AN ESTIMATION APPROACH TO CLOCK AND DATA RECOVERY

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Hae-Chang Lee

November 2006

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as dissertation for the degree of Doctor of Philosophy.

_____

Mark A. Horowitz
(Principal Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as dissertation for the degree of Doctor of Philosophy.

_____

Boris Murmann

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as dissertation for the degree of Doctor of Philosophy.

_____

Stefanos Sidiropoulos

Approved for the University Committee on Graduate Studies

_____

Abstract

High speed I/O is used to increase the bandwidth between chips in a computer or network. The clock and data recovery (CDR) module is responsible for reconstructing the original transmitted bit-stream at the receiver. Until now, the CDR has been viewed as a feedback control system that adjusts its output clock according to the phase movement of the input data. However, there is another way to view the role of the clock recovery circuit. Viewing the CDR as an estimator of the phase position of the next bit rather than as a tracking loop allows one to rethink how a CDR should be designed. It gives one a physical intuition both of what order control loop is needed in conventional applications and of how to construct more complex non-linear control systems. The purpose of this research is to explore how the understanding of the CDR as a phase estimator can improve CDR performance in different applications.

To achieve this, a semi-digital dual loop CDR is modified to better estimate future phase for three applications. Matlab simulations and two test chips in 0.13μm and 0.25μm CMOS were implemented to quantify the improvement. By increasing the order of the loop filter to a second order, we have shown that the timing margin of the link is improved by more than 0.2 UI when a 200ppm frequency offset exists between the transmitter (TX) and receiver (RX). Using a second order CDR with very accurate frequency estimation, a burst mode receiver with zero lock time is made. This CDR can retain lock even when the packets are spaced apart by a million bits. Finally, a higher order estimator for systems using spread spectrum clocking can improve the timing margin by 0.05 UI in comparison to a second order CDR.

Acknowledgments

I recall that my advisor, Mark Horowitz, read my final VLSI draft at 3am on the day of its submission (and in the same email asked me to call him in the next 30 minutes to discuss the changes). While impressive on its own merit, I found out after submission that he had been planning for his cancer surgery during the preceding days. I am fortunate to have had such a dedicated advisor.

I would also like to thank Prof. Boris Murmann for agreeing to be my associate advisor and serving on my reading committee. Dr. Stefanos Sidiropolous who gave much guidance during my work on spread spectrum systems has also graciously made time to read my thesis despite his duties running his company. I am also grateful to Prof. Saraswat who served as the chair for my oral defense on very short notice.

Former and current Horowitz group members have served as teachers as well as friends. Jaeha deserves special mention as he has been a great reference as I put together this thesis. Azita, Dean, Elad, Ken, Ron, Sam, and Vladimir have added tremendously to my engineering knowledge.

Rambus supported me during the last leg of my PhD. As I have now been a full-time employee for almost a year, I can honestly say that I am a lucky person who genuinely enjoys his work. Of course, work would not be enjoyable if it were not for great co-workers such as Jung Hoon, Brian, Yohan, Nhat, Bruno, Carl, and Jared.

I wish to thank MARCO for financial support of my research.

My parents and my brother have been a constant source of encouragement during the decade I have been at Stanford. I thank them for their tireless prayers on my behalf.

On account of my wife Rachel, I have matured socially as well as intellectually during my PhD. With the birth of our son, Isaac, I am even more indebted to her for the completion of this thesis.

This thesis, for all it is worth, is dedicated to my family.

Table of Contents

List of Figures

Chapter 1

Introduction

Technology scaling has dramatically increased the amount of computation that can be integrated onto a small piece of silicon. This increased computation has highlighted the need for chip I/O that can supply the needed information fast enough to keep the compute engine fed. As a result, the design of chip I/O has become increasingly sophisticated, with multi-Gb/s bandwidths now prevalent in high performance computer systems and networks.

These high speed links are composed of a transmitter (TX) and a receiver (RX) communicating over a channel, as shown in Figure 1.1. The blocks must first generate and receive a high bandwidth signal. Then the RX must reconstruct the original transmitted bitstream from the received waveform. The first task spans a wide area of disciplines including channel design [1, 2], package design [9-11], signaling methods (e.g. PAM [3-5]), and equalization [6-8]. The second task is clock and data recovery (CDR), which is the subject of this thesis.
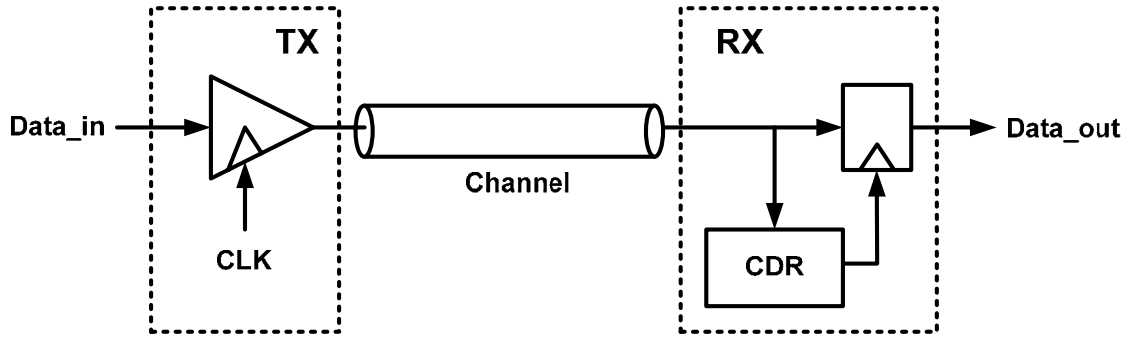
Figure 1.1: High speed link block diagram.

1.1 CDR Overview

An eye diagram, which is created by overlaying consecutive bits onto a single bit time (T_b), is useful in explaining the function of a CDR (Figure 1.2). The 'eye' shape is created by the four possible transitions which are marked with straight lines in Figure 1.2. The first is when the previous bit is high and the current bit is low. The second transition is when the previous bit is low and the current bit is high. The third is when the current bit is high and the next bit is low. The last transition occurs when the current bit is low and the next bit is high. The first two transitions form the left half of the eye. The latter two transitions form the right half. The actual transitions have uncertainty caused by timing and voltage noise in the system. This causes the transitions to appear thick and blurry in the eye diagram.
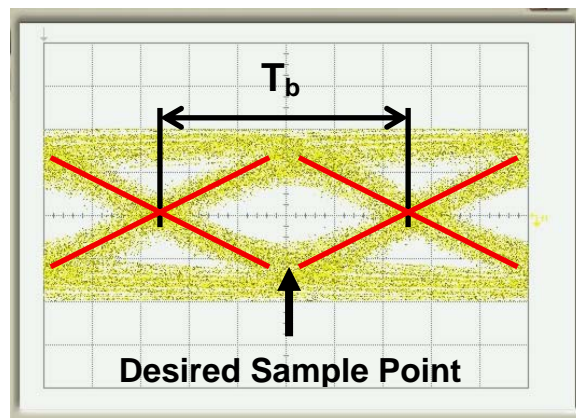
Figure 1.2: Example eye diagram from a sampling oscilloscope. Vertical axis is voltage and the horizontal axis is time.

The objective of the CDR is to recover the data with as few errors as possible. In other words, the goal of the CDR is to minimize the bit error rate (BER) through the channel. To achieve minimum BER, the CDR needs to sample the data where the eye opening is largest. This point ("Desired Sample Point") has the highest signal to noise ratio (SNR) which is in turn the point with the lowest BER. Equation (1.1) governs this relationship for binary signals with additive white Gaussian voltage noise, where erfc(.) denotes the complementary error function [12]. The BER decreases rapidly as the SNR is increased.

$$BER = \frac{1}{2} erfc(\frac{SNR}{2\sqrt{2}}) \tag{1.1}$$

As seen in Figure 1.2, there is a high correlation between the maximum eye opening and the midpoint between the transition crossings. Hence, CDRs are often designed to sample the data half a bit time away from its estimate of where the data transitions are occurring. In the rare case that the data eye is asymmetric, this will not be the best solution, but in most cases it is an efficient way of getting very close to it. Due to this approximation, the CDR needs to recover the clock that is embedded in the data transitions to use it as its reference in sampling the data.

What complicates CDR design is that the data transitions are not stationary with respect to a timing reference. There are two reasons for this movement. The first is due to any deterministic phase offset trajectory that is a result of frequency offsets between the TX and RX reference clocks. The second is due to timing uncertainty or noise that is referred to as jitter. These will be explained next.

1.1.1 Deterministic Phase Offset Trajectory

Frequency offsets exist between the TX and RX when each side of the link has its own independent clock source, such as a crystal or SAW oscillator, that are nominally but not exactly identical in frequency (Figure 1.3). This frequency offset does not have to be a constant but can be a periodic function in time. This causes the phase offset between the

data transitions from the TX and the edges of the clock to the RX to change in a deterministic and predictable manner.
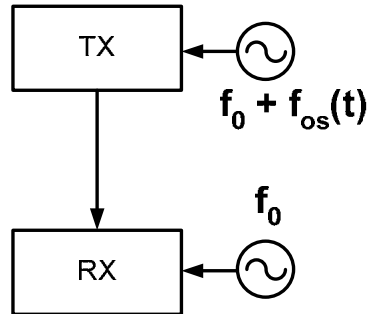
Figure 1.3: Systems with frequency offsets.

The degenerate case is when the average offset in frequency is zero. Such systems are called mesochronous and occur when a global frequency source is shared between all the parties involved. In this situation, the phase offset between the TX and RX is a constant over time (Figure 1.4). A more interesting case is when the frequency offset is a constant. These systems are called plesiochronous. The edges of the clock to the TX (and hence the timing of the data transitions) diverge in a linear fashion with respect to the edges of the clock to the RX (Figure 1.4). The phase offset trajectory is found by plotting the phase difference between the data transitions and the reference clock to the RX versus time. For plesiochronous systems, the phase offset trajectory is a linear ramp with the ramp rate set by the magnitude of the frequency offset.
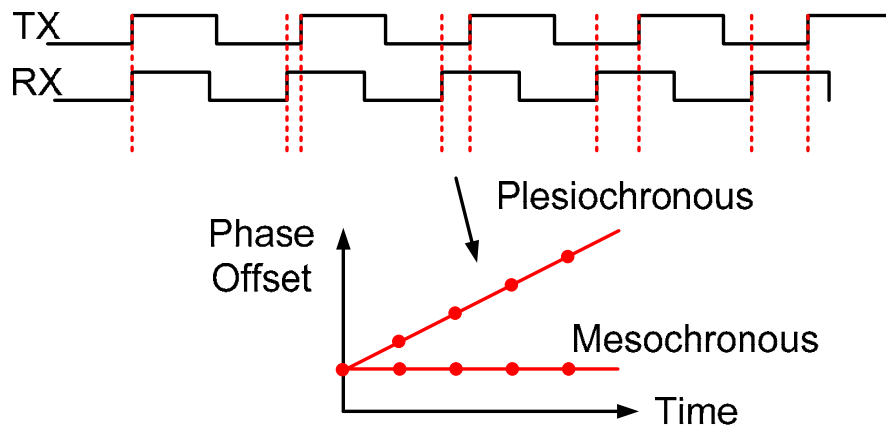
Figure 1.4: Phase offset trajectory of mesochronous and plesiochronous systems.

We have introduced two common clocking systems and their phase offset trajectories. As we will see later, more sophisticated clock schemes have arisen due to electromagnetic interference (EMI) concerns in recent years [13]. The result is a more complex, but still very much deterministic and predictable, phase offset trajectory. For the purposes of CDR design, such deterministic phase offset trajectories must be tracked as accurately as possible.

1.1.2 Jitter

Jitter refers to the uncertainty in the phase of clock (and hence data) edges. There are two categories of jitter – deterministic jitter (DJ) and random jitter (RJ). DJ has bounded statistics while RJ has unbounded Gaussian statistics.[1] These two categories of jitter are demonstrated in Figure 1.5 where the unbounded tail at either side of the time domain histogram results from RJ while the two distinct peaks are due to DJ.
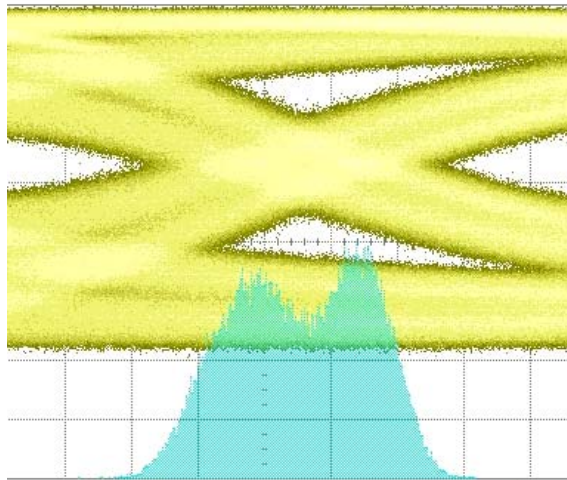
Figure 1.5: Time histogram on sampling scope demonstrating the two major categories of jitter – deterministic and random.

There are three common types of DJ found in real systems: data dependent jitter (a.k.a intersymbol interference or ISI), duty cycle distortion (DCD), and uncorrelated (to the data) bounded jitter such as supply noise induced jitter. In a well designed system, the dominant source of DJ is from ISI. RJ is most commonly caused by fundamental noise

---

[1] Deterministic jitter (DJ) should not be confused with the deterministic phase offset trajectory. DJ and RJ are timing noise super-imposed onto the deterministic phase offset trajectory.

sources of both active and passive devices (such as thermal noise). Most of these jitter components including ISI and RJ show little correlation over time.

The job of the CDR designer is to filter and not attempt to track these uncorrelated jitter sources since their past behavior sheds little light on the position of future transitions.

1.2 Motivation

In general, the CDR has been viewed as a feedback control system that adjusts its output clock phase in response to phase movements of the data. While the deterministic phase offset trajectory and jitter have very different properties, they are often lumped together as timing disturbances. CDR design is complicated as its bandwidth must be optimized between tracking the phase trajectory and filtering uncorrelated jitter such as ISI.

I will show in this thesis that viewing the CDR as an estimator of the phase position of the next data transition rather than as a tracking loop allows us to directly address the different design requirements resulting from the phase offset trajectory and jitter. In answering the question "What should my CDR look like in order for it to predict the behavior of the TX in the future?" we can obtain insight into the optimal structure of the CDR for a given application that will provide independent means for tracking the phase offset trajectory while optimizing the bandwidth for jitter.

How this understanding of the CDR as a phase estimator can be applied beneficially to different application will be demonstrated using the following three applications: systems with a fixed frequency offset between the TX and RX (plesiochronous systems), burst mode communication where the frequency offset is compounded by an extremely low transition density, and systems with complex patterns in the frequency offset such as spread spectrum clocking.

1.3 Organization

This chapter provided an introduction on the objective of CDR design and the various design considerations that must be addressed. We have identified the deterministic phase offset trajectory along with jitter as the primary design consideration for a CDR.

Chapter 2 will first review analog phase-locked loops (PLL) which are the most commonly used CDR architecture. This chapter will then discuss critical component and system properties like transfer function and stability that are of interest to the designer. Finally, we will motivate the architecture chosen to demonstrate phase estimator design - the semi-digital dual loop CDR – by analyzing the noise performance of the analog PLL CDR.

For the semi-digital dual loop CDR, estimator performance is dependent on the linearity of the phase domain digital-to-analog converter (Phase DAC). Chapter 3 will detail the circuits of a high precision phase DAC as well as measured results.

Chapter 4 demonstrates the advantages of building a phase estimator for systems with a fixed frequency offset between the transmitter and receiver. We will show how a second order CDR can improve performance over a first order CDR in the presence of a frequency offset using jitter tolerance as a metric. It will be shown that the reason the second order CDR performs better is because it has the capacity to predict the movement of the data transitions (the deterministic phase offset trajectory) caused by the frequency offset.

Chapter 5 uses burst mode communication to show how an estimation perspective can change the design of the CDR. Burst mode systems arise when multiple transmitters and receivers share a common channel – typically optical fiber – via time division multiplexing (TDM). In these systems, the duration of time between two packets arriving from the same transmitter to a receiver can be very long. Since CDRs were viewed as tracking loops, past efforts to address this absence of timing information focused on building systems that tried to reacquire phase lock as quickly as possible. However, taking an estimation approach leads us to build a CDR that predicts the phase of future packets by obtaining very accurate phase and frequency estimates of each TX. Such a

system can retain lock even if packets are separated by hundreds of thousands of bits and thus achieve zero lock time.

The idea of creating a CDR that predicts future phase position is extended in Chapter 6 for links that use spread spectrum clocking. Spread spectrum clocking is used in wireline communication in order to reduce electro-magnetic interference (EMI). One common implementation of spread spectrum modulates the frequency of the TX with a triangular waveform. In order to make predictions on the timing of future bits, this system estimates all relevant parameters of the spread spectrum clock - i.e. phase, frequency, frequency ramp rate, modulation phase, and modulation frequency. Such a higher order estimator can decouple opposing constraints on the bandwidth of the CDR – namely the need for a large bandwidth to track the deterministic non-constant phase trajectory and the need for a low bandwidth to improve jitter filtering. The implementation of a lower bandwidth without compromising the ability of the CDR to correct the phase movement due to the spread spectrum clock improves the timing margin of this CDR over conventional designs.

Chapter 2

CDR Basics

This chapter will review the fundamentals of CDR design. The objectives of this chapter are two-fold. The first is to provide insight into what factors influence the dynamic behavior of the CDR. This will provide the foundation for understanding the material in later chapters. The second is to motivate the CDR architecture that was chosen to demonstrate phase estimator design in this thesis. Section 2.1 will first review analog phase locked loop (PLL) based CDRs which are the most prevalent in industry. This section will discuss critical component and system properties like transfer function and stability that are of interest to the designer. An analysis of the noise performance of the analog PLL CDR will provide motivation for the semi-digital dual loop architecture. Section 2.2 will then describe this architecture.

2.1 Analog PLL based CDRs

Due to the ever changing phase relationship between the data transitions and the RX clock, the RX needs to constantly adjust the time at which it samples the data. To do this, the CDR needs three major components. The phase detector (PD) is used to determine the phase relationship between the data transitions and its own clock domain. Second, a loop filter removes the noise in the phase detector output and sets the bandwidth of the PLL.

Finally, the voltage controlled oscillator (VCO) provides a method of adjusting the phase of a clock in order to optimally move it to the maximum eye opening of the data. Figure 2.1 is a CDR using a charge pump PLL [23].
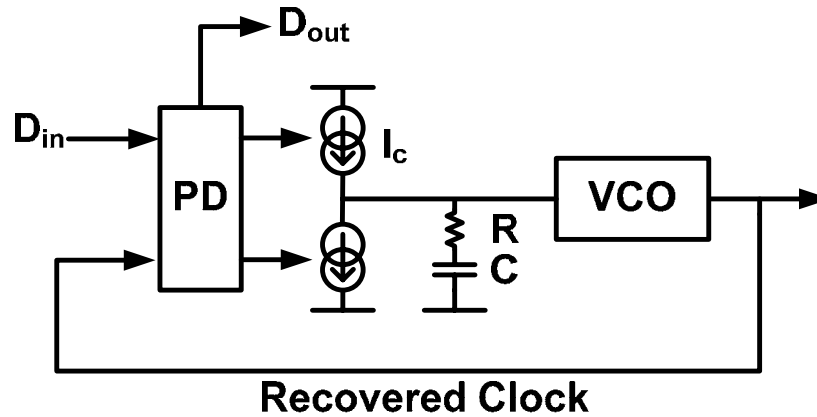
Figure 2.1: Analog PLL based CDR.

2.1.1 Linear Phase Detector

The phase difference between clock domains is determined using a phase detector (PD). The most widely used linear PD was first published by Hogge [17]. Its schematic and transfer function are shown in Figure 2.2. The PD subtracts two pulses, each originating from an XOR gate. The subtraction is done indirectly by the charge pump. The XOR gate on the right generates a reference pulse that is exactly half a bit time in width. The XOR gate on the left generates a pulse whose width depends on the phase error between the edges of Data and the edges of D_{rt}. When the phase difference between Data and D_{rt} (and hence Clk) is half a bit, the two pulses cancel each other thus generating no change in the PD output. The linearity results from the net pulse width at the output being linearly proportional to the phase difference between the center of Data and the edges of Clk (Figure 2.2 (b)). The saw-tooth shape of the transfer function is due to phase being a modulo-2π quantity. Phase detector gain (K_{PD}) is defined as the slope of the transfer function where the phase error is close to zero. The phase detector gain is typically 1 and is unitless.[2]

---

[2] The definition of the PD gain depends on the definition of the VCO gain. In this thesis, VCO gain has units of $Hz{\cdot}V^{-1}$. Alternately, it is common to define VCO gain in $rad{\cdot}(sec{\cdot}V)^{-1}$. In this case, the PD gain is $(2\pi)^{-1}$ and has units of rad^{-1}.

(a)

(b)
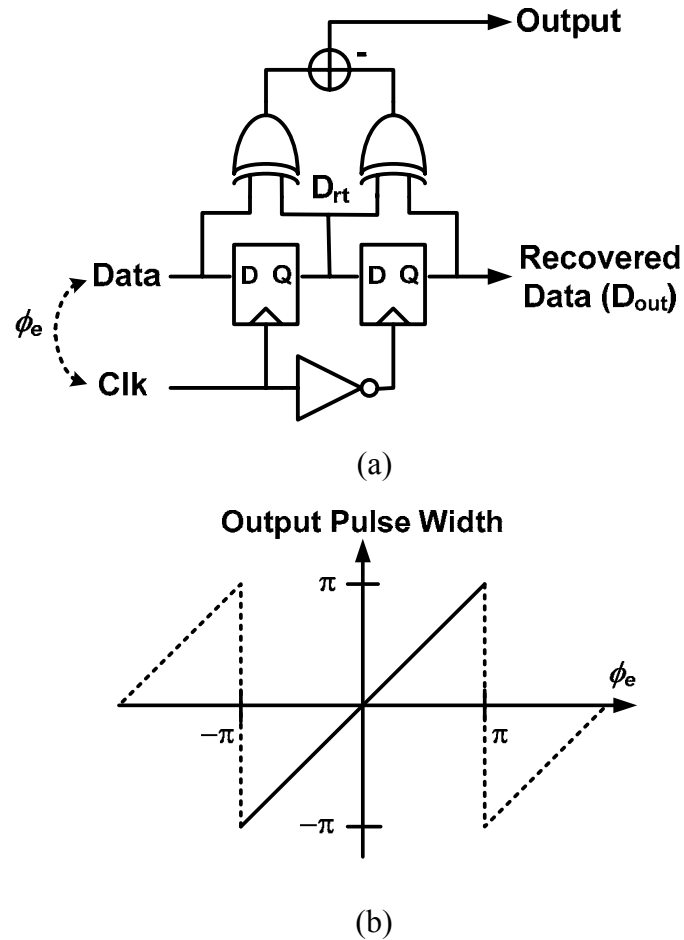
Figure 2.2: Linear Phase Detector (a) schematic and (b) transfer function.

Linear PDs allow the designer to use linear systems theory in analyzing the input-output behavior of the PLL, such as 3-dB bandwidth and the amount of peaking in the transfer function. This can be a benefit when such parameters are defined by a specification such as Synchronous Optical Network (SONET) [15]. Additional advantages include low complexity and low power consumption.

There are several drawbacks to this PD. First, this PD requires conditioning on the swing and duty cycle of the input data via a limiting amplifier. Second, the maximum rate of operation is limited by the intrinsic speed of the flip-flops in current mode logic (CML) and the XOR gate in CMOS. The adaptation of this phase detector for multi-phase operation, a common method of overcoming fundamental circuit speed limitations, is not straight forward [21]. Third, the average PD output when the phase error is zero differs between the case when the input data pattern is a long train of 1's or 0's, referred

to as consecutive identical digits (CID), and when the input is a clock pattern. This in turn causes the average PD output to be pattern dependent which results in data dependent jitter. Designers have addressed this problem but have done so with a significant increase (about twice) the hardware [18-19]. Fourth, the linear PD's analog output necessitates a sigma-delta ADC to make it usable in digital CDRs that are gaining increased interest due to the leakage current in on-chip capacitors as well as the reluctant scaling of analog circuits [20]. Finally and most importantly, the edges of Data do not experience the same clk-q delay seen by D_{rt} and D_{out}. This clk-q delay mismatch causes a phase offset between Clk and the center of the Data in steady state. Hogge recognized this problem and proposed the insertion of a delay on Data to provide a replica of the clk-q delay. Unfortunately, this is an imperfect solution that is affected by process, voltage, and temperature (PVT) variations. Nonetheless, the linear transfer characteristic of this PD compensates for these shortcomings in certain applications leading to its apparent popularity.

2.1.2 PLL Loop Dynamics

Figure 2.3 shows the linear continuous-time model of an analog PLL. We assume for now that the input is a clock pattern. The notable difference to Figure 2.1 is that the charge pump and RC network have been replaced with two branches (proportional and integral) that are summed in parallel [24]. The equivalence of these seemingly different structures will be shown later. The linear model only applies when the phase error between the input data and the recovered clock is small. The acquisition behavior of PLLs is a nonlinear phenomenon that cannot be predicted using this model. Furthermore, the continuous time model requires that the sampling frequency (i.e. the recovered clock frequency) be much larger than the bandwidth of the loop itself. This assumption allows one to ignore the discrete sampling effects and hence apply the Laplace transform [23].
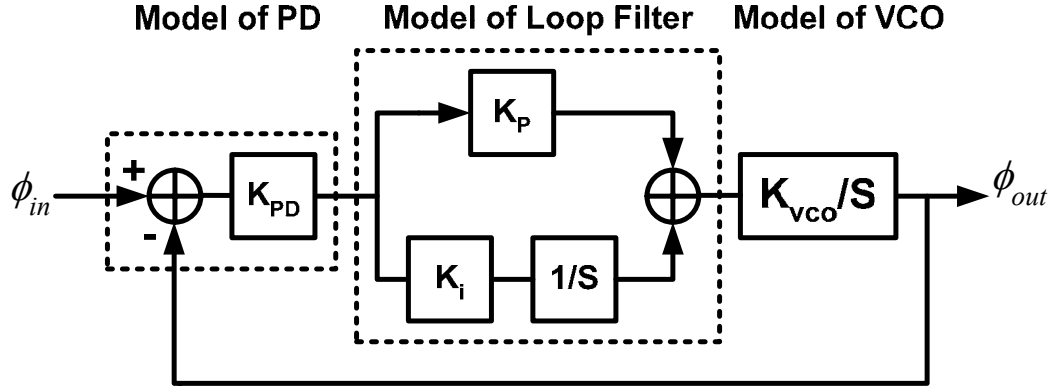
Figure 2.3: Linear PLL model.

The transfer function of the linear PLL has two poles and a zero. The gain of the VCO (Kvco) represents the conversion factor from voltage at its input to a frequency at its output. Hence, its units are in Hz·V^{-1}. K$_{PD}$ is unitless, the proportional gain (K$_P$) is in V, and the integral gain (K$_i$) is in V·sec^{-1}.

$$\frac{\phi_{out}}{\phi_{in}} = \frac{s \cdot K_P \cdot K_{PD} \cdot K_{VCO} + K_i \cdot K_{PD} \cdot K_{VCO}}{s^2 + s \cdot K_P \cdot K_{PD} \cdot K_{VCO} + K_i \cdot K_{PD} \cdot K_{VCO}} \qquad (2.1)$$

The transfer function of the PLL in Figure 2.1 is exactly in the form of (2.1) when we make the following substitutions for K$_P$ and K$_i$. I$_C$ is the charge pump current.

$$K_P = I_C \cdot R \qquad (2.2)$$

$$K_i = \frac{I_C}{C} \qquad (2.3)$$

Equation (2.1) can be rewritten using ζ (damping factor) and ω_n (natural frequency) which are parameters that give insight into the time domain behavior of the PLL. A larger damping factor translates into less peaking and ringing of the output step response. On the other hand, a larger natural frequency translates into a larger 3-dB bandwidth and hence a faster rise time in the step response [25].

$$\frac{\phi_{out}}{\phi_{in}} = \frac{s \cdot (2 \cdot \zeta \cdot \omega_n) + \omega_n^2}{s^2 + s \cdot (2 \cdot \zeta \cdot \omega_n) + \omega_n^2} \tag{2.4}$$

By equating (2.1) and (2.4), ζ and ω_n are found to depend on the various loop gains as follows.

$$\omega_n = \sqrt{K_i \cdot K_{PD} \cdot K_{VCO}} \tag{2.5}$$

$$\zeta = \frac{K_P}{K_i} \cdot \frac{\omega_n}{2} \tag{2.6}$$

In stabilizing the PLL, the two parameters most readily available to the designer are K_P and K_i. Equation (2.6) tells us that to keep constant ζ (a measure of relative stability), K_i must be increased four-fold for every two-fold increase in K_P. This is because ω_n also depends on the square root of K_i.

2.1.3 CDR Loop Dynamics

As a CDR, the PLL loop dynamics found in Section 2.1.2 are modified by the transition density (TD) of the input data [23]. The transition density is the ratio of the number of transitions to the number of bits transmitted in a serial bitstream. It is 1.0 when the input is a clock pattern. Many serial test patterns, such as PRBS, have a transition density close to 0.5 [26-27]. The impact of transition density is to scale K_{PD} by the same factor since the absence of transitions reduces the *average* output pulse width observed over many bits. Equations (2.1) and (2.5) are modified simply by replacing K_{PD} with $TD \cdot K_{PD}$.

$$\frac{\phi_{out}}{\phi_{in}} = \frac{s \cdot K_P \cdot (TD \cdot K_{PD}) \cdot K_{VCO} + K_i \cdot (TD \cdot K_{PD}) \cdot K_{VCO}}{s^2 + s \cdot K_P \cdot (TD \cdot K_{PD}) \cdot K_{VCO} + K_i \cdot (TD \cdot K_{PD}) \cdot K_{VCO}} \tag{2.7}$$

$$\omega_n = \sqrt{K_i \cdot (TD \cdot K_{PD}) \cdot K_{VCO}} \tag{2.8}$$

An approximation of the 3-dB bandwidth can be made when ζ is large. This is a reasonable condition given that CDRs are designed for ζ larger than 5 such that the loop is over-damped and approaches a single pole response [26]. The purpose of over-damping is to minimize any amplification of the jitter at the input of the CDR. Under this condition, (2.4) and (2.7) can be approximated with the following equation.

$$\frac{\phi_{out}}{\phi_{in}} = \frac{K_P \cdot (TD \cdot K_{PD}) \cdot K_{VCO}}{s + K_P \cdot (TD \cdot K_{PD}) \cdot K_{VCO}} = \frac{2 \cdot \zeta \cdot \omega_n}{s + 2 \cdot \zeta \cdot \omega_n} \tag{2.9}$$

Then the 3-dB bandwidth (in Hz) is approximated with (2.10). Notice the absence of K_i from this equation since an infinite damping factor corresponds to zero K_i. In reality, the bandwidth of the CDR shows some dependence on K_i.

$$f_{-3dB} \cong \frac{K_P \cdot (TD \cdot K_{PD}) \cdot K_{VCO}}{2 \cdot \pi} = \frac{\zeta \cdot \omega_n}{\pi} \tag{2.10}$$

2.1.4 Noise to Phase Estimation Error Transfer Function

When used in chip to chip applications where a large number of high speed transceivers are integrated with a noisy digital core onto a single substrate, there are two dominant sources of jitter that concerns the CDR designer. The first is jitter at the input of the CDR due primarily to RJ and ISI.[3] The second is from instability in the VCO frequency caused either by device noise (i.e. thermal and 1/f) or supply and substrate noise. The momentary frequency errors cause jitter accumulation until the CDR is able to correct this disturbance. These observations for the CDR are similar to those made by Mansuri [28] for clock generation PLLs except that the main source of input jitter in that work is from the inherent noise of an off-chip oscillator (e.g. crystal).

