop gain is given by

$$LG(z^{-1}) = \frac{K_{PD} K_{PA} LF(z^{-1})}{1 - z^{-1}}.$$  \hspace{1cm} (2.7)

Similar to the case of the DCO-based CDR, the loop filter is typically has the response

$$LF(z^{-1}) = K_P + \frac{K_I}{1 - z^{-1}}.$$  \hspace{1cm} (2.8)

$H_{JTRAN}$ and $H_{JTOL}$ can be calculated analogously to the case of the VCO-based CDR.
2.3 Jitter in CDRs

Next we discuss the central topic of this thesis: timing jitter. The above discussion has treated the phase $\phi$ as a deterministic signal. In general, the phase error may instead be caused by timing noise referred to as jitter and denoted here as $\psi$. Jitter is a discrete time signal describing the timing error at each transition or edge of a data or clock signal in units of seconds or UI. When jitter becomes too large, the received data can be sampled incorrectly, leading to bit errors. In this thesis we distinguish between three different types of jitter: absolute jitter, relative jitter and period jitter.

2.3.1 Basic Jitter Definitions

Absolute Jitter

As shown in Fig. 2.19, the absolute jitter of a clock $\psi_{CK}$ is the timing error between the transitions of a clock and those of an ideal clock at the same frequency $f = 1/T$. In Fig. 2.19 jitter is only measured on the rising edges of the clock, but in general, the jitter of both edges may be of interest.

Similarly, the absolute jitter of an NRZ data signal $\psi_{DATA}$ is the timing error between its transitions and those of an ideal data stream, where the width of each bit is $T$ as shown in Fig. 2.20. Since jitter only has meaning when there is a transition in the data, we define jitter to be zero whenever there is no data transition. In chapter 3, we describe a method to estimate the absolute jitter of a CDR’s clock and data.
Ideal Ck

Ck with Jitter

\[ \psi_{CK} \]

Figure 2.19: Concept of absolute clock jitter

Ideal Ck

Ideal Data

Data with Jitter

\[ \psi_{DATA} \]

Figure 2.20: Concept of absolute data jitter

**Relative Jitter**

Relative jitter describes the timing error between two non-ideal signals. As an example, in Fig. 2.21, the relative jitter \( \psi_{DATA-CK} \) describes the timing error between a clock and data signal, each having their own absolute jitter. Because only the relative timing difference is used, relative jitter can be either higher or lower than the absolute jitters of the data and clock, depending on whether their absolute jitter adds together or cancels out. In chapter 5, we propose a technique for measuring the relative jitter between the data and clock in a CDR.
Chapter 2. Background

20

Data with Jitter

No Edge

Ck with Jitter

Figure 2.21: Concept of relative jitter between data and clock jitter

Period Jitter

Lastly period jitter, which we denote as $\varphi$, is the deviation between the width of each clock period $T_k$ from its ideal value $T$. As described in Fig. 2.22, because the period of the clock is affected by the absolute jitter of both adjacent edges, period jitter is the first difference of absolute jitter [10].

Period Jitter: $\varphi_{CK,k} = \psi_{CK,k} - \psi_{CK,k-1}$ (2.9)

The concept of period jitter can also be extended to $N$-period jitter, where the period is
measured over $N$ cycles of the clock instead of only one. Since jitter is generally a random process, it can be described in terms of its power spectral density (PSD). The PSD of \textit{N-period jitter} ($S_N(f)$) is related to the PSD of absolute jitter ($S(f)$) by

$$\text{PSD: } S_N(f) = 4\sin^2(\pi f NT) S(f) \quad (2.10)$$

The transfer function between absolute and period jitter $S_N(f) / S(f)$ is plotted in Fig. 2.23 for $N=1$, 10 and 100. As seen in the plot, period jitter lacks the low frequency content of absolute jitter. As $N$ is increased, the periodicity of (2.10) with respect to frequency causes periodic notches to appear in the response.

![Transfer Function Between PSDs of Absolute and Period Jitter](image)

Figure 2.23: Transfer function between absolute and period jitter $S_N(f) / S(f)$

### 2.3.2 Major Sources of Jitter in CDRs

Having defined jitter, we now examine some of the major sources of jitter in CDRs. The first source of jitter is the jitter of the incoming data. Two major sources of this data jitter are the transmitter’s clock jitter and jitter caused by ISI.
Transmit Jitter

The transmitter samples and transmits data using its own jittery clock, imparting its clock jitter to the transmitted waveform. As shown in Fig. 2.24, the width of transmitted bits is affected by jitter from two clock edges. Unlike jitter in the sample clock of the CDR, which simply shifts the sampling position of the data, Tx jitter reduces the horizontal eye-opening of the received data.

![Diagram of transmit clock jitter](image)

Figure 2.24: Data jitter caused by transmit clock jitter

ISI Jitter

A second major source of jitter is ISI caused by the channel. When the channel has insufficient bandwidth, the duration of a pulse transmitted across the channel exceeds 1UI, affecting the amplitude of future bits. This is depicted in Fig. 2.25, which shows a data sequence transmitted through a lossy channel. ISI distorts the rising and falling edges of the transmitted bits, introducing jitter.

In the CDR itself, the VCO or PI introduce jitter into the recovered clock.

VCO Jitter

The jitter of VCOs is traditionally analyzed in terms of phase noise rather than jitter. An ideal VCO outputs a sinewave at a constant frequency $f_0$, but random noise introduces phase deviations $\phi(t)$ so that the output is given by

$$A \sin(2\pi f_0 t + \phi(t))$$  \hspace{1cm} (2.11)
Chapter 2. Background

The so-called excess phase $\phi(t)$ causes the zero-crossings of the sine wave to deviate from their ideal positions at multiples of $T = 1/f_0$. Considering only rising edges, zero crossings occur at times $t_k = kT + \psi_k$, where $\psi_k$ is the jitter at each rising edge of the clock as depicted in Fig. 2.26. $t_k$ is given by:

$$2\pi f_0 t_k + \phi(t_k) = k2\pi$$  \hspace{1cm} (2.12)

Subbing in for $t_k$ gives an expression relating $\psi_k$ and $\phi(t)$.

$$\psi_k = -\frac{\phi(t_k)}{2\pi f_0}$$  \hspace{1cm} (2.13)

While $\phi(t)$ is continuous time signal, $\psi_k$ is a discrete time signal. To study $\phi(t)$, we examine its power spectral density, generally referred to as phase noise. Phase noise is often denoted $L(f)$ or $L(\omega)$ and has been extensively studied using various frequency and time-domain methods. Leeson’s model provides a simple empirical model of the phase noise seen in practical oscillators [11].
In (2.14), $k$ is Boltzmann’s constant, $T$ is the temperature in Kelvin, $P$ is the oscillator’s output power, $Q$ is the quality factor of the oscillator and $\omega_{1/f}$ is the $1/f$ corner frequency. $F$ is an empirical fitting parameter.

Intuitively, the three regions of $\mathcal{L}(\omega)$ can be explained as follows. As described earlier, VCOs operate as integrators in the phase domain, continuously accumulating phase at the rate dictated by the oscillation frequency. Any noise affecting the VCO frequency similarly gets integrated into phase. White noise, whose power spectrum is flat, is therefore integrated by
PI Jitter

We previously described how phase interpolation can be used to generate a clock with variable phase. We now describe this process in more detail and show how phase interpolation can also introduce unwanted jitter. As described earlier, a phase interpolator achieves phase rotation by interpolating between different clocks. For example, by interpolating between two quadrature clocks (I and Q) separated by $\pi/2$ and their inverses (-I and -Q), a phase interpolator can achieve $2\pi$ phase rotation by linearly increasing and decreasing the weights of each clock. The interpolator weights for this case are plotted in 2.28(b).

If the input clocks are ideal triangular waves, the weightings shown in Fig. 2.28(b) lead to the clocks waveforms plotted in Fig. 2.28(a). As seen in the plot, the zero-crossings of the interpolated waveforms are uniformly spaced, leading to a perfectly linear phase response as desired.

Since however, the PI steps have a finite size, the PI output creates jitter in the form of quantization error. Furthermore, nonidealities in the practical implementation can also distort
or cause other errors in the phase response of PIs. As an example, if sinusoidal clocks are used with the same linear weights, the phase of the interpolated waveforms become non-uniformly distributed as shown in Fig. 2.29(a), resulting in a nonlinear phase response shown in Fig. 2.29(b). Additional effects such as I/Q phase mismatch, uneven loading of, or coupling between clocks will cause further errors.

**Bang-Bang PD Jitter**

Bang-bang PDs introduce additional jitter when used in CDRs. As seen previously, the bang-bang PD only outputs the sign of the phase error. Similar to a 1-bit quantizer, the PD output
therefore contains quantization noise, which leads to jitter we denote as $\psi_{PD}$. To understand this jitter, we must first find the effective gain of the BB-PD. To do this we linearize the PD by examining its expected output, given that its input is some deterministic jitter $\psi$, plus random jitter $\psi_{ER}$ with probability density function (PDF) $f_{ER}(\psi_{ER})$.

Since the input to the PD is random data and the PD only produces an output when there is a transition, the maximum PD output is $\alpha_T$, the transition density of the input data. The transition density is the fraction of the time there is a transition in the data and is equal to 0.5 if the data is purely random. Combining these facts, the expected BB-PD output can be found as

$$E[PD_{OUT}] = P[Late] - P[Early]$$

$$= \alpha_T \left[ P[(\psi + \psi_{ER}) > 0] - P[(\psi + \psi_{ER}) < 0] \right]$$

$$= \alpha_T \left[ \int_{-\psi}^{\infty} f_{ER}(u)du - \int_{-\infty}^{\psi} f_{ER}(u)du \right]$$

$$= \alpha_T \left[ 1 - 2 \int_{-\infty}^{-\psi} f_{ER}(u)du \right]$$

(2.16)

The effective BB-PD gain $K_{PD}$ is then taken as the derivative of the expected PD output when $\psi = 0$ [12] as shown in Fig. 2.30(a).

$$K_{PD} = \frac{\partial E[PD_{OUT}]}{\partial \psi} \bigg|_{\psi=0}$$

(2.17)

$$K_{PD} = 2\alpha_T f_{ER}(0)$$

(2.18)

$K_{PD}$ therefore depends on the PDF of $\psi_{ER}$ and the transition density. As will be discussed in chapter 4, this is not desirable as any changes in $\psi_{ER}$ can affect $K_{PD}$, and therefore the CDR’s JTRAN and JTOL. Assuming that the random jitter $\psi_{ER}$ is a zero-mean Gaussian with standard deviation of $\sigma_{ER}$, it can be shown that [12,13]

$$K_{PD} = \sqrt{\frac{2}{\pi}} \frac{\alpha_T}{\sigma_{ER}}$$

(2.19)
Having found the effective gain of the PD, $\psi_{PD}$ can be found as the error between the actual BB-PD output and its linearized version as shown in Fig. 2.31. The standard deviation of $\psi_{PD}$ has been derived as [13]

$$\sigma_{PD} = \sqrt{\alpha_T - \frac{2}{\pi} \alpha_T^2}$$  \hspace{1cm} (2.20)

The resulting linearized model of the bang-bang PD is shown in Fig. 2.32, where the PD takes the difference between two sources of jitter $\psi_{DAT}$ and $\psi_{CK}$, scales it by $K_{PD}$ and adds its own jitter $\psi_{PD}$. We use this model extensively in this thesis to help analyze the bang-bang PD output.

### 2.3.3 Jitter in VCO-Based CDR

In this thesis, we will often analyze and model jitter in the implemented CDRs. To understand the jitter contributions in the previously-described VCO and PI-based CDRs, we reuse the linear models previously discussed and the relevant jitter sources. Here, we simply replace the continuous time phase $\phi$ with the discrete time jitter $\psi$. Using the typical assumption that the sample rate of the system far exceeds the bandwidth of the relevant signal content in $\phi$ [14],
Chapter 2. Background

Figure 2.31: (a) Actual vs. linearized PD output and corresponding (b) error at PD output

Figure 2.32: (a) Functional and (b) linearized model of bang-bang PD

we still approximate the system as being continuous rather than discrete time for simplicity, as shown in Fig. 2.33.

In the case that a BB-PD is used, we must also assume that enough random jitter is present in the system so that linear analysis can be used [13,15]. In the absence of jitter, CDRs using BB-PDs can behave nonlinearly, in which case the model in Fig. 2.33 is no longer valid.

Figure 2.33: Linear model of analog VCO-based CDR showing major sources of jitter

In addition to the above-described jitter from the PD and VCO, additional jitter is caused by
circuit noise from the charge pump $\psi_{CP}$ and loop filter resistance $\psi_{LF}$. Having constructed the jitter model, we can see how individual jitter sources contribute to $\psi_{ER}$, which is the relative jitter between the data and clock and $\psi_{CK}$, the absolute jitter of the recovered clock itself. The jitter can be analyzed in terms of PSDs and the previously discussed transfer functions $H_{JTRAN}(f)$ and $H_{JTOL}^{-1}(f)$. $S_{ER}(f)$ and $S_{CK}(f)$ are the PSDs of $\psi_{ER}$ and $\psi_{CK}$ respectively.

\[
S_{ER}(f) = \left( S_{DAT}(f) + S_{LF}(f) \frac{K_{VCO}}{2\pi f} \right)^2 + S_{VCO}(f) |H_{JTOL}^{-1}(f)|^2 \\
+ \left( \frac{S_{CP}(f)}{K_{PD}^2 I_{CP}^2} + \frac{S_{PD}(f)}{K_{PD}^2} \right) |H_{JTRAN}(f)|^2
\]  

(2.21)

\[
S_{CK}(f) = \left[ S_{DAT}(f) + \left( \frac{S_{CP}(f)}{K_{PD}^2 I_{CP}^2} + \frac{S_{PD}(f)}{K_{PD}^2} \right) \right] |H_{JTRAN}(f)|^2 \\
+ \left( S_{LF}(f) \frac{K_{VCO}}{2\pi f} \right)^2 + S_{VCO}(f) |H_{JTOL}^{-1}(f)|^2
\]  

(2.22)

$S_{ER}(f)$ and $S_{CK}(f)$ see identical contributions from all jitter sources except for $\psi_{DAT}$. The transfer function from $\psi_{DAT}$ to $\psi_{CK}$ is the lowpass transfer function $H_{JTRAN}(f)$ while the transfer function from $\psi_{DAT}$ to $\psi_{ER}$ is the highpass transfer function $H_{JTOL}^{-1}(f)$. As discussed previously, $H_{JTRAN}(f)$ and $H_{JTOL}(f)$ (and therefore $H_{JTOL}^{-1}(f)$) all have the same 3dB corner frequency $f_{3dB,CL}$. Since $H_{JTOL}^{-1}(f)$ is highpass, increasing $f_{3dB,CL}$ reduces the contribution of $\psi_{DAT}$ to $\psi_{ER}$. However, this also increases the bandwidth of $H_{JTRAN}(f)$, increasing $\psi_{DAT}$’s contribution to $\psi_{CK}$.

### 2.3.4 Jitter in PI-Based CDR

We similarly show the jitter model of a PI-based CDR in Fig. 2.34. The PSDs of the jitter can again be analyzed.

\[
S_{ER}(f) = (S_{DAT}(f) + S_{REF}(f) + S_{PI}(f)) |H_{JTOL}^{-1}(f)|^2 + \frac{S_{PD}(f)}{K_{PD}^2} |H_{JTRAN}(f)|^2
\]  

(2.23)

\[
S_{CK}(f) = (S_{REF}(f) + S_{PI}(f)) |H_{JTOL}^{-1}(f)|^2 + \left( S_{DAT}(f) + \frac{S_{PD}(f)}{K_{PD}^2} \right) |H_{JTRAN}(f)|^2
\]  

(2.24)
As in the case of the VCO-based CDR, $\psi_{ER}$ and $\psi_{CK}$ see identical contributions from all jitter sources except for the input data. We will revisit these linear models in this thesis to analyze the jitter of a VCO-based CDR in chapter 3 and a PI-based CDR in chapters 4 and 5.

