UNIVERSITY OF CALIFORNIA

Los Angeles

Low-Power Low-Jitter On-Chip Clock Generation

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Electrical Engineering

by

Mozhgan Mansuri

2003

The dissertation of Mozhgan Mansuri is approved.

Majid Sarrafzadeh

Mau-Chung Frank Chang

Behzad Razavi

Chih-Kong Ken Yang, Committee Chair

University of California, Los Angeles
2003

Dedication

To my parents

Table of Contents

List of Figures

List of Tables

Acknowledgments

During my study and research at UCLA, I have been extremely blessed by God to meet and collaborate with so many people that were so supportive and helpful in this research.

I would like to deeply thank my advisor, professor Ken Yang, for his continuos support, encouragement and help. He has been my best research advisor and it has been a privilege collaborating and working with him these past four years. He has been source of ideas and knowledge, yet, his wisdom allowed me to direct my research successfully.

I would also like to thank professor Behzad Razavi for his support and useful technical discussions. I would like to extend my appreciation to him, professor Frank Chang and professor Majid Sarrafzadeh for serving on my committee and providing me with their fruitful comments.

I would like to express my deepest appreciation to my family. In particular, I am always indebted to my parents for their constant support, love and patience. Without their continued support, I would have not accomplished this effort. I would like to thank my two brothers for being so supportive and encouraging throughout years of my study.

It has been a pleasure to work with so many talented people in UCLA. I wish to thank, in particular, Siamak Modjtahedi, who generously provided me with his help and useful discussions, Jackie Wong and Hamid Hatamkhani, with whom I collaborated in the design of low-power links, and Ali Hadiashar, who helped me with the development of run-time algorithm for jitter optimization. I would also like to thank Dean Liu for his collaboration and great help on the design of phase-frequency detectors.

Also, I am greatly thankful to my friends for their constant support and friendship. I would like to thank, in particular, Hamid Rafati, Esmaeil Heidari, Rahim Bagheri, Ali Karimi, Omid Oliaei, Alireza Razzaghi, Vladimir Stojanovic, Saeed Chehrazi and Pejman Kalkhoran for countless discussions.

I wish to thank National semiconductor, Intel corporation and UCMicro 02-102 for fabrication and their support. Also I would like to thank Makoto Murata for his great help in wire bonding and Dorothy Tarkington for her wonderful help in purchasing the lab equipment and components.

VITA

1972	Born, Tehran, Iran
1995	B.Sc., Electrical Engineering Sharif University of Technology Tehran, Iran
1997	M.Sc., Electrical Engineering Sharif University of Technology Tehran, Iran
1997-1999	Design Engineer KCR company Tehran, Iran
1999-2003	Graduate Researcher Department of Electrical Engineering University of California, Los Angeles

PUBLICATIONS AND PRESENTATIONS

M. Mansuri and CK.K. Yang, “A Low-Power Low-Jitter Adaptive Bandwidth PLL and Clock Buffer,” Submitted for publication, IEEE, Journal of Solid-State Circuits, November 2003

M. Mansuri, A. Hadiashar, and CK.K. Yang, “Methodology for On-chip Adaptive Jitter Minimization in Phase-Locked Loops,” Submitted for publication, IEEE, Journal of Transactions on Circuits and Systems II, November 2003

KL.J. Wong, M. Mansuri, H. Hatamkhani and CK.K. Yang, “A 27-mW 3.6-Gb/s I/O Transceiver,” Proceedings of Symposium on VLSI Circuits, pp. 99-102, Japan, June 2003

M. Mansuri and CK.K. Yang, “A Low-Power Low-Jitter Adaptive Bandwidth PLL and Clock Buffer,” ISSCC Digest of Technical Papers, pp. 430-431, San Francisco, CA, February 2003

M. Mansuri and CK.K. Yang, “Jitter Optimization Based on Phase-Locked Loop Design Parameters,” IEEE, Journal of Solid-State Circuits, vol. 37, no. 11, pp. 1375-1382, November 2002

M. Mansuri, D. Liu and CK.K. Yang, “Fast Frequency Acquisition Phase-Frequency Detectors for GSa/s Phase-Locked Loops,” IEEE, Journal of Solid-State Circuits, vol. 37, no. 10, pp. 1331-1334, October 2002

M. Mansuri and CK.K. Yang, “Jitter Optimization Based on Phase-Locked Loop Design Parameters,” ISSCC Digest of Technical Papers, pp. 138-139, San Francisco, CA, February 2002