Figure 2.4 shows the model used to find the transfer function from the noise source to the phase estimation error (Φ_{ee}) in order to assess the impact of these noise sources on

---

[3] This is true even when equalizers are used in the design. Most equalizers are symbol spaced. Regardless of whether they are implemented in the TX as a finite impulse response (FIR) filter or at the RX as a decision feedback equalizer (DFE), they only minimize the ISI at the data sampling point. ISI for times in between, including the transitions, are not improved significantly [29].

the CDR's performance. Phase estimation error implies that the CDR is sampling the data away from the optimal point in time and hence should be minimized. Figure 2.5 shows the normalized transfer functions from the two noise ports to Φ_{ee}.
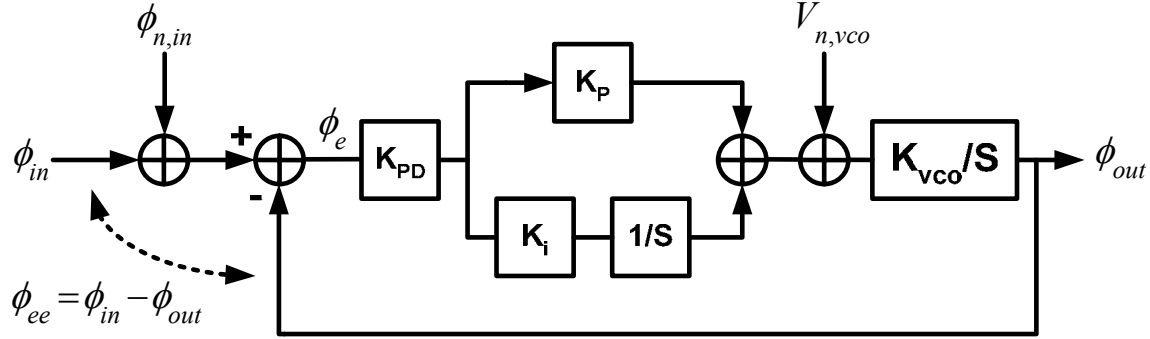
Figure 2.4: Linear model for finding noise to phase estimation error transfer functions. $\Phi_{n,in}$ is the jitter at the input. $V_{n,vco}$ is the device, supply, and substrate noise affecting the VCO frequency. The output of interest is the phase estimation error (Φ_{ee}).

The transfer function from $\Phi_{n,in}$ to Φ_{ee} (2.11) is identical to the input-output transfer function of (2.7) except for an inversion and is a low pass response. Therefore, it is desired to minimize the bandwidth of the CDR in order to reduce the amount of phase estimation error caused by input jitter.

$$\frac{\phi_{ee}}{\phi_{n,in}} = -\frac{\phi_{out}}{\phi_{n,in}} = -\frac{\phi_{out}}{\phi_{in}} \tag{2.11}$$

It is important to note that the BER of the link is ultimately determined by the phase error (Φ_e). However, using Φ_e instead of Φ_{ee} in the preceding analysis leads to the wrong conclusion that a larger CDR bandwidth is desired at all costs. This is because the model ignores the uncorrelated nature of the input jitter and the existence of finite loop delay. For uncorrelated jitter, minimizing Φ_{ee} leads to minimum Φ_e and hence the best BER. As we will see later in the context of jitter tolerance, the phase of the TX (Φ_{in}) can contain correlated jitter (e.g. from the off-chip oscillator) that has low-frequency content. The need to track low frequency changes in Φ_{in} places a practical bound on how small the CDR bandwidth can be before degrading rather than improving the BER.
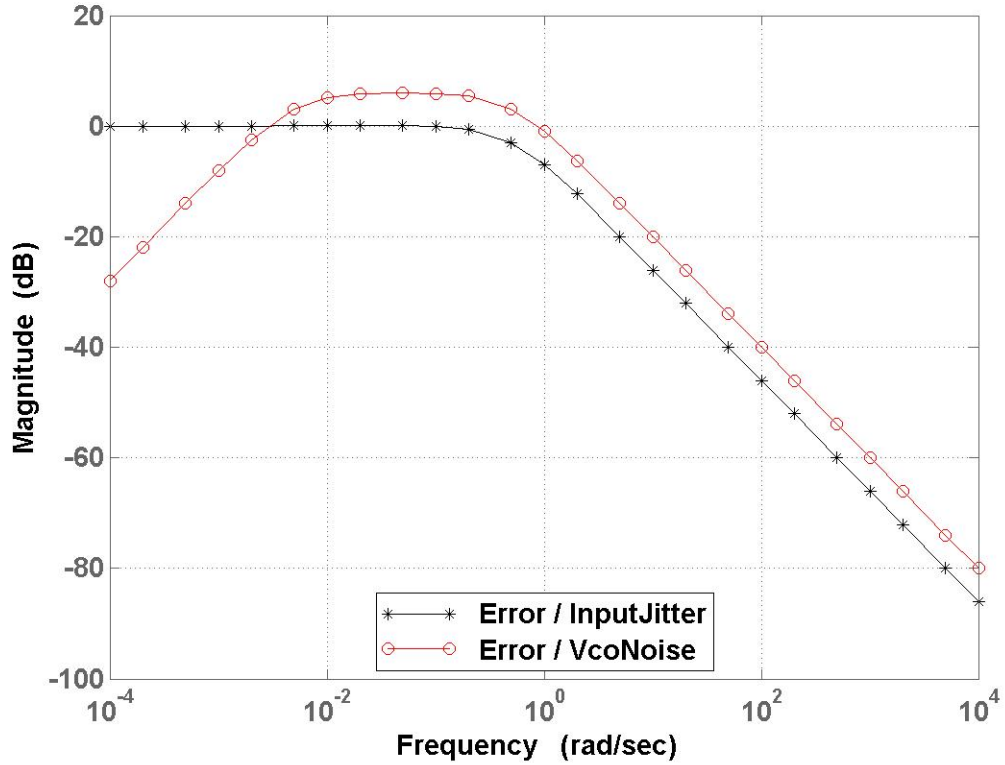
Figure 2.5: Normalized transfer functions from $\Phi_{n,in}$ to Φ_{ee} and from $V_{n,vco}$ to Φ_{ee}. TD = 0.5, $K_{PD} = 1$, $K_P = 1$, $K_i = 0.005$, Kvco = 1, and $\zeta = 5$ in this example.

The transfer function from $V_{n,vco}$ to Φ_{ee} is a band pass response (Figure 2.5). It has a single zero at 0 Hz and two poles at the same location as the input-output transfer function.

$$\frac{\phi_{ee}}{V_{n,vco}} = -\frac{\phi_{out}}{V_{n,vco}} = \frac{-s \cdot K_{VCO}}{s^2 + s \cdot K_P \cdot (TD \cdot K_{PD}) \cdot K_{VCO} + K_i \cdot (TD \cdot K_{PD}) \cdot K_{VCO}} \tag{2.12}$$

Figure 2.6 shows how the bandpass response changes when the bandwidth of the CDR is increased by a factor of 4. It is clear that increasing the bandwidth of the CDR not only reduces the width (in frequency) of the pass band, but also reduces the peak gain of the noise transfer function. This is in line with intuition which says that to minimize

the impact of VCO noise, a large CDR bandwidth is desired so that it can quickly correct for the disturbances in the VCO phase.
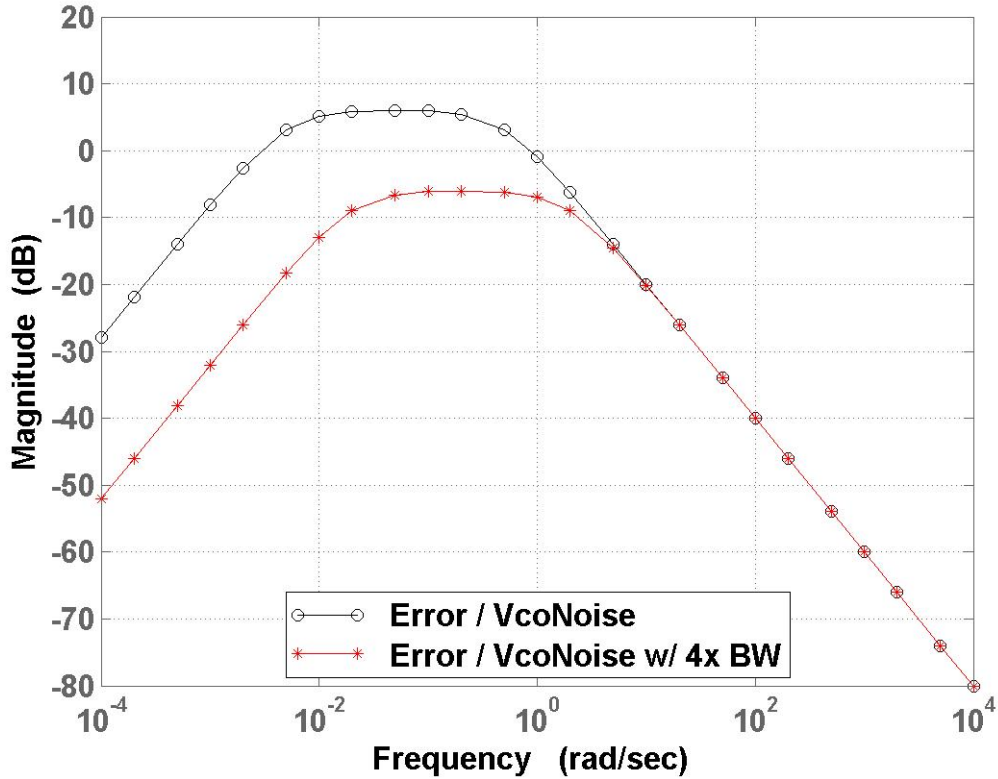
Figure 2.6: Normalized transfer function from $V_{n,vco}$ to Φ_{ee} for two different loop bandwidths. Bandwidth is adjusted by changing $K_P = 1$ and $K_i = 0.005$ to $K_P = 4$ and $K_i = 0.08$. $\zeta = 5$ for both cases for fair comparison.

Link performance degradation from uncorrelated input jitter calls for a small CDR bandwidth whereas VCO jitter calls for a large one. These opposing constraints on the bandwidth of the analog PLL based CDR make its design difficult in highly integrated chip to chip applications.[4]

---

[4] Predicting the characterisitics of supply and substrate noise during the design phase also makes it difficult to optimize the bandwidth of the CDR. Recent advances in supply noise measurement make it possible to at least garner information from a previous chip to improve subsequent revisions [30].

2.2 Semi-Digital Dual Loop CDR

The opposing constraints on the bandwidth of the analog PLL based CDR has led to the increased popularity of the semi-digital dual loop CDR [31, 32]. A simplified block diagram is shown in Figure 2.7. The dual loop architecture decouples these constraints by allowing the designer to set the bandwidth in the two loops independently so as to minimize jitter in the system. The core loop has wide bandwidth to correct noise in its VCO caused by supply, substrate, and device noise while the peripheral loop bandwidth is set low to filter jitter on the incoming data.
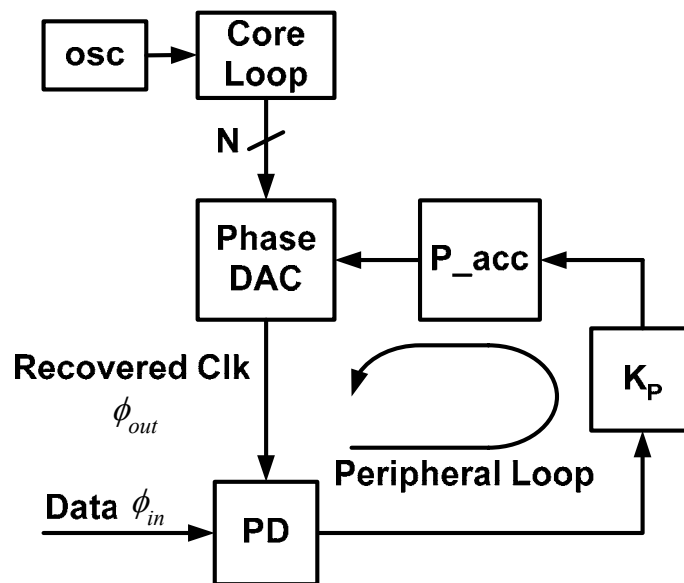
Figure 2.7: Simplified block diagram of the semi-digital dual loop CDR.

The function of the core loop in [31, 32] was to generate multiple phases from an off-chip reference clock for use by the phase DAC (a.k.a. phase interpolator). A delay locked loop (DLL) forms the core loop in [32]. However, the use of a multi-phase PLL as the core loop has become more popular since it can also perform frequency multiplication thus allowing the use of lower frequency off-chip oscillators as its reference [33].

The peripheral loop (i.e. the CDR loop) consists of a phase detector (PD), gain (K_P), accumulator (P_acc), and a phase DAC. The phase DAC converts the digital information from the accumulator into the analog phase of the recovered clock (Φ_{out}) by performing time interpolation between the output phases of the core loop. The phase DAC output is a

set of clocks which sample the data in order to generate timing error information via the PD. The error information is scaled and accumulated to close the CDR loop.

Additional benefits of this architecture deriving from the digital loop filter are reduced pattern dependent jitter caused by leakage currents in the loop filter in the presence of CID, reduced phase offset error caused by charge pump current mismatch, reduced sensitivity to supply noise, and finally loop dynamics that are not affected by process, voltage and temperature variations. For these reasons, the dual loop architecture with a multiplying PLL core is the architecture chosen for this thesis.

2.2.1 Bang-Bang Phase Detector

The issues raised concerning the linear PD in Section 2.1.1 make the bang-bang PD an attractive alternative in CDR applications. The bang-bang PD schematic and transfer function are shown in Figure 2.8. The first bang-bang PD was published by Alexander [22] and is often referred to as a binary PD. Similar to the linear PD it subtracts two pulses each generated from an XOR gate. However, both pulses are a bit period wide. The output of the top XOR is high when the previous data sample and the current edge sample are not equal. The bottom XOR gate is high when the next data sample is not equal to the current edge sample. The PD output is zero when both XOR gates are low (denoting no transitions in the data and hence no timing information) and when both are high (invalid state). When only the top XOR is high, the edge is sampling the data late and the PD output is positive (Figure 2.9). Alternately, when only the bottom XOR is high, the edge sampling is early compared to the transition and the PD output is negative. The PD is binary as it can only decide the early or late relationship but loses the phase error magnitude information. K_{PD} is difficult to define as the slope of the curve through the zero crossing is infinite (Figure 2.8 (b)). A method to approximate K_{PD} exists and will be explored in a later section.

The bang-bang PD is used in this thesis since it provides intrinsic matching of the sampling aperture for the data and edge samples that would otherwise result in a phase offset between Clk and the center of the Data signal in steady state [35]. Furthermore, its binary output simplifies integration with the digital loop filter and allows multi-phase

operation so that the CDR can operate beyond the intrinsic speed limit of a flip-flop in a given process [34].

(a)

(b)
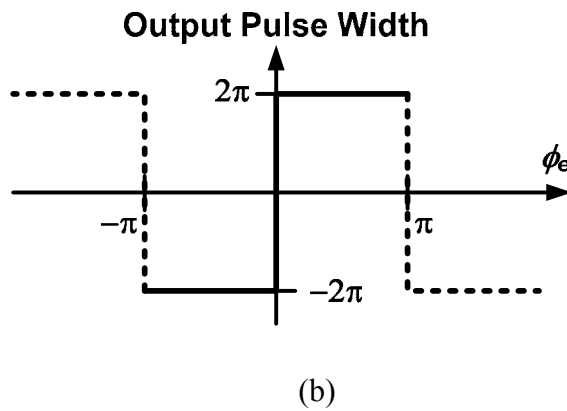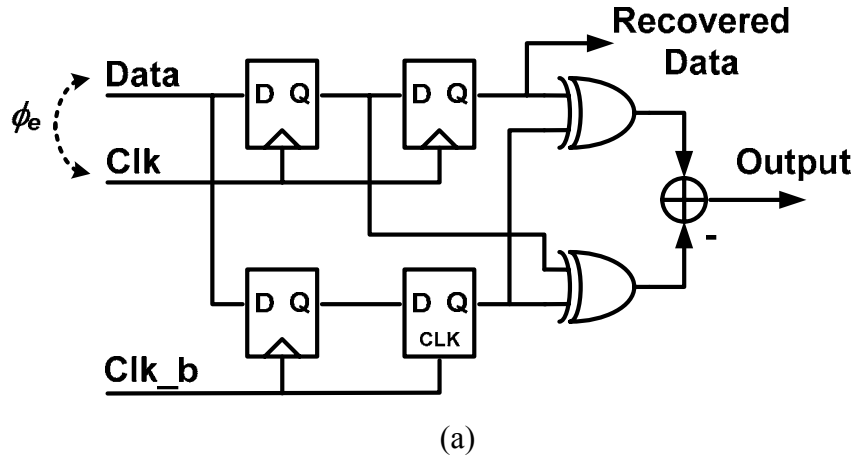
Figure 2.8: Bang-bang phase detector (a) schematic and (b) transfer function.
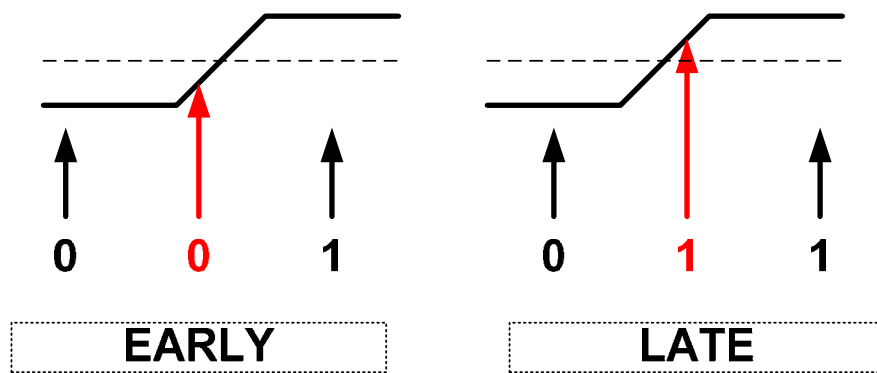
Figure 2.9: Bang-bang phase detector operation.

2.3 Summary

This chapter has reviewed the basic framework for analyzing CDR loops. Using this analysis, we have motivated the semi-digital dual loop architecture by high-lighting the opposing bandwidth constraints on the analog PLL CDR from jitter filtering and VCO noise suppression. The semi-digital dual loop architecture allows the CDR bandwidth to be set low without the concern for jitter accumulation since the VCO is outside the CDR loop. In addition, the bang-bang PD has been chosen for this thesis as it provides several key advantages including intrinsic matching of sampling apertures of the edge and data samplers.

Chapter 3

Phase DAC Design

The previous chapter introduced the semi-digital dual loop architecture. Before exploring complex estimator designs that leverage the digital loop filter, this chapter will first present research on designing phase DACs that enable accurate phase estimation. The primary focus will be on minimizing phase DAC nonlinearity, PLL jitter, and static phase offsets in the multi-phase clocks. Section 3.1 will first provide an overview of the link for which this phase DAC was designed. Section 3.2 will then detail the circuits that were used. The measured results are included in Section 3.3

3.1 Link Overview

A block diagram of the RX using the dual loop CDR is shown in Figure 3.1. The data rate of operation is 3.125Gbps. In the 0.25μm CMOS process that was used, the bit time is only 2·FO-4.[5] Due to the limited speed of this technology, we use parallelism to allow the on-chip circuits to operate at a lower frequency than the off-chip data rate. Furthermore, (de)-multiplexing occurs at the pads that are connected to low impedances (25 or 50 Ohms), which provide high bandwidth despite the large pad capacitance. Hence, 5 bits

---

[5] The minimum pulse width that can reliably propagate through a fanout of 4 (FO-4) inverter chain is 4·FO-4 [60]. This is true for any process and puts a bound on the maximum achievable on-chip data rate using CMOS gates.

are transmitted every reference clock cycle (625MHz) via a multiplexing TX. The RX also performs demultiplexing at the pad. For a multiplexing factor of 5, 10 phases equally spanning a 625MHz period are needed to sample the data stream at the Edge and Data times. Instead of using multiple phase DACs that would increase area and power consumption, a VCO (INJVCO) is injection locked to the single 625MHz output of the phase DAC in order to generate the ten phases. The core PLL serves as a frequency multiplier (factors of 1, 2 and 4) as well as a multiphase generator (10 output phases). As we were able to obtain SAW oscillators at 625MHz, the measured results were taken with the PLL frequency multiplication factor set to 1. A synchronizer (SYNC) aligns the output of the samplers to a single clock domain of 312.5 MHz in which the digital loop filter operates.
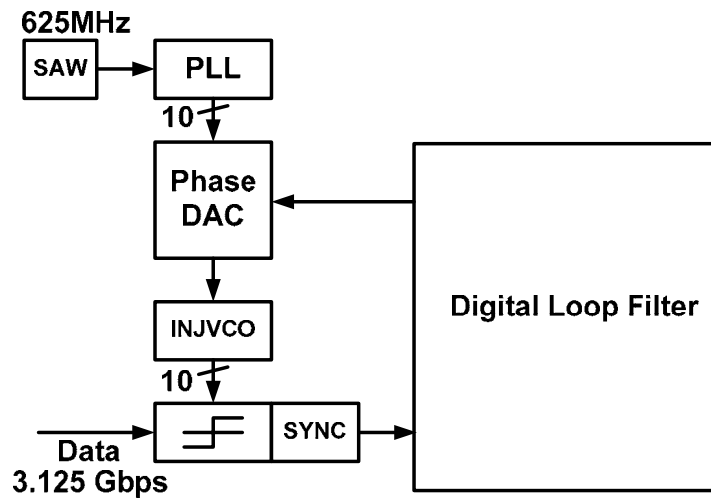
Figure 3.1: Block diagram of the receiver for which this phase DAC is designed.

3.2 Circuit Implementation

3.2.1 Adaptive Bandwidth Phase DAC

Figure 3.2 is the schematic of the phase DAC. The phase DAC consists of two 5:1 clock MUXes and a sixteen step phase interpolator. The clock MUXes select two adjacent phases ($\theta[n]$ and $\theta[n+1]$) from the ten phases ($\theta[9:0]$) of the core PLL. The code to each MUX is one-hot. The interpolator blends $\theta[n]$ and $\theta[n+1]$ to generate $\Phi[n]$ with fine

phase resolution. Thermometer coding is used to ensure monotinicity which would otherwise lead to differential nonlinearity (DNL).

One challenge of phase DAC design is to ensure good linearity over PVT corners. Sidiropoulos showed that the ratio of the RC time constant at the interpolator output and the phase spacing between $\theta[n]$ and $\theta[n+1]$ (called Δt) has a strong influence on its linearity [61]. Weinlader extended that research to show that there is also a dependency on the slew rate of $\theta[n]$ and $\theta[n+1]$ [62]. To minimize nonlinearities, the RC time constant at both $\theta[n]$ and $\theta[n+1]$ as well as at the interpolator output should be greater than $2 \cdot \Delta t$. However, this requirement results in slow clock edges that are susceptible to power supply noise induced jitter. Thus, we set the RC time constants to be greater than Δt. Since Δt is a fixed fraction of a UI, this design constraint requires the component bandwidths of the phase DAC to scale proportionally with the data rate.[6]
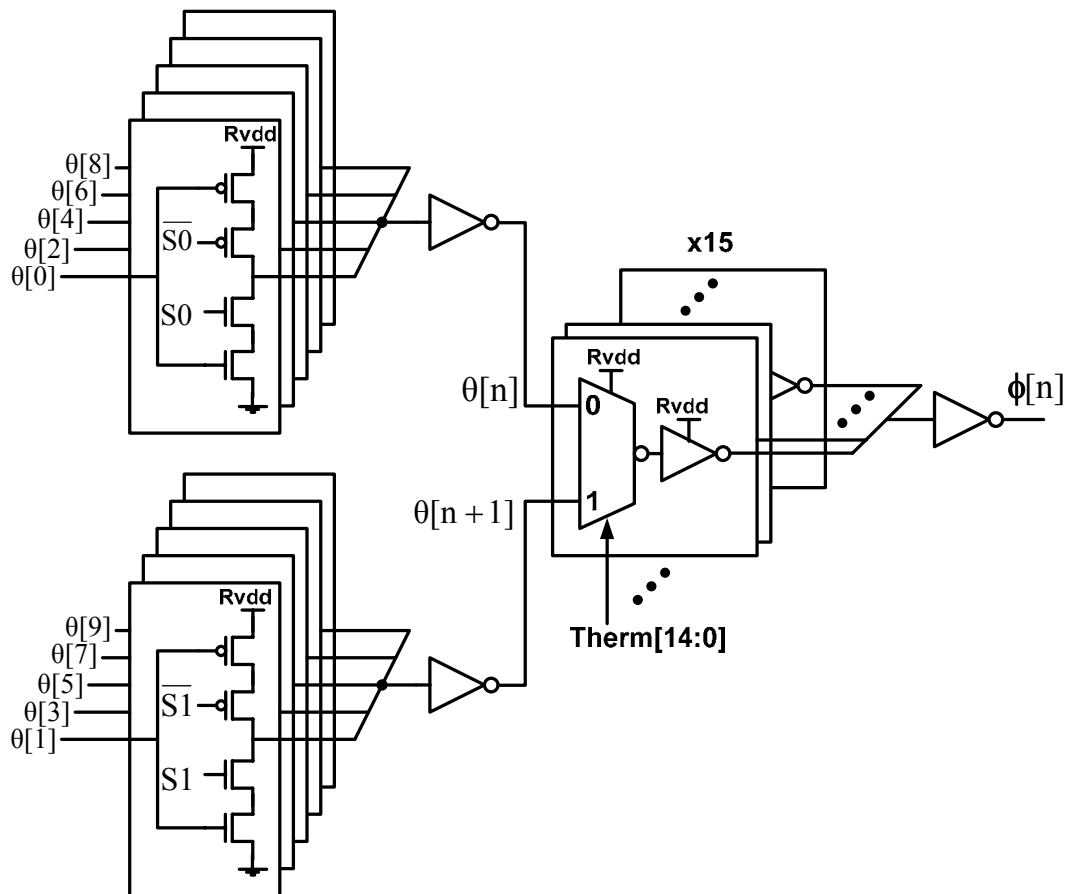
Figure 3.2: Schematic of an adaptive bandwidth phase DAC using CMOS gates. Rvdd is the regulated supply that adjusts the component bandwidths to track the data rate.

---

[6] UI stands for 'Unit Interval' which is a commonly used term for a single bit period.

We achieve this by using a CMOS inverter based ring VCO in our PLL. The control voltage of the PLL is the supply voltage at which the delay of the inverters gives us the correct data rate for the PVT corner. In our system, the delay through two inverters is equal to a UI. Since the speed (and bandwidth) of digital CMOS gates increases almost linearly with supply voltage and the delay of other CMOS gates tracks that of an inverter proportionally across PVT, a fixed ratio between Δt (inverter delay) and the RC time constants can be maintained if the phase DAC is built with CMOS gates that operate on the same regulated supply as the ring VCO [59]. Since the delay through a CMOS gate is approximately equal to its RC time constant, Δt can be made less than RC by ensuring that Δt is smaller than the delay through any of the gates that constitute the phase DAC.[7]

Figure 3.3 is the simulated transfer function of the phase DAC that demonstrates this bandwidth tracking property. The solid line is the ideal transfer function. The five marked lines are for various process corners of the PMOS and NMOS transistors (SS, SF, FS, FF, and TT). The regulated supply voltage (Rvdd) is 1.8V in all cases. Since Rvdd is held constant, the data rate of operation is different in each of the cases (3.5Gbps – 6.15Gbps). The vertical axis is the phase DAC delay normalized to the ideal phase spacing which is a sixteenth of Δt. The integral nonlinearity (INL) varies from 0.76 LSB (SF corner) to 1.9 LSB (FS corner). DNL is about 1 LSB and is worst at the end of the interpolator range. The adaptive bandwidth property results in a relatively consistent transfer function despite the extremely different operating frequencies.

---

[7] By approximating an inverter with a switched current source driving a load capacitance and assuming that the output of a CMOS gate starts to transition when the input is at its 50% point, it can be shown that the rise or fall time of a CMOS gate is twice the delay through the gate. Since the 10%-90% rise time of a RC circuit is about 2.2·RC, RC is approximately equal to the delay through a CMOS gate.
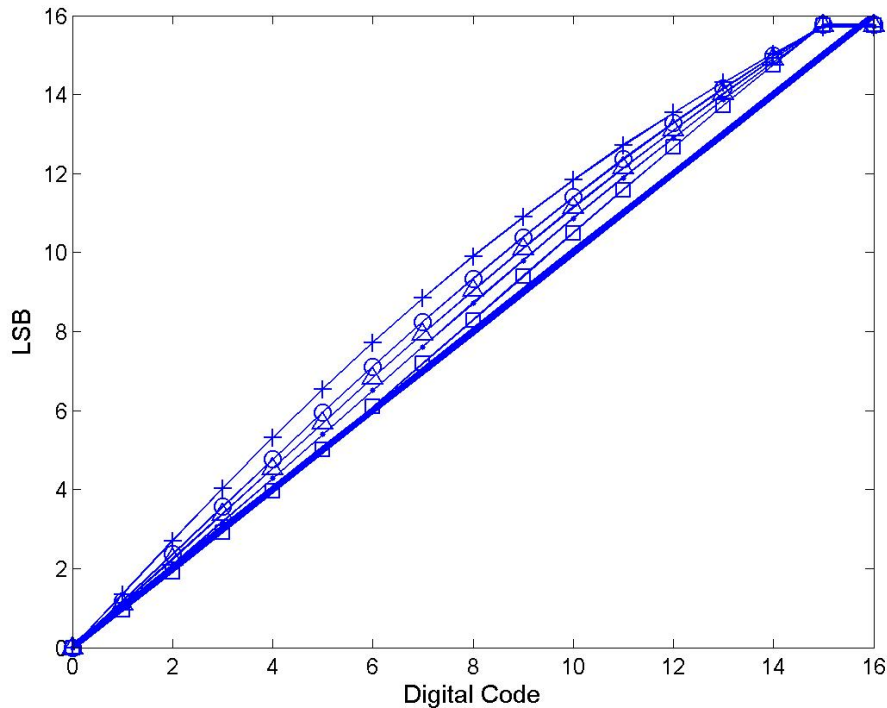
Figure 3.3: Simulated phase DAC transfer function. The solid line is the ideal transfer function. Results are for five process corners (SS, SF, FS, FF, TT) and constant voltage (1.8V) and temperature (25°C).