### 2.4 Summary

This chapter has reviewed the basic concepts of clock recovery and jitter, as well as describing some of the main sources of jitter in VCO and PI-based CDRs. In the next chapter, we propose a technique to estimate the absolute data and clock jitter in a VCO-based CDR.
Chapter 3

On-chip Jitter Measurement for Multilane CDRs

In this chapter, we propose and demonstrate a technique for measuring data and clock jitter on-chip in multilane CDRs. Since they both contribute to a CDR’s bit error rate (BER), the jitter of the received data and recovered clock should both be characterized. At data rates of 10Gb/s and beyond, random clock jitter of well below 1ps RMS and as low as several hundred femtoseconds [16,17] is often needed, requiring jitter measurement circuits to have sub-picosecond resolution. As will be discussed in this chapter, existing techniques for measuring clock jitter [18–21] and data jitter [22,23] often require low-jitter external reference clocks, which may not always be available in a design. The goal of this work is an on-chip jitter measurement system able to characterize both clock and data jitter, without external reference clocks or measurement equipment such as oscilloscopes or spectrum analyzers. It should also add minimal area overhead to circuit layouts.

To characterize jitter, we estimate the autocorrelation functions of data and clock jitter by correlating the phase detector outputs of two 10Gb/s CDRs [24]. We then extract the RMS jitter and estimate the jitter’s PSD. The measurements can be used on-chip, or processed off-chip.

The remainder of this chapter is organized as follows. Section 3.1 reviews existing jitter measurement schemes. Section 3.2 presents the proposed correlation-based jitter measurement
Chapter 3. On-chip Jitter Measurement for Multilane CDRs

3.1 Background

3.1.1 Jitter Measurement Terminology

Before reviewing existing jitter measurement techniques, we recall the three types of jitter discussed in chapter 2: absolute jitter, relative jitter and period jitter. In this work, we want to measure the absolute jitter of both the input data ($\psi_D$) and recovered clock ($\psi_{CK}$) of a CDR. We are interested in absolute jitter because that is typically what is specified in wireline transceiver standards.

In practical cases however, the jitter of the signal of interest (e.g. $\psi_D$) must generally be measured compared to a non-ideal reference clock having its own jitter $\psi_{REF}$. In such cases, only the relative jitter between the signal and reference clock is measured: $\psi_{D-REF} = \psi_D - \psi_{REF}$. The relative jitter ($\psi_{D-REF}$) only approaches $\psi_D$ when $\psi_{REF} << \psi_D$. In other words, absolute jitter ($\psi_D$) can only be measured with a reference clock whose jitter ($\psi_{REF}$) is much smaller than $\psi_D$.

Alternatively, we can measure the time between adjacent zero-crossings of a clock signal to measure its period jitter. However, as described in chapter 2, because period jitter is the first-difference of absolute jitter [10], its spectrum is high-pass filtered compared to that of absolute jitter, making it difficult to observe jitter at lower frequencies. We now review some of the existing techniques for jitter measurement.

3.1.2 Clock Jitter Measurement

Previous works on jitter measurement have largely focused on clock jitter and often fall into three categories, shown in Fig. 3.1: time-to-digital converter (TDC)-based, self-referenced and PD-based. TDC-based circuits measure the relative jitter between a signal and a reference clock. Using delay lines [19,21] or other circuits, they effectively oversample the zero-crossing of a signal with a fine time resolution, converting zero-crossing times to digital codes.
Figure 3.1: Conventional (a) TDC-based (b) self-referenced and (c) PD-based jitter measurement techniques

The concept is illustrated in Fig. 3.2(a) where a data signal is sampled every $\tau$ seconds to determine where the zero crossing occurs. One possible implementation involves using many samplers, each clocked by a slightly delayed reference clock, to sample the data signal. This could however, cause excessive loading on the data. Alternatively, the data rather than the reference clock can be delayed as shown in Fig. 3.2(b). This is the basis of delay-line TDC circuits. An example is shown in Fig. 3.3(a), where the data passes through a delay line to produce many delayed versions of the data, which are each sampled by the reference clock.

The main limitation of this circuit is that TDC’s resolution is limited by the delay of the buffer stages ($\tau$), which is on the order of tens of picoseconds. To avoid this limitation, Vernier delay lines delay both the data and the clock each with different delays $\tau_1$ and $\tau_2$ as shown in Fig. 3.3(b). If $\tau_1$, the delay of the data, is equal to $\tau_2 + \Delta \tau$, the effective resolution becomes $\Delta \tau$, which can be made extremely small. Many other techniques such as time amplifiers [25], can also be used to increase the resolution of TDCs.

The use of TDCs for jitter measurement has two main drawbacks. Firstly, since it can only measure relative jitter, the reference clock jitter must be much lower than that of the signal being measured. Secondly, achieving a high time-resolution from TDC circuits generally limits their operating speed, due to the latency of delay-line structures [19] or cascading of circuits such as time amplifiers [20]. Furthermore, since TDCs have multi-bit outputs, the data
Chapter 3. On-chip Jitter Measurement for Multilane CDRs

Figure 3.2: Concept of time-to-digital conversion by (a) oversampling or (b) sampling delayed versions of clock

Figure 3.3: (a) Delay-line based and (b) Vernier delay-line based time-to-digital converters

generated by a TDC at tens of GHz requires very high data throughput to process. For these reasons, the operating frequency of high-resolution TDC-based jitter measurement circuits is
limited to several GHz.

The second category, self-referenced designs [20], avoids the need for a reference clock by measuring the jitter between a clock and its delayed version, effectively measuring period jitter as shown in Fig. 3.1(b). This approach only works for clock signals as random data does not have a transition in every UI. As mentioned, period jitter also has less content at low frequencies, limiting its usefulness for jitter measurement.

Finally, PD-based circuits use an analog phase detector to convert the relative jitter between the signal of interest and the reference clock, to an analog output [18]. The analog output allows for high-resolution jitter measurement, but is also sensitive to noise. Because analog PDs may also output very small signals, on the order of mV in [18], the output must be measured with either an oscilloscope or spectrum analyzer [18]. Alternatively, an on-chip high-speed, high-resolution ADC would be required to produce a digital output, transforming the PD into a TDC. As in the TDC-based approach, the reference clock jitter must again be much lower than that of the signal to be measured. In this work we also want to measure the jitter of data.

### 3.1.3 Data Jitter Measurement

As mentioned, the self-referenced technique is not applicable to data jitter. Because the TDC and the PD-based [22] approaches both use a clean reference clock, they are also not suitable in plesiochronous links, where CDRs receive data without a reference clock. Generating the required low-jitter clock could be costly in terms of power and area. In CDRs, eye monitor circuits [26] can generate the relative jitter histogram of a data signal by sampling it with a variable phase. This concept is shown in Fig. 3.4. By sweeping the delay of a dedicated eye monitor sampler and comparing its output to the adjacent data samples taken by the BB-PD, the relative jitter histogram can be constructed.

Asynchronous clocking [27] can also be used to sweep the data eye with an external clock having a frequency offset compared to the data. However, both of these methods measure relative jitter and therefore also require low-jitter reference clocks for accuracy.

Table 3.1 summarizes the main limitations of the jitter measurement techniques discussed. In this work, we seek a jitter measurement solution applicable to both data and clock jitter with sub-picosecond accuracy, which does not require a clean reference clock. We propose a
method to do this by estimating the jitter autocorrelation using phase detectors in a CDR.
3.2 Proposed Jitter Measurement Scheme

3.2.1 Proposed Concept for RMS Jitter Extraction

The goal of this work is the extraction of absolute jitter and its power spectral density. As discussed, phase detectors can measure the relative jitter between two signals, but the absolute jitter of each signal is not observable. In this work, we add a third signal. By measuring the relative jitter between each pair of signals using three PDs, we determine the jitter of each source. As shown in Fig. 3.5, with three sources A, B and C, the outputs of three ideal PDs are

\[ e_1 = \psi_A - \psi_B \]  
\[ e_2 = \psi_A - \psi_C \]  
\[ e_3 = \psi_B - \psi_C \]  

Note that \( e_3 \) can be determined from \( e_1 \) and \( e_2 \) as \( e_3 = e_2 - e_1 \). However, this requires subtracting the outputs of two linear PDs. As will be discussed, since bang-bang PDs are used in this work, this is not feasible to implement. We instead use three separate PDs.

Assuming the \( \psi \)'s are zero-mean random processes (\( E[\psi] = 0 \)) that are uncorrelated from each other, we multiply each pair of PD outputs and take the expected value (\( E[\cdot] \)). This eliminates the uncorrelated jitter components, allowing the variance (\( \sigma^2 \)) and RMS value (\( \sigma \)) of each jitter component to be identified. We assume the jitter is ergodic and approximate \( E[\cdot] \) using the time-average, implemented by a low-pass filter.

\[ E[e_1 e_2] = E[\psi_A^2] = \sigma_{\psi_A}^2 \]  
\[ E[e_1 e_3] = -E[\psi_B^2] = -\sigma_{\psi_B}^2 \]  
\[ E[e_2 e_3] = E[\psi_C^2] = \sigma_{\psi_C}^2 \]  

Unlike prior PD-based jitter measurement approaches, no clock is used as an ideal reference, eliminating the need for a clean reference clock.

This approach relies on the jitter of the clocks being uncorrelated. Any correlated jitter in
the two clock sources would add an offset error to the measurement, that would have to be calibrated out. If for example, clocks $B$ and $C$ both contain some jitter $\psi_{\text{CORR}}$, (3.4) would become $E[\psi_A^2 + \psi_{\text{CORR}}^2]$. To minimize any such correlation, the two clock sources should be well isolated from each other through careful layout and separation of their power grids using regulators or separate supply pads. This isolation should ensure that any correlated jitter caused by mutual coupling contributes only a fraction of the total clock jitter. In this work, we assume any correlated jitter is negligible compared to the total jitter being measured.

In the remainder of this section, we first discuss how the described method can be extended to measure the jitter’s autocorrelation and PSD. We then describe how it can be implemented using bang-bang PDs and applied to multilane CDRs by locking two CDRs to the same data.

### 3.2.2 Jitter Autocorrelation and PSD Measurement

The above scheme provides the measured RMS jitter of the signal of interest but does not provide information about its frequency content. To extract spectral information, we estimate the jitter’s autocorrelation function. By delaying $e_1$ by $n$ before correlating it with $e_2$, (3.4) becomes

$$E[e_1(k-n)e_2(k)] = E[\psi_A(k-n)\psi_A(k)] = R_{\psi_A}(k-n,k)$$

where $R_{\psi_A}(k-n,k)$ represents the autocorrelation function of the jitter $\psi_A$. Here, we have assumed uncorrelated jitter sources as before. If the jitter is wide-sense stationary (WSS), which is true in oscillators [28], then $R_{\psi_A}(k-n,k)$ is not a function of time $k$ and can be
replaced by $R_{\psi_A}(n)$. The Fourier transform of $R_{\psi_A}(n)$ gives the PSD \[29\] of the jitter, providing information about the jitter’s frequency content. We approximate $E[\cdot]$ in \[3.7\] by taking the average of $e_1(k – n)e_2(k)$ over time $k$, for different values of $n$. This method of estimating $R_{\psi_A}(n)$ is similar to the Blackman-Tukey \[30\] method for spectral estimation.

If the jitter is not WSS but cyclostationary, which is true for example when jitter is caused by periodic noise from clocked digital circuits \[31\], the autocorrelation function becomes a periodic function of $k$. In this case, our averaging approach gives an estimate of the time-averaged autocorrelation \[32\] with respect to $n$.

$$R_{\psi_A, \text{average}}(n) = \frac{1}{N} \sum_{k=1}^{N} R_{\psi_A}(k – n, k) \quad (3.8)$$

As a result of this averaging, the measured autocorrelation preserves the amplitude and frequency of periodic jitter, but not its phase. In the remainder of this chapter, we assume jitter is wide-sense stationary. If the jitter is cyclostationary, the results will be subject to this averaging effect.

### 3.2.3 Application to Bang-Bang PD

So far, we have ignored the gain of the PDs. The ideal PDs in Fig. 3.5 are replaced with PDs having gains $K_{P1}$ and $K_{P2}$ in Fig. 3.4. If bang-bang PDs are used, the correlation and filtering can be done on-chip using logic and counters. To measure autocorrelation, the phase offset $n$ can be adjusted using FIFOs. Accounting for the PD gains, and assuming $\psi_A$ is WSS, \[3.7\] becomes
\[ E[e_{12}] = K_{P1}K_{P2}R_{\psi_A}(n) \] (3.9)

To estimate \( \sigma_{\psi_A} \), we set \( n \) to zero giving

\[ \sigma_{\psi_A} = E[\psi_A^2] = \sqrt{\frac{E[e_{12}]}{K_{P1}K_{P2}}} \bigg|_{n=0} \] (3.10)

For a bang-bang PD however, and as described in chapter 2, \( K_{P1} \) and \( K_{P2} \) depend on the distribution of the relative input jitter (\( \psi_A - \psi_B \) for PD1 and \( \psi_A - \psi_C \) for PD2) [12]. We estimate the PD gain using an edge monitor circuit, which consists of an auxiliary edge sampler driven by a clock with a variable phase offset. Comparing the edge monitor samples to the edge samples from the PD allows the PD output to be measured as a function of the phase offset, without affecting the lock position of the CDR. This allows the cumulative distribution function (CDF) of the PD output to be measured on-chip using counters. The PD gain can then be measured as the slope of this CDF, and the RMS jitter calculated (off-chip) from (3.10). Because linearizing the PD response is an approximation, the value of \( \sigma_{\psi_A} \) as determined by (3.10) must be divided by a constant that depends on the type of the jitter distribution. When all of the jitter sources are Gaussian, Matlab simulations estimate this constant to be 1.34. This factor also accounts for some error in the method used to estimate the PD gain and is described in Appendix A. When \( \sigma_{\psi_A} \) is greater than or equal to \( \sigma_{\psi_B} \) and \( \sigma_{\psi_C} \), the scaling constant changes to 1.97 if \( \psi_A \) is sinusoidal jitter (SJ), as the jitter distribution changes shape and no longer appears Gaussian. This is also described in the Appendix.

### 3.2.4 Complete System for Multilane CDR

In summary, the proposed technique characterizes jitter using the correlation between each pair of PDs. The CDF of each PD output is used to extract the PD gain. Spectral information is obtained by sweeping the delay of one of the PD outputs, to produce the autocorrelation function. This data can be sent off-chip, and its FFT taken, to obtain the jitter’s PSD.

This jitter measurement scheme can be applied to CDRs by replacing the signals \( A, B \) and
C with the input \( DATA \) and two clocks \( CK1 \) and \( CK2 \), respectively, having jitters of \( \psi_D \), \( \psi_{CK1} \) and \( \psi_{CK2} \). In this work, \( CK1 \) and \( CK2 \) are generated by two adjacent analog CDRs both locked to \( DATA \). Fig. 3.7 shows an example system applied in a multilane CDR where an adjacent lane could be taken offline and reconfigured using a MUX, to provide \( CK2 \) in a diagnostic mode. To maintain full operation of the link, a redundant diagnostic lane could also be added to the system, amortizing the cost of circuits across many lanes. The PD outputs from each CDR are used for jitter measurement. Adding PD3 allows the jitter of \( CK1 \) and \( CK2 \) to also be measured. Edge monitors are added for PD gain measurement. Since PD3 is not part of a CDR loop, its sampling phase is adjustable and can serve as its own edge monitor. The outputs of all of the PDs can be correlated and analyzed digitally. The CDRs have a filtering effect on the jitter being measured, which is analyzed in section 3.3 below.

### 3.3 Analysis of Jitter Measurement with Two CDRs

#### 3.3.1 Linear Model

To determine the effect of the CDRs on the PD correlation signal, we examine the frequency content of the PD signals using a linear phase model of the CDR as shown in Fig. 3.8. Note that this linear model is only applicable when the CDR sees sufficient jitter to linearize its response [13,15]. In this model, \( \psi_{N1} \) and \( \psi_{N2} \) represent all of the jitter contributions from the VCO, PD, charge pump (CP) and loop filter (LF) of CDR1 and CDR2 respectively. The CDR
loops filter $\psi_D$, $\psi_{N1}$ and $\psi_{N2}$ and produce $e_1$ and $e_2$ with corresponding Laplace transforms $E_1(s)$ and $E_2(s)$

$$E_1(s) = \frac{K_{P1}}{1 + K_{P1}H_1(s)} [\Psi_D(s) - \Psi_{N1}(s)]$$

$$E_2(s) = \frac{K_{P2}}{1 + K_{P2}H_2(s)} [\Psi_D(s) - \Psi_{N2}(s)]$$

Where $H_1(s)$ and $H_2(s)$ represent the combined transfer functions of the CP, LF and VCO of CDR1 and CDR2 respectively. In CDRs, $H_1(s)$ and $H_2(s)$ have a low-pass response, therefore each PD output contains high-pass filtered versions of the corresponding data jitter $\psi_D$ and CDR jitter $\psi_{N1}$ or $\psi_{N2}$. The measurement of data and recovered clock jitter are analyzed separately.