M. Mansuri, D. Liu and CK.K. Yang, “Fast Frequency Acquisition Phase-Frequency Detectors for GSa/s Phase-Locked Loops,” Proceedings of the European Solid-State Circuits Conference, Vienna, September 2001

ABSTRACT OF THE DISSERTATION

Low-Power Low-Jitter On-Chip Clock Generation

by

Mozhgan Mansuri

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2003

Professor Chih-Kong Ken Yang, Chair

Phase locked-loops (PLLs) are widely used to generate well-timed on-chip clocks in high-performance digital systems. Any timing jitter or phase noise significantly degrades the performance of these systems, especially as operating frequency increases. Switching activity in large digital systems introduces power supply or substrate noise which perturb the more sensitive blocks in a PLL, in particular, voltage-controlled oscillators (VCOs) and clock buffers.

Power dissipated by PLLs is often a small fraction of total active power. However, during sleep modes where the PLL must remain in lock, it can be a significant fraction of dissipated power. Also, for some applications such as high speed parallel links and distributed synchronous clocking, multiple PLLs are employed to minimize the timing uncertainty. Therefore, demand for low-power PLLs has been increasing. The low-power requirement makes the design of a low-jitter PLL even more challenging.

This research describes the design of a fully-integrated low-jitter PLL for low-power applications. To achieve the low-jitter performance, this work proposes jitter minimization methods at both system and circuit levels.

At the system level, this work investigates the effects of PLL design parameters, such as bandwidth and peaking in the frequency response, on timing jitter of PLL output clock. The analysis includes several common noise sources in a PLL and develops an intuition for selecting design parameters to obtain minimum output jitter based on the dominant noise source. The proposed PLL is equipped with digitally-controllable loop parameters that independently adjusts the loop parameters. Based on jitter analysis, a methodology for on-chip adaptive jitter minimization in PLLs is developed. The proposed method measures the output jitter and adjusts the PLL loop parameters toward minimizing the jitter by a closed loop control system. The experimental results verify the success of the proposed method in minimizing jitter to within 5ps of the minimum long-term peak-to-peak jitter.

At the circuit level, two new supply rejection techniques for VCOs and clock buffers are developed. Both methods demonstrate the delay sensitivity of ≤0.1%-delay/%-V_{DD} due to both static and dynamic supply noise. While the jitter performance is comparable with prior state-of-art work, the proposed VCO and clock buffer consume less power with smaller area than previous designs. The VCO is designed to operate over a wide frequency range and has a linear voltage-to-frequency gain. The PLL is designed with scaling loop parameters that track over a 10x frequency range of the VCO and allow the adaptive loop bandwidth. The PLL is implemented in 0.25-μm CMOS technology and consumes 10mW from a 2.5-V supply.

Chapter 1

Introduction

High-performance digital systems use clocks to sequence operations and synchronize between functional units and between ICs. Clock frequencies and data rates have been increasing with each generation of processing technology and processor architecture. Figure 1.1 shows the clock frequency versus technology generation according to 2002 ITRS[1]. Within these digital systems, well-timed clocks are generated with phase-locked loops (PLLs) and then distributed on-chip with clock buffers. The rapid increase of the systems' clock frequency poses challenges in generating and distributing the clock with low uncertainty and low power. This research presents innovative techniques at both system and circuit levels that minimize the clock timing uncertainty with minimum power and area overhead.