To minimize quantization noise from the phase DAC, we want a resolution of 64 steps per UI. The resolution of Figure 3.2 is only half that. To achieve a two-fold increase in resolution, an additional interpolator leg with half the drive strength (binary weighted) is used. In addition, the transfer function of Figure 3.3 exhibits significant nonlinearity. A 100 iteration monte-carlo simulation to account for random mismatches of transistor properties [56] shows that the standard deviation (σ) of the maximum DNL is 0.06 LSB and the σ of the maximum INL is 0.7 LSB. To correct for these, the phase DAC resolution is increased by an additional four times. By making more phase positions possible, we can pick out the 64 digital codes that gives the best INL and DNL performance. The 'picking' will be performed by a lookup table (Section 3.2.4 The additional binary weighted interpolator legs are shown in Figure 3.4. Binary weighting is achieved by proportionally increasing the channel length of the CMOS inverters. It is important then to increase the widths of the transistors in the MUXes to preserve constant bandwidth at each stage. While full thermometer coding would be ideal to avoid

discontinuities caused by binary coding and device mismatches, the routing area of the digital signals becomes significant, hence leading to this compromise.
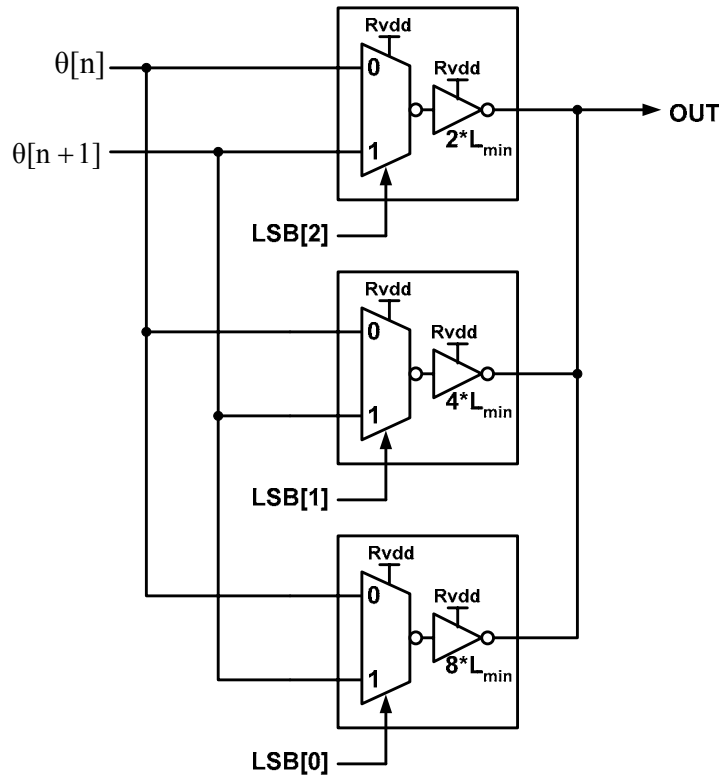
Figure 3.4: Binary weighted interpolator legs to increase the resolution eight-fold.

The supply to the phase DAC (Rvdd) is controlled using the replica compensated linear regulator of [63]. The linear regulator serves not only to adapt the bandwidth of the phase DAC components but also to isolate them from power supply noise that would result in jitter. This regulator provides better power supply rejection (PSRR) for a given power budget in comparison to prior art. The superior performance derives from the additional feedback loop (V_{bp} to V_{rep}) that rejects supply induced disturbances quickly (Figure 3.5). For this scheme to work, the I-V characteristic of the replica load must match that of the phase DAC across various operating conditions. The replica is a diode connected PMOS load in series with an NMOS. The source and bulk of the PMOS as well as the gate of the NMOS are connected to V_{rep}. The current of the replica tracks the phase DAC to within 5%. Simulations show a PSRR better than 15 when the supply voltage is reduced by 10% with a fall time of 10ps.
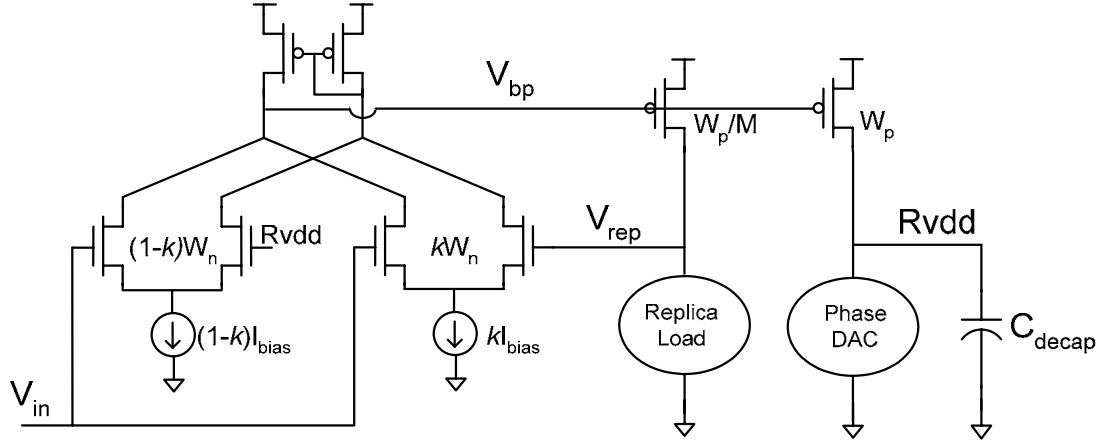
Figure 3.5: Replica compensated linear regulator of [63]. V_{in} is the control voltage of the PLL. M=32 and k=0.5.

3.2.2 Adaptive Bandwidth Phase Locked Loop

As mentioned earlier, the control voltage of the core PLL adjusts the bandwidth of the phase DAC components according to the data rate in order to achieve good linearity. Since a larger PLL bandwidth reduces jitter due to VCO frequency instability (from device or supply noise), we would like the PLL bandwidth ($\sim\omega_n$) also to scale proportionally with the off-chip oscillator frequency (ω_{ref}). Furthermore, it is desired to maintain a fixed ζ so that the PLL does not become unstable at certain data rates or PVT corners. Sidiropoulos showed that for PLLs using regulated-supply inverter based VCOs, these design goals are achieved by scaling the charge pump current and the output resistance of the linear regulator that supplies current to the VCO [57]. More specifically, the charge pump current must scale proportionally with the VCO current (I_{vco}) and the resistance must scale inversely with $\sqrt{I_{vco}\cdot\beta}$. β is the process transconductance ($\mu\cdot C_{ox}$). The charge pump and linear regulator circuits used in our test chip are identical to those in [57]. Simulations show that over a ω_{ref} range from 385 to 770 MHz, ζ is 0.9 to within 2.5% while the ratio between ω_n and ω_{ref} is 0.025 with 15% error.[8]

The VCO schematic is shown in Figure 3.6. It is composed of two 5 stage ring oscillators that are coupled so as to provide differential clocks. For example, the rising

---

[8] Researchers have often cited the ideal ratio between ω_n and ω_{ref} (for clock generation applications) to be 0.1. Due to the low feedback divider ratio in our design and the ensuing high ω_{ref}, ratios above 0.025 exhibited significant phase margin degradation from loop delay. Since loop delay is a fixed amount of time, its phase contribution is proportional to the bandwidth. The phase margin of our design is 75°.

edge of θ[0] and the falling edge of θ[5] occur at the same time. The 10 phases are spaced by a F0-4 delay (when parasitic wire capacitances are accounted for). The complex cross coupling is for forward interpolation which increases the oscillation frequency at a given regulated supply (Rvdd) [36].
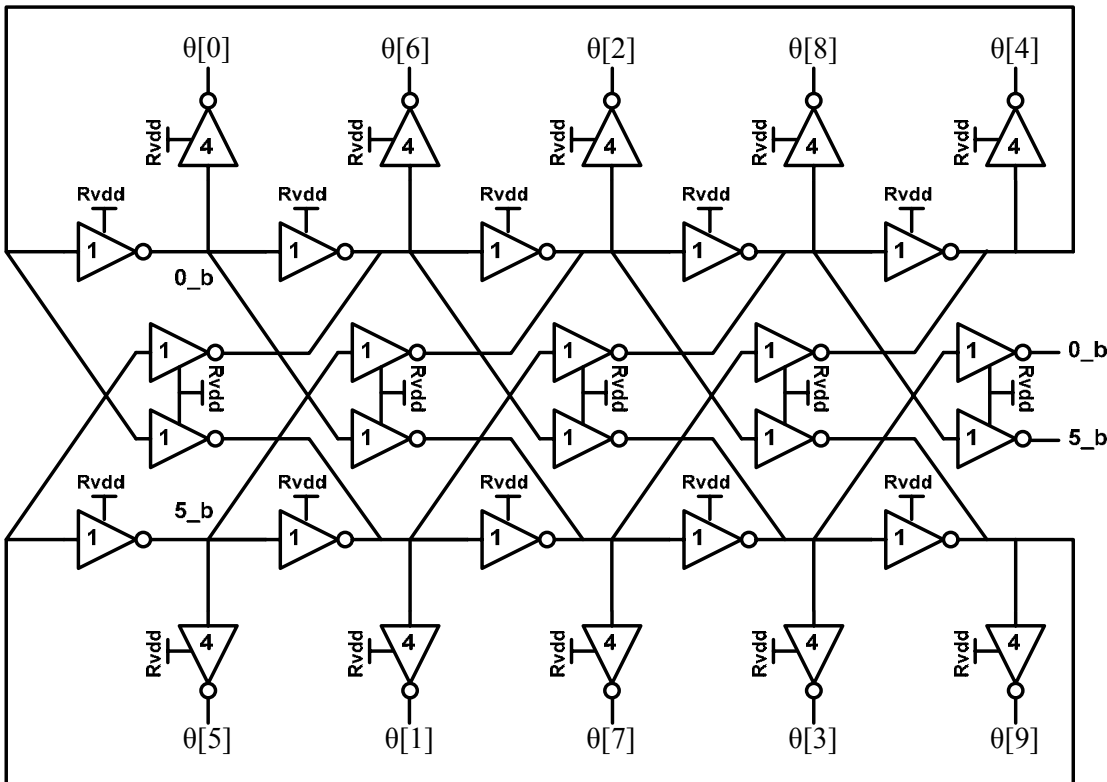
Figure 3.6: Forward interpolating coupled ring oscillator.

Great care is taken to minimize static phase offsets caused by layout mismatches. The VCO is laid out in a single row of ten units each consisting of three inverters (one for the main ring, one for cross coupling, and one for driving the output). A replica dummy unit is added at each end for poly density matching. Below these unit elements is a wiring channel consisting of 21 wires. Ten of the wires are for the internal nodes of the VCO (input of the output drivers). The other 11 wires are ground shields placed between and outside these clock wires. The clock phases in the wiring channel are arranged such that its own transitions do not coincide with those of its closest neighbors. This minimizes any offsets due to Miller capacitance. Even if the layout is perfect, a monte-carlo simulation of 100 runs shows that the phase spacing has a σ of 6% (normalized to the nominal spacing). These offsets will result in INL in the phase DAC transfer function and must be

corrected. The phase offset correction circuit is shown in Figure 3.7. It consists of binary weighted PMOS and NMOS capacitors with switches in series. The correction circuit uses both PMOS and NMOS to avoid excessive duty cycle distortion. The correction circuit has a resolution of 3ps and a range of 48ps at 625MHz. Thus its correction range is +/-2.6·σ. The nominal control signal is 4'b0111 to allow both positive and negative phase correction.
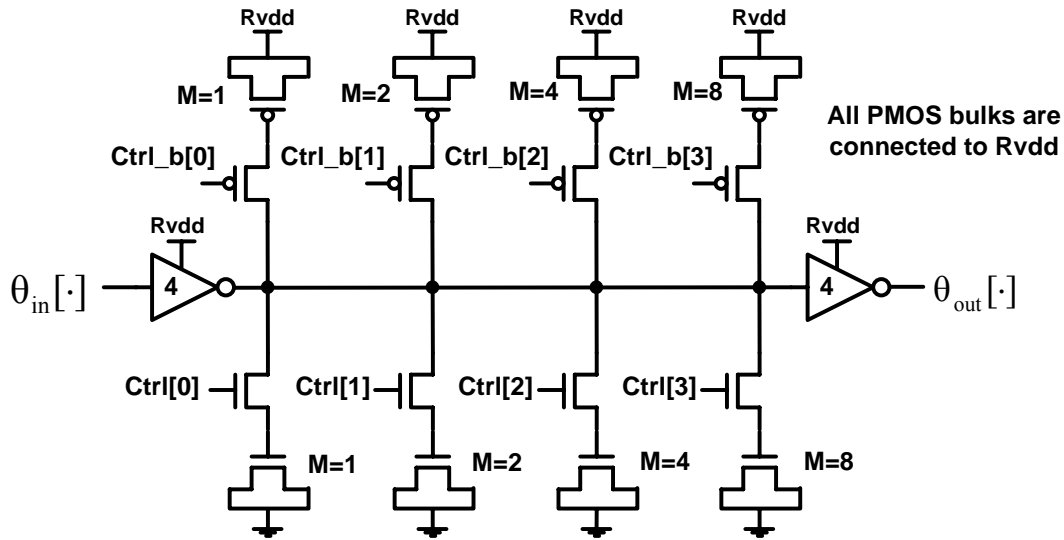
Figure 3.7: Circuit for correcting phase offsets of the VCO.

The target ω_{ref} of 625MHz pushes the performance of the phase-frequency detector (PFD). The latch-based PFD of [55] is used to enable a higher operating frequency than that achievable using a flip-flop based design. The minimum period that the PFD will operate is determined by the delay of its reset path. By replacing the D-flip flops with latches, the operating frequency range of the PFD is increased by a factor of 2. This translates into a minimum period equal to the reset path delay. For good linear operation in the presence of jitter, PFD operation should be limited to half of this maximum frequency. In the 0.25μm process used, this PFD works reliably beyond 800 MHz over PVT.

3.2.3 Injection Locked VCO

The schematic of the injection locked VCO (INJVCO) is shown in Figure 3.8. Prior work [42] used an INJVCO in order to filter high frequency jitter. Their phase-domain analysis showed that the INJVCO can be viewed as a single pole filter. Reducing the drive strength of the input relative to those of the ring elements lowers the INJVCO bandwidth and hence increases the amount of filtering it provides. This is because the circuit retains more memory of its past state.

While not addressed by [42], a low bandwidth INJVCO used in a CDR loop creates an unwanted pole that degrades its stability. On the other hand, making the input too strong results in unequal phase spacing. As the primary purpose of our INJVCO is to re-generate 10 phases from a single phase clock, it is designed to favor its input as much as possible without causing excessive static phase offsets. Simulations of the circuit in Figure 3.8 show that the maximum phase error occurs at θ[3] and is less than 10ps over a wide range of operating frequencies. The phase spacing of INJVCO also has a σ of 6% due to random transistor mismatches. These phase spacing errors will increase CDR jitter and are thus corrected using the same correction circuit as for the PLL VCO. The ten phases of the INJVCO clock ten samplers for timing and data recovery.
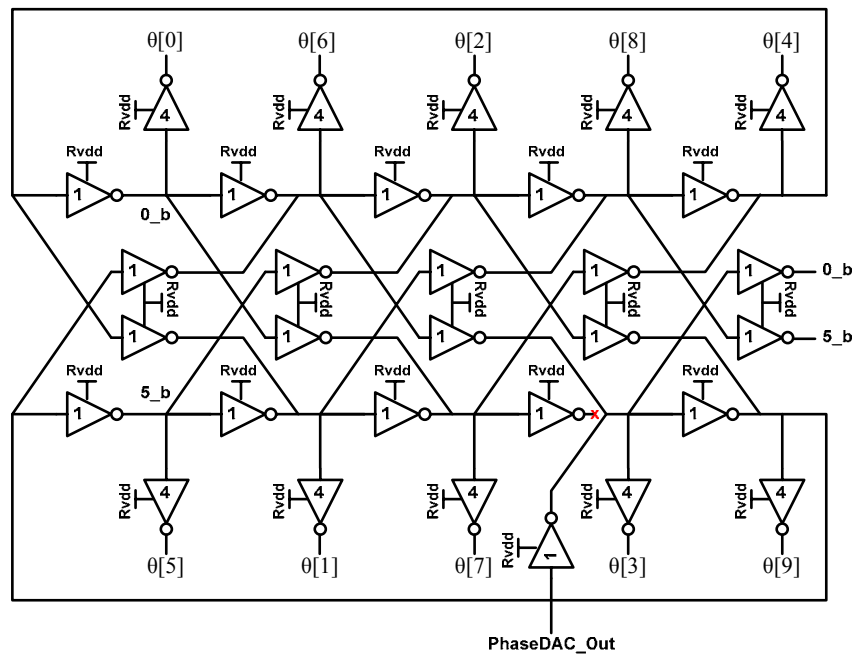
Figure 3.8: Injection locked VCO for re-generating 10 phases from the single phase DAC output.

3.2.4 FSM and Lookup Table

The FSM has two states: Coarse and Fine (Figure 3.9). The Coarse state is 4 bit binary and represents the two phases ($\theta[n]$ and $\theta[n+1]$) to interpolate between. Since there are only ten possible Coarse settings, the FSM transitions between state 4'b0000 and 4'b1001. Digital logic converts this binary number into two sets of 5 bit one hot code to be used by the clock MUXes (Figure 3.2). For example, when Coarse is 4'b0000, $\theta[0]$ and $\theta[1]$ are chosen by the MUXes. When Coarse transitions to 4'b1001 (e.g. due to a frequency offset), $\theta[0]$ and $\theta[9]$ are chosen. In this way, only one of the two MUX settings can change in any given clock cycle.
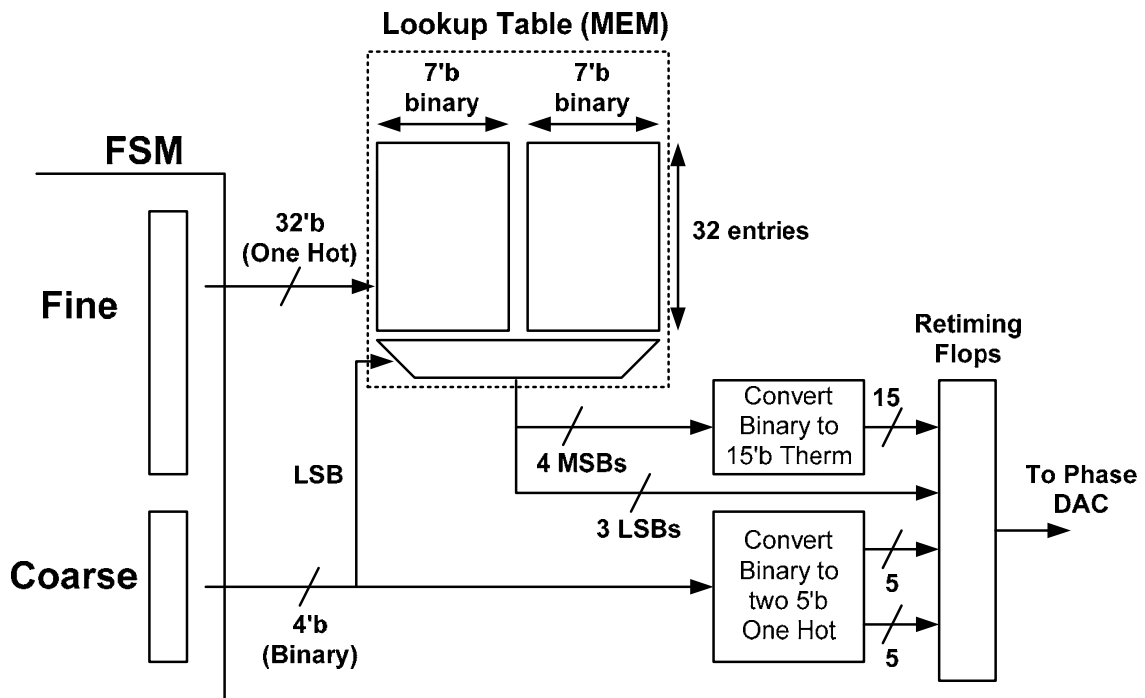
Figure 3.9: Interface circuits between the FSM and the phase DAC.

The Fine state is in 32 bit one hot code. The hot bit selects one out of 32 rows (each 14 bits wide) in the lookup table. The LSB of the Coarse state denotes whether the earlier phase is an even or odd phase of the core loop. This bit is used to select the relevant half of the 14 bit word. Since the interpolator control contains 15 bits of thermometer code (equivalent to 4 bits of binary) and 3 bits of binary code (for the LSBs), digital logic converts the binary output of the table into the appropriate format. The Fine state is

implemented as a barrel shifter which shifts the hot bit toward the MSB when the delay is to be increased. When the barrel shifter wraps around, the Coarse setting is increased (when wrapping from MSB to LSB) or decreased (when wrapping from LSB to MSB) accordingly. The lookup table is implemented as a FIFO using D flip flops. Its entries are written once during a calibration stage. Section 3.2.5 discusses measurement circuits that help us determine which codes minimize the INL and DNL.

A by-product of this hardware is that unlike prior implementations [61] which allowed no more than one control bit to change each clock cycle, multiple bits can change simultaneously. This requires us to time the transitions of these control signals as close to each other as possible so that the phase DAC does not see invalid control signals no matter how brief since this will lead to unwanted clock jitter at its output. Retiming flip-flops are placed right before the phase DAC for this purpose.

3.2.5 Phase Measurement Circuits

In order to properly set the various phase correction circuits, additional circuits are needed to measure phase spacing. To measure the phase spacing between any two clocks, the circuit of Figure 3.10 consisting of two samplers, an XOR gate, and two 20-bit counters is used [62].

The operation of these circuits will first be explained for the VCO output phases. In this case, $\Phi 1$ and $\Phi 2$ are the outputs of the two clock MUXes ($\theta[n]$ and $\theta[n+1]$). The Time Counter is used to set the acquisition time. Until this counter reaches all ones, the Histogram Counter tallies the number of clock cycles for which the output of the XOR gate is high. The XOR gate is high when a transition has occurred between $\theta[n]$ and $\theta[n+1]$. As long as the frequency of the "Random Signal" is chosen such that its transitions occur at any time in the period of the phases with equal probability, the Histogram Counter value is proportional to the phase spacing. The same measurement is repeated for each of the 10 settings for $\theta[n]$ and $\theta[n+1]$ and the counter values are read out using a scan chain. The difference in the Histogram Counter values is therefore proportional to the phase spacing error and is used to correct them. The 20 bit length of the time counter is large enough to ensure that the measurement resolution is better than

the smallest phase spacing of interest (less than 1ps) and that any effect of jitter on the measurement is averaged out.

In order to measure the phase DAC nonlinearity, additional MUXes are placed at $\Phi1$ and $\Phi2$ such that the phase DAC output can be routed to either sampler. MUXes are needed in both paths so as avoid introducing a phase offset between $\theta[n]$ and $\theta[n+1]$ for the VCO measurement above. The histogram counts of all possible codes are gathered and the codes that give increments closest to the ideal are chosen to be written into the lookup table.

The INJVCO phase spacing is measured in much the same way as the VCO. The only difference is that the existing RX front end samplers are used. Ten XOR gates are added to the RX front-end. The XOR gate output in question is selected via a MUX and routed to a Histogram Counter.
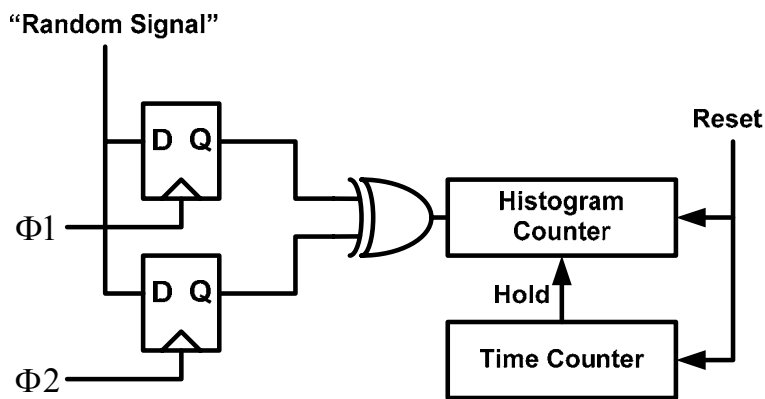
Figure 3.10: On-chip phase measurement circuit.

3.2.6 High Speed Sampler

The samplers used for on-chip phase measurement are clocked regenerative latches (Figure 3.11) which was first published in [58]. When Clock is low, the tail device is turned off and the sampler is disabled. Furthermore, its outputs are pre-charged to Vdd and are equalized to each other to completely reset the sampler. On the rising edge of Clock, the sampler amplifies the differential voltage caused by the inputs via positive feedback. The output of the sampler is full-swing digital. Since the output is only valid for a short time during each clock period, the sampler is followed by an SR latch. This

sampler is widely used due to its lower power and high speed performance. It is also used in the RX front end for Data and Edge sample resolution in this chip.
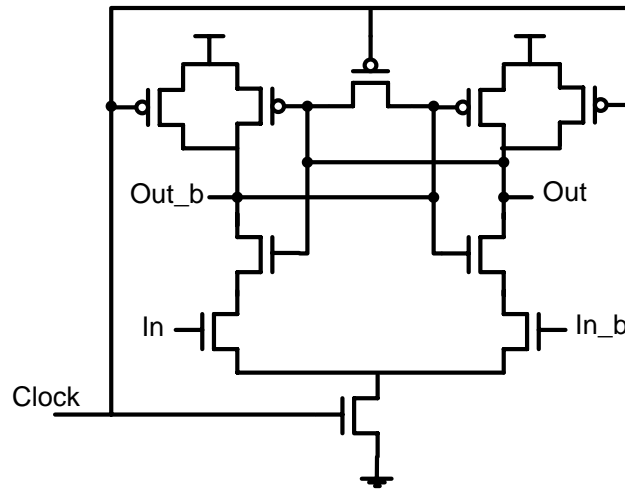
Figure 3.11: High speed sampler used in RX front-end and on-chip measurement circuits.

3.3 Measured Results

The first thing we noticed when the test-chip arrived is that despite all the accommodations made to correct for nonlinearities in the phase DAC, the transfer function showed a large step at the edge of the interpolation boundary that is uncorrectable using our correction circuits (Figure 3.12). The phase offsets of the VCO which would lead to INL were corrected and the resulting measured performance was DNL=31.4ps (6.28 LSB), INL=33.2ps (6.64 LSB). This type of nonlinearity is typically from capacitive coupling between the input and the output of the interpolator. However, our circuit in Figure 3.2 does not exhibit this type of coupling since the phase DAC output is completely isolated from $\theta[n]$ and $\theta[n+1]$. A post-layout simulation shows that there is instead significant capacitive coupling directly between $\theta[n]$ and $\theta[n+1]$ which pulls these two clocks together and reduces Δt. This is because these two clocks run in parallel along the complete length of the phase DAC in order to distribute them to all the unit elements in the interpolator. This is a design error and should have been caught before tapeout. Not anticipating this type of problem, post layout simulation was run only on the interpolator portion of the phase DAC. A simple ground shield between the two

signals would also have prevented this error. Because of this problem, the time difference between when the interpolator code is all zeros and when it is all ones is 20ps less than the ideal Δt. The other 10ps of nonlinearity does not appear in post-layout simulations. The measured jitter of the core loop PLL is 1.57ps rms.
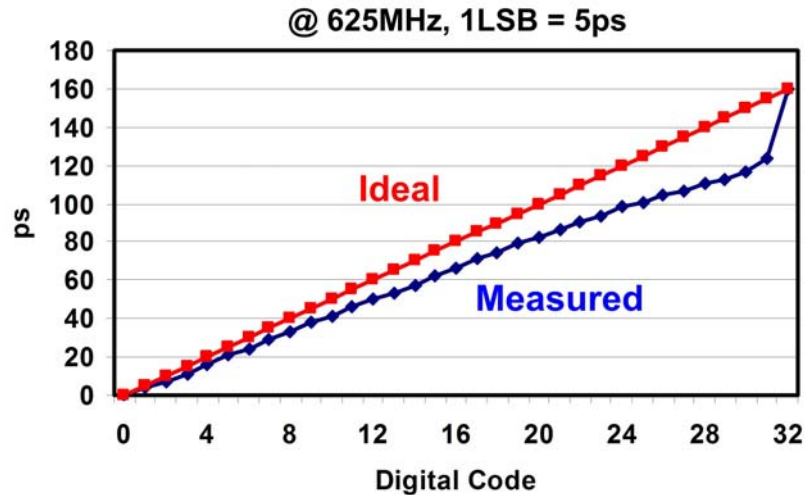
Figure 3.12: Measured phase DAC transfer function. 32 steps span half of a UI. The other half turned out to be almost symmetric and hence is not shown.

This phase DAC was used in Chapter 5 and was the first chip we taped out. For work done in Chapter 4 and Chapter 6, we used a phase DAC made available to us by Rambus which does not have this large nonlinearity. This circuit is described in Chapter 4.

3.4 Summary

We have shown circuits that achieve low phase DAC nonlinearity, PLL jitter, and static phase offsets in the multi-phase clocks. Using an adaptive bandwidth phase DAC, linearity can be maintained over PVT. This necessitates the use of an adaptive bandwidth PLL to adjust the supply of the phase DAC according to the operating conditions. This PLL has the added benefit of maintaining relatively constant bandwidth and stability over PVT so as to minimize supply noise induced jitter.

With advanced processes, random transistor mismatches dominate INL and DNL of the phase DAC. This chapter has discussed circuits that enable us to correct these after

chip fabrication. Unfortunately, a particular form of nonlinearity that these circuits could not fix limited the measured performance of this phase DAC.

Chapter 4

Plesiochronous Systems

Plesiochronous systems are those that have a fixed frequency offset between the TX and RX. Such systems arise when the TX and RX each have their own independent clock sources that are nominally but not exactly identical in frequency. The frequency offset is typically less than a few hundred parts per million (ppm) [14, 16]. This chapter shows that phase estimator design can improve the performance of CDRs by decoupling the need for a large bandwidth to track the frequency offset from the need for a low bandwidth to filter the jitter on the data. The design of an appropriate phase estimator for plesiochronous systems is detailed in Section 4.1 Section 4.2 provides an analysis of this phase estimator using the framework of Chapter 2. This will help us understand the measured results. Finally, the benefit of using an appropriate phase estimator for a given application is demonstrated in Section 4.3 by using jitter tolerance, which is an industry standard method of specifying RX performance [14-16].