3.3.2 Analysis of Data Jitter Measurement

We first examine the correlation signal $e_{12}$, used to measure data jitter. In the general case, one of the PD signals is delayed by $n$, as in (3.7).

$$E[e_{12}] = E[e_1(k-n)e_2(k)]$$

Figure 3.8: Linear model of PD correlation with two CDRs
Assuming that the CDR jitter sources $\psi_{N1}$ and $\psi_{N2}$ can be modelled as uncorrelated, zero-mean Gaussian random processes, multiplying the outputs of PD1 and PD2 and taking the expected value cancels out the uncorrelated jitter sources, leaving only the data jitter. If $\psi_D$ is wide-sense stationary then (3.13) is only a function of $n$. Consequently, if the PSD of the jitter $\psi_D$ is $S_D(f)$, then using the Wiener-Khinchin theorem [29], it can be shown that the Fourier transform of $E[e_{12}]$ (taken with respect to time $n$) can be written as

$$F\{E[e_{12}]\} = \frac{K_{P1} K_{P2} S_D(f)}{(1 + K_{P2} H2(f))(1 + K_{P1} H1(f))^2} \quad (3.14)$$

The Fourier transform of the averaged PD autocorrelation signal is therefore a high-pass filtered version of $S_D(f)$, the PSD of $\psi_D$. Since in-band jitter is suppressed by the CDR loops, this scheme characterizes the out-of-band data jitter responsible for performance degradation in the CDR. This method is suitable for measuring wideband jitter such as jitter on PRBS data.

Although this highpass filtering effect may seem similar to that of existing self-referenced jitter measurement schemes that measure period jitter, the response of (3.14) remains flat above the highpass filter’s corner frequency. If the CDR’s closed loop bandwidth is 10MHz, absolute jitter can therefore be measured from 10MHz to the Nyquist frequency ($F_{Nyquist}$) of the system, which is half of the data rate. In contrast, the highpass filtering effect seen by self-referenced measurement schemes starts rolling off immediately below $F_{Nyquist}$, and attenuates jitter over a much wider bandwidth. If for example period jitter is measured, absolute jitter is already attenuated by 3dB below $F_{Nyquist}/2$, which is 2.5GHz for a data rate of 10Gb/s. This can be seen from (2.10) in section 2.3.1 of chapter 2. Next, we consider PD3, which allows the CDR’s recovered clock jitter to be estimated.

### 3.3.3 Analysis of Clock Jitter Measurement

PD3 measures the phase difference between $CK1$ and $CK2$ and when combined with PD1 and PD2, allows us to measure the jitter in $CK1$ and $CK2$. The output of PD3 is

$$e_3 = K_{P3} (\psi_{CK1} - \psi_{CK2}) \quad (3.15)$$
CK1 and CK2 both contain filtered versions of the data jitter so $\psi_{CK1}$ and $\psi_{CK2}$ can be written as

$$\psi_{CK1} = (\psi'_{D1} + \psi'_{N1}), \quad \psi_{CK2} = (\psi'_{D2} + \psi'_{N2})$$

(3.16)

where $\psi'_{D1}$ and $\psi'_{N1}$ represent the contributions to the recovered clock jitter of CDR1 from $\psi_D$ and $\psi_{N1}$ respectively, and $\psi'_{D2}$ and $\psi'_{N2}$ are the corresponding terms for CDR2.

$$\Psi'_{D1}(s) = \frac{K_{P1}H1(s)\Psi_D(s)}{1 + K_{P1}H1(s)}, \quad \Psi'_{N1}(s) = \frac{\Psi_{N1}(s)}{1 + K_{P1}H1(s)}$$

(3.17)

$$\Psi'_{D2}(s) = \frac{K_{P2}H2(s)\Psi_D(s)}{1 + K_{P2}H2(s)}, \quad \Psi'_{N2}(s) = \frac{\Psi_{N2}(s)}{1 + K_{P2}H2(s)}$$

(3.18)

If the two CDRs are identical, i.e. $H1(s) = H2(s)$, and $K_{P1} = K_{P2}$, then the $\psi_D$ terms cancel out in (3.15), leaving

$$e_3 = K_{P3}(\psi'_{N1} - \psi'_{N2})$$

(3.19)

PD3 therefore provides a measure of the filtered CDR jitter $\psi'_{N1}$ and $\psi'_{N2}$. Now correlating PD3 from (3.19) with the PD1 output given by (3.11) and making the same assumptions about jitter being uncorrelated gives

$$E[e_{13}] = E[e_1(n - k)e_3(n)]$$

$$= -K_{P1}K_{P3}E[\psi'_{N1}(n - k)\psi'_{N1}(n)]$$

(3.20)

(3.21)

The correlation signal $E[e_{13}]$ contains the high-pass filtered CDR jitter $\psi'_{N1}$. Assuming that $\psi_{N1}$ is wide-sense stationary, the Fourier transform of $E[e_{13}]$ is then

$$\mathcal{F}\{E[e_{13}]\} = \frac{-K_{P1}K_{P3}S_{N1}(f)}{|1 + K_{P1}H1(f)|^2}$$

(3.22)

Correlating PD1 and PD3 therefore allows us to measure the out-of-band portion of $\psi_{N1}$, which represents the portion of the recovered clock jitter ($\psi_{CK1}$) contributed by CDR1’s circuits. This measurement is decoupled from the data jitter, allowing an assessment of CDR1’s intrinsic
jitter performance. To minimize the high-pass filtering effect on clock jitter measurement, a low CDR loop bandwidth should be used. Correlating PD3 with PD2 yields the analogous result for $\psi_{N2}$.

Dividing equations (3.13) and (3.20) by the PD gains and taking the square root yields the RMS values of the high-pass filtered versions of $\psi_D$ and $\psi_{N1}$. The proposed approach therefore allows us to estimate the RMS value of both the data and CDR clock jitter without any clean reference clock. Taking the Fourier transform of the correlation signals also gives us the estimated PSD.

### 3.3.4 Overhead of Proposed Technique

Since edge monitors are still required to measure the PD gain, we may want to compare the overhead of the proposed system to the alternative of simply using edge monitors clocked by a clean reference clock. Since we only expect jitter measurement circuits to be active occasionally for diagnostics or adaptation, the area rather than the power consumption of any added circuits is the primary concern. As described above, the proposed technique only adds a small hardware overhead if applied to a multilane system. In cases where no other CDR lanes are available, the difference between the two approaches becomes the cost of clock generation, which we assume must be performed on-chip. Unlike in the proposed scheme, the jitter of an eye monitor’s clean reference clock must be much lower than that of the data and clock being measured. For example, assuming all jitter is uncorrelated, to measure 1ps RMS jitter with 100fs accuracy would require the reference clock jitter to be less than 460fs.

Since the reference clock must also be phase-aligned to the data and clock being measured, it must be generated by a CDR with less than half the jitter of the CDR being tested. This could be difficult to achieve, given that the CDR under test has probably already been optimized for jitter performance. To achieve this likely requires a VCO with larger area than that used in the CDR being tested, either to reduce the jitter of the ring VCO, or because an LC oscillator may be needed, whose inductor area alone could exceed the area of the ring VCO used in this work.

Therefore generating a clean reference clock on-chip would likely consume more power and area than the second CDR lane used in this work, even if the second CDR was added only for
measurement purposes. Furthermore, an edge monitor alone does not provide the ability to estimate the jitter’s PSD. To do so would require a TDC, which also requires a clean reference clock and would likely consume much more area than the eye monitor.

Note that all techniques considered typically require some off-chip post-processing to calculate the RMS jitter either from the measured jitter histogram or in the proposed work, based on the correlation of the PD outputs.

3.4 Implementation

3.4.1 Test Chip Implementation

As shown in Fig. 3.9, a test chip was fabricated consisting of a continuous-time linear equalizer (CTLE) driving two 10Gb/s half-rate CDRs, DMUXes and a digital core. PD3 is added to allow estimation of the CDR’s recovered clock jitter. A variable delay block deskews CK2 compared to CK1, ensuring correct operation of PD3.

3.4.2 CDR Implementation

Fig. 3.10 shows the CDR1 architecture, consisting of a half-rate bang-bang PD, charge pump, loop filter and a 4-stage ring VCO operating at 5GHz. To estimate the PD gain with sufficient accuracy, the edge monitor must have a phase resolution on the order of the RMS jitter being measured. In this case, the variable-phase edge monitor clock ($CK_{EDGE}$) is generated by a
5-bit CML phase interpolator (PI\textsubscript{E}), which interpolates between two phases of the VCO with a resolution of $25\text{ps}/31 \approx 0.8\text{ps}$. Two PI blocks (PI\textsubscript{I} and PI\textsubscript{Q}) with fixed interpolation ratios buffer $CK_I$ and $CK_Q$ with a fixed delay to ensure that $CK_{EDGE}$ and $CK_Q$ are nominally aligned. Fig. 3.11 shows the two types of PI blocks; one with 5-bit control and the other with both inputs equally weighted.

Differential-to-single-ended (D2S) converters convert the CML clocks to CMOS levels for use in the half-rate PD shown in Fig. 3.12. The PD outputs two half-rate UP/DN signals with rail-to-rail swing to dual charge pumps that drive the loop filter. This relaxes the design
requirements of the charge pump and avoids the high-speed muxes required in [33]. The PD uses sense-amp based latches due to their narrower sampling aperture [34]. Double-tail latches based on those used in [35] and shown in Fig. 3.13 are used. Compared to the design in [35], an additional NMOS keeper cell is used in the second stage to maintain pull-down current when the first stage outputs go low, and a reset switch is added to reduce hysteresis. As shown in Fig. 3.12, to maintain timing margin, additional latches resample and align all outputs to a single clock phase before all the outputs are resampled with conventional CMOS flip-flops. The PD outputs are then DMUXed by 8 and sent to the digital core, which is clocked at 625MHz.

As shown in Fig. 3.10, PD3 is a second bang-bang PD whose edge clock phase $CK_{Q,PD3}$ is also driven by CDR1’s VCO through a phase interpolator. Note that although only the edge sample is needed for PD3 to detect the phase of $CK2$, a complete PD was used to ease debugging of the test chip. In CDR2, instead of driving PD3, the VCO drives the variable delay block (see Fig. 3.9) used to feed $CK2$ into PD3. All of the additional measurement-related circuits can be disabled by disablimg the D2S circuits, thereby gating the clock for all front-end latch circuits.

3.4.3 Phase Offset Compensation

Effects such as charge pump mismatch and comparator offset could cause the CDR to lock with residual phase offset with respect to the data. The PD output could therefore have a
non-zero mean, introducing error into the PD correlation. Although this can be managed with careful design and offset compensation, in this work, residual offset was compensated by using the edge monitor samples, rather than the PD edge samples for PD correlation. Phase offset was compensated by digitally adjusting the edge monitor phase using either the on-chip DLL function (described below) or manually, by examining the CDF of the edge monitor output. Duty-cycle-distortion (DCD) in the half-rate architecture could also cause even/odd mismatch in the PD. To compensate, RMS jitter was measured separately for even and odd samples, with the blue edge monitor phase optimally adjusted in each case. The even and odd results were then averaged.

### 3.4.4 Digital Core

The digital core is shown in Fig. 3.14 and consists of FIFOs, digital PD blocks, a filter mask block, correlation counters and a programmable DLL counter. Instead of sending the PD outputs to the digital core directly, the raw data and edge samples are sent, requiring two bits per UI including the recovered data, instead of three if the PD outputs were sent in addition to the recovered data. FIFO stages allow data to be phase-shifted between PDs to generate the autocorrelation function. The filter mask block can filter the PD data based on even and odd samples, as well as several data patterns that can be used to analyze the effect of intersymbol interference (ISI) on jitter. For example, by measuring only the 010 data pattern, the effect of the first post-cursor ISI on jitter can be suppressed in the measurement.

The filtered PD outputs are then sent to digital correlation counters. In this block, a 17-bit edge counter counts the total number of data transitions, while the correlation counters
count how many of the PD outputs are correlated. The ratio of the correlated to the total number of transitions gives the PD correlation. To achieve accurate measurements, enough PD samples should be correlated to span several time constants of the CDR, ensuring that any CDR dynamics are averaged out. Additional histogram counters count the number of DN transitions, allowing the relative jitter histogram to be measured as the edge monitor phase is swept. In this chip, one edge counter and six additional counters were used to simultaneously process and correlate data from all three PDs. To reduce power and area, fewer counters could be implemented and reused for different measurements.

The DLL counter with programmable division ratio is reconfigurable to accept the input of any of the PDs. In conjunction with the edge monitor PI blocks, the counter could be used to lock the edge monitors to the edge of the data eye. When driving the variable delay block (see Fig. 3.9), the DLL could also be used to deskew $CK_2$ with respect to $CK_1$. 

![Figure 3.14: Overview of digital core](image)
\section*{3.5 Measurement Results}

\subsection*{3.5.1 CDR Functionality}

Fig. 3.15 shows the die photo of the chip fabricated in Fujitsu’s 65nm CMOS technology. CDR1 consumes 62mW and occupies 0.084mm$^2$ while CDR2 consumes 57mW due to fewer circuits. The edge monitor blocks add 11\% measured power and 9\% area overhead to CDR1. The DMUXes occupy a total of 0.013mm$^2$ and consume 7mW. The total area overhead including DMUXes, of all jitter-related analog circuitry is approximately 18\%. The digital core occupies 0.106mm$^2$ and consumes 31mW.

To demonstrate the CDR’s functionality and typical performance, Fig. 3.16(a) shows a recovered half-rate PRBS7 data eye. The real-time scope is able to pattern-lock to the PRBS7 pattern. Fig. 3.16(b) shows the jitter histogram of the recovered clock, showing typical RMS jitter of 1.8ps. Fig. 3.17 shows the CDR’s jitter tolerance for 10Gb/s PRBS31 data at a bit error rate (BER) of $10^{-12}$. High-frequency jitter tolerance is 0.19UIPP.

\subsection*{3.5.2 Jitter Measurement Test Setup}

The test setup is shown in Fig. 3.18. To validate the proposed concept, PRBS31 data from a Centellax TG1B1-A BERT was used to drive the CDRs. The BERT was clocked by a TG1C1-A clock synthesizer with internal SJ injection. Random jitter (RJ) was applied by driving the synthesizer’s external modulation input with a NoiseCOM noise generator. The bandwidth of this input was 20MHz to 100MHz, allowing RJ in this frequency range to be injected.
Figure 3.16: (a) Half-rate recovered PRBS7 data eye and (b) clock jitter (pink is jitter spectrum measured by scope)

Figure 3.17: Measured jitter tolerance

Figure 3.18: Test setup

3.5.3 PD Gain Measurement

The PD gain is measured as part of each jitter measurement. Fig. 3.19 is an example of a CDF of PD1’s output, measured by sweeping the edge monitor phase. As described in section 3.2.3, the PD gain is calculated from the slope of the CDF in its linear region and combined with the
PD correlation in calculating RMS jitter.

### 3.5.4 Data Jitter Measurement Results

Fig. 3.20 shows the measured RMS data jitter as RJ is injected into the data. The plot compares the RMS jitter as estimated by on-chip measurement against the jitter measured by an Agilent DSAX91604A 80GS/s (16GHz bandwidth) real-time oscilloscope, which has a 150fs jitter measurement noise floor. Fig. 3.21 shows measurement results when SJ is injected into the data at 100MHz. In both cases, the estimated jitter differs from the real-time scope’s measurement by no more than 0.6ps over the entire range of injected jitter amplitudes. Using this approach, jitter levels well below that of the CDR’s recovered clock jitter of 1.8ps RMS can be estimated. The results shown in Fig. 3.20 and Fig. 3.21 are slightly different than those reported in [24] as we previously used a scaling factor of 2 for both RJ and SJ cases. These scaling factors are now updated to 1.34 for RJ and 1.97 for SJ, as discussed in section 3.2.3.

Some of the discrepancy between the estimated and scope measurement results is likely attributed to coupling of the data jitter, possibly through the power supplies of the test chip. When injecting SJ into the data, measurements showed that the injected SJ was coupling to the CDR output clocks, causing spurs in their spectra. Since the coupling was to both CDR outputs, this would cause correlated jitter between the two CDR outputs, leading to an offset in the estimated jitter as described in section 3.2.1.