4.1 Second Order Estimator

When viewed as a phase estimator, the conventional semi-digital dual loop CDR of Figure 2.7 predicts the future phase position of the data to be same as the current. A simple thought experiment will make this clear to the reader. Imagine that after the CDR

is locked to the data, no timing information is available for some period of time. This can occur in the presence of CID. When this happens, this CDR will maintain the position of the recovered clock (Φ_{out}) at the last known phase position of the data. Put in a different way, the digital estimate of the current phase position of the data stored in the accumulator is not changed since there is no new error information. Due to this property, we expect this CDR to be sub-optimal when the phase of the data is changing over time, which is the case when a frequency offset exists between the TX and RX. Recall from Section 1.1.1 that when a frequency offset exists, the deterministic phase offset trajectory is a linear ramp.

For such systems, a better CDR would estimate the frequency offset and use it to predict the future phase position. This is exactly what the second order dual loop CDR does by adding in a frequency tracking loop consisting of a gain (K_i) and a frequency accumulator (F_acc) as shown in Figure 4.1. The phase and frequency accumulators contain digital estimates of the respective quantities. K_P and K_i refer to the proportional and integral gains that also represent the rate of update of the phase and frequency estimates. The operation of the second order phase estimator is captured by equations (4.1)-(4.4). K_D is the conversion gain of the phase DAC in units of UI·LSB^{-1}. The LSB is that of the phase DAC and not the phase accumulator (P_acc) which we will see later can have higher resolution. The digital phase estimate (Φ_{est}) is a real number in units of (phase DAC) LSB. Both the digital frequency estimate ($Freq_{est}$) and the proportional gain (K_P) are in units of LSB·cycle^{-1}. The integral gain (K_i) is in units of LSB·cycle^{-2}.

$$\phi_{out}(n) = K_D \cdot \phi_{est}(n) \tag{4.1}$$

$$\phi_{est}(n+1) = \phi_{est}(n) + Freq_{est}(n) + \phi_e(n) \cdot K_P \tag{4.2}$$

$$Freq_{est}(n+1) = Freq_{est}(n) + \phi_e(n) \cdot K_i \tag{4.3}$$

$$\phi_e(n) = \text{sgn}(\phi_{in}(n) - \phi_{out}(n)) \tag{4.4}$$
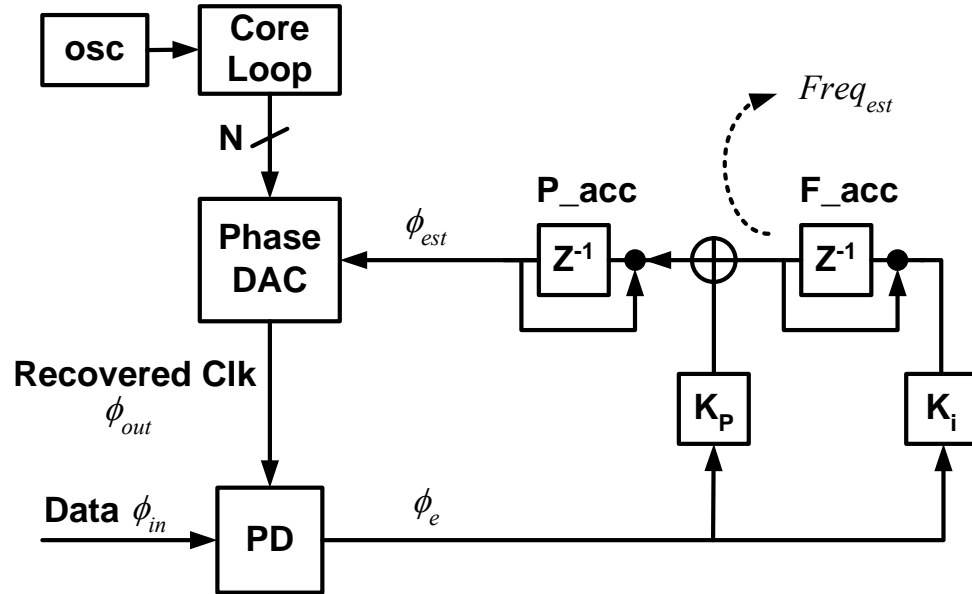
Figure 4.1: Simplified block diagram of the second order dual loop CDR.

We apply the previous thought experiment to the second order CDR by assuming that there is no timing information for a certain length of time. Contrary to before, the digital estimate of the current phase position of the data (Φ_{est}) continues to be updated by the frequency estimate ($Freq_{est}$). Since the addition of a constant number every clock cycle creates a ramp, this CDR is able to predict where the data transitions will be in subsequent clock cycles.

Interestingly, this CDR loop has the same structure as the PLL consisting of proportional and integral branches and two integrators (equivalently accumulators). The only difference is that the VCO implicitly performs one of the integrations in the PLL while the second order dual loop CDR has an explicit accumulator driving a phase DAC. Hence, the PLL also shares the predictive qualities of the second order dual loop CDR in plesiochronous systems. However, the substitution of the VCO with a digital accumulator and phase DAC removes the noise port at the VCO input which allows the designer to set the CDR bandwidth low without concern for jitter accumulation.

4.2 CDR Loop Dynamics

The bang-bang PD, due to its undefined K_{PD}, makes the analysis of this CDR difficult. This single bit quantizer leads to non-linear loop dynamics in bang-bang CDRs. Several researchers have recently tried to analyze such systems. Since the PD can only generate a single bit binary update every clock edge, the CDR loop will exhibit slew rate limiting in the presence of large phase or frequency offsets. Walker first demonstrated this behavior for analog bang-bang CDRs, and derived approximate expressions for the minimum phase and frequency offset at which slew rate limiting occurs [35]. J. Kim further extended the analysis by taking into account the effect of loop delay on slew rate limiting [36]. In addition, he derived an expression for the worst case limit cycle (i.e. steady state dither jitter) by using phase portrait analysis assuming the absence of any noise sources. A similar analysis to [36] was done for digital CDRs in [37] leading to similar results.

However, it has proven difficult thus far to capture the effect of slew rate limiting, limit cycles, and external jitter altogether within a single analytical framework. Hence, time-step simulations are still used to predict the CDR's behavior.

While an approximation, we can continue to apply linear systems theory to bang-bang CDRs by recognizing that the bang-bang PD is linearized in the presence of high frequency jitter when the CDR is locked [35, 38-40]. This allows us to apply the equations found earlier in Section 2.1 to gain intuition into the CDR loop dynamics when there is sufficient jitter in the system. The effective transfer function of the PD is the convolution of the noise probability density function (PDF) with the ideal transfer function of the PD (Figure 2.8 (b)). A graphical example is shown in Figure 4.2 for the case when the noise PDF is uniform. As the noise increases (i.e. the noise PDF becomes broader), the effective K_{PD} is reduced which is consistent with the findings of [36].[9]

---

[9] Note that the instantaneous transfer function of the bang-bang PD is still accurately described by Figure 2.8 (b). This linearized transfer function is the *statistical average* transfer function.
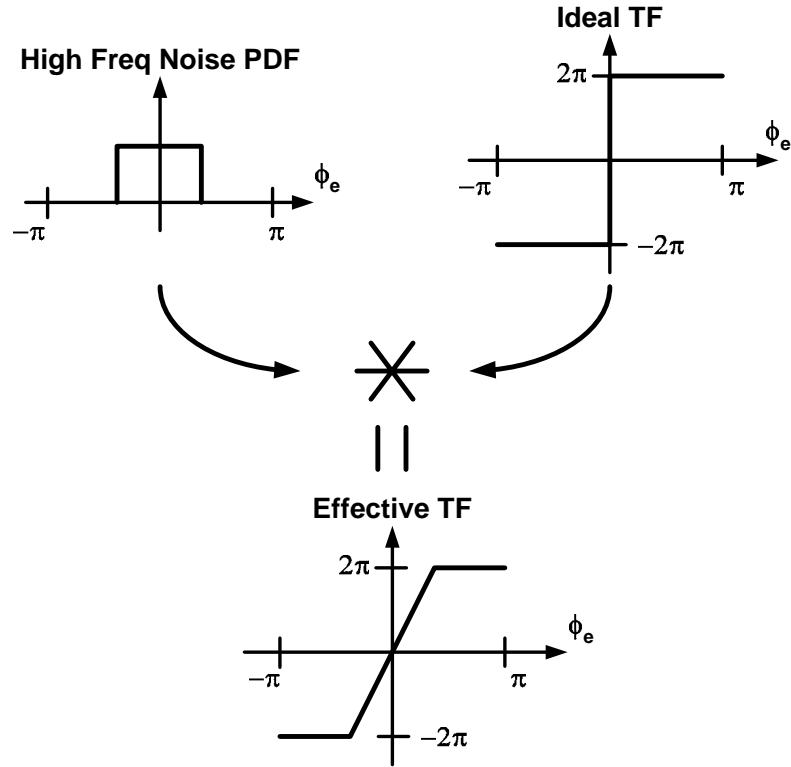
Figure 4.2: Effective transfer function of the bang-bang PD in the presence of high frequency jitter. This example uses a noise PDF that is uniform.

Under these circumstances, equation (4.5) is a reasonable approximation for K_{PD}.

$$K_{PD,eff} = \frac{4 \cdot \pi}{J_{pp}} \tag{4.5}$$

J_{pp} is the peak-to-peak jitter in radians. When the noise has a Gaussian PDF, it has been shown that the effective transfer function has a linear region spanning roughly +/- 2σ [39]. In this case, 4σ can be used as an approximation for J_{pp} resulting in $K_{PD,eff}$ of $\pi \cdot \sigma^{-1}$.

With the approximation that the bang-bang PD is linearized, we can apply s-domain analysis to this digital CDR assuming that its loop bandwidth is much lower than the digital clock frequency ($1/T_S$). Effective proportional and integral gains ($K_{P,eff}$ and $K_{i,eff}$) are defined to convert the gains to the continuous domain. These quantities are in units of LSB·sec^{-1} and LSB·sec^{-2}. The derivation uses impulse invariance by approximating z^{-1}, which is equal to e^{-sT}, as $(1-sT_s)$ [54].

$$K_{P,eff} = \frac{K_P}{T_S} \tag{4.6}$$

$$K_{i,eff} = \frac{K_i}{T_S^2} \tag{4.7}$$

The equations of Section 2.1.3 can now be applied to this phase estimator. The new equations are summarized below. $K_{PD,eff}$, $K_{P,eff}$, and $K_{i,eff}$ are defined in (4.5), (4.6), and (4.7). Note that both the relative stability (ζ) and bandwidth are changed if the digital clock frequency changes without appropriate adjustment of K_P and K_i.

$$\frac{\phi_{out}}{\phi_{in}} = \frac{s \cdot K_{P,eff} \cdot (TD \cdot K_{PD,eff}) \cdot K_D + K_{i,eff} \cdot (TD \cdot K_{PD,eff}) \cdot K_D}{s^2 + s \cdot K_{P,eff} \cdot (TD \cdot K_{PD,eff}) \cdot K_D + K_{i,eff} \cdot (TD \cdot K_{PD,eff}) \cdot K_D} \tag{4.8}$$

$$\omega_n = \sqrt{K_{i,eff} \cdot (TD \cdot K_{PD,eff}) \cdot K_D} \tag{4.9}$$

$$\zeta = \frac{K_{P,eff}}{K_{i,eff}} \cdot \frac{\omega_n}{2} \tag{4.10}$$

$$f_{-3dB} \cong \frac{K_{P,eff} \cdot (TD \cdot K_{PD,eff}) \cdot K_D}{2 \cdot \pi} = \frac{\zeta \cdot \omega_n}{\pi} \tag{4.11}$$

4.3 Performance Comparison using Jitter Tolerance

4.3.1 Test Chip

To understand how this predictive property impacts CDR performance, two serial link transceivers, one with a first order and the other with a second order CDR, were taped out in a TSMC 0.13μm CMOS process. Rather than building the entire chip, an existing Rambus 3.125Gb/s RX was modified to operate instead with our digital loop filters that were synthesized using a standard cell library (Figure 4.3).
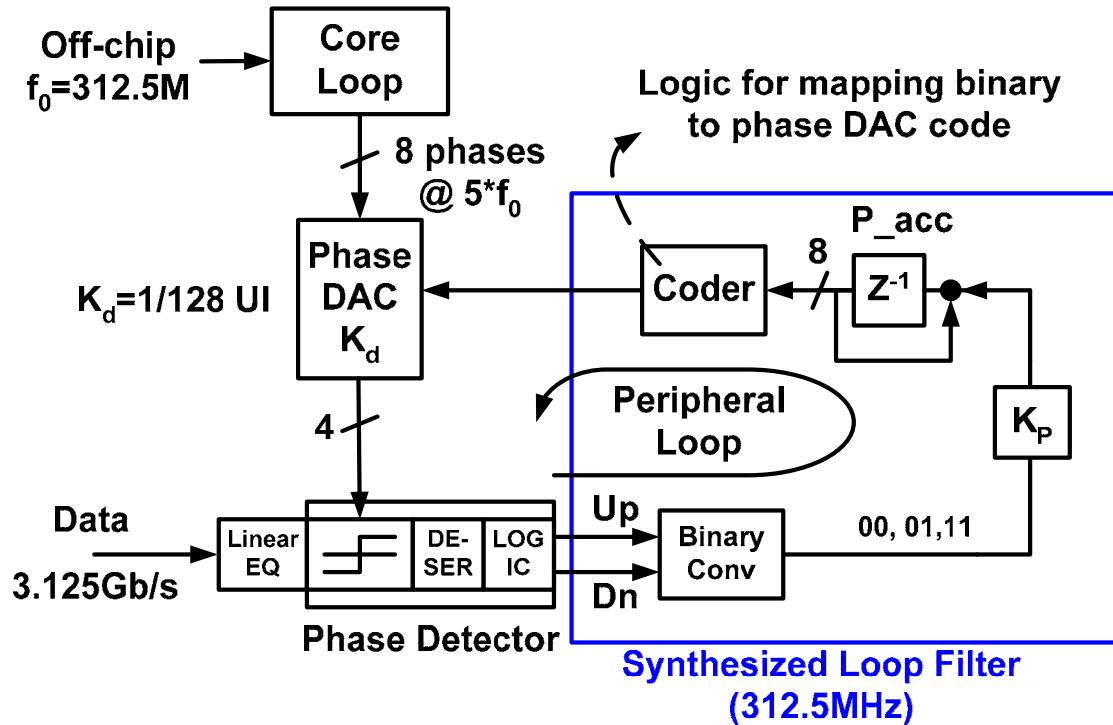
Figure 4.3: RX with a first order CDR that was fabricated.

The reference clock to the chip is nominally at 312.5 MHz. The core loop multiplies this frequency up by five times and at the same time generates 8 phases that equally span a period. The phase DAC performs time interpolation on these 8 phases to achieve an effective resolution of 7 bits per UI. Data is received on both phases of the recovered clock to relax the speed requirements on the circuits. This reduces the highest clock frequency in the chip by a factor of 2. Hence, a pair of differential phase DACs controlled by the digital loop filter generates four clocks (two for Data and two for Edge).

The input data is conditioned by a linear equalizer to reduce the DJ and ISI caused by channel limitations. This is done by introducing a zero that causes peaking in the transfer function of an amplifier. This peaking, when combined with the low pass response of the channel, extends the total 3-dB frequency. The linear equalizer is implemented with two stages of differential amplifiers, of which the first stage is source degenerated by an RC network. At low frequencies, the capacitor is an open circuit and the amplifier's gain is reduced by the resistive source degeneration. At higher frequencies, the capacitor is a short which removes the effect of these resistors thus creating a zero. The equalization gain (i.e. amount of dB of peaking) is controlled by the

source resistance which is implemented using poly resistors in series with MOS switches. A larger resistance translates into a larger equalization gain. The second stage is a simple differential pair amplifier. The linear equalizer is designed to provide 5dB of peaking at 2GHz which is approximately the Nyquist frequency of the input data (Figure 4.4(b)).

(a)   (b)
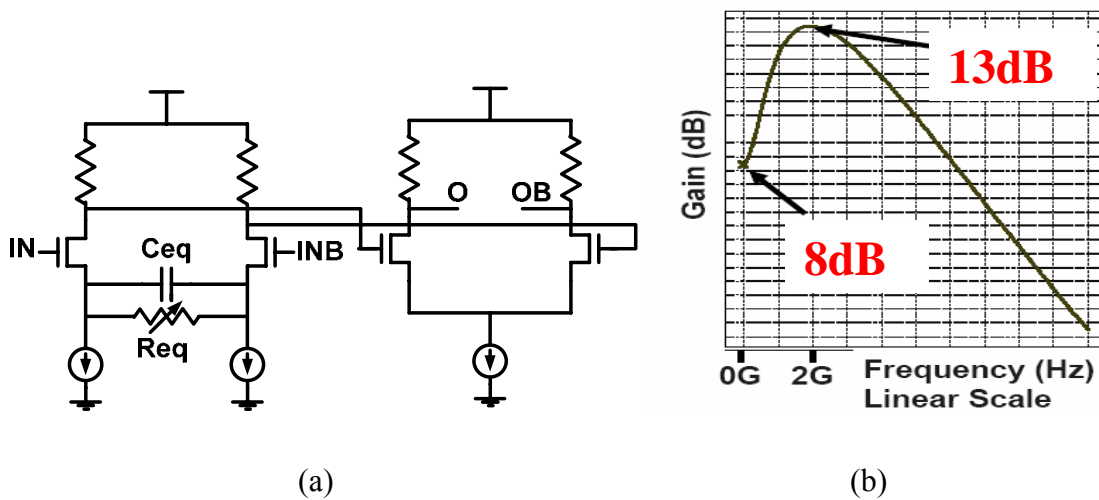
Figure 4.4: Circuit schematic of linear equalizer and its simulated transfer function.

The output of the linear equalizer is sampled by the phase DAC output clocks. These samples are aligned to a single clock domain via the deserializer (DESER) whose frequency is one tenth of the data rate. The deserialized Edge and Data samples are passed to the PD logic (LOGIC) that decides whether to advance (Dn), delay (Up), or hold constant the recovered clock. These three decisions are encoded into two binary bits as 11, 01, and 00. K_P is limited to be a binary number so it can be implemented as a shift and sign extension. This signed binary number is accumulated by the phase accumulator (P_acc). Some logic (Coder) converts the binary accumulator output into the appropriate format required by the phase DACs which is a mixture of thermometer (MSBs) and binary (LSBs). While it is preferable to have full thermometer encoding for the sake of phase DAC linearity, the amount of routing makes it a prohibitive solution. The phase DAC shown in Figure 4.5 is a PMOS-input variation of that published in [32]. A replica bias generator (not shown) adjusts the total current and the output swing via V_{cp}. In turn, V_{cn} sets the output time constant by adjusting the effective load impedance. It performs time interpolation between two adjacent phases from the core loop (denoted θ[n] and

$\theta[n+1]$) by current summing the output of two differential pairs. The total current in the two differential pairs is always constant. When all the current is steered to the left pair, the output phase is a delayed version of $\theta[n]$. When it is steered fully to the right, then the output is fully dependent on $\theta[n+1]$. Naturally, by distributing the current between the two phases, the output phase becomes a weighted sum of the two. Not shown for clarity is the clock MUX that selects the two adjacent phases from the 8 available at the output of the core loop PLL. As with all DACs, the linearity of this circuit is of particular interest. We will investigate this later in this chapter.
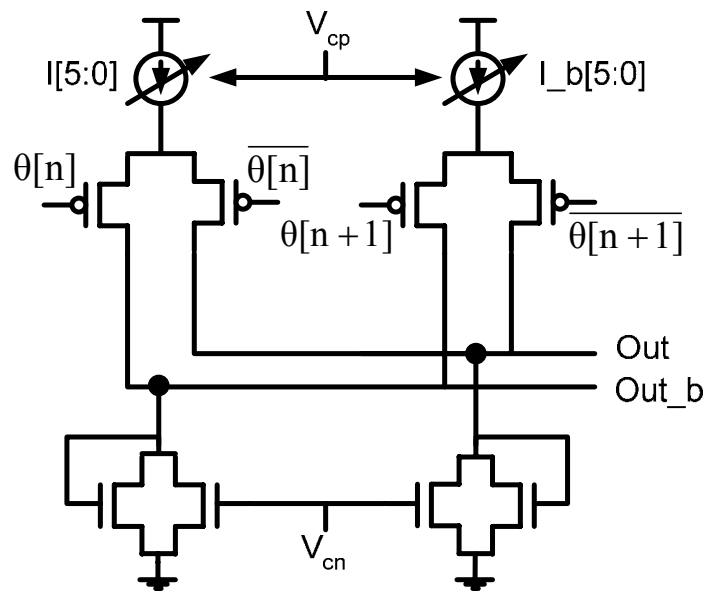
Figure 4.5: Circuit schematic of phase DAC.

The second order CDR that was taped out is otherwise identical to the first order CDR except for the addition of the frequency tracking loop inside the digital loop filter that is synthesized. The gains K_P and K_i are programmable to provide flexibility in setting the damping factor and bandwidth. The ratio of the two gains can be as high as 256.
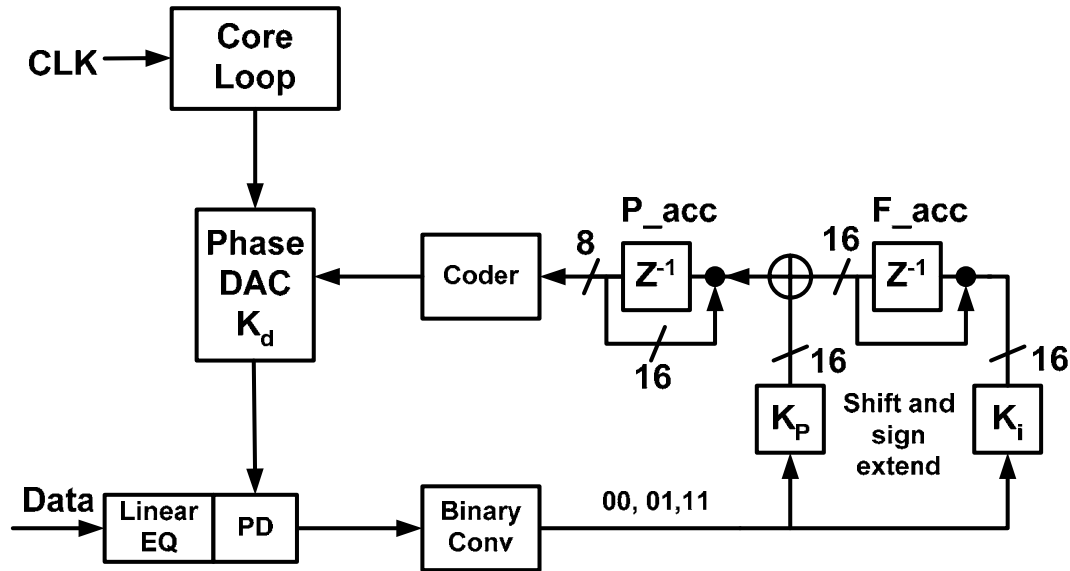
Figure 4.6: RX with a second order CDR that was fabricated.

4.3.2 Jitter Tolerance

Jitter tolerance (JTOL) is an industry standard method of evaluating RX performance. Figure 4.7 illustrates the test setup for measuring JTOL. Random (RJ) and sinusoidal jitter (SJ) is added to the TX data by modulating the delay of the signal generator clock output. The BERT outputs a differential pseudo random bit sequence (PRBS) data which is further corrupted by passing it through a 32 inch backplane. This adds about 0.4 UI_{pp} (i.e. peak-to-peak) of DJ. An on-chip PRBS error counter is polled at regular intervals to estimate the BER through the link. The RJ is calibrated to about 4ps RMS. The SJ frequency and amplitude is varied to obtain JTOL.[10]

---

[10] SJ does not exist in real systems as do RJ and DJ. It is simply a method of evaluating CDR performance. In addition, the RJ and DJ we are adding here are slightly in excess of those set by the XAUI spec [16].
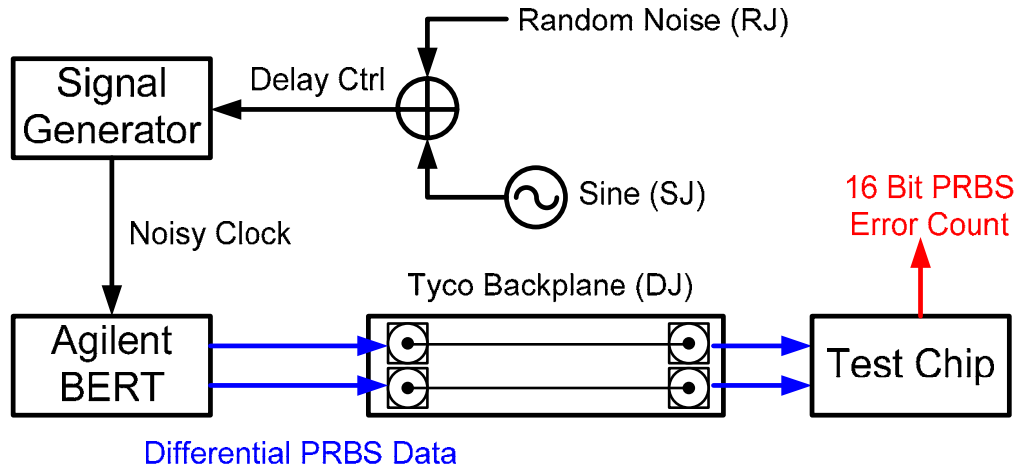
Figure 4.7: Test setup for JTOL measurement. Random and sinusoidal jitter is added by modulating the clock to the BERT. Deterministic jitter is added by passing the differential PRBS data through a 32 inch TYCO backplane.

When SJ is added, the data transitions at the input of the RX are being moved back and forth with respect to its undisturbed position. The amplitude of the SJ is how much the transitions are being moved whereas the frequency of the SJ represents how quickly the transitions are being moved (Figure 4.8). At a given SJ frequency, the SJ amplitude is increased until a certain BER is measured. A BER of 10^{-12} is commonly used. The peak-to-peak deviation of the SJ at which this BER is measured is recorded in the y-axis. This is repeated over a range of different SJ frequencies to form the JTOL curve like the one in Figure 4.9.
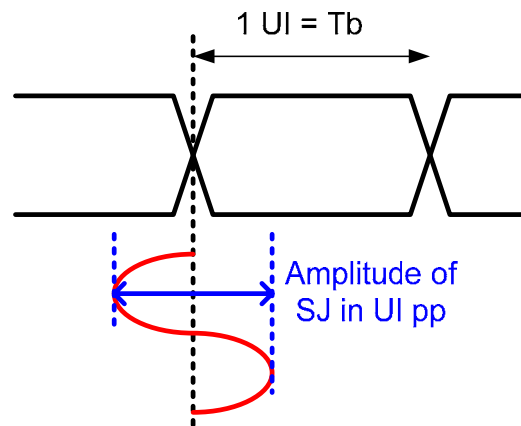
Figure 4.8: Visualizing sinusoidal jitter.
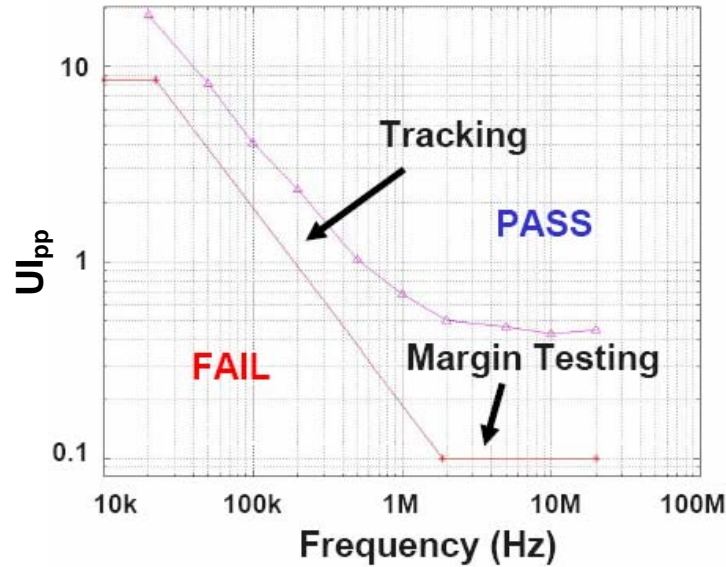
Figure 4.9: Example JTOL plot shown is for the first order CDR. JTOL has two regions that test the timing margin of the link and the CDR's tracking performance. The x-axis is the SJ frequency in Hz. The y-axis is the peak-to-peak SJ amplitude in UI_{pp}.

The JTOL mask (marked with straight lines) is the specification that all parts of the measured curve must be above to pass. The mask consists primarily of a sloped region and a flat one at high frequencies. At very high SJ frequencies, the CDR cannot respond to the jitter since it is outside its bandwidth. Hence, it is measuring how much peak-to-peak timing noise we can add before measuring a certain BER. This is defined as the timing margin of the link which is an indicator of how wide the data eye opening is in the time domain. As the SJ frequency is reduced, the JTOL becomes larger as the CDR is able to track more of the disturbance. At very low frequencies, it tells us how the CDR behaves in the presence of low frequency phase and frequency offsets. In effect, JTOL gives us insight into the frequency domain behavior of the CDR.[11]

4.3.3 Measured Results

We first measured the JTOL of the first and second order CDRs with no frequency offset between the BERT and the RX (Figure 4.10). We find that the addition of the frequency tracking loop helps the low frequency JTOL. This is expected since very low frequency

---

[11] The fact that the JTOL specification does not endorse minimizing the bandwidth may be confusing to the reader. Minimizing CDR bandwidth is the optimum for uncorrelated noise sources. In practice, some bandwidth is still required to accommodate slow variations in the frequency of the TX PLL.

phase variations will appear as instantaneous frequency offsets. However, we find that as K_i is increased beyond a certain level the JTOL degrades near its bandwidth.
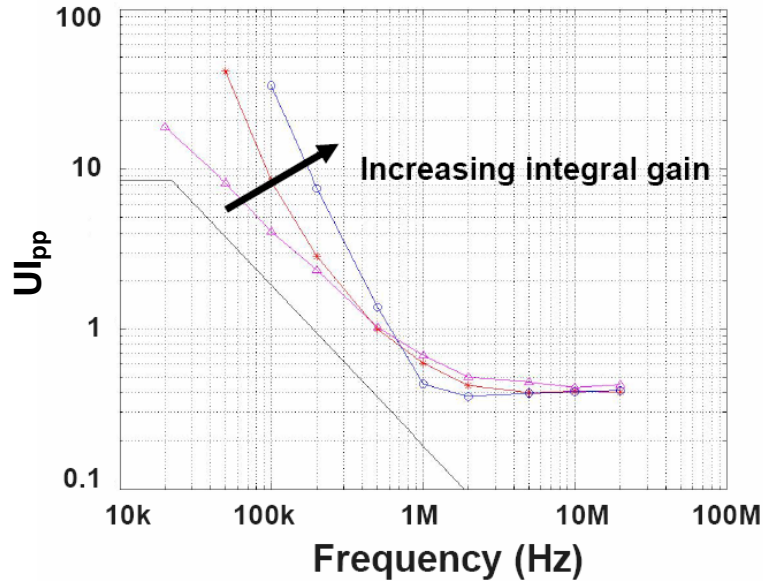
Figure 4.10: JTOL with varying integral gain. The three curves are for $K_i = (0, 1/512, 1/128)$. $K_P = 0.5$ for these measurements. $K_i = 0$ is the first order CDR.