The data jitter’s PSD is estimated from the FFT of the measured PD autocorrelation. Fig. 3.22(a) shows the measured data jitter autocorrelation $R_{\psi_D}(n)$ with no additional jitter added to the data. DCD in the half-rate CDR causes variation between the even and odd values of $R_{\psi_D}(n)$. Fig. 3.22(b), plots the even and odd samples of $R_{\psi_D}(n)$ separately. In this figure the delta function centered at $n = 0$, indicates that the data’s random jitter is nearly white. (The autocorrelation function of white noise is a delta function.) Fig. 3.23 shows the measured data jitter autocorrelation when 0.05UI$_{PP}$ SJ injected at 100MHz, corresponding to a period of 100UI. The SJ at 100MHz is clearly visible in the autocorrelation function as a sinusoid. Fig. 3.24 compares the FFT of $R_{\psi_D}(n)$ to the jitter spectrum measured by the scope with SJ injected. Both show large spurs at 100MHz, demonstrating that individual SJ components can be identified with this approach.
Figure 3.19: Measured CDF of PD1’s relative jitter $\psi_D - \psi_{CK1}$ with no RJ or SJ added

Figure 3.20: Measured RMS data jitter with 20-100MHz injected RJ

Figure 3.21: Measured RMS data jitter with SJ injected at 100MHz
3.5.5 CDR Clock Jitter Measurement Results

Correlating the outputs of PD1 with PD3 allows estimation of the CDR’s output jitter. As shown in (3.22), the correlation signal is a high-pass filtered version of the CDR clock’s output.
jitter. Unlike the data jitter, which has a high bandwidth, due to modulation by the random data pattern, the jitter of the CDR clock has a lower bandwidth. The high-pass characteristic of the correlation signal attenuates the low-frequency content of this jitter and measures only the high-frequency portion of the CDR clock’s output jitter. Despite this, the measurement is useful for diagnostics as it can still reveal changes in the CDR’s jitter performance. To test this, SJ was injected at 47MHz (close to the CDR’s jitter transfer corner frequency) into the CDR’s VCO by coupling an external clock source into the VCO’s bias control circuit. As shown in Fig. 3.25, the jitter estimated from PD correlation closely tracks the CDR’s output jitter as measured by the scope but has an offset of about 0.8ps due to the high-pass filtering effect of the CDR.

Fig. 3.26 shows the estimated clock jitter autocorrelation and PSD with and without jitter injected at 1GHz. First, unlike the data jitter autocorrelation, instead of a delta function, a much wider pulse is centered at $k = 0$. This indicates that the clock jitter has a limited bandwidth with a time constant related to the spread of the pulse in UI. Second, once injected, the 1GHz SJ is visible in Fig. 3.26(a), superimposed on the original autocorrelation function. Fig. 3.27 compares the estimated jitter PSD to the jitter spectrum measured by the scope. Despite the high-pass filtering effect, the plotted PSD not only shows the injected 1GHz spur,
Figure 3.25: Measured RMS clock jitter with SJ injected into VCO at 47MHz

![Graph showing measured vs. estimated jitter with SJ injected into VCO at 47MHz.](image)

Offset error $\approx 0.8\text{ps}$

Figure 3.26: Estimated CDR clock jitter autocorrelation $R_{\psi_{CK1}}(n)$ (a) with and (b) without SJ injected at 1GHz ($1\text{UI}=100\text{ps}$)

![Graphs showing estimated CDR clock jitter autocorrelation with and without SJ injection.](image)

but also the low-pass nature of the clock’s jitter spectrum and the CDR’s loop bandwidth.

Table 3.2 compares this work to previous works on jitter measurement.
Figure 3.27: PSD of CDR clock jitter with SJ at 1GHz as measured by (a) scope and (b) on-chip measurement using FFT of $R_{\psi_{CK1}}(n)$

Table 3.2: Comparison to Previous Jitter Measurement Circuits

<table>
<thead>
<tr>
<th>Reference</th>
<th>Input signal</th>
<th>Ext. Ref. Clock</th>
<th>Output Signal</th>
<th>Frequency / Data Rate</th>
<th>Measurement Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>[19]</td>
<td>Clock</td>
<td>Yes</td>
<td>Digital</td>
<td>250MHz</td>
<td>&lt;1ps</td>
</tr>
<tr>
<td>[20]</td>
<td>Clock</td>
<td>No</td>
<td>Digital</td>
<td>3.36GHz</td>
<td>350-700fs*</td>
</tr>
<tr>
<td>[21]</td>
<td>Clock</td>
<td>Yes</td>
<td>Digital</td>
<td>1GHz</td>
<td>101fs*</td>
</tr>
<tr>
<td>[18]</td>
<td>PRBS</td>
<td>Yes</td>
<td>Analog</td>
<td>2.5Gb/s</td>
<td>1.56ps†</td>
</tr>
<tr>
<td>This Work</td>
<td>PRBS</td>
<td>No</td>
<td>Digital</td>
<td>10Gb/s</td>
<td>≤ 580fs</td>
</tr>
<tr>
<td>This Work</td>
<td>Clock</td>
<td>No</td>
<td>Digital</td>
<td>5GHz</td>
<td>≤ 1ps</td>
</tr>
</tbody>
</table>

*Jitter only reported for one case
† Value is standard deviation of measurement error
3.6 Application Example: Jitter-Based Equalization

The on-chip nature of this jitter measurement technique lends itself well to on-chip diagnostics or adaptation. In this section, we describe an example of how the proposed techniques can be used to help equalize a lossy channel. By simply masking the measured PD data, we can extract information on ISI. The channel used is a 14" backplane channel with two daughter card connectors and 10.6dB loss at 5GHz.

Similar to [36], we choose different data patterns corresponding to different ISI levels to estimate the portion of jitter contributed by ISI. Making the assumption that the channel has a large first post-cursor, we select PD data matching the data patterns “110” and “010”. Because of ISI, the average phase position of the “110” pattern will be slightly delayed compared to that of “010”. We confirm this on-chip by extracting the locked phase position from jitter histograms corresponding to the different patterns. Fig. 3.28 shows the different lock phases for four data patterns as the CTLE boost setting is varied. As the boost increases, the locked phase of the “110” and “010” patterns converge and then diverge again as the channel becomes over-equalized.

Similarly, we can compare the data jitter of different patterns, as estimated with our proposed technique. Fig. 3.29 compares the measured jitter of the same four patterns compared to the case of unfiltered data marked as “all bits”. The results show that the lowest jitter is found with the “1010” and “0010” patterns where the effect of the first two post-cursors is suppressed.
Figure 3.29: Measured RMS data jitter for different patterns using on-chip measurement (since all data have the same ISI). The jitter of these patterns is also much lower than that of the combined data, as well as being less sensitive to changes in the CTLE setting.

Fig. 3.30 compares the estimated data jitter, CDR clock jitter and measured high-frequency jitter tolerance of the CDR for different CTLE settings. The minimum measured jitter is observed at a CTLE boost setting of 11. The lower plot shows that this setting also maximizes the high-frequency jitter tolerance of the CDR.

Comparing Figures 3.28 and 3.29, we also see that the lowest data jitter occurs when the phases of the different data patterns most closely converge, between CTLE codes 10 and 11. Using this observation, a simple CTLE adaptation loop is implemented which drives the mean phase difference between the “110” and “010” patterns to zero in an approach similar to [37] and shown in Fig. 3.31. Two filter blocks mask the PD data going into a counter so that the mean PD outputs for “110” and “010” patterns are driven to be the same. Fig. 3.32 shows that the adaptation loop response for the same channel with PRBS31 data. The loop converges within 1μs to between codes 10 and 11. This is slightly less optimal than code 11. Accounting for the additional postcursors with more patterns could lead to improved performance. For example a weighted function of each pattern [36] could be used. This simple application demonstrates how on-chip jitter measurement can be used to drive or verify system parameters such as equalizer gain. In fact the jitter measurement itself could also be used as a cost-function for least mean squares (LMS) type adaptation loops.
Figure 3.30: Measured RMS data and CDR1 clock jitter (using on-chip measurement) and jitter tolerance vs. CTLE pattern

Figure 3.31: Pattern-filtering based CTLE adaptation

Figure 3.32: Pattern jitter-based CTLE adaptation curve
3.7 Summary

In this chapter, we have proposed and demonstrated a jitter measurement scheme using PD correlation. By correlating the PD outputs from two CDRs locked to the same data, the RMS clock and data jitter can be measured with sub-picosecond accuracy. Compared to prior techniques, this approach achieves comparable accuracy at the highest data rate, is applicable to both clock and data jitter measurement, and does not rely on any clean external clock source. Using autocorrelation, the jitter’s PSD can also be estimated. This approach is applicable to multilane CDRs where CDRs could be reconfigured in a diagnostic mode and allows for monitoring and optimization of the CDR’s jitter performance.
Chapter 4

Adaptive Loop Gain CDR for Jitter Tolerance Optimization

In this chapter, we move from measuring jitter, to trying to mitigate its effects by adapting the loop gain of a digital CDR to optimize jitter tolerance. Digital CDRs are popular in part for their robustness, but their use of bang-bang phase detectors makes their performance sensitive to changes in jitter caused by PVT variations, crosstalk or power supply noise. This is because the gain of a BB-PD depends on the CDR’s input jitter, causing the CDR’s loop gain to change if the jitter’s magnitude or spectrum varies. This problem is illustrated in Fig. 4.1 where small jitter leads to excessive loop gain and hence to an underdamped behaviour in the CDRs jitter tolerance, while large jitter leads to insufficient loop gain and hence to low overall JTOL. To prevent this, we propose a CDR with an adaptive loop gain, $K_G$, as shown in Fig. 4.1.

As will be discussed in this chapter, prior works adapt the CDR’s loop filter by either directly measuring or estimating the amplitude or bandwidth of the CDR’s jitter. These approaches require either dedicated jitter measurement circuits, which can be costly to implement, or some prior knowledge of the expected jitter profile seen by the CDR. In contrast, in this work, we simply increase $K_G$ and therefore the CDR’s loop bandwidth to suppress the most jitter and maximize jitter tolerance. To prevent the CDR from becoming too underdamped, we monitor the autocorrelation function of the CDR’s BB-PD output for any ringing, which appears if the CDR’s phase margin drops too low.
Figure 4.1: (a) Conventional bang-bang CDR and proposed adaptive loop gain CDR and the impact of adaptation on jitter tolerance when jitter is (b) too small or (c) too large.

The remainder of this chapter is organized as follows. Section 4.1 first provides background on existing adaptive loop gain strategies. Next, after modelling the jitter of the proposed PI-based CDR in section 4.2, our analysis in section 4.3 shows that our approach achieves near-optimal jitter performance while maintaining the high loop bandwidth desired for meeting jitter tolerance requirements. Section 4.4 describes how the proposed adaptation prevents the CDR from becoming too underdamped by monitoring the autocorrelation function of the PD output. Section 4.5 then describes additional adaptation features allowing it to handle sinusoidal jitter. Section 4.6 describes the test chip implemented in 28nm CMOS and section 4.7 describes measurement results, which confirm that the proposed technique optimizes high-frequency jitter tolerance for several different jitter profiles.
4.1 Background

Several prior works have adapted the loop gain of a PLL or CDR to minimize its jitter or BER. In [38], jitter is measured off-chip and fed to a gradient descent algorithm to optimize the parameters of a PLL. This technique could be combined with on-chip jitter measurement circuits such as eye monitors [39], leading to the system shown in Fig. 4.2(a). In [12], jitter is measured using a PD with an adjustable dead-zone, and used to regulate the CDR’s loop gain. We have seen in chapter 3 that high-precision jitter measurement is difficult to implement on-chip. Adding jitter measurement circuits also increases power and area, and can increase loading on high-speed clock and data paths, making these approaches less attractive.

Other techniques avoid the use of jitter measurement circuits by instead monitoring the PD or loop filter outputs. In [40], the jitter’s magnitude is estimated using the FFT of the loop
filter output as shown in Fig. 4.2(b). This is then used to calculate the optimum loop gain using Kalman theory. Assuming that the jitter at the CDR input is of the form shown in Fig. 4.3(a), it was shown in [41] that $K_{B,OPT}$, the optimum gain of the first-order PI-based CDR shown in Fig. 4.3(b) is approximately given by

$$K_{B,OPT} = \sigma_W N_{PI}$$  \hspace{1cm} (4.1)

$\sigma_W$ is the $\sigma$ of $\psi_W$, the fictitious Gaussian jitter that would be accumulated to create the observed jitter having a -20dB/decade roll-off as shown in Fig. 4.3(a). $N_{PI}$ is the number of phase steps per UI of the PI. Since the loop filter largely tracks the -20dB/decade jitter, its output is used to estimate $\sigma_W$ and used to calculate the optimum loop gain using (4.1). Unfortunately, this technique only applies to this particular jitter profile and loop filter. It’s reliance on FFT (performed off-chip in [40]) also makes it challenging to integrate on-chip.

Another approach shown in Fig. 4.2(c) is to estimate the jitter bandwidth from the PD output. In [42], the amplitude of the lowpass-filtered PD output is monitored. Since high-frequency jitter gets filtered more, the amplitude of the filter output indicates if the jitter bandwidth is high or low. This is illustrated in Fig. 4.4, which shows how the amplitude at
the LPF output drops off if jitter is at higher frequencies. The amplitude of the filter output is then compared to a reference level to determine if the jitter bandwidth is higher or lower than optimal. In [43] and [44], the consistency of the PD output is monitored over some window, also providing an estimate of the jitter bandwidth. Having estimated its bandwidth, if the jitter appears to be predominantly low frequency, the CDR increases its loop gain to suppress it, decreasing it otherwise.

The difficulty with this technique is that the jitter bandwidth which leads to optimal performance can depend on whether for example, jitter is dominated by a VCO whose jitter is concentrated at low frequencies, or ISI jitter which is broadband. Unless these jitter sources are known a priori, it is difficult to determine when the jitter bandwidth is optimal.

In this work, we show that simply maximizing the CDR’s loop gain and therefore loop bandwidth, leads to near-optimal jitter for a variety of jitter profiles, while achieving the high loop bandwidth desirable for meeting jitter tolerance requirements. To prevent the CDR from becoming underdamped, the autocorrelation function of the BB-PD output is monitored.

### 4.2 Analysis of Jitter in PI-based CDR

To first step in developing our loop gain adaptation strategy, is to analyze how the loop gain $K_G$ impacts the jitter of the CDR. To minimize the CDR’s BER, the relative jitter between the input data and CDR’s recovered clock $\psi_{ER}$ must be minimized. In this section, we identify and quantify the jitter sources contributing to $\psi_{ER}$ in a PI-based digital CDR. In addition to the jitter of the data ($\psi_{DAT}$) and reference clock ($\psi_{REF}$), jitter is introduced by quantization error from the PI ($\psi_{PI}$), BB-PD ($\psi_{PD}$) and majority voting blocks ($\psi_{MV}$).
4.2.1 PI Quantization Noise

$\psi_{PI}$ is caused by the nonlinearity and quantization error of the PI. The quantization noise for a PI with $N_{PI}$ phase steps per UI sets a lower bound for the $\sigma$ of $\psi_{PI}$ in units of UI.

$$\sigma_{PI} \geq \frac{1}{\sqrt{12N_{PI}}} \quad (4.2)$$

In this work, $N_{PI}$ is 64 steps per UI, which gives $\sigma_{PI} \geq 4.5\text{mUI}$ or 0.16ps at 28Gb/s. We approximate $\psi_{PI}$ as white noise for simplicity, although in reality, PI nonlinearity can lead to deterministic jitter in the presence of frequency offset.

4.2.2 Bang-bang PD Model

As described in chapter 2, the BB-PD can be modelled as a linear gain with additive quantization noise at its output [13]. We repeat the equations for $K_{PD}$ and the standard deviation of the PD quantization noise here for convenience.

$$K_{PD} = \sqrt{\frac{2}{\pi} \frac{\alpha_T}{\sigma_{ER}}} \quad (4.3)$$

$$\sigma_{PD} = \sqrt{\alpha_T - \frac{2}{\pi} \alpha_T^2} \quad (4.4)$$

As mentioned, $\alpha_T$ is the transition density of the input data and $\sigma_{ER}$ is the standard deviation of $\psi_{ER}$.