The impact of K_i on JTOL can be explained using the linear approximation of the second order CDR. The relationship between the phase error observed at the PD (Φ_e) and $\Phi_{n,in}$ (noise at the input) can be found as (4.12) which is a high pass response with a DC zero and two poles at the same location as the transfer function (Φ_{out}/Φ_{in}). The RX will start detecting bit errors when Φ_e starts to approach half the timing margin (*TM*). Noting that JTOL is defined as the peak-to-peak $\Phi_{n,in}$ at which this occurs, JTOL can be found to satisfy (4.13). The effective Φ_{out}/Φ_{in} is defined in (4.8). The JTOL profile is the inversion of a high pass response as expected. As K_i is increased, ζ is reduced by the square-root of the factor translating into a less stable loop. This manifests itself in the frequency domain as peaking in (4.12) and convergence of the two poles closer in frequency. Hence, when the loop is heavily damped (low K_i) there is little or no "dipping" (the inverse of peaking) in the JTOL and three regions with slope -40dB/dec, -20dB/dec and 0dB/dec will exist [39]. Conversely, when the loop is less stable (high K_i), we will observe dipping along with only two regions with slope -40dB/dec and 0dB/dec. This agrees with the measurements of Figure 4.10.

$$\phi_e = (1 - \frac{\phi_{out}}{\phi_{in}}) \cdot \phi_{n,in} = \frac{TM}{2} \tag{4.12}$$

$$JTOL(s) = 2 \cdot \phi_{n,in} = \frac{TM}{(1 - \frac{\phi_{out}}{\phi_{in}})} \tag{4.13}$$

What is of particular interest is the JTOL when a frequency offset exists (Figure 4.11). For the first order CDR, JTOL degrades at all SJ frequencies as the frequency offset is increased. However, the JTOL of the second order CDR is not affected significantly even when the offset is increased to 5000ppm.
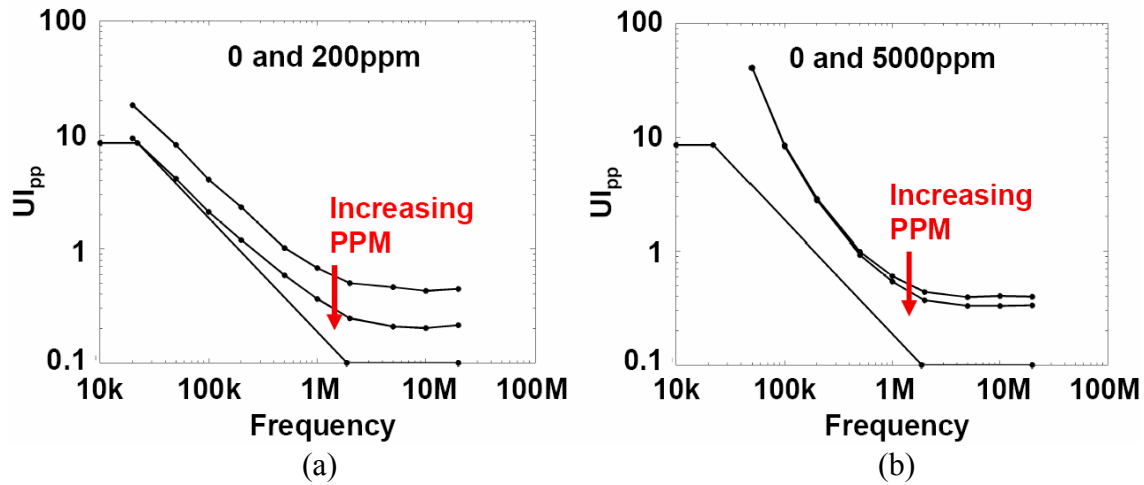
Figure 4.11: Measured JTOL with different frequency offsets for (a) first order CDR and (b) second order CDR. $K_P = 0.5$ for both CDRs. $K_i = 1/256$ for the second order CDR [41].

The superiority of the second order CDR in the presence of a frequency offset is a direct result of its predictive ability. The first order CDR can only take corrective action after the error has occurred. For this reason, a time lag exists between the TX data transitions and the recovered clock in steady state which in turn offsets the data sampling point from its optimum. The second order CDR learns the phase offset ramp rate (frequency offset) from past bits and takes predictive correction on the deterministic phase offset trajectory. This allows the second order CDR to drive the average steady

state phase estimation error to zero. This is expressed graphically in Figure 4.12. One additional observation is that this steady state phase estimation error in the first order CDR increases with the reduction in its bandwidth. This insight comes from applying the final value theorem (4.14) to the equivalent linear transfer function. This is in line with intuition since in the absence of predictive correction, a lower bandwidth (equivalent to increased filtering) means that the error has to be observed for a longer duration of time before action is taken to correct it. This is particularly a problem since a low bandwidth is desired to filter the jitter at the CDR input.

$$H(s) = \frac{\phi_{out}}{\phi_{in}} = \frac{K \cdot K_P}{s + K \cdot K_P} \qquad\qquad H(s) = \frac{\phi_{out}}{\phi_{in}} = \frac{s(K \cdot K_P) + K \cdot K_i}{s^2 + s(K \cdot K_P) + K \cdot K_i}$$
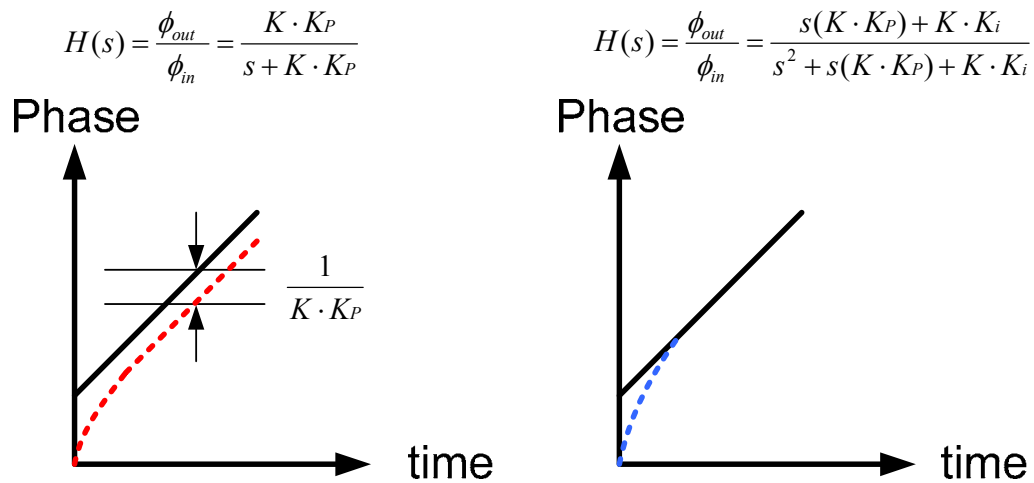
Figure 4.12: Steady state phase estimation error in the first order and second order CDRs when a frequency offset exists. Here, $K = K_D \cdot K_{PD} \cdot TD$ in the expressions for the equivalent linear transfer functions. The first order CDR has a steady state error that is inversely proportional to its bandwidth. The second order CDR has no steady state error.

$$f(\infty) = \lim_{s \to 0} s \cdot E(s) = \lim_{s \to 0} s \cdot R(s) \cdot (1 - H(s)) \qquad\qquad (4.14)$$

$R(s)$ is the phase movement at the input of the CDR. It is s^{-1} for a unit phase step and s^{-2} for a unit phase ramp.

The opposing constraints put on the bandwidth of the first order CDR from tracking the deterministic phase ramp and jitter filtering are decoupled in the second order CDR due to its ability to estimate the deterministic phase offset trajectory. In Figure 4.13, a loop gain (and hence bandwidth) reduction of a factor of four improves the *TM* by 0.1

UI_{pp}. The larger bandwidth setting has a lower *TM* but it still passes the XAUI specification at 200ppm. On the other hand, the lower bandwidth setting despite the *higher TM* does not even pass the specification at high SJ frequencies at 200ppm thus demonstrating this tradeoff. It should be pointed out that at lower bandwidths, slew rate limiting in bang-bang CDRs becomes a bigger problem for a given frequency offset. The frequency range of operation (in units of ppm) due to slew rate limiting for a first order CDR is proportional to the loop gain according to (4.15). Near this limit, the first order CDR exhibits performance degradation much larger than that predicted by (4.14). Since the JTOL of the second order CDR does not degrade with the frequency offset, its *TM* can be increased by reducing the CDR bandwidth without this concern.

$$f_{slew} = \frac{K_P \cdot K_D}{T_S} \cdot 1\text{UI} \cdot 10^6 \tag{4.15}$$
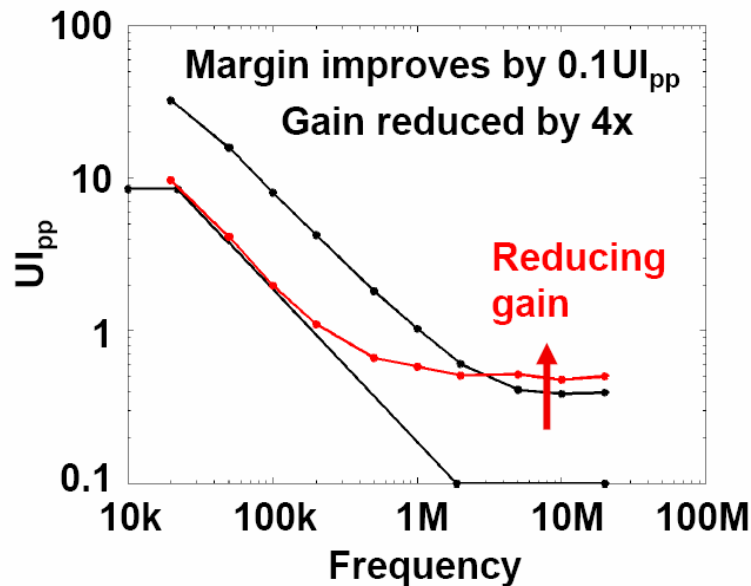
Figure 4.13: JTOL with varying K_P (1 and 0.25) in the first order CDR. The timing margin improves with the reduction in loop gain / bandwidth due to the increased jitter filtering.

Before concluding this chapter, I would like to delve into the cause of the slight *TM* degradation in the second order CDR in the presence of large frequency offsets as observed in Figure 4.11(b). The first cause is a result of the CDR taking large phase steps

every clock cycle to compensate for the large frequency offset. Since the CDR loop drives the error to zero every slow logic clock cycle of 10UI, there is residual phase estimation error for bits in between as shown in Figure 4.14. The slanted line is the phase drift due to the frequency offset whereas the staircase represents the phase estimate from the discrete time CDR. At 5000ppm, this degradation in margin is about 5% of a UI. This error can be reduced by increasing the update rate of the CDR.
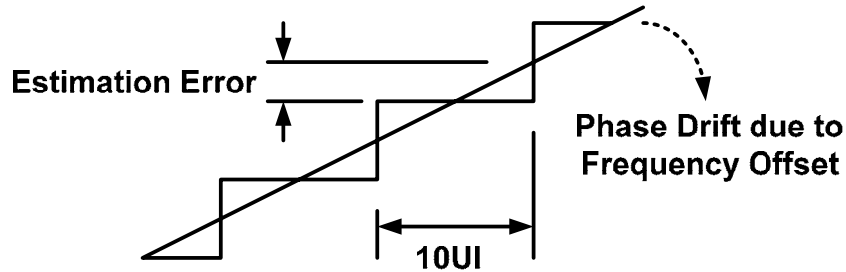
Figure 4.14: Estimation error caused by the discrete time nature of the CDR.

The second source of phase estimation error is from the nonlinearity of the phase DAC transfer function. This is due to the capacitive coupling from the input clocks ($\theta[n]$ and $\theta[n+1]$) to the output clock which effectively reduces the range of the phase DAC within each quadrant. The nonlinearity is therefore largest at the quadrant boundaries where the phase DAC is switched fully to one of the input clocks. We can figure out the *TM* penalty due to the nonlinearity of the phase DAC by looking at the integral nonlinearity (INL). The INL can be viewed as a noise source whose frequency depends on how quickly the CDR traverses through the codes. At high frequency offsets, INL becomes a high frequency phase noise source that is outside the CDR bandwidth. In this case, the penalty to the *TM* is approximately the peak-to-peak amount of 3% UI (Figure 4.15). These two sources of phase estimation error can account for the 8% UI_{pp} margin degradation observed at 5000ppm. Interestingly, this means that the effect of phase DAC nonlinearity is worsened at large frequency offsets.
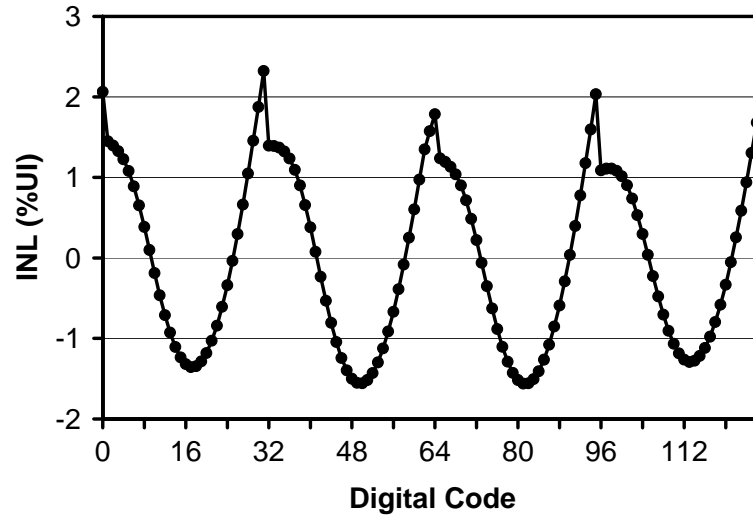
Figure 4.15: Simulated INL of the phase DAC.

4.4 Summary

Matching the order of the CDR to the operating environment can improve CDR performance by removing the opposing bandwidth constraints on the CDR from phase trajectory tracking performance and jitter filtering. For systems with frequency offsets, a second order CDR improves JTOL due to its ability to predict the phase movement of the data which also drives the steady state error to zero.

A linear approximation of the second order CDR was found useful in explaining the JTOL results. We find that the second order dual loop CDR has similar dynamics to the classic analog PLL CDR. However, the absence of a VCO in the CDR loop allows the timing margin to improve as the bandwidth is reduced.

Finally, the impact of the phase DAC nonlinearity on CDR performance was analyzed. The penalty on the timing margin depends, surprisingly, on the frequency offset in the system.

Chapter 5

Burst Mode Communications

In the previous chapter, we showed that a second order phase estimator can predict the TX data phase in plesiochronous systems. We expect that by building a more accurate estimator, a CDR for plesiochronous operation complicated by a very low transition density can be built. Low transition density data (TD≈0.01) occurs in packet based networks due to the bursty nature of network traffic. It is possible that packets from a given TX to a given RX will be separated by almost a million UI. For this reason, these systems are referred to as burst mode communication systems. This chapter will show that by building a very accurate second order phase estimator, a CDR with zero effective lock time can be made without sacrificing the need to sufficiently filter the jitter on the data. Section 5.1 provides an overview of the application and prior art. Section 5.2 will then detail the second order semi-digital dual loop CDR that was implemented along with architectural modifications to enable its operation in this low transition density environment. Finally, the measured results will be presented in Section 5.3

5.1 Background

Internet traffic occurs in occasional bursts such as when a user downloads a new webpage. This traffic pattern results in less efficient channel usage closer to the end users

since the available bandwidth goes unused for long periods of time.[12] This inefficiency is especially true in optical fiber that has a total bandwidth greater than 1 Tbps (i.e. 1000 Gbps) which is more than a single user can possibly use continuously. To reduce this waste, industry and academics have investigated ways to share the high optical bandwidth among multiple clients. One method is wavelength division multiplexing (WDM). Researchers have demonstrated improved network efficiency using WDM in optically switched packet networks [45, 46]. However, the still large bandwidth in each wavelength (>10Gbps) necessitates methods of further sharing the bandwidth among many users.[13] For this, time domain multiplexing (TDM) is most popular.

5.1.1 TDM Optical Networks

One example application using TDM is fiber-to-the-home (FTTH). The general industry consensus is to use a passive optical power combiner/splitter to broadcast a common data stream to all the clients downstream while the TDM packets from the clients are combined onto a single fiber to the central office (CO) upstream [43, 44]. The upstream is TDM and each TX located on the customer premises is given a time slot to send data to the single RX at the CO (Figure 5.1). This approach still has low utilization on the short lengths of fiber from the power combiner to the individual users. However, the bandwidth of the long fiber from the power combiner to the CO is used much more efficiently.

Whereas in typical high speed links the communication between a TX and a RX is continuous, the data stream from a *given* TX to the RX still occurs in short bursts in such TDM scenarios. Furthermore, in the more general case, each node in the network has its own independent clock source that has a different phase and frequency offset with respect to the RX. This means that the CDR in the receiver must resynchronize at the start of every packet. For this reason, lock time becomes a critical parameter since it reduces the effective bandwidth of the network.

---

[12] Further away from the end user, the aggregated traffic of millions of users is more continuous. This is the case for the Internet backbones.

[13] The achievable optical bandwidth depends on the distance of transmission, the number of wavelengths used, and the type of optical fiber. 10Gbps assumes transmission over 10km, on a single wavelength, through a single mode fiber.
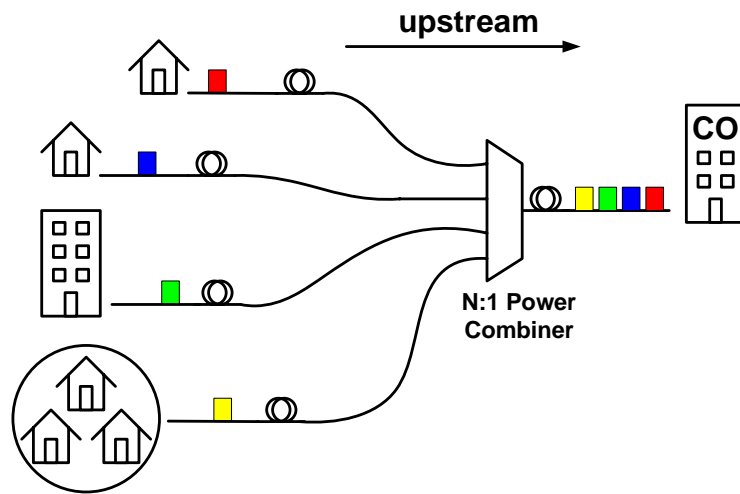
Figure 5.1: An example TDM optical packet network is found in the upstream of FTTH.

In one specification, the CDR is expected to recover data with a BER less than 10^{-10} after 44 UI [44]. The RX must perform signal threshold level recovery (called level recovery) as well as clock recovery within 44 bits. Level recovery is needed since the packets from the different TXs can experience substantially different attenuation. Since optical signals are by nature single-ended, the RX must figure out the optimal decision threshold for each packet. This means that the CDR must relock to packets with 100ppm frequency offsets in about 20 UI. This is very challenging given that analog PLL lock times can run well above 1000 bits.

5.1.2 Prior Art

Previous approaches have focused on building CDRs with very small lock time. Two well known ones are the oversampling receiver [47, 48] and the gated oscillator [49]. In the oversampling receiver, the input data is oversampled by a factor greater than 3. The samples are stored and the transitions are detected by comparing adjacent samples. When adjacent samples are different, it implies that a transition has occurred between them. This transition information is passed through a post processing filter (i.e. phase picking logic) that chooses as the recovered data the samples with the best timing margin (Figure 5.2). In [48], a digital PLL was implemented in the phase picking logic as a way of filtering noise so as to make the best decisions.
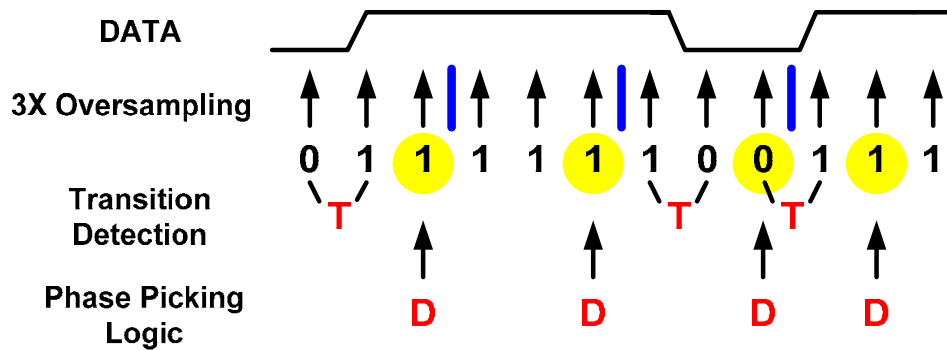
Figure 5.2: Conceptual view of the oversampling receiver. This example is for 3x oversampling.

The primary drawback of the oversampling receiver is its inherent power performance tradeoff. The timing margin (*TM*) of the link increases with the phase resolution that is proportional to the oversampling ratio. Unfortunately, increasing the oversampling increases proportionally the power consumption of the CDR since more clock phases need to be generated and distributed. The size of the register file that holds the samples while the transition detection and phase picking logic completes also increases proportionally. As a reference, a 4Gbps 3x oversampling receiver consumed 0.9W when realized in 3.3V 0.6um CMOS [51].

Interestingly, when comparing to A/D converter (ADC) design the oversampling receiver is similar to the flash architecture. Both have a multiplicity of comparators (either in the phase or voltage domain) to obtain the relevant information in one clock cycle. On the other hand, the classic closed loop CDR, such as the semi-digital dual loop CDR, is very much like a tracking ADC which is low power but is also very slow. Just as the successive approximation register ADC is a good compromise between these two extremes, a binary search algorithm has been implemented in [50, 52] to improve lock time without increasing the power consumption. However, the clock to the RX makes large phase jumps at the beginning of the algorithm making reliable timing an issue. This limits the cycle time of the digital logic and will prevent these designs from achieving a lock time below 20 bits in multi-Gbps operation.

An alternative design uses gated oscillators (Figure 5.3). A reference PLL locks gated oscillator C to an external reference frequency. The control voltage of the PLL is

used to coarsely set the replica gated VCOs (A and B) to the right frequency. When data is zero, oscillator A is enabled and it samples the data. When the data is a one, oscillator B samples the data. While having the benefits of low complexity and virtually zero lock time, this CDR provides no jitter filtering which will result in degraded *TM*. Finally, there is no active feedback mechanism to ensure that the delay between the data and the recovered clock due to the buffer delay through the gated VCO and the NOR gate is half a bit time. For these reasons, this design is not robust across PVT variations or to noise sources in the system.
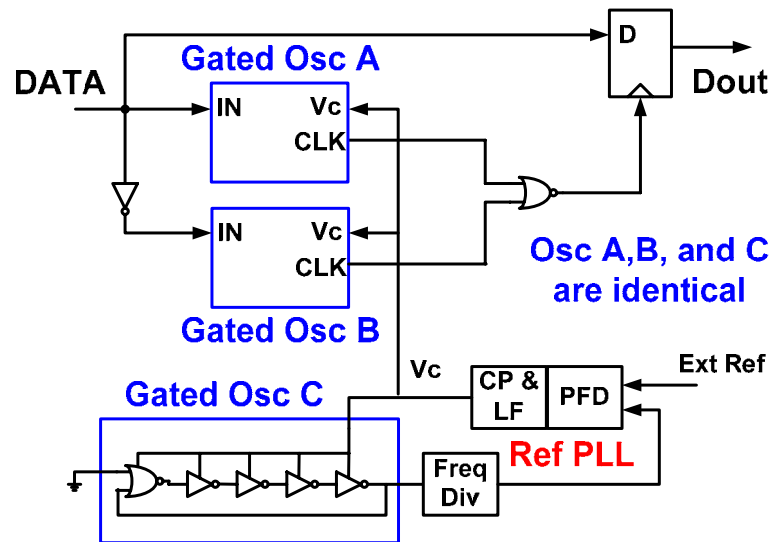
Figure 5.3: Burst mode packet receiver using gated oscillators.

5.2 Architecture

The common theme in previous work is that they were fast phase acquisition systems that assumed no prior knowledge about the TX. Instead, the approach taken in this thesis is to build very precise phase and frequency offset estimators that use past information to predict the phase of bits in future packets so that both zero lock time and good jitter filtering can be achieved without increasing the power significantly compared to conventional CDRs. We have seen in the previous chapter that the second order dual loop CDR can predict the phase of future bits in the presence of a frequency offset. This section will detail modifications that are required to enable its operation as a burst mode receiver.

In formulating the architecture of the burst mode receiver, I will assume that the TDM network has up to 32 TXs each sending 10k bit packets.[14] Then the time between packets arriving from a given TX to the RX will be 320k bits. To achieve this level of estimation accuracy, five dominant factors must be considered. They are quantization noise due to the finite precision of components, jitter in both the RX and TX, limit cycles caused by loop delay, phase DAC nonlinearity, and initial convergence of the CDR in the presence of extremely sparse data. This section will expand on these issues which I first presented in [53].

A block diagram of the second order dual loop CDR that forms the basis of this burst mode receiver is shown in Figure 5.4. The specifics of the circuits including the PLL, phase DAC, and injection locked VCO were discussed in Section 3.2 The data rate of operation is 3.125Gbps. 5 bits are transmitted every reference clock cycle (625MHz) via a multiplexing TX. The RX also performs demultiplexing at the pad. For a multiplexing factor of 5, 10 phases equally spanning a 625MHz period are needed to sample the data stream at the Edge and Data times.
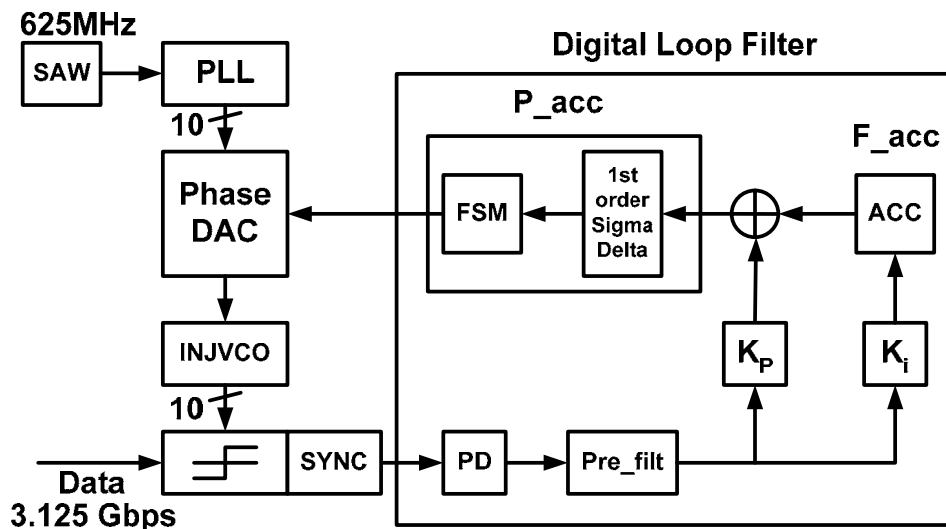
Figure 5.4: The second order dual loop CDR core of the burst mode packet receiver.

---

[14] An additional assumption is that the receiver knows which TX is sending the packet *a priori*. This information is usually available in these systems through a higher network layer that negotiates the TDM. This assumption along with those concerning the number of clients and the typical packets sizes were based on discussions with researchers in optical networks.

5.2.1 Quantization Noise

The need to predict the phase of bits that are a million UI away necessitates a digital approach. While the analog PLL based CDR has similar predictive qualities as the second order dual loop CDR, the control voltage (frequency estimate) is especially susceptible to leakage in the capacitor as well as supply and substrate noise in the absence of data transitions. Using a CDR with digital components can avoid these problems but they in turn suffer from quantization noise due to their finite precision.

The phase DAC resolution determines the minimum phase quantization noise of the system. Our prototype's resolution was a 64th of a UI (approximately 0.0156 UI). This phase resolution also sets the maximum frequency drift that can be tracked, since the integral control at best can increment the phase one step each clock cycle (one tenth the bit rate or 312.5MHz) in our prototype.[15] Thus we can rotate through one bit every 640 UI, which is equal to a frequency offset of 1/640 or 1562.5 ppm. It is straight forward then to budget for the resolution of our frequency accumulator (F_acc). For example, if we want a maximum phase estimate error of 0.1 UI after one million bits, 15 bits (including one bit for the sign) is required to achieve a frequency resolution better than 0.1ppm. The digital frequency information inside F_acc represents the fractional phase DAC step that needs be compensated each clock cycle to stay phase locked with the TX.

We use a first order digital $\Sigma\Delta$ modulator to convert the higher resolution frequency information to the lower resolution input of the FSM. The modulator achieves this by encoding the (N+1) bit information into the duty cycle of its 1 bit output.[16] The FSM serves both as the accumulator for the six MSBs of P_acc as well as the coder to generate a non-binary output code. The details of the FSM and its output format were discussed in Section 3.2.4

---

[15] It is possible to track beyond this limit by allowing the integral control to update the phase by more than one step each clock cycle. This was done in Chapter 4 and Chapter 6 to track 5000ppm. As the expected offset is only 100ppm in this project, the integral control was capped.

[16] In hindsight, due to the absence of any phase domain filter after the phase DAC, the CDR would have performed equivalently with an accumulator with a signed carry-out to perform error accumulation.

5.2.2 Jitter in TX and RX

Jitter at both the TX and RX adds uncertainty to the phase and frequency estimates of the CDR. Since these errors should be random, their effect on phase estimation error can be minimized by decreasing K_P. Put in different terms, we can improve CDR performance by reducing the bandwidth so as to increase jitter filtering. Furthermore, increasing the resolution of the frequency accumulator (i.e. reduce K_i) ensures that the uncertainty only corrupts bits that are well below the 15th. For this reason, the system was designed with at least 21bits of frequency precision with programmability up to 24 (including one bit for sign). The smallest achievable frequency step is better than 0.2ppb.

Interestingly in bang-bang CDRs, the frequency resolution is tightly coupled to the damping factor through the magnitude of the integral gain. Since in this application a high frequency resolution is required, it also means that a heavily damped CDR is necessary. For the case of our prototype (assuming that $K_{PD,eff}$=10, K_P=1, and K_i=2^{-20}), the effective damping factor using Equation (4.10) is about 140 in the presence of a continuous PRBS pattern. For TD≈0.01, the effective damping factor is still about 20. The dynamics of this CDR is clearly close to a single pole system.