4.2.3 Majority Voting Noise Model

Majority voting among $N$ consecutive BB-PD outputs, each having a value of $\pm 1$ or 0, consists of summing the PD outputs and taking the sign of the result. It can therefore be modelled as a moving average filter $M(z^{-1})$, whose output $\psi_A$ is followed by a slicer. $M(z^{-1})$ is given by

$$M(z^{-1}) = \sum_{k=1}^{N} z^{-k} \quad (4.5)$$
The slicer can be modelled identically to a BB-PD, as a linear gain $K_{MV}$ followed by a quantization noise source $\psi_{MV}$. Instead of using simulation to find $K_{MV}$ as in [45], we follow the analysis of the BB-PD, and find $K_{MV}$ based on the probability density function of $\psi_A$.

If we assume $PD_{OUT}$, the output of the BB-PD is a random process with independently and identically distributed samples, then if $N$ is large, by the central limit theorem [29], summing up the PD outputs leads to a random process with a zero-mean Gaussian distribution. This gives a lower bound for $\sigma_A$.

$$\sigma_A \geq \sqrt{N} \sigma_{PD_{OUT}}$$  \hspace{1cm} (4.6)

Note that (4.6) is only a lower bound, since the BB-PD outputs could be correlated with each other, potentially increasing $\sigma_A$. Analogous to the analysis in [13], and since there is no effect from transition density, $K_{MV}$ and $\sigma_{MV}$, the $\sigma$ of $\psi_{MV}$ are then

$$K_{MV} = \sqrt{2 \pi} \frac{1}{\sigma_A}$$  \hspace{1cm} (4.7)

$$\sigma_{MV} \approx \sqrt{1 - \frac{2}{\pi}} \approx 0.6$$  \hspace{1cm} (4.8)

(4.8) assumes that $N$ is large enough to approximate $\psi_A$ as Gaussian and is somewhat pessimistic for smaller $N$. For $N=32$ in this design, directly calculating the PDF of $\psi_A$, leads to $\sigma_{MV} \approx 0.52$.

Figure 4.5: Jitter model of PI-based CDR
4.2.4 Complete Noise Model

The complete jitter model is shown in Fig. 4.5. Note that we have assumed that enough jitter is present to allow the system to be linearized [13, 15]. The combination of the demultiplexer (DeMUX) and majority voting operation leads to a downsampling operation since MV only produces one output for every $N$ PD outputs. This downsampling can cause aliasing of high-frequency jitter. Since the loop filter operates at a lower frequency, its output is also upsampled and followed by a zero-order hold (ZOH) to convert the jitter back to a full-rate signal. The dependence of $K_{PD}$ and $K_{MV}$ on the $\sigma$ of their inputs means this model must be solved iteratively, by recalculating $K_{PD}$ and $K_{MV}$ at each step. Using the model, we can determine $S_{ER}(f)$, the PSD of $\psi_{ER}$.

$$S_{ER}(f) = [S_{DAT}(f) + S_{REF}(f) + S_{PI, ZOH}(f)] \left| H^{-1}_{JTOL}(f) \right|^2 + \left[ \frac{S_{MV}(f)}{|K_{PD}M(f)K_{MV}|^2} + \frac{S_{PD}(f)}{|K_{PD}|^2} \right] \left| H_{JTRAN}(f) \right|^2$$

(4.9)

Here we have assumed that each jitter source is uncorrelated so that their noise power can be added together. As described in chapter 2, $H^{-1}_{JTOL}(f)$ and $H_{JTRAN}(f)$ are given by

$$H^{-1}_{JTOL}(f) = \frac{1}{1 + LG(f)}$$

(4.10)

$$H_{JTRAN}(f) = \frac{LG(f)}{1 + LG(f)}$$

(4.11)

$LG(f)$ is the CDR’s loop gain while $LF(f)$ is the loop filter response. The digital loop filter’s clock period is $NT$, while its latency is $N_D NT$. ($1/T$ is the data rate.)

$$LG(f) = K_{PD}K_GM(f)K_{MV}LF(f)N_P^{-1}e^{-j2\pi f N_D NT}$$

(4.12)

$LF(f)$ includes the effect of the phase accumulator.

$$LF(f) = \left( K_P + \frac{K_I}{1 - e^{-2\pi j NT}} \right) \frac{K_{PA}}{1 - e^{-2\pi j NT}}$$

(4.13)
As discussed in chapter 2, $H_{JTOL}^{-1}(f)$ has a highpass response while $H_{JTRAN}(f)$ has a lowpass response. $S_{PI,ZOH}$ includes the effect of the ZOH on $\psi_{PI}$. Since the output of the loop filter is already lowpass, the ZOH operation has little effect on its output and is otherwise ignored.

The jitter model is validated by comparing the PSDs of $\psi_{ER}$ as simulated in Simulink using the proposed jitter model, as well as a nonlinear model, where the BB-PD, majority voting and PI are not linearized. The results for several cases are shown in Fig. 4.6, which also plots the PSDs when calculated based on (4.9) using Matlab. The simulation results match quite well but the jitter model fails to predict some jitter at high frequencies in Case 2, likely caused by PI nonlinearity. Some inconsistencies in the values of $K_{PD}$ and $K_{MV}$ also cause small discrepancies between the simulated (Simulink) and calculated (Matlab) results of the jitter model.

**4.3 Finding Optimal Loop Gain**

We now use the jitter model to develop a strategy to optimize $K_G$. We first examine how to minimize $\sigma_{ER}$, and then consider how loop gain affects the CDR’s phase margin and loop bandwidth.
4.3.1 Jitter vs. Loop Gain

Increasing $K_G$ raises the corner frequencies of $H_{JTOL}^{-1}(f)$ and $H_{JTRAN}(f)$ as shown in Fig. 4.7. This reduces the contributions to $\psi_{ER}$ from $\psi_{DAT}$, $\psi_{REF}$ and $\psi_{PI}$, but increases the contributions from $\psi_{PD}$ and $\psi_{MV}$. We explore this tradeoff using two examples.

In Case I, we assume that $\psi_{DAT}$ consists of 500fs RMS white random jitter and that $\psi_{REF}$ comes from an oscillator with -80dBc/Hz phase noise at 1MHz offset and a -20dB/decade roll-off. To analyze the contribution of each jitter source to $\sigma_{ER}$, we use the jitter model described in the previous section to calculate and integrate the PSDs of each jitter source in Matlab. Fig. 4.8 plots each jitter contribution as $K_G$ varies. Raising $K_G$ from 1 to 5 significantly reduces $\sigma_{ER,DAT+REF}$, the jitter contributed by $\psi_{DAT}$ and $\psi_{REF}$. Because $\sigma_{ER,MV}$ and $\sigma_{ER,PD}$ are small, increasing $K_G$ reduces $\sigma_{ER}$ despite increasing the contributions from $\sigma_{ER,MV}$ and $\sigma_{ER,PD}$. However, when $K_G$ is increased too much, peaking starts to occur in both $H_{JTOL}^{-1}(f)$ and $H_{JTRAN}(f)$ as shown in Fig. 4.7, causing $\sigma_{ER,DAT+REF}$, $\sigma_{ER,PI}$ and overall jitter $\sigma_{ER}$ to
increase. In this example, $\sigma_{ER}$ can therefore be minimized by increasing $K_G$ until peaking in $H_{JTOL}^{-1}$ starts to increase overall jitter.

If however, $\psi_{DAT}$ and $\psi_{REF}$ are very small, or relatively broadband, increasing $K_G$ may not necessarily reduce $\sigma_{ER}$, as seen in Case II, where $\psi_{REF}$ represents the jitter of a PLL with -110dBc/Hz in-band phase noise and a 3dB bandwidth of 5MHz, while $\psi_{DAT}$ is 500fs RMS white random jitter. Although $\sigma_{ER,PD}$ and $\sigma_{ER,MV}$ have a minimal contribution to $\sigma_{ER}$, increasing $K_G$ still increases $\sigma_{ER}$, as shown in Fig. 4.9. To arrive at our proposed approach, we also consider the impact of the CDR’s loop gain on its loop bandwidth and phase margin.
4.3.2 Loop Bandwidth vs. Loop Gain

A high CDR loop gain and bandwidth are desirable and may even be required to meet stringent jitter tolerance masks [46]. In view of this, Fig. 4.10 re-plots the results for Case II, showing $\sigma_{ER}$ and the CDR’s phase margin as a function of its unity-gain frequency. Although $\sigma_{ER}$ is minimized when $K_G$ and the CDR’s bandwidth are at their lowest values, increasing $K_G$ can significantly improve loop bandwidth with only a small degradation in jitter. For example, moving from point A to B in Fig. 4.10 increases $\sigma_{ER}$ by less than 1% while the unity-gain frequency increases by 2.8x. Jitter only starts to increase significantly as the CDR’s phase margin drops below roughly 60°.

These examples show that a high loop gain is generally desirable in terms of both jitter and loop bandwidth. Even if the absolute lowest jitter is not achieved, maximizing the CDR’s loop gain can still achieve near-optimal jitter performance, while attaining the high loop bandwidth desired for meeting jitter tolerance requirements.

![Figure 4.10: Jitter and phase margin vs. unity gain frequency for Case II](image)

4.4 Proposed Loop Gain Adaptation Strategy

Given the above, our proposed strategy is to simply maximize $K_G$. The limit on increasing $K_G$ comes from having to maintain sufficient phase margin. Due to loop latency, increasing $K_G$ too much degrades the phase margin of digital CDRs, causing peaking in $H_{JTOL}^{-1}(f)$ and $H_{JTRAN}(f)$, and increasing jitter as we have seen. Poor phase margin also leads to undershoot in jitter tolerance and ringing in the impulse response of the CDR as shown in Fig. 4.11. When
Chapter 4. Adaptive Loop Gain CDR for Jitter Tolerance Optimization

Figure 4.11: (a) Normalized impulse response and (b) ideal jitter tolerance of CDR for different phase margins

the ringing becomes large its period, which we denote as $2n_{peak}$ approaches the inverse of $f_{180}$, the frequency at which the phase of the CDR loop gain reaches $-180^\circ$ (or $-\pi$ in radians), and at which the CDR can oscillate if it becomes unstable. We write $n_{peak}$ in units of UI.

$$n_{peak} \approx \frac{1}{2f_{180}T} \quad (4.14)$$

$$\angle LG(f_{180}) = -180^\circ \quad (4.15)$$

Any ringing in the CDR’s impulse response causes corresponding damped oscillations to occur in $\psi_{ER}$. This can be observed by measuring the CDR’s step response [47], but this is difficult to accomplish on-chip. Instead, the ringing in $\psi_{ER}$ can be observed by monitoring the autocorrelation function of the BB-PD output $R(n)$. Ringing causes $R(n)$ to dip at $n_{peak}$ as shown in Fig. 4.12. This is consistent with the peaking observed near $f_{180}$ in $S_{ER}(f)$, as shown in the same figure. Our adaptation strategy is therefore to increase $K_G$ and therefore the CDR’s loop bandwidth, while monitoring and preventing any ringing in $R(n)$.

4.4.1 Proposed Adaptive Loop Gain CDR

The proposed adaptive loop gain CDR is shown in Fig. 4.13. The adaptation logic monitors the autocorrelation function of the lowpass-filtered PD output and increases the CDR loop gain
Figure 4.12: (a) Spectrum and (b) corresponding autocorrelation function of BB-PD for various loop gain settings showing peaking caused by excessive loop gain

Figure 4.13: Overview of proposed adaptive loop gain CDR

$K_G$, as long as $R(n_{\text{peak}})$ is greater than the decision threshold $R_{TH}$, decreasing it otherwise. Since ringing causes $R(n_{\text{peak}})$ to fall below zero, and based on simulation results such as those in Fig. 4.11, setting $R_{TH} = 0$ leads to a phase margin of approximately 60°, providing a good tradeoff between CDR bandwidth and undershoot in jitter tolerance.

As shown in Fig. 4.13, $R(n)$ is estimated by correlating the output of the lowpass-filtered BB-PD $PD_{OUT}$ with delayed versions of itself and taking the average. This gives the time-
averaged autocorrelation function:

\[
R_{\text{Estimated}}(n) = \frac{1}{L} \sum_{k=1}^{L} PD_{\text{OUT}}(k)PD_{\text{OUT}}(k - n) \quad (4.16)
\]

Since the BB-PD only outputs ±1 or 0, the correlation operation is greatly simplified. By only counting samples where \( PD_{\text{OUT}} \neq 0 \), the measured \( R(n) \) waveform always has a fixed maximum amplitude equal to 1 (i.e. \( R(0) = 1 \)).

### 4.4.2 Limitations of Proposed Technique

The proposed technique has certain limitations. Firstly, if the jitter of the reference clock, PI and data are not much larger than the jitter caused by majority voting, the proposed adaptation will become sub-optimal in minimizing \( \sigma_{ER} \).

In addition, in observing \( R(n_{\text{peak}}) \), it is assumed that any ringing in \( R(n) \) is caused by the CDR’s impulse response. Ringing caused by other jitter sources could therefore interfere with adaptation. This could happen if for example, the reference clock is generated by a severely underdamped PLL, whose jitter already contains ringing. We show later how this can be handled if the jitter is largely sinusoidal, but other cases could be more difficult to address. Lastly, the profile of the data and reference clock jitter can have an effect on the value of \( R(n_{\text{peak}}) \), causing the CDR’s phase margin to vary between 55° and 60° after adaptation, according to simulations.

Next, we discuss two components of the adaptive loop gain CDR. First, we explain why the BB-PD output must be lowpass-filtered before estimating \( R(n) \). We then describe how \( n_{\text{peak}} \) can be found adaptively on-chip.

### 4.4.3 PD filtering

As seen in Fig. 4.12(b), \( R(n) \) generally includes a delta-function, which is the autocorrelation function of any white random jitter in the BB-PD output, including PD and MV quantization noise and ISI jitter. Because \( R(n) \) has a fixed maximum amplitude, in cases where large white jitter is present, the delta-function becomes large compared to the rest of \( R(n) \), which represents the autocorrelation of the other jitter sources. This makes it difficult to detect ringing as shown
Figure 4.14: $R(n)$ for different $K_G$ values measured using (a) raw and (b) lowpass filtered BB-PD output in the presence of high white random jitter and (c) corresponding values of $R(n_{\text{peak}})$

in Fig. 4.14(a) which plots $R(n)$ for different $K_G$ values when a uniformly distributed white jitter of 0.1UI$_{PP}$ is present. The changes in $R(n)$ caused by varying $K_G$ are difficult to detect.

To prevent this, we lowpass filter the BB-PD output to suppress white jitter, reducing the height of the delta-function compared to the rest of $R(n)$. The result is shown in Fig. 4.14(b) where filtering makes the peaking in $R(n)$ much more evident. The effect is further illustrated in Fig. 4.14(c), which plots $R(n_{\text{peak}})$ with and without filtering of the PD output. As shown in the figure, the slope of $R(n_{\text{peak}})$ as a function of phase margin is dramatically increased when the PD output is filtered.

The remaining question is how to determine $n_{\text{peak}}$. We first gain some intuition by estimating $n_{\text{peak}}$ using a simplified analysis, before describing how $n_{\text{peak}}$ can be found more reliably by measuring it on-chip.
4.4.4 Estimating $n_{\text{peak}}$ Analytically

We have defined $n_{\text{peak}}$ as half of the oscillation period when the CDR undergoes a damped oscillation. While nonlinear analysis [48] has been used to analyze such oscillations, we adopt a simpler linear analysis using the linear model of Fig. 4.15, where majority voting is ignored for simplicity, and the loop filter has a latency of $N_D$ cycles as before. The simplified loop gain is given by:

$$LG(z^{-1}) = K_{PD} \left[ K_P + \left( \frac{K_I}{1 - z^{-1}} \right) \right] \frac{z^{-N_D}}{1 - z^{-1}}$$  \hspace{1cm} (4.17)

Recalling that the loop filter’s sample rate is $1/NT$, we can replace $z^{-1}$ by $e^{-j2\pi fNT}$. By assuming that $f << \frac{1}{2\pi NT}$, we can approximate $e^{-j2\pi fNT}$ as $1 - j2\pi fNT$. We also assume that the CDR has some additional analog delay $T_D$, giving

$$LG(f) \approx K_{PD}K_I \left[ 1 + j2\pi fNT \frac{K_P}{K_I} \right] \frac{1}{-4\pi^2(NT)^2} e^{-j2\pi f(NDNT+TD)}$$  \hspace{1cm} (4.18)

The CDR can oscillate if $|LG(f_{180})| = 1$, where $\angle LG(f_{180}) = -\pi$ and $\angle LG(f)$ is given by

$$\angle LG(f) \approx -\pi + \tan^{-1} \left( 2\pi fNT \frac{K_P}{K_I} \right) - 2\pi f(NDNT + T_D)$$  \hspace{1cm} (4.19)

$f_{180}$ can then be found from

$$\tan^{-1} \left( 2\pi f_{180}NT \frac{K_P}{K_I} \right) = 2\pi f_{180}(NDNT + T_D)$$  \hspace{1cm} (4.20)
If \( K_P \gg K_I \) and we approximate the loop filter as being first order, the LHS of (4.20) is approximately \( \pi/2 \), giving

\[
f_{180} \approx \frac{1}{4(N_DNT + T_D)} \tag{4.21}
\]

This result is similar to that of [48], where inserting our variable names, the equivalent \( f_{180} \) was found to be \( 1/(2 + 4N_DNT) \) for a first order loop filter. \( n_{peak} \) is then calculated from (4.14).

\[
n_{peak} \approx \frac{2(N_DNT + T_D)}{T} \tag{4.22}
\]

The above shows that \( n_{peak} \) is mainly a function of CDR loop latency \( N_DNT + T_D \), but relies on several simplifications, ignoring the effect of \( K_I \) and majority voting. Furthermore, \( T_D \) must include the delays of all phase interpolator, clock tree, retiming, demux and digital circuitry within the CDR feedback loop. These analog delays can be challenging to accurately characterize, and may also be sensitive to PVT variation. For the above reasons, we choose to find \( n_{peak} \) adaptively on-chip, avoiding the need for simplifying assumptions or detailed, mixed-mode simulations of CDR loop latency.

Figure 4.16: (a) Concept behind adaptation of \( n_{peak} \) and (b) feedback loop used to identify \( n_{peak}/2 \)
4.4.5 Adaptation of $n_{\text{peak}}$

We determine $n_{\text{peak}}$ adaptively by initially setting $K_G$ to its highest value and finding the period of the resulting damped oscillation that can be observed in $R(n)$. $R(n_{\text{peak}})$ could be found as the minimum of $R(n)$, but that would require a relatively slow gradient descent or search algorithm. Instead, we use the fact that $R(n_{\text{peak}}/2) \approx 0$ to estimate $n_{\text{peak}}/2$ using a feedback loop as shown in Fig. 4.16. Once $n_{\text{peak}}$ is measured, it is stored as $n_{\text{peak,REF}}$ and used for the rest of the adaptation process.

4.4.6 Previous Works Using Autocorrelation

While this work was being developed, several works were independently published, also using the autocorrelation function of the BB-PD output in attempting to minimize the jitter of a PLL [49, 50] or CDR [51]. Our technique was developed independently of these works, which use similar concepts but suffer from several limitations. The approach of [49] is similar to this work, attempting to drive the autocorrelation function of a PLL’s BB-PD output to zero at $n = 2D + 1$. $D$ is the delay of the digital loop filter, making $2D + 1$ equivalent to $n_{\text{peak}}$ defined above when using the nonlinear analysis of [48]. Although [49] claims that this minimizes the PLL’s output jitter, based on our analysis, this is only the case if the reference clock jitter is extremely low (i.e. when $\psi_{\text{DAT}}$ is small), which is not true for example, in PLLs used to filter jitter. In [50], $R(n)$ for a PLL is observed at several arbitrarily chosen points, based on the analysis in [52], which assumes that the PLL always remains stable, ignoring the possibility of instability, which we have shown can be the limiting factor in minimizing jitter.

Observing $R(n)$ at calculated and fixed, rather than adapted values of $n$ makes these works sensitive to variations in loop latency as discussed previously. Their lack of filtering of the BB-PD output also makes them sensitive to white jitter such as ISI jitter, making them ill-suited for use in CDRs. The system proposed in [51], where similar techniques are applied to a CDR, suffers from the above and additional limitations.

The CDR in [51] monitors and attempts to drive $R(D + 1)$ to zero in a CDR. $D + 1$ is approximately $n_{\text{peak}}/2$ in our analysis, meaning that $R(D + 1)$ will generally be near zero even when the CDR’s phase margin is poor, making it a poor criterion for optimizing the CDR loop.
gain.

Finally, all previous approaches are also ineffective in the presence of sinusoidal jitter. We show next how additional logic enables our scheme to operate in the presence of sinusoidal jitter.

4.5 Adaptation in Presence of Sinusoidal Jitter (Dynamic Mode)

We have thus far assumed that the adaptation scheme described in section 4.4 and henceforth referred to as basic adaptation, operates without any large periodic or sinusoidal jitter being present. This requires that adaptation occur while major periodic noise sources such as nearby I/O drivers are disabled. Having completed basic adaptation, the results now stored as $n_{\text{peak,REF}}$ and $K_{G,\text{REF}}$, can then be used to enhance the CDR’s tracking performance when it is subjected to sinusoidal jitter, by enabling additional logic in what we describe in this section as dynamic mode.

4.5.1 Proposed Dynamic Adaptation Scheme

While basic adaptation optimizes the CDR’s jitter performance when random jitter dominates, the selected loop gain $K_{G,\text{REF}}$ may not be the best value if the CDR is then subjected to sinusoidal or periodic jitter. If SJ is dominant, and its frequency is within the CDR’s bandwidth, $K_G$ should increase to better suppress it. If the SJ is out-of-band, the CDR should simply ignore it and default to $K_{G,\text{REF}}$. This is similar to the loop gain adaptation schemes based on jitter bandwidth [42–44] and is effective in the case of SJ, but as mentioned in section 4.4.6, is otherwise only applicable if the jitter spectrum is known a priori. Note that very high-frequency SJ will already have been suppressed by lowpass filtering the PD output, and will be ignored as desired.

Dynamic mode operates as in basic mode unless SJ is detected, in which case $K_G$ is chosen based on the bandwidth of the SJ. To detect SJ in dynamic mode, $R(n_{\text{peak}})$ is monitored while $n_{\text{peak}}$ continuously adapts to detect oscillations at any frequency. If $R(n_{\text{peak}})$ stays below $R_{TH}$ even when $K_G$ is reduced, the ringing must not be caused by poor phase margin. Accordingly, the system treats the jitter as SJ and switches to the bandwidth-based strategy.
This bandwidth-based strategy relies on being able to estimate both the SJ frequency, and the maximum tracking bandwidth of the CDR $f_{3dB,MAX}$. We explain how these can be found next.

### 4.5.2 Detection of Sinusoidal Jitter

If SJ $\psi_{ER} = A_0 \sin(2\pi f_0 t + \phi)$ becomes the dominant source of jitter, the time-averaged autocorrelation function $R(n)$ defined by (4.16) will have the form

$$R(n) \approx \frac{K^2_{PD} A_0^2}{2} \cos(2\pi f_0 nT)$$

(4.23)

We have assumed that $f_0 << 1/T$, so that averaging $R(n)$ over many samples removes the effect of the phase offset $\phi$. Because $R(n)$ is now mostly sinusoidal, the $n_{peak}$ adaptation scheme, which finds the zero-crossing of $R(n)$ can be used to estimate the frequency of the SJ as seen in Fig. 4.17(a).

$$f_{SJ,Estimated} = \frac{1}{2n_{peak}T}$$

(4.24)

Note that because $R(n)$ now contains a large sinusoid at $f_0$, the ringing in $R(n)$ used to drive basic adaptation may no longer be visible. This is why dynamic mode still requires the previously obtained values $n_{peak,REF}$ and $K_{G,REF}$ as references. Next we discuss how to find $f_{3dB,MAX}$.

### 4.5.3 Determining Maximum Tracking Bandwidth

As discussed in section 4.4.4, $n_{peak,REF}$ (adapted without SJ) depends on $f_{180}$. Using the same analysis, we can define $f_{3dB,MAX}$ as the bandwidth corresponding to 60° phase margin. $f_{3dB,MAX}$ is then

$$\angle L(f_{3dB,MAX}) = -\frac{2\pi}{3}$$

(4.25)

Assuming as in section 4.4.4, that $K_P >> K_I$, and using (4.19) gives

$$f_{3dB,MAX} = \frac{f_{180}}{3}$$

(4.26)
Equations (4.24), (4.14) and (4.26) can then be combined to give the condition on $n_{\text{peak}}$ (measured with SJ present) under which the adaptation scheme should increase the CDR loop gain:

$$n_{\text{peak}} > 3n_{\text{peak,REF}}$$  \hspace{1cm} (4.27)

*Dynamic* mode requires some additional circuitry compared to *basic* mode. Whereas in *basic* mode, $n_{\text{peak}}$ is only adapted once initially and stored as $n_{\text{peak,REF}}$, in *dynamic* operation, $n_{\text{peak}}$ is continuously monitored. *Dynamic* adaptation also monitors both $R(n_{\text{peak}})$ and $R(n_{\text{peak,REF}})$ since ringing could be caused either by ringing at $f_{180}$ (detected from $R(n_{\text{peak,REF}})$), or SJ at another frequency (detected from $R(n_{\text{peak}})$).

### 4.5.4 Effect of Multi-Tone SJ

The above becomes more complicated if SJ is present at multiple frequencies. If $\psi_{\text{ER}}$ is the sum of $M$ sinusoids each with amplitude $A_i$ and frequency $f_i$, the resultant $R(n)$ will become

$$R(n) \approx \frac{K_{PD}^2}{N} \sum_{i=1}^{M} \frac{A_i^2}{2} \cos(2\pi f_i n T)$$  \hspace{1cm} (4.28)

Clearly if the amplitude at one SJ frequency is dominant, that frequency will dominate $R(n)$, but if several $A_i$’s are similar the result is not obvious. We take as an example, SJ at two frequencies $f_1$ and $f_2$, each with equal amplitude $A$. $R(n)$ is then plotted for two conditions.

Fig. 4.17(b) plots $R(n)$ when $f_1$ and $f_2$ are close together. The dominant oscillation in $R(n)$ near $n = 0$, is at the average of the frequencies. This is convenient as the zero-crossing of $R(n)$ provides a good estimate of the SJ frequencies. In Fig. 4.17(c), $f_1 \gg f_2$ and the oscillation at $f_1$ appears superimposed on the lower frequency oscillation at $f_2$. Since $R(n)$ does not cross zero at $1/4f_1 T$, estimating the SJ frequency from the zero crossing of $R(n)$ may yield $f_2$ instead of $f_1$.

Underestimating the SJ frequency could drive *dynamic* adaptation to an excessive loop gain setting. We therefore want to bias the SJ frequency estimate towards higher frequency $f_1$. As shown in Fig. 4.17(c), this can be done by estimating $n_{\text{peak}}$ using the width of $R(n)$ (where $R(n) = 0.5$), instead of its zero crossing. In *dynamic* mode, we therefore apply an offset in the
Figure 4.17: Autocorrelation of (a) single-tone SJ, (b) two-tone SJ with equal amplitudes and \( f_1 \approx f_2 \) and (c) two-tone SJ with equal amplitudes and \( f_1 \gg f_2 \)

\( n_{\text{peak}} \) adaptation block when estimating the SJ frequency, providing an estimate that will be closer to the higher frequency \( f_1 \), as desired.

4.6 Implementation

4.6.1 Analog Front-End

Both the basic and dynamic adaptive loop gain schemes were implemented in a 28Gb/s half-rate PI-based digital CDR shown in Fig. 4.18.

The CDR’s analog front-end includes the continuous time linear equalizer (CTLE) shown in Fig. 4.19, which combines active feedback [53] with an inverter-based second stage to improve bandwidth and DC gain. This configuration is problematic though, since the combination of the inverter stage with a conventional differential feedback amplifier such as a differential pair, will create a positive common-mode feedback loop. To prevent this, the feedback stage is replaced with a pseudo differential transconductor with positive common mode gain.

Half-rate quadrature clocks are generated from an external reference clock using a two-stage injection-locked oscillator (ILO), which feeds 7-bit CMOS phase interpolators. The upper 2 bits
of the PI code select the polarities of the two input clocks to each PI while the lower 5 bits control their weighting by enabling and disabling inverter slices for each input. The 5-bit inverter banks consist of 16 thermometer-coded slices and a half-size LSB slice to reduce area.

The half-rate data, edge and eye monitor samples are demuxed by 16, forming 32-bit buses that are sent to the synthesized digital core operating at 875MHz.

### 4.6.2 Digital Backend

The PD takes each set of 32 demuxed edge and data samples and generates 32 corresponding early and late signals. Majority voting is used to convert these to a single 2-bit value of +1, 0 or -1. This is then fed to the adaptation block and LF whose implementations are thus greatly
The second-order digital loop filter is shown in Fig. 4.21. Because majority voting is used, implementing the gain $K_G$ only requires a mux instead of a multiplier. In this work $K_G$ is a 4-bit value that changes in linear steps from 1 to 15. Since adaptation compensates for variations in jitter, which is not expected to change by more than an order of magnitude, this should offer sufficient programmability. A finer $K_G$ step size could be used but that would also increase power consumption. To avoid using multipliers, the integral and proportional gains $K_I$ and $K_P$ are powers of two. The outputs of both the majority voting and decoder blocks are resampled, adding two additional cycles of latency and giving a total digital latency of four cycles.
Chapter 4. Adaptive Loop Gain CDR for Jitter Tolerance Optimization

The implementation of $R(n)$ measurement is shown in Fig. 4.22. The block takes the 2-bit output of the majority voting block and further lowpass filters it using another FIFO and majority voting stage. Majority voting conveniently provides the desired lowpass filtering effect while maintaining a binary output, which is easily processed by the $R(n)$ block. The LPF bandwidth is adjusted by selecting the range of FIFO samples over which voting occurs. The filtered signal is then correlated with itself delayed by an adjustable FIFO. As in [54], $R(n)$ itself is generated by two counters, one which counts the total number of transitions and another which counts the number of correlated samples, producing $R(n)$ as a digital code. In this work,
4.7 Measurement Results

The test chip was fabricated in 28nm CMOS and consumes 106.6mW or 3.82pJ/bit at 28Gb/s with the eye monitor circuits (only used for diagnostics) consuming roughly 12% of the total power. Although the eye monitors cannot be completely disabled, removing them should improve power efficiency to 3.35pJ/bit. The die photo, area and power breakdowns are shown in Fig. 4.23.

4.7.1 Test Setup and Chip Functionality

The test chip was wire-bonded to a PCB as shown in Fig. 4.24, making the wire-bond the dominant source of attenuation on the input data. As shown in Fig. 4.25, the half-rate 14GHz reference clock is supplied by a Rhode and Schwarz SMB100A signal generator. The SMB100A
was phase and frequency modulated with random noise from a NoiseCom noise source to generate different phase noise profiles.

Fig. 4.26 shows the quarter-rate (7Gb/s) PRBS31 data and half-rate (14GHz) clock recovered by the CDR. Error-free operation was verified by feeding the recovered quarter-rate data to a Centellax TG1B1-A BERT. Since however, error-checking could not be performed simultaneously on all four quarter-rate recovered data streams, the external BERT results were optimistic compared to those from the on-chip BERT, which checks the entire deserialized data.
4.7.2 Adaptation Performance (Basic Mode)

The first step of adaptation is finding $n_{\text{peak}}$. As discussed, this is done by setting $K_G$ to its highest setting and enabling the $n_{\text{peak}}$ adaptation block. The result of adaptation is shown in Fig. 4.27(a) while Fig. 4.27(b) shows the complete $R(n)$ waveform measured for the same condition. Since the adaptation circuits operate on the output of the majority voting block, the precision of $n_{\text{peak}}$ is limited to the nearest 32UI. $n_{\text{peak}}$ converges to an average value of approximately 300UI, which is close to where $R(n)$ reaches its minimum in Fig. 4.27(b). After adaptation, $n_{\text{peak,REF}}$ is rounded to 320UI.

The rest of adaptation was then performed in basic mode. Measurements were performed with the reference clocks having three different phase noise profiles and frequency offsets. In Case 1, the SMB100A was phase modulated to give a lowpass phase noise characteristic similar to that of a PLL, with -80dBc/Hz in-band phase noise. In Cases 2 and 3, the SMB100A was FM-modulated, producing reference clock phase noise profiles with -20dB/decade roll-off similar to a free-running VCO. Due to equipment limitations, the modulation bandwidth was limited to 1MHz. The above-described phase noise profiles are plotted in Fig. 4.28(a).

Jitter tolerance was measured with the $K_G$ set to minimum, maximum, and following adaptation. The results plotted in Fig. 4.28(b) show that the maximizing $K_G$ causes undershoot in the jitter tolerance in all three test cases. When $K_G$ is minimized, BER remains above $10^{-12}$ in all cases as the CDR is unable to suppress the phase noise of the reference clock.
adaptation, well-behaved jitter tolerance is achieved in all three cases.