5.2.3 Limit Cycles

Limit cycles occur in feedback systems with comparators such as the bang-bang PD. The smallest limit cycle occurs when the CDR dithers between the two adjacent phases that straddle the actual phase position of the data. Any loop delay increases the limit cycle beyond this minimum as expressed in (5.1). This equation is a modified version of that originally derived for analog bang-bang PLLs [36] to account for the discrete time nature of the CDR. D represents the loop delay in units of T_S (the period of the digital clock). The peak-to-peak dither is in units of UI.

$$Dither_{pp} = K_P \cdot K_D \cdot D \cdot \frac{2 \cdot (K_P/K_i \cdot T_S) - 1}{(K_P/K_i \cdot T_S) - 1} \tag{5.1}$$

For heavily damped systems, the worst case peak-to-peak dither jitter becomes:[17]

$$Dither_{pp} \approx K_P \cdot K_D \cdot D \cdot 2 \tag{5.2}$$

Clearly, one way to reduce the limit cycle is to reduce the loop delay. Thus, the gains in the loop have been limited to exponents of 2 such that multiplication is simply a shift operation that incurs marginal delay. Still, the loop delay is equal to 8 cycles (shown with bars in Figure 5.5). This large delay is due to the limitations of the process coupled with the high precision required in our estimates. The final two delays are due to an offset correction memory and its decoder that are discussed in 3.2.4 Loop delay will increase *both* the phase and frequency wander of the CDR.

Another method of reducing the dither is to use an attenuator. A programmable attenuation is provided by the Pre_filt block. This block prevents additional phase and frequency updates until the last update has taken effect. It is an accumulator that generates a positive or negative carry when the accumulated error reaches a programmable threshold (+/- 2, 4, 8, or 16). In implementation, instead of adjusting the threshold, the input is shifted before accumulation. It acts as a linear attenuator (1/2, 1/4, 1/8, or 1/16). So long as the amount of delay in the loop is not greater than this attenuation, the limit cycle does not trigger a carry at the output of this attenuator. Hence, the limit cycle is suppressed.

---

[17] (5.1) and (5.2) assume that the phase DAC resolution is equal to that of the phase accumulator. When it is not, the minimum achievable peak to peak jitter is 2 times K_D.
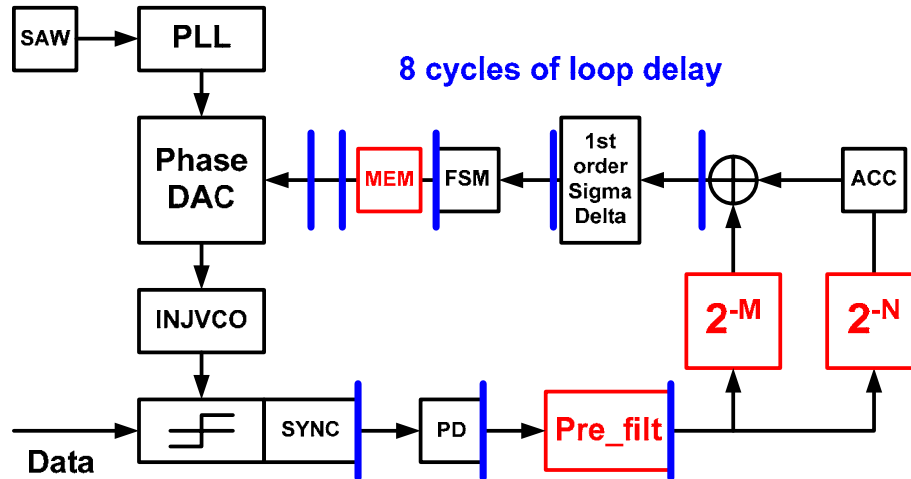
Figure 5.5: The 8 cycles of delay in the CDR feedback loop. Each delay is marked with a blue bar. Two delays are incurred by the offset correction memory and its decoder.

Figure 5.6 shows a Matlab simulation of this second order CDR employing this filter. The peak-to-peak dither jitter is approximately equal to the minimum achievable of 2 phase DAC steps (=10ps) as expected.
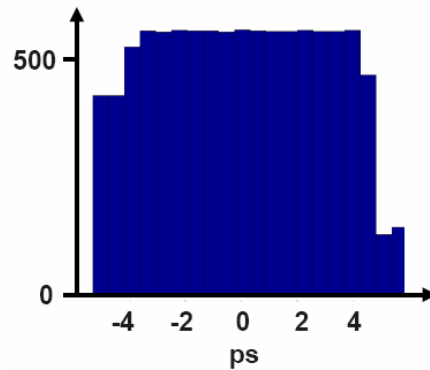
Figure 5.6: Jitter histogram from simulation showing the limit cycle of the second order CDR with K_P=1, K_i=2^{-20}, D=8, K_D=1/64, and T_S=3.2ns. Pre_filt gain is set to 1/8. Frequency offset of 100ppm exists between the TX and RX.

5.2.4 Phase DAC Nonlinearity

When this CDR is receiving continuous data and the frequency offset is smaller than 1562.5 ppm, differential nonlinearity (DNL) of the phase DAC will add to the peak-to-peak phase estimation error distribution by an amount roughly equal to 2 times the

maximum step size.[18] Thus the worst-case step is generally chosen to be small compared to the expected jitter. In this scenario, integral nonlinearity (INL) is not an issue – the control loop never depends on the accuracy of the phase DAC since it is constantly measuring the actual phase error.

The situation changes with packet reception. Between packets, there is no input from the TX and the loop estimates the phase with the assumption that the phase DAC is linear. For this reason, the error caused by the INL adds onto the jitter caused by the DNL. If the intervals between packets and the frequency offset are large enough for the receiver to traverse the full range of the phase DAC, the peak-to-peak jitter distribution will increase by the amount of the maximum INL. The worst case occurs when the CDR receives the last bit of a packet with the worst phase location caused by the DNL and then makes an additional estimation error of INL when the next data arrives. In terms of frequency accuracy, INL and DNL does not matter much since they appear as noise and corrupt the lower bits which we have already been allocated to guard against noise.

Given the importance of linearity, the prototype contains several circuits to correct nonlinearity as described in Section 3.2

5.2.5 Acquisition Aid

So far in this section, we have addressed how to build an accurate phase and frequency estimator so that we can predict the phase in the distant future (i.e. at the start of the next packet). Put in a different way, we have discussed how to build a CDR that can retain lock in the absence of data for long periods of time. Figure 5.7 shows how this CDR can be leveraged to build a burst mode receiver. The digital phase/frequency estimator gathers snippets of timing information from the multiple TXs simultaneously from the burst mode packets. The estimator will converge to the correct phase and frequency values for all TXs after some calibration period. After each packet, the last phase and frequency estimate of that TX is stored away into memory. When that TX is sending data again, the last frequency estimate for that TX is loaded back from memory, and the phase

---

[18] This observation does not contradict the previous assertion that the penalty on the timing margin from the phase DAC nonlinearity can be found from the INL. When the frequency offset is small, the phase variation caused by the nonlinearity is within the bandwidth of the CDR. In this case, only the DNL contribution to the total INL is observed.

estimate is computed by incrementing the last known phase by the product of the frequency estimate in memory and the elapsed time. After the calibration period, the CDR reaches steady state and future packets from any TX can be recovered with zero lock time.

We have assumed thus far that the system will converge to the correct phase and frequency offset estimates for each of the TXs. However, false locking to frequency harmonics can occur due to the sparse timing information. We can ensure that the system will converge if the frequency estimate after the first packet is good enough to limit the maximum phase error between packets to 0.5UI. To do this, the first packet after system reset is used to obtain a fast but coarse frequency estimate and the CDR is modified appropriately as shown in Figure 5.9.
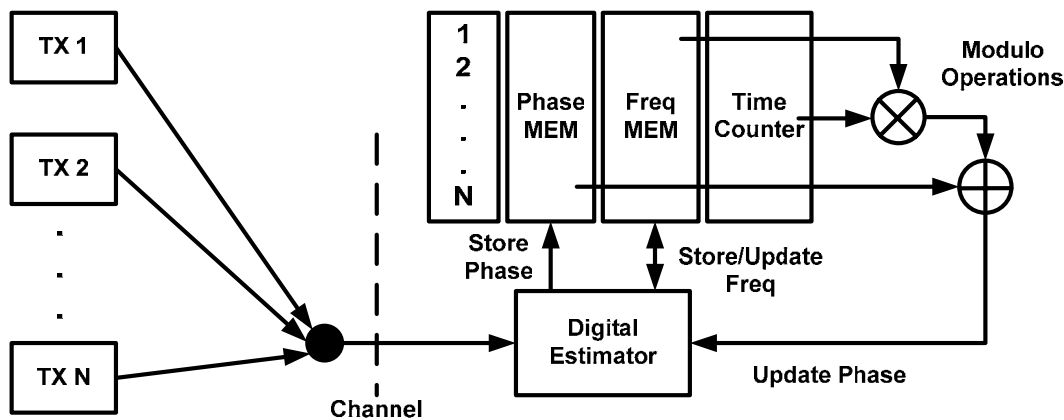
Figure 5.7: Complete burst mode receiver leveraging a high accuracy digital phase and frequency estimator.

Since the frequency offset is equal to the phase drift between the TX and RX divided by the elapsed time, we can simply count the phase updates that the RX makes to keep in step with the TX and divide it by the observation time to estimate the frequency offset. The accuracy of this estimate increases with the observation time. This method requires the RX to phase lock to the data first since phase update information during acquisition is uncorrelated to the frequency offset.

Thus, the first packet is divided into 2 parts (Figure 5.8). The first part of the packet is long enough to guarantee phase lock. The second part is long enough to give us an accurate enough estimate of frequency. In our prototype, we simply divide the packet into

halves. The packet length is chosen as an exponent of 2 (2^P) multiple of the low frequency digital clock cycle (10 UI). In this system, this means that the packet length is 10×2^P UI.
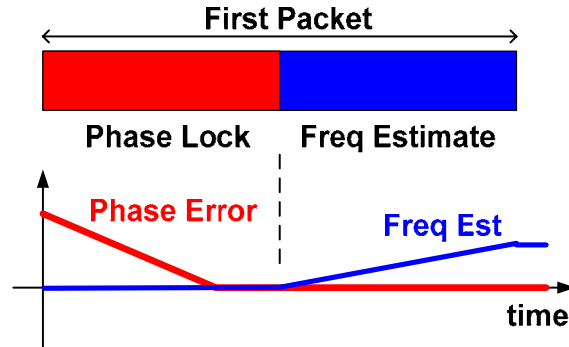
Figure 5.8: Sub-division of first packet to enhance the lock range. The first half is allocated for the RX to obtain phase lock while the second half is used to obtain a fast but coarse frequency offset estimate.

During the first half of the first packet, the integral branch is turned off (via MUX2 and MUX4) and the CDR acquires phase lock as a first order CDR. Since the objective is to obtain phase lock as quickly as possible, the programmable attenuator (Pre_filt) is set to its maximum. During the second half of the first packet, the loop continues to track as a first order loop. However, the error information is scaled and accumulated into the frequency accumulator. The input to the integral branch is enabled (via MUX2) while the integral control into the adder is zeroed (via MUX4). The input to the frequency accumulator is scaled by 2^{-Q} (via MUX3).

If the CDR can make one phase DAC step for every Pre_filt output, then Q should be (P-1). The −1 results from using half of the packet in our frequency estimation. But when the proportional gain is 2^{-M}, then 2^M updates from Pre_filt are needed before an actual phase DAC step occurs. Since we want to calculate the number of phase DAC updates divided by the observation time, the input to the frequency accumulator must be further scaled down by 2^M. Thus Q is set to (M+P-1). This coarse frequency estimate also suffers from quantization effects. $\pm 2^{-Q} \times 1562.5$ppm is the maximum frequency estimate error of this method. So long as the product of the residual frequency error and the interval between packets (320k bits) is less than 0.5, then the system will converge.
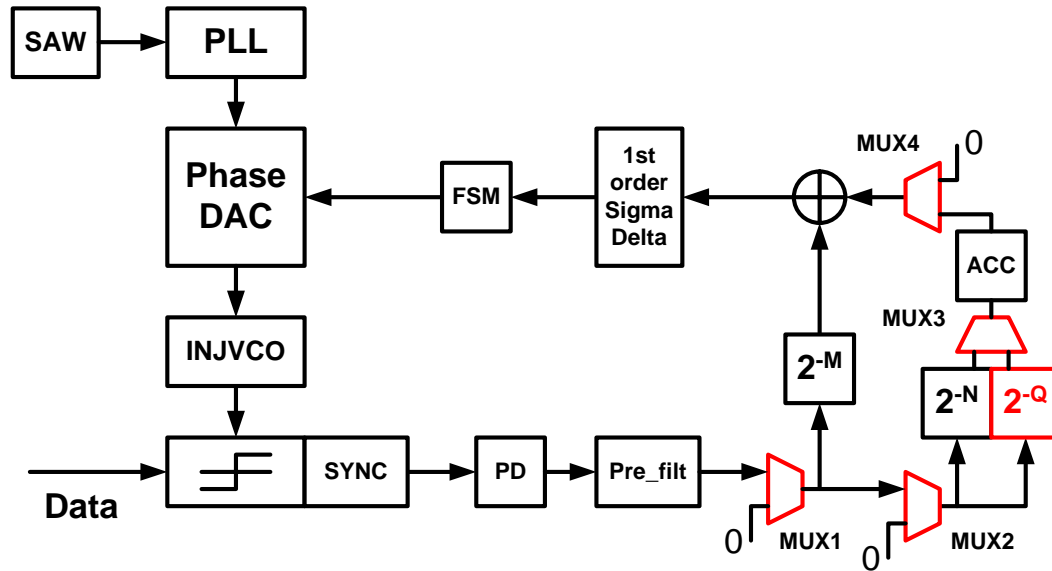
Figure 5.9: Modifications to the CDR to enhance the lock range in packet mode operation.

Packets and the intervals between them are emulated by selectively zeroing the output of the PD (via MUX1). Matlab is used to simulate the acquisition behavior (Figure 5.10). The settings used for the packet receiver are M=1, N=20, Q=10, packet length is 10k bits (i.e. P=10), and the interval between packets is 320kbits.[19] The phase acquisition plot shows the phase estimation error sampled at 10k bit intervals. Initially, the frequency estimate is not very good causing the accumulated phase estimation error at the start of each packet to be large. However, the phase estimation error due to drift decreases over time as the frequency estimate converges to the final value of 96.7ppm. The frequency acquisition plot shows the CDR successfully acquiring frequency lock. The CDR has a heavily damped response as expected.

---

[19] The packet length is 10240 bits to be exact. The interval is 32 times the packet length.
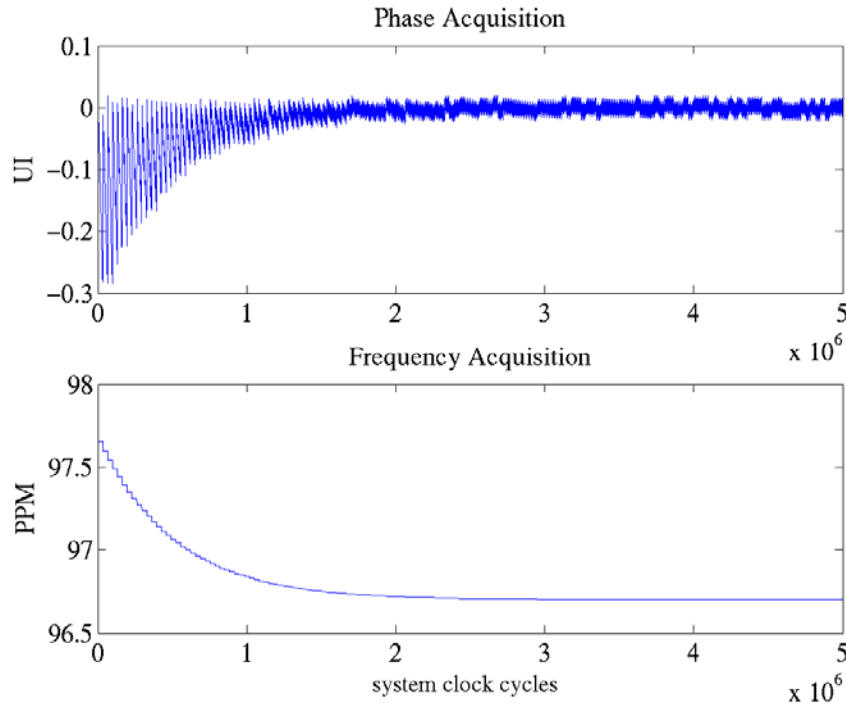
Figure 5.10: Matlab simulation demonstrating the phase and frequency acquisition behavior of the second order CDR in the presence of packets. The packets were 10k bits long with 320k bit spacing in between. The time axis is in digital clock cycles whose frequency is one tenth the data rate.

Though after initial convergence this packet receiver will achieve zero lock time, there is an overhead that is incurred once at system bootup. Figure 5.10 shows that the phase estimation error is equal to the steady state in about 20 million UI for 32 TXs. This means that only 62 packets from each TX are consumed by this process. For 32 TXs, this convergence time is about 6ms. Even if the system needs a reboot every one hour, the overhead is only 1.78e-4%.

5.3 Measurement Results

In this application, JTOL is not a meaningful performance metric since the packets are short relative to the period of the SJ. The real test of this CDR will be whether it can achieve and maintain phase lock in the presence of sparse packets thus demonstrating its viability as a burst mode receiver. The CDR shown in Figure 5.9 was implemented with the circuits detailed in Section 3.2 in a National Semiconductor 0.25μm CMOS process.

Unfortunately, the nonlinearity of the phase DAC which was measured in Figure 3.12 dominates the phase estimation accuracy of our burst mode receiver. However, we can account for this limitation by putting the measured phase DAC transfer function into our otherwise ideal Matlab model of the system. As we will see, we can get reasonable agreement between the measured and simulated behaviors giving us confidence in our model of the burst mode receiver.

When receiving continuous PRBS (2^{10}-1) data, the measured phase estimation error is 9.2ps rms and 61ps peak-to-peak (Figure 5.11). As expected, the peak-to-peak jitter is about 2 times the maximum step size. Figure 5.11 (b) is the phase estimation error from Matlab using the measured phase DAC transfer function. Since the measured jitter on the core loop PLL is 1.57ps rms and that on the TX PLL is 1.8ps rms, we combine them with the assumption that they are independent. Thus, a total of 2.4ps rms is applied to the input of the CDR model. It shows rough agreement with the measured results in that the lopsided error profile is seen in both and the peak-to-peak distribution is almost identical. This gives us confidence in verifying that the phase DAC nonlinearity is the issue and that our model is correct. It is difficult to match the measured and simulated profiles exactly. For instance, we do not know the frequency content of the PLL jitter and so it was assumed to be additive white Gaussian. Most likely, the jitter will have more lower frequency content due to the PLL loop bandwidths and a portion will be tracked by the CDR. This may explain why the peak-to-peak distribution is smaller in measurement.

As expected, the peak-to-peak phase estimation error increases by about 30ps (the INL) when the receiver is emulating the presence of burst mode packets. Figure 5.12 shows the measured jitter histogram of the recovered clock with the same setting as Figure 5.10 except that the interval is 1 million bits (instead of 320k bits). The packets are 10k bits long. This is the first demonstration of a CDR that is able to retain lock on data with such low transition density, thus proving the viability of its use in burst mode applications.

(a)   (b)
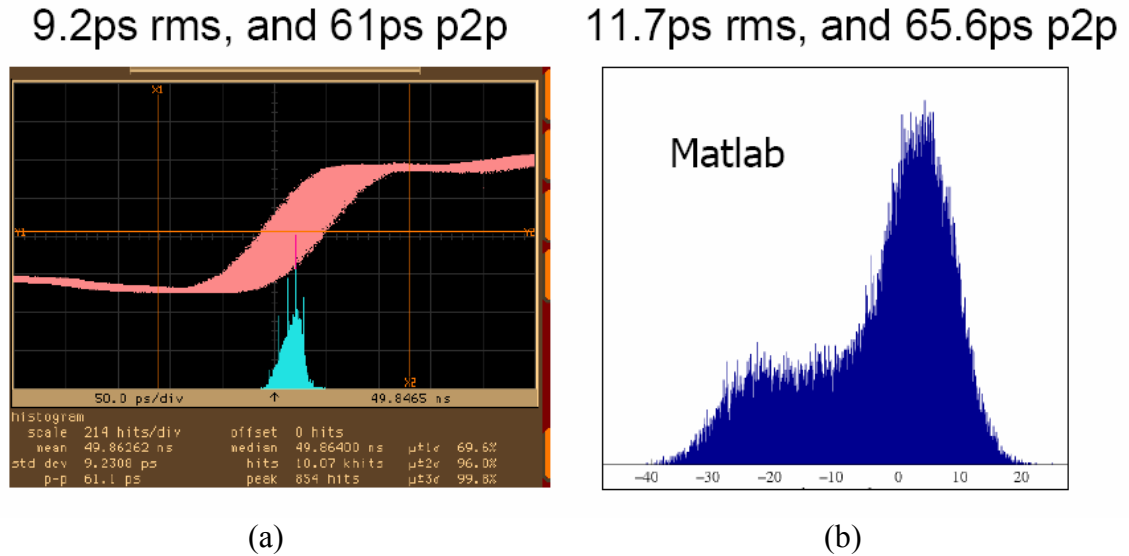
Figure 5.11: (a) Measured and (b) simulated phase estimation error in the presence of PRBS data (2^{10}-1).
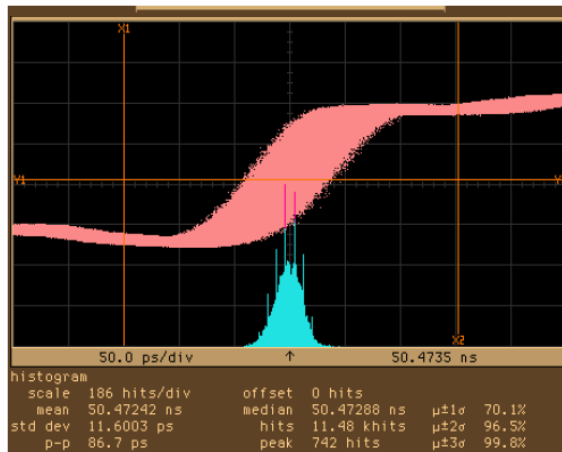
Figure 5.12: Measured phase estimation error in the presence of burst mode packets (10k bit packets and 1 million bit intervals).

While we have shown how to build hardware that can estimate frequency to high precision, a key question that remains is whether commercial oscillators are stable enough to support long idle times between packets. Measurement results using an EPSON EG2102 SAW oscillator show that they are. The frequency of the SAW oscillator is plotted over time in Figure 5.13. This is obtained by clocking the RX with a FLUKE clock generator (6060) which has much better stability than the SAW oscillator.

The SAW oscillator output is the input to the CDR. By reading out the content of the frequency accumulator through a scan chain, we can plot the frequency variations of the oscillator over time. This plot shows fifty measurements taken at 7.6 second intervals. There are two components to the variation. There is a quickly varying component on the order of 0.01ppm superimposed on a slow gradient. This slow gradient varies by only 0.03ppm over 4 minutes, and will be tracked by the CDR, even in packet mode.[20] The somewhat random 0.01ppm variation limits the frequency accuracy of our system since it cannot be tracked. This level of stability is more than sufficient for our system. With 0.01ppm error, the phase prediction error will only be 0.01 UI after 1 million bits. Furthermore, it is below the phase quantization level of 0.0156 UI set by the phase DAC.
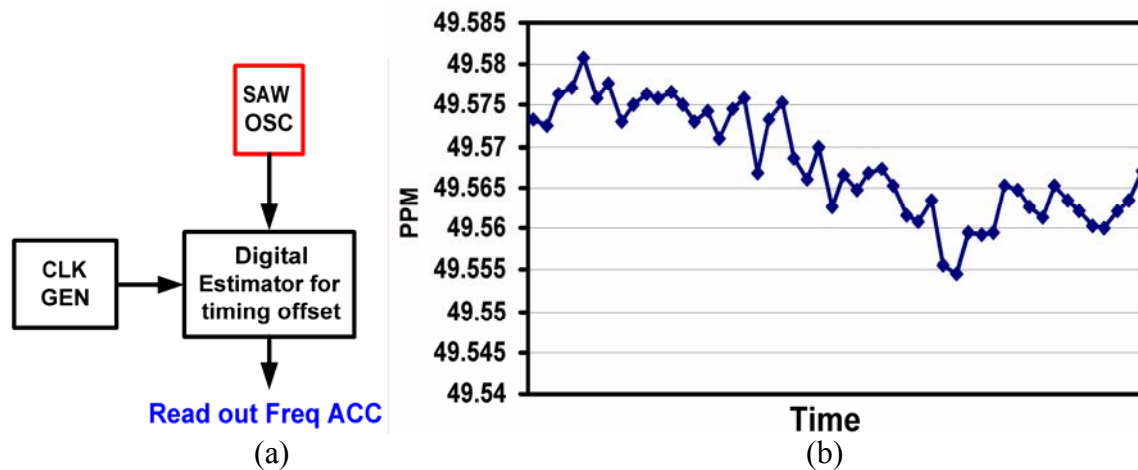
Figure 5.13: (a) Measurement setup for oscillator stability and (b) the measured frequency stability of a commercial SAW oscillator (Epson EG2102) at 7.6 sec intervals.

The die photo of the test chip is shown in Figure 5.14. Implemented in National Semiconductor's 0.25μm CMOS technology, its area is 3mm by 2mm. The area of the RX is increased considerably by the various offset measurement and correction circuits such as the lookup table (MEM). The synthesized digital logic includes not only the digital loop filter for the CDR but also the FSM that emulates the burst mode packets as well as the counters for on-chip phase-measurement. Using a 2.5V supply, the power of the RX is 150mW when operating at 3.125 Gbps.[21]

---

[20] This slow gradient is likely caused by temperature variations.
[21] The total TX power is 140mW with a swing of 1V$_{ppd}$.
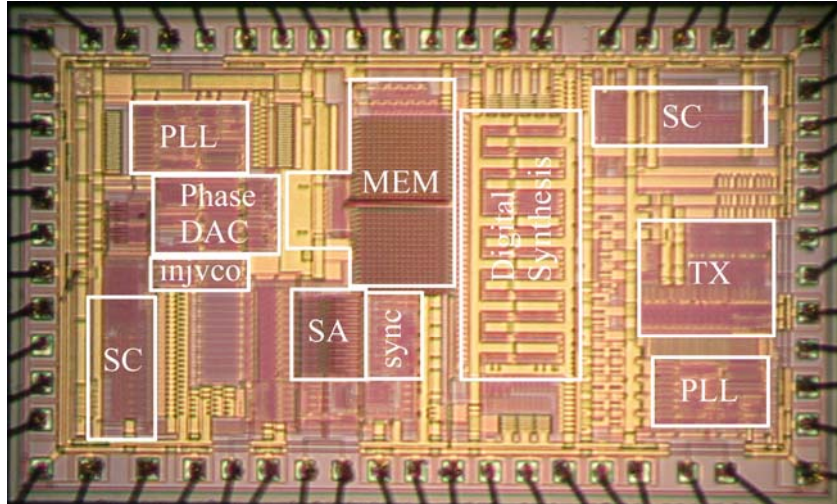
Figure 5.14: Die photo of test chip.

5.4 Summary

We have demonstrated in this chapter the ability to estimate phase and frequency accurately. The true limitation in building phase estimators is the frequency stability of the oscillator. Our measurements indicate that commercial SAW oscillators allow us to build phase estimators that can predict the phase 1 million bits away to within 0.01 UI. In order to achieve this level of accuracy, the phase DAC linearity is critical. Most importantly, we have demonstrated a CDR that can retain lock on 10k bit packets that are spaced apart by a million bits.

For TDM optical networks, we have shown that a highly accurate second order dual loop CDR can become a zero lock time burst mode CDR with appropriate modifications. Finally, this CDR also demonstrates the techniques necessary for building a CDR for very low transition density data.

Chapter 6

Spread Spectrum Clocking

So far, we have demonstrated the application of phase estimation in systems with a frequency offset. As I had mentioned earlier, when using a second order estimator for systems with a fixed frequency offset, a low bandwidth is optimum since the CDR is able to completely estimate the phase movement due to the frequency offset. However, the frequency offset is not fixed in some cases. For example, the clock frequency can be modulated to reduce electro-magnetic interference (EMI). This is called spread spectrum clocking (SSC). In this scenario, the low bandwidth of the second order CDR can degrade performance rather than helping it. To address this problem, this chapter will demonstrate a CDR which can predict the future phase position in the presence of SSC. We will show that by building a phase estimator that acquires all the relevant parameters of the SSC, it is possible to once again decouple the opposing constraints on the CDR from tracking the phase offset trajectory and filtering jitter. This in turn allows us to improve timing margin in comparison to the second order CDR. Section 6.1 provides a background of SSC and identifies an appropriate example for this study. Section 6.2 will then show the limitations of applying a lower (i.e. second) order CDR to this higher order trajectory. Section 6.3 will then describe the design of an appropriate estimator for SSC. Finally, the measured results will be presented in Section 6.4

6.1 Background

For ease of synchronization, it is often the case that the frequency of both the TX and the RX are both relatively stable. As the data rate of high speed links increases, the electromagnetic interference (EMI) caused by these clock sources becomes a problem as their output spectrum starts overlapping with wireless frequency bands. Such EMI is regulated by the Federal Communications Commission (FCC) as it disrupts the operation of radio-frequency (RF) circuits operating near various harmonics of these clocks.

Spread spectrum clocking (SSC) solves this problem by varying the frequency of the clock so as to spread its power over a range of frequencies such that the average power emitted at a specific frequency is reduced (Figure 6.1). An RF signal whose frequency is close to the nominal clock frequency of the link as well as its higher harmonics becomes more easily discernable due to this reduction in EMI.
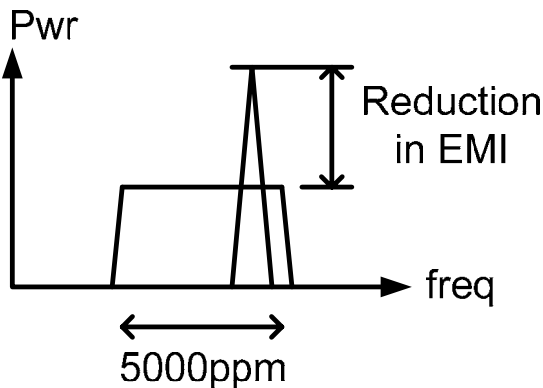
Figure 6.1: Frequency domain view of SSC.

A widely adopted SSC profile proposed in an industry standard, SATA [13], is shown in Figure 6.2. Figure 6.2 (a) shows a TX clock whose nominal frequency is varied by 5000ppm in a triangular profile with a modulation frequency of 30 to 33 kHz. This time-varying frequency offset results in the phase offset trajectory of Figure 6.2 (b) between the TX and RX. The abrupt switching from a positive slope in the frequency offset to a negative slope in the frequency offset, and vice versa, makes this a nonlinear process. The following sections will address the possible performance improvement as well as the cost of building CDRs able to estimate the future phase in this SSC scheme.