To better quantify the adaptation’s performance, and given the CDR’s bandwidth of approximately 10MHz, we examine the lowest out-of-band JTOL measured from 10MHz to 100MHz for a BER<10^{-12} and PRBS31 data. The lowest JTOL over this range is taken to ensure that no undershoot is present. As shown in Fig. 4.28(c), adaptation leads to near-optimal high-frequency jitter tolerance in all cases. Note that in these measurements, $K_G$ is adapted prior to applying SJ and held at the adapted value ($K_{G, REF}$) during JTOL tests.

The relationship between $K_G$ and the jitter of the CDR’s recovered half rate clock is also
Figure 4.28: (a) Phase noise profiles of Ref Ck, (b) jitter tolerance for adapted, min and max $K_G$ (c) minimum jitter tolerance measured between 10-100MHz after adaptation and (d) recovered clock jitter vs. $K_G$ for three test cases.

plotted in 4.28(d). The CDR clock jitter was measured by an Agilent DCA-86100D sampling scope with precision timebase module as well as an Agilent EXA 9010A spectrum analyzer, by integrating the measured phase noise from 10kHz to 100MHz. As seen in Fig. 4.26, the rising edge of the of the recovered clock suffered from deterministic jitter likely caused by PI-induced
DCD, so the scope measurement was taken on the falling edges, whose jitter closely matched with the spectrum analyzer measurements.

The results show that the adapted $K_G$ values achieve close to the lowest achievable clock jitter. However, they also reveal that clock jitter is a relatively weak predictor of the jitter tolerance performance plotted in Fig. 4.28(c). In all cases, setting $K_G$ to its highest setting lead to significant undershoot in the jitter tolerance response but in Cases 1 and 3, the recovered clock jitter remains close to optimal. This demonstrates that recovered clock jitter alone is not a very strong predictor of good jitter tolerance.

The measured $K_G$ adaptation curves for each of the three test cases are plotted in Fig. 4.29. The adaptation process is relatively slow due to the extensive use of filtering and a slow loop filter in the adaptation logic. Hysteresis was used to prevent dithering after adaptation has converged.

The effect of lowpass filtering on the $R(n)$ measurement is shown in Fig. 4.30. To generate $R(n)$ without filtering, sub-sampled PD data was taken off-chip and correlated on an FPGA. Filtering with 1x BW corresponds to $R(n)$ measured on-chip after the first MV stage in the CDR. The 1/3x and 1/9x BW curves are taken after additional filtering by the second MV stage. Note that all of the filtered $R(n)$ curves use downsampled data and only have a time resolution of 32UI.

The figure confirms that without filtering, the oscillations in $R(n)$ are almost impossible to
discern as the unfiltered $R(n)$ curve is dominated by a large delta-function and has large ripple caused by DCD. Filtering with 1x and 1/3x BW improves visibility of the oscillation while filtering with 1/9x becomes excessive, adding undesirable phase distortion and moving the zero crossing and peak locations.

![Effect of Filtering on R(n)](image)

Figure 4.30: Measured $R(n)$ with and without lowpass filtering of PD output

### 4.7.3 Adaptation Performance (Dynamic Mode)

In the above measurements, $K_G$ is adapted prior to applying SJ and held at the adapted value ($K_{G,REF}$) during JTOL tests. To demonstrate the effectiveness of dynamic adaptation, where $K_G$ can adapt based on the SJ frequency, Fig. 4.31 plots the highest SJ amplitude for which the CDR can maintain $\text{BER} < 10^{-12}$, measured for the same test cases described earlier. The results of basic adaptation (where $K_G$ stays at $K_{G,REF}$ after adaptation) are compared to the dynamic case. Both results are also compared to the best possible result, where $K_G$ is manually tuned at each frequency. While basic adaptation optimizes the CDR's ability to tolerate high-frequency SJ, enabling dynamic operation further improves tracking of in-band jitter, nearly matching the performance achieved if $K_G$ is manually tuned.

Finally, Table 4.1 compares this work to prior adaptive loop gain CDRs. Because prior works use several different methods to characterize their adaptation performance, such as jitter, BER or jitter tolerance, we are limited to a qualitative comparison. Other characteristics such as adaptation time are also generally not reported in prior works and can therefore not be
Test Case 1
Measured Maximum Tolerable SJ (BER<10^{-12})

- **Max over all** $K_G$
- **Adapted**: Basic
- **Adapted**: Dynamic

$K_G$ increases when jitter is mostly in-band.

Test Case 2
Measured Maximum Tolerable SJ (BER<10^{-12})

- **Max over all** $K_G$
- **Adapted**: Basic
- **Adapted**: Dynamic

Test Case 3
Measured Maximum Tolerable SJ (BER<10^{-12})

- **Max over all** $K_G$
- **Adapted**: Basic
- **Adapted**: Dynamic

Figure 4.31: Maximum SJ the CDR can tolerate; comparing max over all $K_G$ settings, to basic adaptation ($K_G$ adapted and fixed before applying SJ) and dynamic adaptation ($K_G$ dynamically adapts to input jitter).

4.8 Summary

In this chapter, an adaptive loop gain PI-based CDR has been demonstrated, using the autocorrelation of the BB-PD output to maximize loop gain while preventing the CDR from becoming too underdamped. The proposed CDR adapts to achieve near-optimal high-frequency jitter tolerance and recovered clock jitter, without requiring knowledge of the input jitter profile. Unlike prior works using autocorrelation, the proposed work does not use any pre-calculated parameters, making it robust to variations and easily portable between CDR designs. Filtering
of random jitter and the option to employ dynamic adaptation makes the system robust to any large white RJ or SJ that may be encountered.
Chapter 5

Jitter Injection for On-Chip Jitter Measurement in PI-Based CDRs

The jitter measurement technique presented in chapter 3 provides observability of clock and data jitter in multilane CDRs, but is relatively complex and only applicable when multiple CDRs are available. In chapter 4, we showed that the same autocorrelation concepts used in chapter 3 can be used to optimize the jitter tolerance of a CDR, without actually measuring the jitter's amplitude. While effective, this technique could only be applied to the CDR’s loop gain and not other parameters such as equalizer settings. Without the ability to measure jitter, this approach also cannot be guaranteed to always achieve the lowest possible jitter. In contrast, on-chip jitter measurement allows any CDR parameter to be optimized, by directly minimizing jitter using gradient descent [38] or other optimization techniques. To achieve the lowest BER, the relative jitter between the CDR’s input data and recovered clock must be minimized.

In this chapter, we propose a simple technique to measure the RMS value of the relative jitter in PI-based CDRs. The proposed technique does not rely on dedicated analog circuits such as eye monitors and makes use of existing circuits in the implemented half-rate PI-based CDR.

Section 5.1 of this chapter provides a brief introduction while section 5.2 describes the proposed technique in more detail. The implementation and measurement results are presented in sections 5.3 and 5.4.
Chapter 5. Jitter Injection for On-Chip Jitter Measurement

5.1 Introduction

As described in chapter 3, eye monitors [39,54–56] have traditionally been used to measure the relative jitter between the data and clock. An example of this scheme is shown in Fig. 5.1(a) where a dedicated eye sampler is added to a CDR and driven with an adjustable clock phase $CK_{EYE}$. As the clock phase is swept, the jitter histogram can be plotted by measuring the eye sampler’s BER or comparing its output to adjacent data samples taken by the CDR. Since no ideal reference clock is used, this approach measures the relative jitter between the input data and clock. However, eye monitor circuits consume additional area and power, and also increase loading on critical high-speed data and clock paths.

In this work, we avoid the use of eye monitor circuits and rely instead on the bang-bang PD (BB-PD) to measure jitter in a PI-based digital CDR. To do this, we add a square wave signal to the PI code, thereby injecting square wave jitter into the CDR’s edge clock $CK_{EDGE}$ as shown in Fig. 5.1(b).

Figure 5.1: (a) Conventional eye monitor-based vs (b) proposed autocorrelation-based jitter measurement

loading on critical high-speed data and clock paths.
in Fig. 5.1(b). Given that the injected jitter amplitude is known, the RMS relative jitter can be calculated by measuring the effect of this injected jitter on the autocorrelation function of the BB-PD output $R(n)$. We implement by adding digital circuits to a 28Gbps 2x half-rate CDR as shown in Fig. 5.1(b), without impacting the design of the analog frontend. Measured results from the test chip, fabricated in 28nm CMOS, demonstrate that this scheme can estimate the RMS relative jitter at the CDR input with an accuracy well below one picosecond. In addition, this scheme correctly predicts the loop gain, and equalizer settings that minimize jitter.

### 5.2 Proposed Technique

Fig 5.2(a) shows a model of the CDR where the BB-PD is replaced with its linear model, which consists of a linear gain $K_{PD}$ and some added quantization noise $\psi_{PD}$. If $\psi_{INJ} = 0$, the relative jitter between the clock and data $\psi_{ER}$ has contributions from jitter present in the input data ($\psi_{DAT}$), reference clock ($\psi_{REF}$), PI ($\psi_{DAT}$) and PD ($\psi_{PD}$). In this work, we monitor the BB-PD output $PD_{OUT}$ and estimate its autocorrelation function as shown in Fig 5.2(a). If $\psi_{ER}$ is mostly white, $R(n)$ will be dominated by a delta-function as shown in Fig. 5.2(b). In this case, the RMS value of $\psi_{ER}$ ($\sigma_{ER}$) can be extracted from $R(n)$, which is given by

\[
R(n) = E[PD_{OUT}(k)PD_{OUT}(k-n)] \\
= K_{PD}^2 E[\psi_{ER}(k)\psi_{ER}(k-n)] + \sigma_{PD}^2 \delta[n]
\]

Here, we have assumed that the jitter is stationary so that $R(n)$ does not depend on the time index $k$, and is only a function of the time shift $n$. It is also assumed that $\psi_{PD}$ is white so that its autocorrelation function is a delta-function. Evaluating $R(n)$ at $n = 0$ gives

\[
R(0) = K_{PD}^2 \sigma_{ER}^2 + \sigma_{PD}^2
\]

Because the BB-PD only outputs ±1, or zero when there is no transition, $R(0) = E[PD_{OUT}^2]$ is also equal to $\alpha_T$, the transition density of the data. Setting (5.3) equal to $\alpha_T$, we can solve
for $\sigma_{ER}$

$$
\sigma_{ER} = \sqrt{\frac{\alpha_T - \sigma_{PD}^2}{K_{PD}^2}}
$$  \hspace{1cm} (5.4)

The expressions for $\sigma_{PD}$, the RMS value of $\psi_{PD}$ and $K_{PD}$ are repeated below for convenience.

$$
\sigma_{PD} = \sqrt{\alpha_T - \frac{2}{\pi} \sigma_T^2}
$$  \hspace{1cm} (5.5)
To determine $K_{PD}$, an eye monitor could be used \[54\], but that would require additional analog hardware. Instead, in this work, we inject square wave jitter $\psi_{INJ}$ with amplitude $A$ into the CDR and estimate $R(n)$. Since $\psi_{INJ}$ has a 50% duty cycle, $\sigma_{INJ} = A$. As we explain next, the injected jitter causes a triangular wave to appear in $R(n)$, whose amplitude can be used to extract $K_{PD}$ and $\sigma_{ER}$.

In the presence of $\psi_{INJ}$ and assuming that $A$ is small compared to $\psi_{ER}$, the probability distribution of the total jitter $\psi_{EFF} = \psi_{ER} + \psi_{INJ}$ still appears mostly Gaussian, but now has a $\sigma$ equal to $\sigma_{EFF} = \sqrt{\sigma_{ER}^2 + \sigma_{INJ}^2}$. Due to $\psi_{INJ}$, a small triangular wave, which is the autocorrelation function of a square wave, also appears in $R(n)$ with amplitude $\Delta$ as seen in Fig 5.2(b). Since $\psi_{INJ}$ is uncorrelated with $\psi_{ER}$, which is random, $\Delta$ is simply the power of $\psi_{INJ}$ at the output of the PD, whose gain we now denote as $K_{PD,EFF}$.

$$\Delta \approx K_{PD,EFF}^2 \sigma_{INJ}^2$$ (5.6)

Measuring $\Delta$ therefore allows us to find $K_{PD,EFF}$, and by extension $\sigma_{ER}$ as we now see. $R(0)$ now has an added contribution from $\psi_{INJ}$.

$$R(0) = \alpha_T = K_{PD,EFF}^2 \left[ \sigma_{ER}^2 + \sigma_{INJ}^2 \right] + \sigma_{PD}^2$$ (5.7)

Combining (5.5), (5.6) and (5.7) gives $\sigma_{ER}$ as a function of $\Delta$.

$$\sigma_{ER} \approx A \sqrt{\frac{2 \alpha_T^2}{\pi \Delta}} - 1$$ (5.8)

Since the BB-PD output is binary, measuring $R(n)$ and $\Delta$ only requires simple logic and digital counters. Because $\alpha_T$ is known, $\sigma_{ER}$ can be precomputed and stored in a lookup table as a function of $\Delta$.

### 5.3 Implementation

The proposed technique was implemented by adding digital logic to the half-rate PI-based 28Gb/s digital CDR described in chapter 4. As described in chapter 4 and shown again in Fig.
5.3, the demuxed data is sent to the digital core clocked at 875MHz. Majority voting (MV)

![Diagram](image-url)

Figure 5.3: Overview of Half-Rate CDR from chapter 3 showing added jitter injection function

reduces the parallel PD outputs to a single binary PD output, which drives the loop filter and $R(n)$ measuring block. This greatly simplifies the $R(n)$ measurement block, reducing the number of bits that must be correlated.

Majority voting also acts as a moving average lowpass filter (LPF), followed by a slicer, which adds quantization noise $\psi_{MV}$ similar to a BB-PD as shown in Fig. 5.4(a). Because it gets heavily suppressed by the LPF, we assume that $\psi_{PD}$ is negligible and lump the PD and MV gains $K_{PD}$ and $K_{MV}$ together. The result effectively acts as a single BB-PD with gain $K_{EFF}$, preceded by a LPF as shown in Fig. 5.4(b). The previous analysis of a BB-PD still holds except that $\alpha_T = 1$, since after majority voting, there is only a small chance of having zero output. Because of the filtering effect, wideband jitter such as jitter from intersymbol interference (ISI) is also suppressed. The frequency response of majority voting is plotted in Fig 5.5, and limits the measurement bandwidth to approximately 387MHz in this implementation. If a wider bandwidth is needed, $R(n)$ could instead be calculated from the full-rate PD output as in [54] at the cost of more power.

Inside the $R(n)$ measurement block, the average correlation is measured using two counters as in [54]: one which counts the total number of samples and another which counts the correlated samples over the same interval. After measuring $R(n)$ on-chip, $\Delta$ was calculated off-chip by taking the average amplitude of the triangular wave present in the $R(n)$ waveform.
5.3.1 Jitter Injection

Jitter is injected into the CDR’s edge clock phase at 437.5MHz by selectively passing the CDR’s loop filter output through a jitter injection stage, which adds or subtracts an offset from the phase code of $CK_{EDGE}$ at each cycle. Note that although the jitter frequency is beyond the measurement bandwidth shown in Fig 5.5, as long as the phase of the injected jitter is aligned to the window over which majority voting is performed, it will not be attenuated when measuring $R(n)$. As shown in Fig. 5.3, since the half-rate CDR has independent PI’s for the edge and data samplers, the edge clock phase can be changed without affecting the CDR’s data samplers, allowing error-free operation to be maintained. Furthermore, since the injected jitter frequency is well beyond the CDR’s loop bandwidth of approximately 10MHz, it is heavily suppressed by the loop filter, and should not affect CDR operation.
5.4 Measurement Results

The proposed technique was tested for several different cases while the CDR is locked to 28Gb/s PRBS31 data. Fig. 5.6 shows an example of a measured $R(n)$ waveform with and without jitter injection enabled. As seen in the figure, enabling jitter leaves the shape of $R(n)$ relatively unchanged, but causes a triangular waveform to appear as expected.

5.4.1 Measuring the Effect of Loop Gain Setting on Relative Jitter

To measure the effect of the CDR’s loop gain setting on relative jitter, Fig. 5.7 plots the relative jitter as estimated by on-chip measurement using the proposed technique for two test cases, compared to the relative jitter estimated from measurement equipment using

$$\sigma_{ER,Meas} \approx \sqrt{\sigma_{CK,Meas}^2 + \sigma_{DAT,Meas}^2}$$  \hspace{1cm} (5.9)

where $\sigma_{CK,Meas}$ and $\sigma_{DAT,Meas}$ are the RMS jitter of the clock and data as measured using an Agilent DCA-86100D sampling scope, equipped with a precision timebase module with less than 200fs jitter.

Note that $\sigma_{ER,Meas}$ is an estimate, since the relative jitter at the input of the CDR cannot be directly measured with external test equipment. However, (5.9) is a valid approximation whenever the clock and data jitter are essentially uncorrelated. This is true in our case since simulations show that the CDR output clock jitter $\psi_{CK}$ is dominated by PI and reference clock
Figure 5.7: Estimated relative CK vs. Data jitter at CDR input as CDR loop gain setting is swept for two test cases.

The recovered clock jitter was measured using a breakout path with jitter injection disabled. Data jitter was measured at the input of the test board, which could differ from the jitter that CTLE output. In all cases, the data jitter was approximately 1ps RMS at the input of the test board.

In each test case, the CDR’s reference clock was modulated with a different jitter profile and frequency offset. In both cases, the jitter estimated using the proposed technique closely agrees with estimates based on oscilloscope measurements. The ability to observe the effect of loop gain on jitter, and identify the settings that lead to the lowest jitter, demonstrates the potential use for this technique in on-chip adaptation.

5.4.2 Measuring the Effect of Data Jitter

Next, we plot the estimated relative jitter as a function of the input data jitter. Random jitter (\(\psi_{PI}\) and \(\psi_{REF}\) in Fig. 5.2(a)), and not \(\psi_{DAT}\).

Since the random data jitter is band-limited to 350MHz in the test setup, the filtering effect
of majority voting does not have a major impact on the results. If RJ is injected over a much wider bandwidth, it could be filtered by majority voting, leading to offset and/or gain error.

5.4.3 Measuring the Effect of CTLE on Jitter

The last set of measurements in Fig. 5.8(b) plot the estimated relative jitter as the CTLE code was swept. By largely suppressing broadband ISI jitter, the LPF effect of majority voting in on-chip measurement allows us to observe the subtle effect of the CTLE on the CDR’s clock jitter. Since data jitter is measured at the input of the test board, the oscilloscope measurements also do not contain the CTLE’s effect on ISI, making Fig. 5.8(b) a fair comparison.

Both sets of measurements identify CTLE code 11 as having the lowest jitter. To confirm the validity of these measurements, jitter tolerance at 100MHz is also plotted in Fig. 5.8(b). As expected, CTLE code 11 leads to the highest jitter tolerance, confirming that the proposed technique can be used to optimize CTLE performance.

5.4.4 Effect of Jitter Injection on CDR Operation

Lastly we examine the effect of the injected jitter on jitter tolerance. Since only the edge phase is modulated, we expect the data samples to be relatively unaffected. To verify this,
Figure 5.9: Jitter tolerance (BER $< 10^{-12}$, PRBS31) with and without jitter injection enabled. We perform jitter tolerance measurements with and without jitter injection enabled. Although some degradation is seen, the CDR is able to maintain error free operation with BER $< 10^{-12}$ making it possible (although not necessarily desirable) to perform measurements while the CDR remains operational. Note that jitter tolerance was measured with the adaptive loop gain algorithm from chapter 4 enabled. Without the adaptive loop, the CDR loop gain may have to be re-tuned to maintain performance while injection is on.

### 5.5 Summary

In this chapter, a jitter measurement scheme has been proposed which makes use of existing hardware in a half-rate PI-based digital CDR. The proposed scheme is able to measure the effects of loop gain, CTLE settings and input data jitter on the relative jitter of the CDR, potentially allowing CDR performance to be optimized, without requiring any dedicated eye monitor circuits.
Chapter 6

Conclusion

This thesis has presented two new techniques for on-chip jitter measurement in bang-bang CDRs, along with a method to adapt the loop gain of digital CDRs to optimize their jitter tolerance.

In chapter 3, a technique was developed and demonstrated, that allows absolute clock and data jitter to be measured in multilane CDRs. By locking two CDRs with different VCOs to the same data, and correlating the outputs of their PDs, the absolute jitter of the data can be measured without using any clean external reference clocks. By adding another PD which measures the phase difference between the two CDRs’ recovered clocks, the absolute jitter of each clock can also be measured. Separately measuring data and clock jitter makes circuit performance more observable, making it possible to distinguish between jitter on the transmit data path and jitter generated by the CDR itself.

In chapter 4, a digital CDR was designed whose loop gain automatically adapts to maximize jitter tolerance. By monitoring the autocorrelation function of the BB-PD output to prevent any ringing, the CDR loop gain can be increased, improving jitter suppression while preventing the CDR from becoming too underdamped. Tracking of sinusoidal jitter is further optimized by increasing the CDR’s loop gain whenever sinusoidal jitter is detected within the bandwidth of the CDR.

In chapter 5, a simplified jitter measurement was proposed and implemented which estimates the relative jitter between the data and clock at the input of a PI-based digital CDR. A known amount of square wave jitter is injected by adjusting the phase code of the CDR’s edge clock,
allowing the BB-PD gain and total jitter to be calculated from autocorrelation function of the BB-PD output, without the need for an eye monitor.

## 6.1 Thesis Contributions

The contributions of this thesis are summarized below.

- A technique to measure the absolute clock and data jitter in multilane CDRs was presented, which does not require a clean external reference clock. The proposed technique was implemented in a 65nm test chip containing two VCO-based 10Gb/s analog CDRs locked to the same input data. Sadegh Jalali helped to code the initial Verilog for the chip’s digital backend. This work was presented at the IEEE VLSI Symposium in 2014 [24] and published in IEEE Journal of Solid State Circuits (JSSC) in 2015 [54].

- An adaptive loop gain digital CDR was presented which automatically optimizes jitter tolerance. The proposed technique was implemented in a half-rate PI-based 28Gb/s digital CDR fabricated in 28nm CMOS. The work has been accepted for presentation at the 2017 IEEE International Solid State Circuits Conference (ISSCC) [57].

- A low-cost jitter measurement technique was proposed where a known amount of jitter is used to calculate the gain of a BB-PD. The proposed scheme was implemented in the same 28Gbps CDR described above. This work has been submitted to the 2017 IEEE Custom Integrated Circuits Conference (CICC).

## 6.2 Future Work

This thesis has presented two techniques for jitter measurement, in addition to a method to optimize the jitter tolerance of a digital CDR by adapting its loop gain. Areas exist to improve or further develop each of these works.

- **Possible Improvements to PD Correlation-Based Jitter Measurement:**

  The obvious weakness of the jitter measurement approach described in chapter 3 is its reliance on clocks from two CDRs. The system becomes much more attractive if a simple
method can be found to generate the second uncorrelated clock source. Two possibilities could be to generate the clock with a simpler circuit such as an ILO, that could be included for test purposes without consuming large area, or to generate the second clock by somehow randomizing the phase of the first clock to make its jitter uncorrelated with that of the original.

- **Possible Improvements to Adaptive Loop Gain CDR:**
  The results of chapter 4 showed that the proposed adaptive loop gain CDR is effective when random jitter is dominant. However, to handle sinusoidal jitter, adaptation relies on settings adapted when no large SJ is present. An open research problem is to remove this reliance on previously adapted values. This would require a way to detect ringing in the CDR’s response even when large sinusoidal jitter dominates $R(n)$.

  While able to handle sinusoidal jitter, the existing dynamic mode may also be less effective when subjected to other periodic jitter profiles, possibly at multiple frequencies. The proposed CDR uses a very simple method to estimate the dominant SJ frequency, which becomes less effective when multiple tones are present. A more versatile bandwidth estimation scheme would make the adaptation scheme more robust by more accurately estimating the dominant frequency content of more complex periodic jitter profiles.

- **Possible Improvements to Jitter Injection-Based Jitter Measurement:**
  As discussed in chapter 5, majority voting had a filtering effect on the jitter injection-based jitter measurement technique, attenuating broadband ISI jitter. In some applications, this may not be desirable. Future implementations could include the ability to bypass majority voting and estimate $R(n)$ directly from the full-rate PD output. Furthermore, in chapter 5, averaging was used to obtain more accurate jitter measurement results. This averaging should be included on-chip in order to create a fully on-chip solution. Measurement of the triangular waveform amplitude $\Delta$ should also be incorporated on-chip.

  This work can also be extended by applying it to on-chip adaptation. This jitter measurement technique provides direct observability of the relative jitter at the CDR input. Using this measurement, a gradient descent algorithm could be used to optimize CDR or wireline receiver characteristics, such as loop gain, equalizer settings, or any other circuit...
parameters that can impact jitter performance. One of the main challenges to developing such an adaptation scheme, is ensuring convergence in cases where jitter is a relatively weak function of the parameter being adapted, making the performance surface shallow.
Appendix
Appendix A

Derivation of Scaling Factors for Correlated-Based Jitter Estimation

In this appendix, we more carefully analyze what happens when the outputs of two bang-bang PDs, driven by the same data, and by different clocks, each with uncorrelated jitter are correlated to estimate the jitter of the data. Whereas in chapter 3, we used linear BB-PD models, here we analyze the correlation directly using the PDFs of the jitter of each signal. This analysis gives rise to the scaling factors referred to in Section 3.2.3 of chapter 3.

A.1 Impact of Bang-Bang PD Non-linearity

In chapter 3, we correlated the outputs of two bang-bang PDs, modelled as linear gains as shown in Fig. A.1(a). We assumed that the jitter of each signal $\psi_A$, $\psi_B$, and $\psi_C$ are all uncorrelated. Based on the linear model, the PDs output $e_1 = K_{P1}(\psi_A - \psi_B)$ and $e_2 = K_{P2}(\psi_A - \psi_C)$. However in reality, bang-bang PDs only output the sign of $\psi_{ER1} = \psi_A - \psi_B$ and $\psi_{ER2} = \psi_A - \psi_C$ as +1, -1 (or 0 if there is no transition) as shown in Fig. A.1(b). Correlating two bang-bang PD outputs therefore gives $E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})]$, whose value can be found by integrating
Figure A.1: (a) Linear model and (b) actual nonlinear model of two bang-bang PD outputs being correlated

\[ E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})] = \int \int f_{\psi_{ER1},\psi_{ER2}}(\psi_{ER1}, \psi_{ER2}) \, d\psi_{ER1} d\psi_{ER2} \]

(A.1)

\[ = \int_{\text{sgn}(\psi_{ER1})=\text{sgn}(\psi_{ER2})} f_{\psi_{ER1},\psi_{ER2}}(\psi_{ER1}, \psi_{ER2}) \, d\psi_{ER1} d\psi_{ER2} - \int_{\text{sgn}(\psi_{ER1})\neq\text{sgn}(\psi_{ER2})} f_{\psi_{ER1},\psi_{ER2}}(\psi_{ER1}, \psi_{ER2}) \, d\psi_{ER1} d\psi_{ER2} \]

(A.2)

Assuming that the bang-bang PD can be linearized, and based on the discussions in chapter 3 we expect \( E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})] \) to be proportional to \( K_{P1}K_{P2}\sigma_{\psi_A}^2 \), with the uncorrelated jitter components having been eliminated. \( \sigma_{\psi_A} \) can then be extracted as

\[ \sigma_{\psi_A}^2 = \frac{1}{C} \sqrt{\frac{E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})]}{K_{P1}K_{P2}}} \],

(A.3)

where \( C \) is a proportionality constant we would like to find. As in chapter 2, we define the gain of a bang-bang PD with relative input jitter \( e_i \), as [12]
Appendix A. Derivation of Scaling Factors for Jitter Estimation

\[ K_{P_i} = 2f_{\psi_i}(0) \]  \hspace{1cm} (A.4)

where \( f_{\psi_i}(\psi_i) \) is the PDF of \( \psi_i \). The PDF is the slope of the PD response at zero phase offset. If \( \psi_i \) is Gaussian with \( \sigma_i \), this becomes

\[ K_{P_i} = \sqrt{\frac{2}{\pi}} \frac{1}{\sigma_i} \]  \hspace{1cm} (A.5)

as we have seen in chapter 2. To determine the relationship between \( \sqrt{E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})]} \) and \( \sigma_{\psi_A} \), we evaluate (A.2), \( K_{PD1} \) and \( K_{PD2} \), and plot \( \sqrt{E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})]/(K_{P1}K_{P2})} \) vs. \( \sigma_{\psi_A} \) in Fig. A.2. \( \psi_B \) and \( \psi_C \) are zero-mean Gaussians with \( \sigma = 1 \), while \( \psi_A \) is assumed to be either Gaussian, sinusoidal, or uniformly distributed with a \( \sigma \) that is swept. When \( \psi_B \) and \( \psi_C \) are Gaussian, (A.2) has no closed form solution and must be integrated numerically.

![Figure A.2](image)

Figure A.2: \( \sqrt{E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})]/(K_{P1}K_{P2})} \) as a function of \( \sigma_{\psi_A} \) using numerical integration. \( \psi_B \) and \( \psi_C \) are Gaussian with \( \sigma_{\psi_B} = \sigma_{\psi_C} = 1 \)

Each curve in Fig. A.2 remains highly linear as \( \sigma_a \) varies by an order of magnitude, with
a slope that is dependent on the PDF of $\psi_A$. We can therefore find the constant C from the slope of the plots. The value of C depends on the PDF of $\psi_A$ because the integral in (A.2) compares the relative distribution of $f_{\psi_{ER1}\psi_{ER2}}(\psi_{ER1},\psi_{ER2})$ between areas where $\psi_{ER1}$ and $\psi_{ER2}$ have the same sign, and where their signs differ. The PDF of $\psi_A$ changes the shape of $f_{\psi_{ER1}\psi_{ER2}}(\psi_{ER1},\psi_{ER2})$. To illustrate this, Fig. A.3(a) plots $f_{\psi_{ER1}\psi_{ER2}}(\psi_{ER1},\psi_{ER2})$ where

$$\psi_A = 0.$$ The PDF is symmetric. When jitter is added, the PDF stretches in the diagonal direction. This causes more of the PDF to enter quadrants 1 and 3, where $\psi_{ER1}$ and $\psi_{ER2}$ have the same sign; increasing the value of $E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})]$. Depending on the PDF of $\psi_A$, the amount of distortion in the PDF changes. For the same $\sigma$, Fig. A.3(c) shows that sinusoidal jitter causes the strongest distortion in the PDF, leading to the highest value of $E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})]$.

![Contour plots](image-url)
Figure A.4: $\sqrt{E[\text{sgn}(\psi_{ER1})\text{sgn}(\psi_{ER2})]/(K_P1K_P2)}$ as a function of $\sigma_{\psi_A}$ using Matlab simulation

### A.2 Effect of PD Gain Calculation

The method of calculating the PD gain can also affect the slopes in Fig. A.2. Evaluating (A.4) is actually quite difficult. The relative jitter PDF is found from the relative jitter histogram, but depending on the bin size of the histogram data, and number of hits, the PDF waveform can appear noisy, making it difficult to find its value precisely at zero. To avoid this, we calculate the average slope over the 68% of samples closest to zero in the PDF. In this Gaussian case, this leads to a gain that is up to 15% lower than (A.5). To account for this effect, Matlab simulations were conducted using this method of calculating PD gain, with the same histogram bin and sample sizes used in the actual system. The results of Fig. A.2 are repeated using this gain calculation method and plotted in Fig. A.4. Because the PDF in the uniformly distributed case is very flat, the PD gain calculation has minimal impact on the constant $C$. The Gaussian and Sinusoidal cases are affected more, causing $C$ to increase by 5-10%. The results of Fig. A.4
show that for this system, $\sigma_{\psi_A}$ can therefore be extracted as

$$\sigma_{\psi_A}^2 = \frac{1}{C} \sqrt{\frac{E[\psi_{12}]}{K_{p1}K_{p2}}}, \quad C = \begin{cases} 
\frac{1}{1.34} \text{ for } \psi_A \text{ Gaussian} \\
\frac{1}{1.97} \text{ for } \psi_A \text{ sinusoidal} \\
\frac{1}{1.7} \text{ for } \psi_A \text{ uniform}
\end{cases} \tag{A.6}$$

Where $C$ is a fixed factor can be chosen based on the shape of the relative jitter histogram, which should show if the jitter is Gaussian or not. The autocorrelation function can be used to look for any large sinusoidal jitter components.
References