(a)

(b)
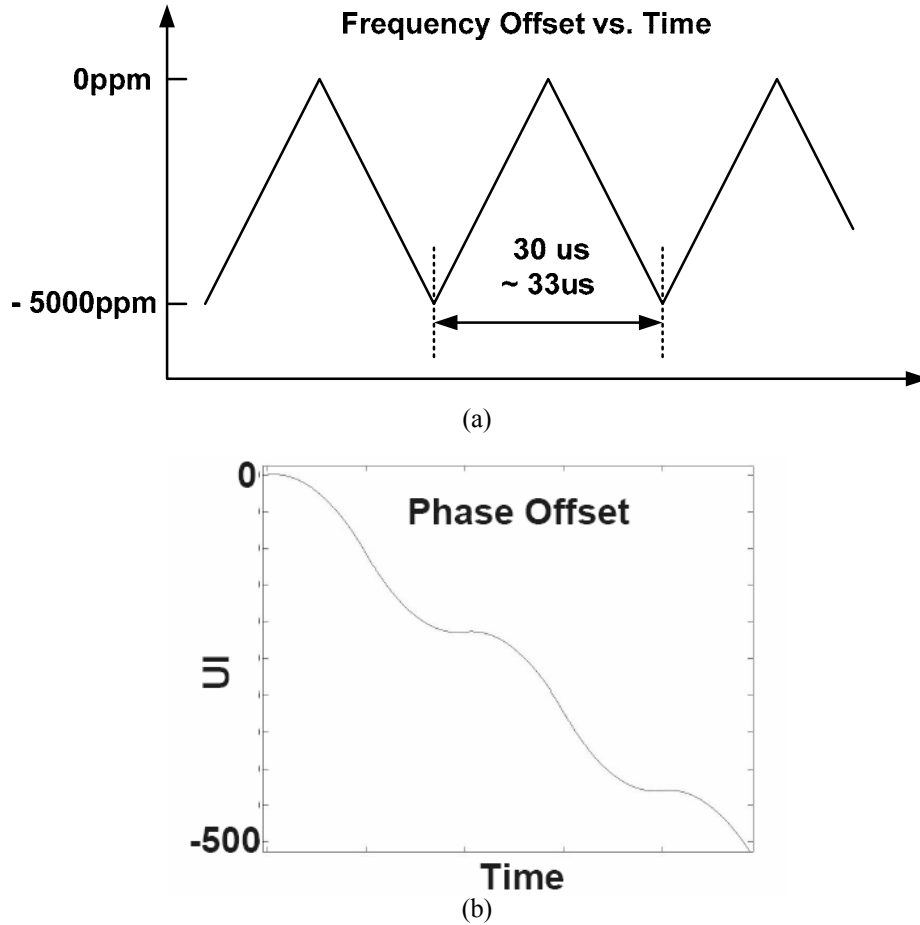
Figure 6.2: Time domain view of an example SSC. (a) Frequency offset and (b) phase offset between the TX and RX vs. time.

6.2 Performance of the Second Order CDR

Whereas the modulation frequency of the SSC was intentionally set low so that a second order phase estimator can be used to synchronize [13, 64], there is a performance penalty as the second order estimator can not predict the phase movement of a frequency ramp. This is found by applying the final value theorem of (4.14) as we had done in Section 4.3.3 R(s) is s^{-3} for a unit frequency ramp. The penalty from lacking predictive correction is inversely proportional to the integral gain as shown in (6.1). Here, K is $K_D \cdot K_{PD} \cdot TD$. This is analogous to the steady state phase estimation error when a first order CDR is used to track a frequency offset.

$$\phi_{ee,ss} = \frac{1}{K_i \cdot K} \qquad (6.1)$$

Hence, there is once again a tradeoff between wanting a lower bandwidth to filter jitter more and the need for larger bandwidth to minimize error in tracking the phase offset trajectory due to the SSC. We see this tradeoff in the simulations shown in Figure 6.3. These simulations compare the phase estimation error between the CDR clock and the transitions of the TX data (whose clock is SSC) when a second order CDR is used. Comparing the variance of a single peak in each case, it is clear that a lower integral gain results in a tighter distribution due to increased filtering. However, it also worsens the ability of the CDR to track the deterministic phase trajectory of the SSC resulting in a bimodal distribution. The bimodal distribution is because the polarity of the error in (6.1) depends on the polarity of the frequency ramp. In fact, instead of reducing phase estimation error, filtering can *worsen* the peak-to-peak distribution (in this example by almost two-fold). This estimation error will subtract from the timing margin of our link.
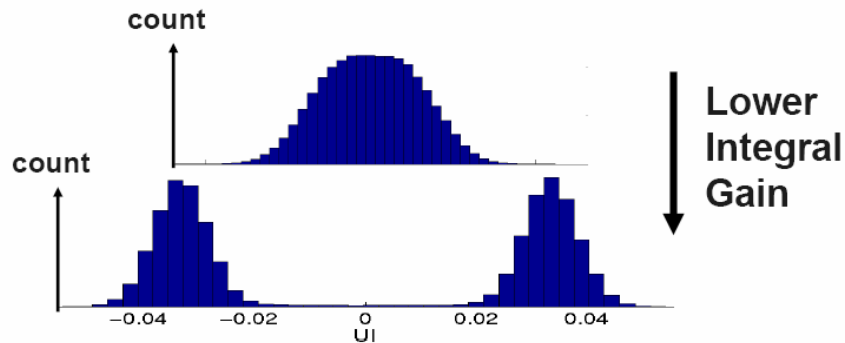
Figure 6.3: Histogram of the phase estimation error for the second order CDR tracking data from a TX using SSC. Results are from Matlab simulations. The integral gain is reduced by four times.

To remove this tradeoff in filtering and tracking, we can build a higher order estimator appropriate for SSC.

6.3 Estimator Design for SSC

The semi-digital dual loop architecture of Chapter 4 along with its circuits (such as the phase DAC) is once again used in this study. The only difference is in the implementation of the digital loop filter. As a reminder, the CDR along with its key components is shown in Figure 6.4. For simplicity, we only show the digital estimator portion as we develop the CDR for SSC systems.
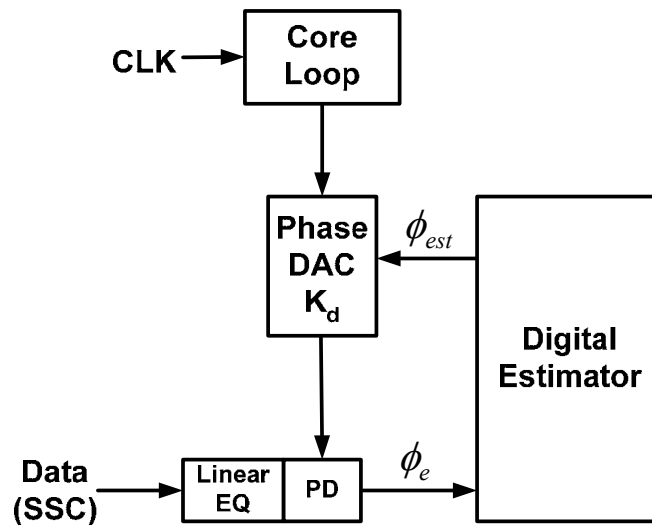
Figure 6.4: Semi-digital dual loop architecture used in this chapter.

6.3.1 Third Order Estimator

A third order estimator can predict the phase of future bits within each linear segment of the triangular profile by estimating the phase, frequency, and frequency ramp rate of the TX (Figure 6.5). These estimates are contained respectively in the three accumulators (P_acc, F_acc, and R_acc). A CDR using this estimator can track a frequency ramp with zero mean phase error unlike the second order CDR. However, this CDR will perform worse than a second order when the polarity of the frequency ramp rate changes abruptly (i.e. at the switching points of the SSC). This is because the ramp rate accumulator continues to push the frequency estimate in one direction when in fact it should be pushing it the other way causing significant overshoot in the frequency estimate. Hence, it is clear that this estimator needs to be augmented with an estimate of

when the frequency ramp will change polarity. This will enable the CDR to switch the polarity of its frequency ramp estimate at the appropriate time. To do this, one must obtain both the phase and frequency of the triangular modulation. Sections 6.3.2 and 6.3.3 will detail two different methods of obtaining this information.
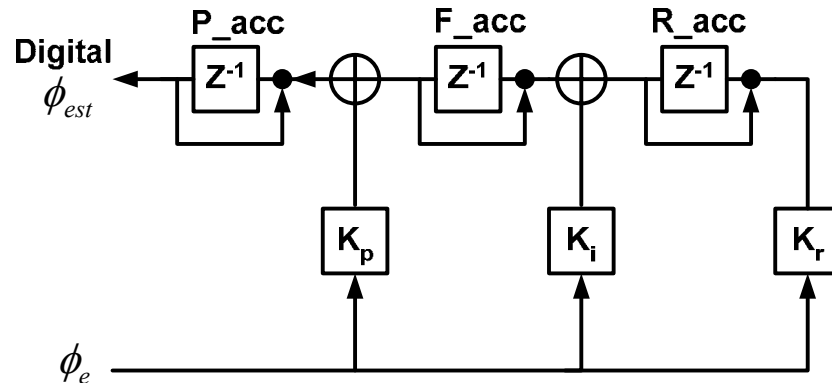
Figure 6.5: Third order estimator which acquires the phase, frequency, and frequency ramp rate of the TX data with respect to the RX clock.

6.3.2 Modulation Estimation using the Frequency Mean

The first method of estimating the modulation phase and frequency compares the frequency estimate (in F_acc) to its mean (i.e. -2500ppm). The frequency estimate before the system has fully converged draws a triangular profile akin to that of the SSC but with a time lag, rounded corners, and noise (Figure 6.6). The comparison of the frequency estimate with its mean produces a square wave whose transitions are approximately a quarter period (90°) delayed with respect to the switching points of the SSC. A second order digital PLL (DPLL) is locked to this signal, so as to acquire its phase and frequency, and its output clock is phase advanced by 90 degrees to produce an estimate of when the frequency ramp changes polarity. More details of this DPLL will be provided in Section 6.3.4

When the DPLL is locked, then the switching points between the frequency ramps can be predicted. In fact, the rising and falling edges of the DPLL output will coincide with the switching points. Furthermore, when the DPLL output is high, then it is an indication that the frequency ramp slope is positive. On the other hand, if it is low then it indicates that the frequency ramp of the SSC has a negative slope.
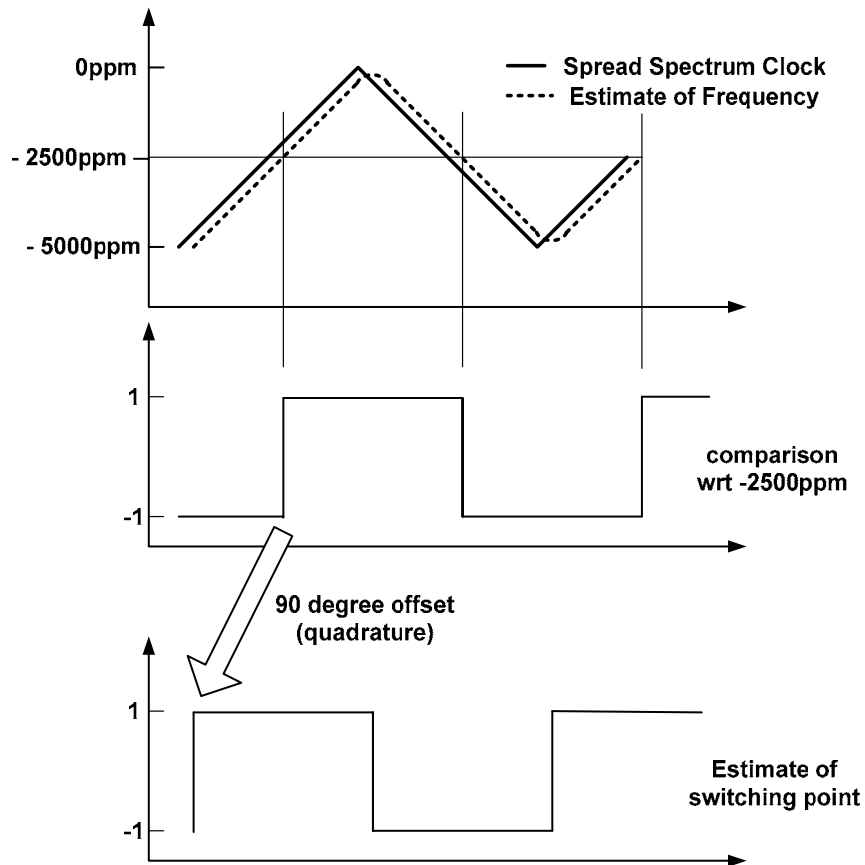
Figure 6.6: Graphical view of modulation estimation by comparing the frequency estimate with its mean.

Figure 6.7 shows the SSC estimator that uses this approach of estimating the two distinct regions of the SSC. The output of the DPLL is used to enable one of two frequency ramp rate accumulators (Rp_acc, Rn_acc). Rp_acc is the third order accumulator containing an estimate of the positive frequency ramp rate whereas Rn_acc is that for the negative. The output of the DPLL is used to enable the input and output of the correct frequency ramp rate accumulator using MUXes. When the DPLL output is high, Rp_acc is enabled. When DPLL output is low, Rn_acc is enabled. In steady state, this system operates as a third order CDR with Rp_acc when the ramp is positive and Rn_acc when it is negative. The ramp accumulators saturate to maintain the correct polarity to help convergence. The possibility of using a single ramp accumulator was investigated but it did not converge as the phase error from the positive ramp (negative

error) cancels that from the negative ramp (positive error) causing a small stable oscillation.
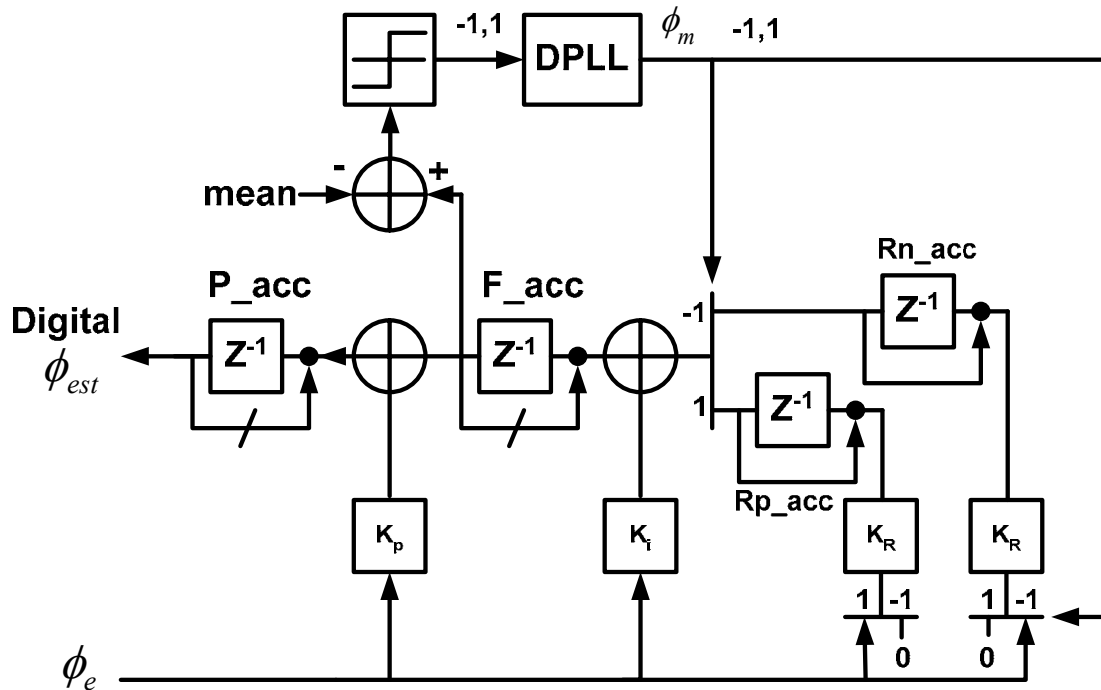
Figure 6.7: SSC estimator using the frequency mean to derive the modulation information.

Figure 6.8 shows the acquisition behavior of this SSC estimator. The gains in the loop are chosen so that the states are acquired in the following order: phase, frequency, modulation phase, modulation frequency, and finally the frequency ramp rates. The corresponding gains are K_P, K_i, K_{MP}, K_{MI}, and K_R. These gain magnitudes are proposed because the modulation estimation requires F_acc to first track the frequency of the TX. Furthermore, only after the modulation estimation is completed can the frequency ramp rate be estimated since the error information accumulated into Rp_acc and Rn_acc will be invalid otherwise. This condition has been seen in simulations to be required for convergence and stability.

Before Rp_acc and Rn_acc converge to their final values, the estimator behaves as a second order estimator. However, as the ramp rates converge, the estimator reduces the mean of the error to zero. The time lag and corner rounding of the frequency estimate seen in Figure 6.6 also disappear as the CDR is able to more accurately predict the

trajectory. When this estimator has acquired all states, it will take predictive correction based on previous error rather than depending only on current error information.

(a)

(b)

(c)
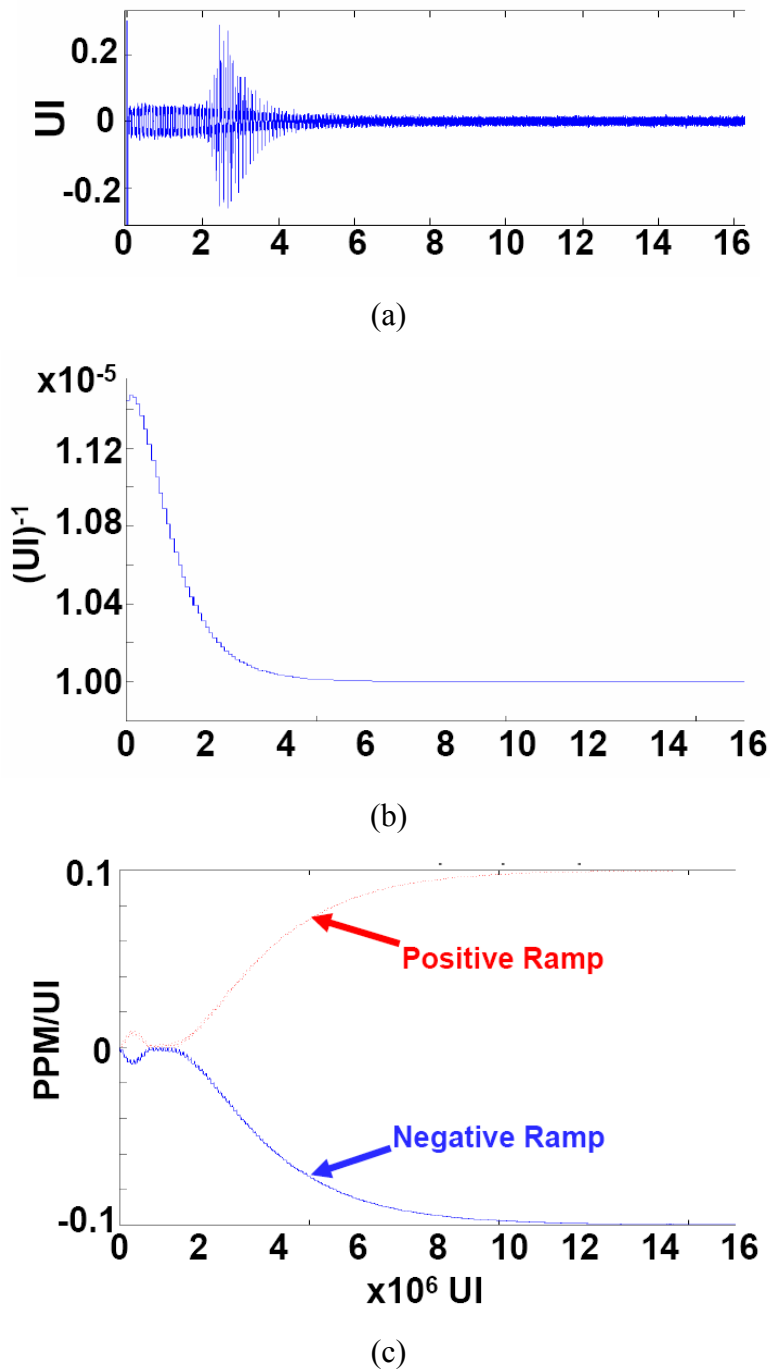
Figure 6.8: Acquisition behavior of the SSC estimator using the frequency mean. (a) phase estimate error, (b) estimated modulation frequency, and (c) estimated frequency ramp rate vs. time. The gains used are $K_P=1/2^3$, $K_i=1/2^9$, $K_R=1/2^{29}$, $K_{MP}=1/2^{15}$, and $K_{MI}=1/2^{32}$. Phase DAC has 128 steps per UI.

Unlike the second order estimator, the SSC estimator has only one peak in its phase estimation error distribution regardless of its bandwidth. This is a direct result of building a higher order estimator that acquires all the necessary parameters of the trajectory. Furthermore, its peak phase estimation error reduces monotonically with the reduction in K_i (Figure 6.9). This implies that the timing margin of the SSC estimator will also improve monotonically. In contrast, we find that the second order estimator has a convex shape since an optimum exists between tracking and filtering. The key observation is that the SSC estimator removes the opposing forces that the deterministic trajectory and jitter posed to the second order estimator.
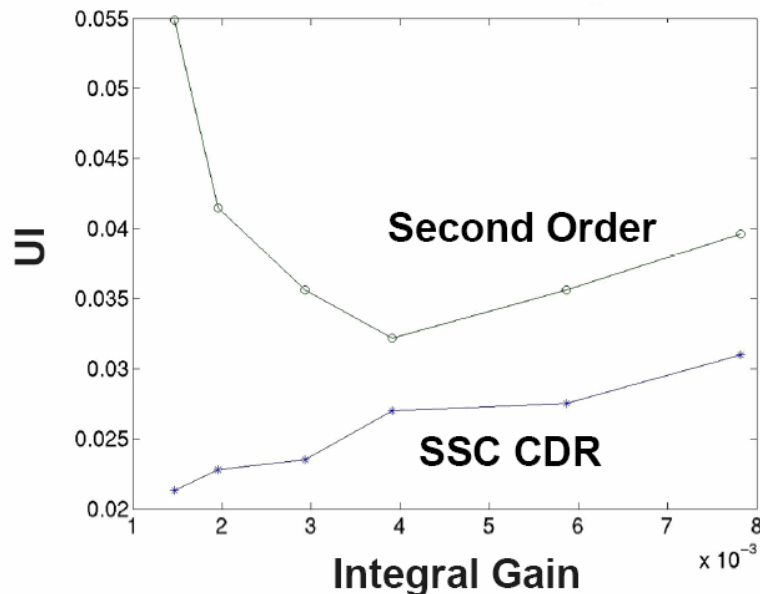
Figure 6.9: Comparison of the peak phase estimation error (UI) for the second order and SSC estimators when tracking data from a TX using SSC. Results are from Matlab simulations. $K_P=1/2^3$ for both estimators. $K_R=1/2^{29}$, $K_{MP}=1/2^{15}$, and $K_{MI}=1/2^{32}$ for the SSC estimator. Phase DAC has 128 steps per UI. Random jitter σ is 0.0214 UI. The peak error is that observed in 16e6 bits.

The single drawback of this estimator is that the mean of the frequency is not known *a priori*. SATA specifies an uncertainty of +/- 350ppm in the mean. This uncertainty leads to error in estimating the switching points of the SSC which would greatly increase the phase estimation error. Solving this necessitates another filter. To circumvent this, we investigated an alternative method of modulation estimation which is the one that was included in the chip.

6.3.3 Modulation Estimation using Frequency Differentiation

In the design we implemented, we estimate the modulation phase and frequency by differentiating the output of the frequency accumulator (F_acc). This is equivalent to subtracting each sample of F_acc from a previous one. The sign of this subtraction tells us the polarity of the frequency ramp and draws out a square wave whose transitions coincide with the switching points of the SSC (Figure 6.10). Similar to before, a DPLL is locked to this switching point estimate to acquire its phase and frequency.[22]

Since the ramp rate is not large, glitches can occur in random locations as the differentiation tends to amplify noise. Most glitches can be removed by running this differentiation at a much slower sample rate than the rest of the CDR logic such that the change in frequency will be larger between samples. Glitches occurring at the switching points due to the initially rounded corners (i.e. small slope) are removed with a simple filter with hysteresis at the input of the DPLL. This filter is a 3 bit accumulator whose output triggers a high if the state is above 5 and a low if it is below 2.
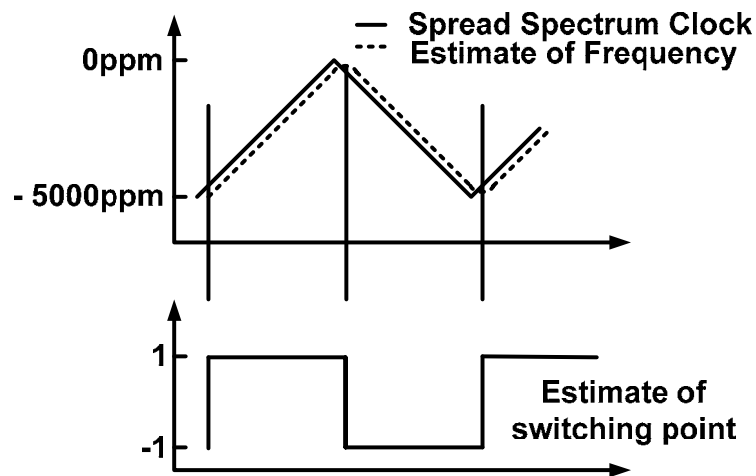
Figure 6.10: Graphical view of modulation estimation by differentiating the frequency estimate.

Figure 6.11 shows the implemented SSC estimator. The complete CDR is shown in this case. The portion of the estimator in the red box operates at a clock frequency that is 128 times slower than the rest of the CDR to mitigate the effect of noise. The

---

[22] However, this method does not require a phase advance of 90 degrees.

aforementioned filter with hysteresis between the comparator and the DPLL is not shown for simplicity. Furthermore, the estimator has the ability to operate as a second order CDR so as to provide a fair performance comparison.
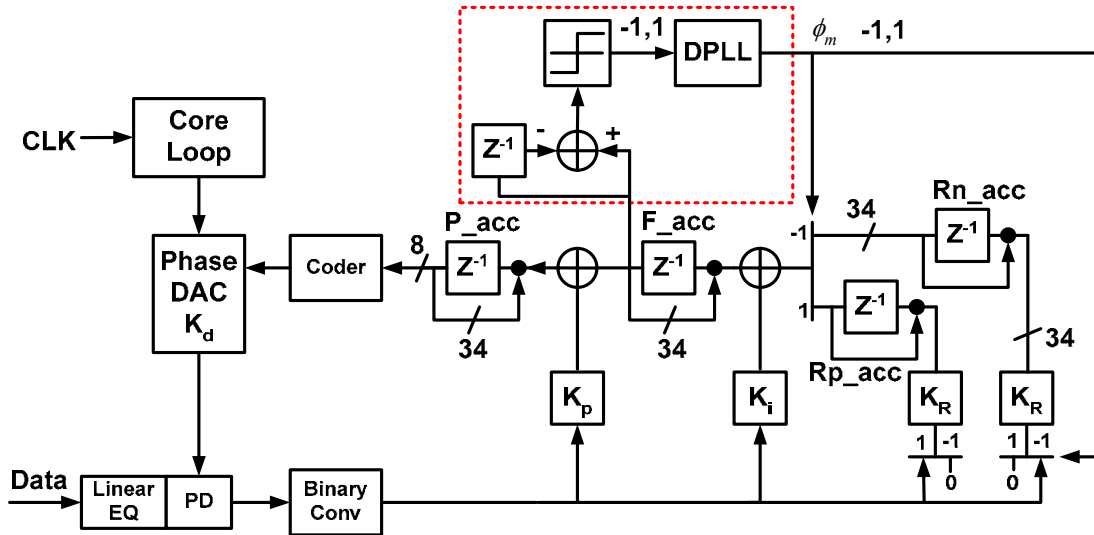
Figure 6.11: SSC estimator using the derivative of the frequency estimate to perform the modulation estimation. Each of the gains can be programmed over a range of 16x.

Unfortunately, we found out after tapeout that a very slow phase estimate error beat generated by this modulation estimation technique limits the performance improvement from this system (Figure 6.12). The problem arises from the slower clock rate of the modulation estimation. This effectively increases the quantization error of the modulation phase as the DPLL observes samples of the modulation waveform that are spaced wider apart in time. In this design, the DPLL resolution is approximately 1% of the period of the modulation (~0.3μs).

To illustrate, let us assume that the switching point of the SSC occurs 0.5% of a period later than the modulation phase estimate of the DPLL. The SSC estimator accumulates phase error as the frequency ramp rate will have the wrong polarity for 0.5% of the modulation period. However, as the modulation frequency is estimated with finite precision, the time relationship between the estimated modulation phase and the actual switching point will drift over time. If the DPLL underestimated the modulation frequency by some fraction of an LSB, then after some time the switching point will be estimated perfectly. This instance results in the least phase estimation error. After this

happens the switching point will occur earlier than the estimate thus increasing the phase error again. When the switching point has moved yet another 0.5%, the DPLL will detect that the modulation phase estimate is too late and will correct it. This process repeats to produce the phase estimate error beating.

The beat frequency is a function of the sub-sampling factor of the differentiation and the frequency resolution of the digital PLL resulting in a period of 10 million bits in this example. Due to this very long period, this bug was not found before tapeout. As K_i is reduced, the phase error increases since the CDR has less ability to correct for the error caused by the incorrect estimation of the frequency ramp polarity. This results in a convex phase error profile much like that seen in the second order CDR. Hence, the performance of this SSC estimator is degraded in the region at which it is supposed to show the most benefit. The impact of this design error on the performance of the CDR will be shown in Section 6.4
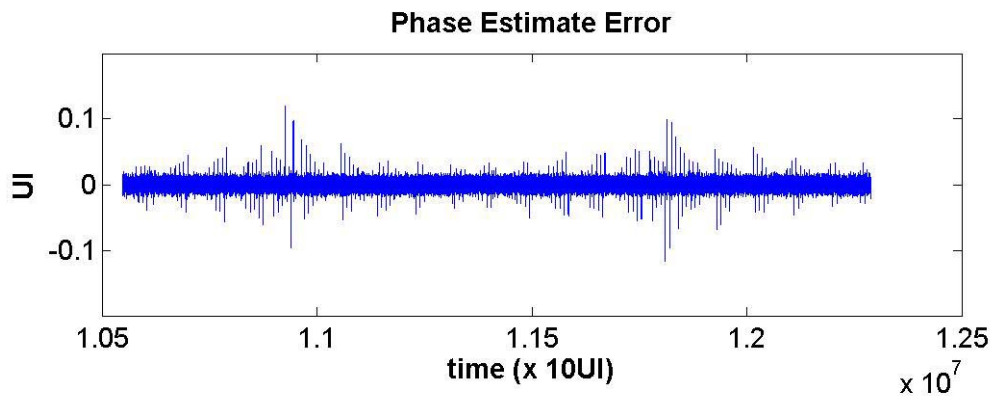
Figure 6.12: Phase estimate error beating due to the interaction of the quantization error of multiple loops. Results are from Matlab simulations. $K_P=1/2^3$, $K_i=1/2^9$, $K_R=1/2^{29}$, $K_{MP}=1/2^8$, and $K_{MI}=1/2^{16}$. Phase DAC has 128 steps per UI. Random jitter σ is 0.0214 UI. The peak error is about 0.12 UI.

6.3.4 Digital PLL for Modulation Estimation

Figure 6.13 is the DPLL used in the modulation estimation. The structure is similar to the second order CDR but for a few differences. The first difference is that the feedback clock is simply the MSB of the phase accumulator (DP_acc). The digital number inside DF_acc sets how quickly DP_acc overflows. Hence it represents the frequency of the

MSB of DP_acc. The second difference is the phase detector. The digital phase detector (DPD) emulates a linear analog phase detector. When the DPD detects a change in the state of its input (i.e. a transition), it starts counting up the number of cycles until a transition occurs on the feedback clock. When the feedback clock transitions first, then the DPD counts down until the input transitions. Since a positive number advances the phase of the feedback clock, the DPD output is positive when the feedback clock comes late. This counter value is dumped to the loop filter and the DPD is reset when the later transition occurs. Advancing the phase by 90 degrees (for the modulation estimation using the frequency mean) is achieved by adding a digital number to the output of DP_acc. This requires an additional adder outside the loop. The MSB of the output of this adder is used to control the MUXes in the SSC estimators.
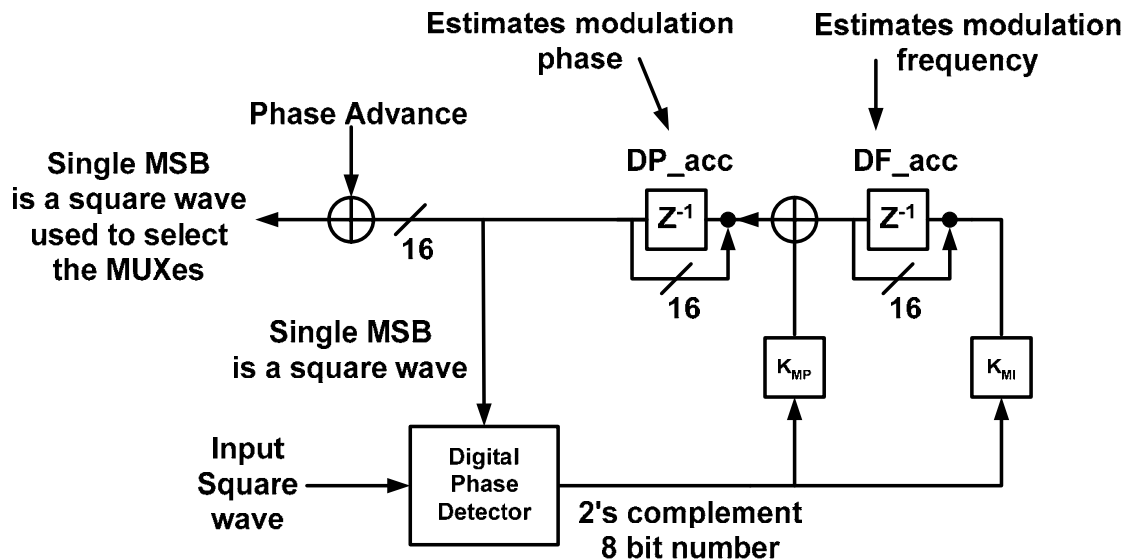
Figure 6.13: DPLL used in the modulation estimation. K_{MP} is its proportional gain and K_{MI} is its integral gain.

6.4 Measured Results

A similar test setup as that for jitter tolerance testing is used (Figure 6.14). The TX clock is an SSC clock part that runs at 100MHz (ICS9FG104). The differential TX output is passed through a backplane to add 0.4 UI of DJ. The clock to the RX test chip is sinusoidally modulated at 10MHz to measure the timing margin. RJ added is about 4ps

RMS. The SJ amplitude is increased until the on-chip PRBS error counter gives us the target BER of 10^{-11}. This BER was chosen to speed up testing.
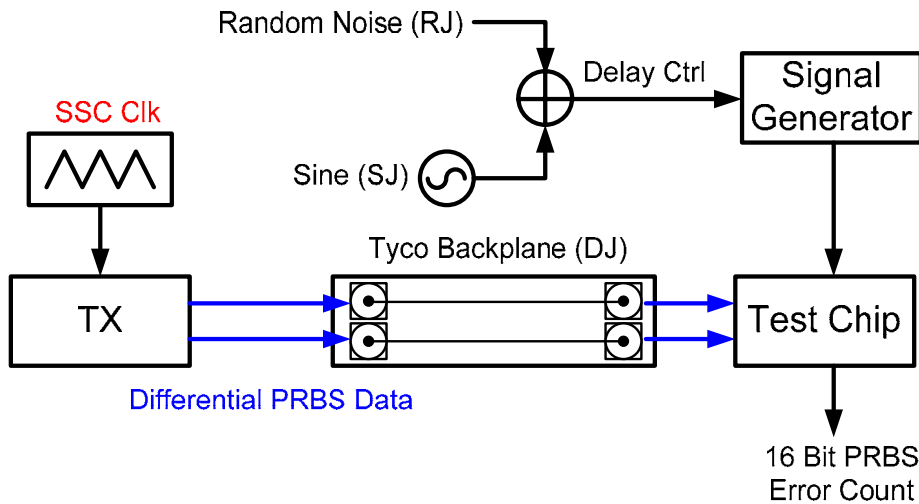
Figure 6.14: Test setup for measuring the timing margin of the SSC and second order estimators.

We are able to verify that the SSC estimator works as intended by scanning out the states. The best indicators are the frequency ramp rates and the modulation frequency since they do not vary much over time. Comparing them to the expected values from an ideal SSC profile shows that they have converged to their expected values.

Figure 6.15 shows the measured timing margin of the second order and the higher order CDR as the integral gain (K_i) is varied when receiving SSC data. An optimum exists for the second order estimator because of the opposing bandwidth constraints from jitter filtering and tracking the SSC. When testing with and without the channel, we notice that the optimum gain setting changes. This is because the tradeoff between tracking the deterministic trajectory and filtering the jitter also changes with the amount of jitter in the system.

The margin of the SSC estimator improves at lower gain settings in comparison to the second order estimator as expected. The margin improvement is 0.05UI for the range of K_i that we tested. However, the improvement is limited at lower gains due to the phase error beating problem I described earlier. Since simulations of the SSC estimator that uses the frequency mean indicated that the timing margin of a properly designed SSC

estimator will improve monotonically with reduction in integral gain, the measured results suggest that the achievable margin improvement at lower integral gain is on the order of 0.1UI.
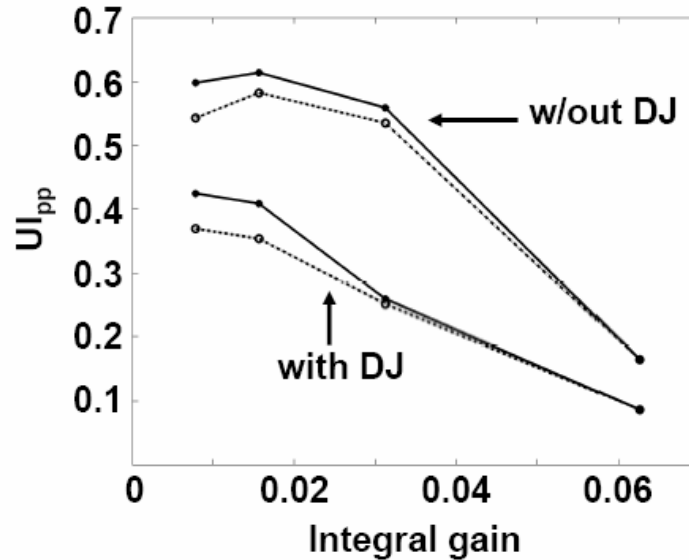
Figure 6.15: Timing margin vs. integral gain. TX is using a SSC clock. Second order CDR (dotted, °) and SSC estimator (solid, *). The timing margin is measured at a BER of 10^{-11}. $K_P=1$, $K_R=1/2^{26}$, $K_{MP}=1/2^8$, and $K_{MI}=1/2^{16}$. Phase DAC has 128 steps per UI.

During testing, we observed that neither CDR operated at the lowest settings of K_P and K_i ($1/2^3$ and $1/2^{10}$). At this setting, the CDRs fail to converge. In fact, both gains had to be increased by a factor of 4 for the CDRs to even provide an eye opening at the target BER of 10^{-11}. Furthermore, the best timing margin was observed when K_P was further increased by a factor of 2. This is the reason that the data was taken with $K_P=1$.

With a triangular SSC profile and relatively large jitter ($\sigma=0.036$ UI) modeled with a Gaussian distribution, it has not been possible to replicate this behavior in simulations. Furthermore, simulations indicate a shallow optimum in the timing margin under these conditions rather than a steep roll off as seen in the data. These observations indicate that some other factor that was not accounted for in our simulations is the culprit.
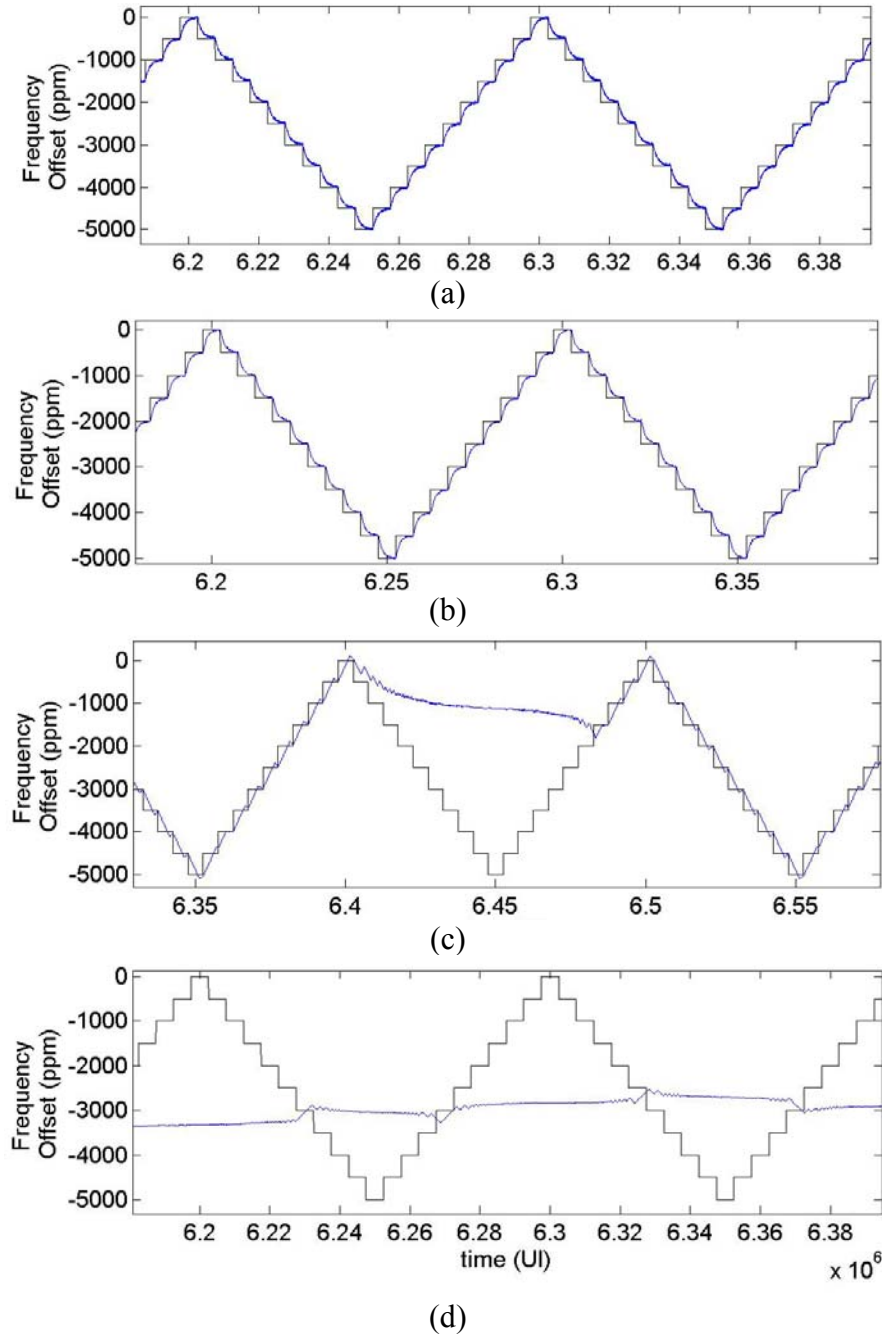
Figure 6.16: Time domain tracking behavior of the second order CDR in the presence of a staircase SSC that would result from a fractional-N PLL. The frequency step size is 500ppm. Random jitter σ is 0.0214 UI. Phase DAC has 128 steps per UI. $K_P = 1$ and $K_i = 1/2^7$ for (a). Both gains are decreased by a factor of two for each successive plot. The CDR operates only at $K_P = 1$ and 1/2.

As it turns out, the triangular profile is only *suggested* by the SATA standard. The standard only specifies the frequency spectrum which can look almost identical despite

very different time domain characteristics. For example, a SSC generated by a fractional-N PLL with coarse frequency resolution will result in a very different phase offset trajectory in comparison to a triangular profile. Other researchers have also noted this problem with SSC synthesizers based on fractional-N PLLs [65]. Coarse frequency resolution can explain why the CDRs failed unexpectedly at lower bandwidths. Figure 6.16 shows Matlab simulations of the second order CDR with different gains settings in the presence of a staircase SSC profile typical of a fractional-N PLL. For a frequency step size of 500ppm, the CDR only operates with higher K_P settings (1 and 1/2) as observed in the lab.

The test chip was implemented in a TSMC 0.13um LV process. The supply was 1.0V and the link operates at 3Gbps. In this process, the area of the digital loop filter for the second order CDR is $8600\mu m^2$ while that of the SSC estimator is $66000\mu m^2$.

6.5 Summary

Estimation provides insight into the optimal structure of the CDR. For the SSC profile that we looked at, five parameters are needed (phase, frequency, frequency ramp rate, modulation phase, and modulation frequency) to predict the phase of future bits. If the structure of the CDR matches the phase trajectory, then the required update gain can be set very low (so as to maximize the timing margin) as the opposing bandwidth constraints due to tracking the deterministic phase trajectory and filtering jitter are decoupled. Unfortunately, it is difficult to capture the characteristics of the phase trajectory completely since it is not directly specified in many cases. For instance, a common method of creating the SSC is to employ a fractional-N frequency synthesizer. While the SSC produced will meet the specifications set forth by the standard in terms of modulation depth and frequency, the actual phase offset trajectory can vary significantly based on the frequency resolution of the synthesizer. Without a direct specification on the deterministic phase offset trajectory, the effectiveness of higher order estimators will be limited.

Chapter 7

Conclusions

This research has investigated the benefit of applying an estimation approach to clock and data recovery. We have shown that the phase movement of the data can be partitioned into the deterministic phase offset trajectory and jitter. These two components of phase movement pose very different design objectives.

In today's high speed links, jitter is dominated by deterministic jitter (DJ) caused by channel limitations and random jitter (RJ). For such uncorrelated jitter, it is best to minimize the CDR bandwidth since past information sheds little light into future behavior. This is clearly not the case for deterministic phase offset trajectories that are the result of any frequency offsets between the TX and RX. Chapter 4 and Chapter 6 showed that by matching the structure of the CDR to the phase trajectory, one can exploit its determinism and produce improved timing margin for a variety of applications. If the correlated trajectory is known, a CDR with the correct matching structure will allow the CDR designer to increase its filtering such that the effect of DJ and RJ is minimized. Furthermore, we have shown that the predictive nature of such CDRs can also be leveraged to decouple the opposing constraints of lock time and jitter filtering. Chapter 5 showed the feasibility of a zero lock time CDR using this approach.

Apart from the limitations due to the phase DAC linearity, additional drawbacks of this approach were highlighted in Chapter 6. First, the complexity of the structure

increases very quickly. This leads to behavior that is harder to fully verify without the silicon and is difficult to capture with mathematical models. In the absence of intuition that equations often provide, it is difficult to know if the simulation is capturing all the behavior that affects performance. The second is that producing a model of the phase offset trajectory is also a challenge. This may be either due to implementation differences (as in SSC) or simply due to aging and drift common in all circuits. This uncertainty means that it is never optimum to reduce the CDR bandwidth as much as theoretically possible. In practice, it is always beneficial to leave some bandwidth to track these imperfections in the model.

This leads us to the question of what the optimal gain setting is. While the optimal structure is set by the deterministic phase offset trajectory, the optimal gains are then set by the uncertainty of this trajectory. Since noise is not always known a priori, estimator design theory has moved onto adaptive gain systems that adjust their bandwidth in accordance with the observed noise/uncertainty to minimize estimation error. As more complicated estimator designs are used, both the optimal structure and the optimal gain should be considered. Finding an optimal algorithm for loop bandwidth adaptation is a possible area of research in both CDR and PLL design.

Another possible area of research is to build a CDR that can perform piece-wise estimation on the phase offset trajectory. Such an estimator can handle different types of SSC modulation including triangular, trapezoidal (which occurs when both the TX and RX have independent triangular modulation), and staircase and would not require a priori knowledge of the modulation.

Finally, the continued scaling of CMOS technology will accentuate the effect of random transistor mismatches on the phase DAC linearity. Applying techniques from voltage domain DAC design to address this issue will be a fruitful area of research.

Bibliography

[1] R. Kollipara, B. Chia, Q. Lin, J. Zerbe, "Impact of Manufacturing Parametric Variations on Backplane System Performance," 6-WA2, High-Performance Backplane System Design, DesignCon2005, Santa Clara, CA, USA.

[2] R. Kollipara *et al*, "Practical Design Considerations for 10 to 25 Gbps Copper Backplane Serial Links" High-Performance Backplane System Design, DesignCon2006, Santa Clara, CA, USA.

[3] J. Zerbe, C. Werner, V. Stojanović, F. Chen, J. Wei, G. Tsang, D. Kim, W. Stonecypher, A. Ho, T. Thrush, R. Kollipara , M. Horowitz, K. Donnelly, "Equalization and Clock Recovery for a 2.5-10Gb/s 2-PAM/4-PAM Backplane Transceiver Cell," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, Dec. 2003, pp. 2121-2130.

[4] J. L. Zerbe *et al*, "1.6Gb/s/pin 4-PAM signaling and circuits for a multidrop bus," *IEEE J. Solid-State Circuits*, vol. 35, May 2001, pp. 752-760.

[5] J.T. Stonick *et al*, "An adaptive pam-4 5-Gb/s backplane transceiver in 0.25-μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 3, March 2003, pp. 436-443.

[6] V. Stojanović *et al*, "Autonomous Dual Mode (PAM2/4) Serial Link Transceiver with Adaptive Equalization and Data Recovery," *IEEE J. Solid-State Circuits,* April 2005, pp. 1012 -1026.

[7] M. Meghelli *et al*, "A 10Gb/s 5-Tap-DFE/4-Tap-FFE Transceiver in 90nm CMOS," *IEEE International Solid-State Circuits Conference,* Feb. 2006, pp. 80-81.

[8] R. Payne *et al*, "A 6.25Gb/s Binary Adaptive DFE with First Post-Cursor Tap Cancellation for Serial Backplane Communications," *IEEE International Solid-State Circuits Conference,* Feb. 2005, pp. 68-69.

[9] M. Wang, A. Langari, H. Hashemi, "Advanced Packaging for GHz Switching Applications," *2002 Electronic Components and Technology Conference*

[10] R. McBride, S. Rosser, R. Nowak, "Modeling and Simulation of 12.5Gb/s on a HyperBGA Package," *2003 IEEE/CPMT/SEMI Int'l Electronics Manufacturing Technology Symposium*

[11] H. Liaw, P. Yue, R. Emigh, D. Shin, "Package and Test Environment Design for a 10-Gigabit Ethernet Transceiver," DesignCon2004, Santa Clara, CA, USA.

[12] L. DeVito, "A versatile clock recovery architecture and monolithic implementation," in *Monolithic Phase-Locked Loops and Clock Recovery Circuits: Theory and Design*, B. Razavi, Ed. New York, NY: IEEE Press, 1996, pp. 405–42.

[13] Serial ATA Workgroup, "SATA: High Speed Serialized AT Attachment", Rev. 1.0, Aug. 2001.

[14] "Fiber Channel – Methodologies for Jitter and Signal Quality Specification," T11.2/Project 1316-DT/Rev 1.0, 2001.

[15] "SONET OC-192 Transport System Generic Criteria," Telcordia, Piscataway, NJ, Tech. Rep. GR-1377-CORE, Mar. 1998.

[16] "IEEE Std 802.3ae-2002," Aug. 2002.

[17] C.R. Hogge, "A self-correcting clock recovery circuit," *IEEE J. Lightwave Technology.*, vol. 3, Dec. 1985, pp. 1312–1314.

[18] T.H. Lee, J.F. Bulzacchelli, "A 155-MHz clock recovery delay- and phase-locked loop," *IEEE Journal of Solid-State Circuits,* vol. 27, no. 12, Dec. 1992, pp. 1736-1746.

[19] L. DeVito, J. Newton, R. Croughwell, J. Bulzacchelli, F. Benkley, "A 52MHz And 155MHz Clock-recovery PLL," *IEEE International Solid-State Circuits Conference,* Feb. 1991, pp. 142, 306.

[20] M. Perrott *et al*, "A 2.5 Gb/s, Multi-rate, 0.25um CMOS Clock and Data Recovery Circuit Utilizing a Hybrid Analog/Digital Loop Filter," *IEEE International Solid-State Circuits Conference,* Feb. 2006.

[21] J. Savoj, B. Razavi, "A 10-Gb/s CMOS clock and data recovery circuit with a half-rate linear phase detector," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 761 - 767, May 2001.

[22] J.D.H. Alexander, "Clock Recovery from Random Binary Signals", *Electronic Letters*, vol. 11, October 1975, pp. 541-542.

[23] F. Gardner, "Charge-pump phase-lock loops", *IEEE Trans. Communications*, vol. COM-28, no. 11, pp. 1849-1858, Nov. 1980.

[24] J.G. Maneatis, "Low-jitter process-independent DLL and PLL based on self-biased techniques," *IEEE Journal of Solid-State Circuits,* vol 31, no. 11, Nov. 1996, pp. 1723-1732.

[25] G. Franklin, D. Powell, A. Emami-aeini, *Feedback Control of Dynamic Systems*, Prentice Hall, Upper Saddle River, NJ, USA, 2005.

[26] Y. Greshishchev, P. Schvan, "SiGe clock and data recovery IC with linear-type PLL for 10-Gb/s SONET application", *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 1353 - 1359, Sept. 2000.

[27] M. Miller, "Transition Density: What does it affect and why is it explicitly specified within   the   LeCroy   SDA", http://www.lecroy.com/tm/Library/WhitePapers/PDF/WP_TechBrief_Trans_Den.p df

[28] M. Mansuri, C-K.K. Yang, "Jitter optimization based on phase-locked loop design parameters," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, Nov. 2002, pp. 1375 –1382.

[29] V. Stojanović, A. Ho, B. Garlepp, F. Chen, J. Wei, E. Alon, C. Werner, J. Zerbe, M.A. Horowitz, "Adaptive Equalization and Data Recovery in a Dual-Mode (PAM2/4) Serial Link Transceiver," *IEEE Symposium on VLSI Circuits*, June 2004, pp. 348-351.

[30] E. Alon, V. Stojanović, M.A. Horowitz, "Circuits and Techniques for High-Resolution Measurement of On-Chip Power Supply Noise," *IEEE Symposium on VLSI Circuits*, June 2004, pp. 102-105.

[31] M. Horowitz, A. Chan, J. Conbrunson, J. Gasbarro, T. Lee, W. Leung, W. Richardson, T. Thrush, Y. Fujii, "PLL design for a 500 MB/s interface," *IEEE International Solid-State Circuits Conference,* Feb. 1993, pp. 160-161.

[32] S. Sidiropoulos, M. Horowitz, "A semidigital dual delay-locked loop," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, Nov. 1998, pp. 1683-1692.

[33] K. K.-Y. Chang, W. Ellersick, T.-S. Chuang, S. Sidiropoulos, M. Horowitz, "A 2 Gb/s/pin CMOS asymmetric serial link," *VLSI Circuits Symposium*, June 1998, pp. 216-217.

[34] R. Walker *et al*, "A 10 Gb/s Si-bipolar TX/RX chipset for computer data transmission," *IEEE International Solid-State Circuits Conference*, Feb. 1998.

[35] R. Walker, "Designing Bang-Bang PLLs for Clock and Data Recovery in Serial Data Transmission Systems," in *Phase-Locking in High-Performance Systems*, B. Razavi, Ed. New Jersey: IEEE Press, 2003, pp. 34-45.

[36] J. Kim, *Design of CMOS Adaptive-Supply Serial Links, Ph.D. Thesis*, Stanford University, December 2002.

[37] N. Da Dalt, "A Design-Oriented Study of the Nonlinear Dynamics of Digital Bang-Bang PLLs," *IEEE Transactions on Circuits and Systems I*, Jan. 2005.

[38] Y. Choi, D. Jeong, W. Kim, "Jitter Transfer Analysis of Tracked Oversampling Techniques for Multigigabit Clock and Data Recovery," *IEEE Transactions on Circuits and Systems II*, Nov. 2003.

[39] J. Lee, K. Kundert, B. Razavi, "Analysis and Modeling of Bang-Bang Clock and Data Recovery Circuits," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, Sept. 2004, pp. 1571–1580.

[40] S. Norsworthy, R. Schreier, G. C. Temes, *Delta-Sigma Data Converters*, IEEE Press 1997.

[41] H. Lee, A. Bansal, Y. Frans, J. Zerbe, S. Sidiropoulos, M. Horowitz, "Improving CDR Performance via Estimation," *IEEE Solid-State Circuits Conference,* Feb. 2006, pp. 332-333.

[42] M.-J.E. Lee, W.J. Dally, J. Poulton, T. Greer, J. Edmondson, R. Farjad-Rad, N. Tiaq, R. Rathi, R. Senthinathan, "A second-order semi-digital clock recovery circuit based on injection locking," *IEEE Solid-State Circuits Conference,* Feb. 2003, pp. 74-75

[43] IEEE P802.3ah, "Media Access Control Parameters, Physical Layers and Management Parameters for subscriber access networks", Draft 1.732.

[44] ITU-T Recommendation G.984.2, "Gigabit-capable passive optical networks (GPON): Physical media dependent (PMD) layer", Mar. 2003.

[45] C. Qiao, "Labeled Optical Burst Switching for IP-over-WDM Integration," *IEEE Communications Magazine*, vol. 38. no. 9, Sep. 2000, pp. 104-114.

[46] I.M. White, M.S. Rogge, K. Shrikhande, L.G. Kazovsky, "A summary of the HORNET project: a next-generation metropolitan area network", *IEEE Journal on Selected Areas in Communications*, Vol. 21, Issue: 9, pp. 1478 – 1494.

[47] C.-K. Yang, M. Horowitz, "A 0.8um CMOS 2.5Gb/s oversampling receiver and transmitter for serial links," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 12, Dec. 1996, pp. 2015-2023.

[48] B. Kim *et al*, "A 30-MHz Hybrid Analog/Digital Clock Recovery Circuit in 2μm CMOS," *IEEE Journal of Solid-State Circuits*, Dec. 1990.

[49] M. Banu, A. Dunlop, "A 660 Mb/s CMOS clock recovery circuit with instantaneous locking for NRZ data and burst-mode transmission", *IEEE Solid-State Circuits Conference,* Feb. 1993.

[50] G.-K. Dehng, J.-W. Lin, S.-I. Liu, "A Fast-Lock Mixed-Mode DLL Using a 2-b SAR Algorithm", *IEEE Journal of Solid State Circuits,* Oct. 2001, pp. 1464-1471.

[51] C.-K. Yang, R. Farjad-Rad, M. Horowitz, "A 0.6um CMOS 4Gb/s Transceiver with Data Recovery using Oversampling," *IEEE Symposium on VLSI Circuits*, June. 1997, pp. 71-72.

[52] H. Partovi *et al*, "Data Recovery and Retiming for the Fully Buffered DIMM 4.8Gb/s Serial Links," *IEEE Solid-State Circuits Conference,* Feb. 2006, pp. 336-337.

[53] H. Lee, C. H. Yue, S. Palermo, K. W. Mai, M. Horowitz, "Burst mode packet receiver using a second order DLL," *IEEE Symposium on VLSI Circuits*, June 2004, pp. 264-267.

[54] J. Kovacs, "Analyze PLLs with Discrete Time Modeling," *Microwaves & RF*, May 1991, pp. 224-229.

[55] M. Mansuri, D. Liu, C.-K. Yang, "Fast frequency acquisition phase-frequency detectors for Gsamples/s phase-locked loops," *IEEE Journal of Solid-State Circuits*, Oct. 2002, pp. 1331-1334.

[56] M.J. Pelgrom, A.C.J. Duinmaijer, A.P.G. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, Oct. 1989, pp. 1433-1439.

[57] S. Sidiropoulos, D. Liu, J. Kim, G. Wei, M. Horowitz, "Adaptive Bandwidth DLLs and PLLs using Regulated Supply CMOS Buffers," *IEEE Symposium on VLSI Circuits*, June 2000, pp. 124-127.

[58] D. Dobberpuhl, "The design of a high performance low power microprocessor," *IEEE Int'l Symposium on Low Power Electronics and Design Dig. Tech. Papers*, Aug. 1996, pp. 11-16.

[59] G. Wei, J. Kim, D. Liu, S. Sidiropoulos, M. Horowitz, "A Variable-Frequency Parallel I/O Interface with Adaptive Power-Supply Regulation," *IEEE Journal of Solid-State Circuits*, vol. 35, no.11, Nov. 2000, pp.1600-1610.

[60] S. Sidiropoulos, C-K. K. Yang, M. Horowitz, "High-Speed Inter-Chip Signaling," in *Design of High-Performance Microprocessor Circuits*, A. Chandrakasan *et al*, Ed. Piscataway, NJ: IEEE Press, 2001, pp. 397–425.

[61] S. Sidiropoulos, *High Performance Inter-Chip Signaling, Ph.D. Thesis*, Stanford University, April 1998.

[62] D. Weinlader, *Precision CMOS Receivers for VLSI Testing Applications, Ph.D. Thesis*, Stanford University, Nov. 2001.

[63] E. Alon *et al*, "Replica Compensated Linear Regulators for Supply-Regulated Phase Locked Loops," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 2, Feb. 2006, pp. 413-424.

[64] M. Aoyama *et al*, "3Gbps, 5000ppm spread spectrum SerDes PHY with frequency tracking phase interpolator for serial ATA," *IEEE Symposium on VLSI Circuits*, June 2003.

[65] H.-R. Lee *et al*, "A Low-Jitter 5000ppm Spread Spectrum Clock Generator for Multi-channel SATA Transceiver in 0.18μm CMOS" *IEEE Solid-State Circuits Conference,* Feb. 2005, pp. 162-163.