1. International technology roadmap for semiconductors

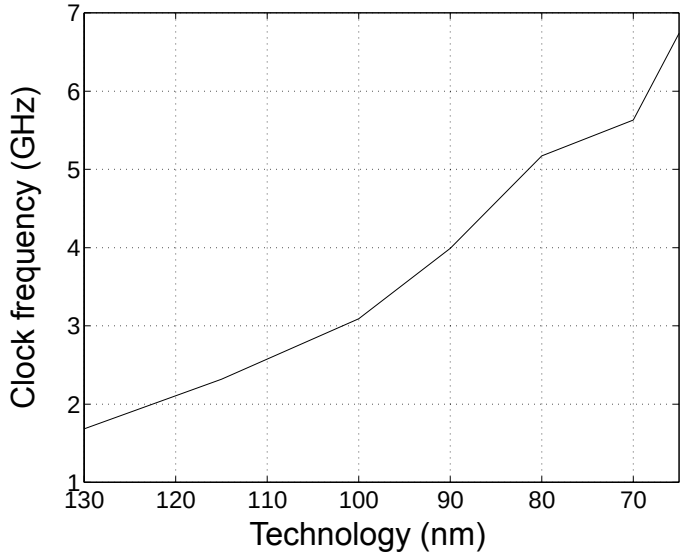


Figure 1.1 Clock frequency versus technology generation

1.1 Motivation

A PLL is essentially a feedback loop that locks the on-chip clock phase to that of an input clock or signal. Because the on-chip clock toggles a large capacitive load, a series of clock buffers efficiently increases the drive strength of the PLL output to drive the load. High-performance PLLs and clock buffers are widely used within a digital system for two purposes: clock generation, and timing recovery.

For clock generation, since off-chip reference frequencies are limited by the maximum frequency of a crystal frequency reference[1], a PLL receives the reference clock and multiplies the frequency to the multi-gigahertz operating frequency. The high-

1. Typically from tens of MHz to a few hundred of MHz

frequency clock is then driven to all parts of the chip. Timing recovery pertains to the data communication between chips. As data rates increase to satisfy the increase in on-chip processing rate, the phase relationship between the input data and the on-chip clock is not fixed. To reliably receive the high-speed data, a PLL locks the clock phase that samples the data to the phase of the input data.

Timing uncertainty impacts the performance of both applications. In order to maintain proper synchronization, large timing uncertainty would result in lower frequency of operation. Jitter is due to both intrinsic random noise (i.e. thermal noise and flicker noise), and systematic supply/substrate noise. Particularly in large digital systems, switching activity introduces power-supply or substrate noise which perturbs the PLL elements and clock buffers. Supply or substrate noise is the dominant source of jitter in these systems. This research focusses on the design of the most sensitive blocks in a PLL and clock buffer with high immunity to supply/substrate noise. The research also represents a powerful noise-filtering technique that minimizes jitter through adjusting the key loop parameters of a PLL based on the dominant noise source in the PLL.

The power performance of a PLL is a growing concern for many applications. Power dissipated by PLLs is often a small fraction of the total active power. However, it can be a significant fraction of the power dissipated in the sleep mode where the PLL must remain in lock. Also, as operating clock frequency of digital systems is increasing, the systems become less tolerable to clock skew. There is an increasing demand for using distributed phase-locking systems such as PLLs for applications such as high-speed parallel links [8]-[10] and distributed synchronous clocking [1]-[7]. In both applications,

multiple PLLs are employed to reduce the timing uncertainty across the entire system with the cost of power and area overhead due to each PLL.

The block diagram of a high-speed parallel link is shown in Figure 1.2. To

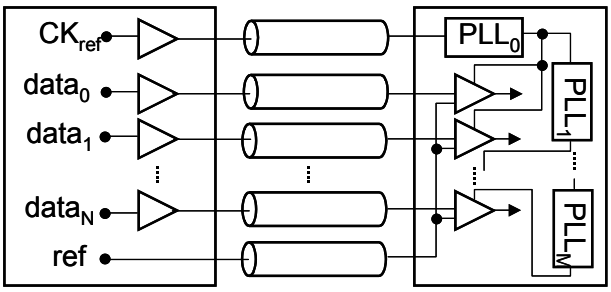


Figure 1.2 The block diagram of high-speed parallel link

increase the bandwidth, the architecture utilizes a set of parallel data signals. The synchronization is achieved through transmitting a reference clock with the parallel data signals. In the receiver, the on-chip clock is locally generated by multiple PLLs from the transmitted clock to recover the data. Locally distributed PLLs reduces the timing uncertainty and minimizes bit-error-rate (BER).

In conventional clock distribution networks, a well-aligned generated on-chip clock is distributed to many locations on the chip over a tree-like or grid-like network (Figure 1.3-(a) or (b)) with repeaters at necessary intervals. These networks are passive because it does nothing to reduce the uncertainty of the clock delivered to the sequential elements. As the clock frequency goes up, the number of required repeaters increases and shielding the interconnect segments becomes more difficult; thus, the timing uncertainty inevitably increases. Skew compensation [13]-[14] is used to reduce the delay mismatches

introduced during fabrication. However, this technique does not suppress jitter. A possible solution to the jitter accumulation problem is distributed synchronous clocking [1]-[7].

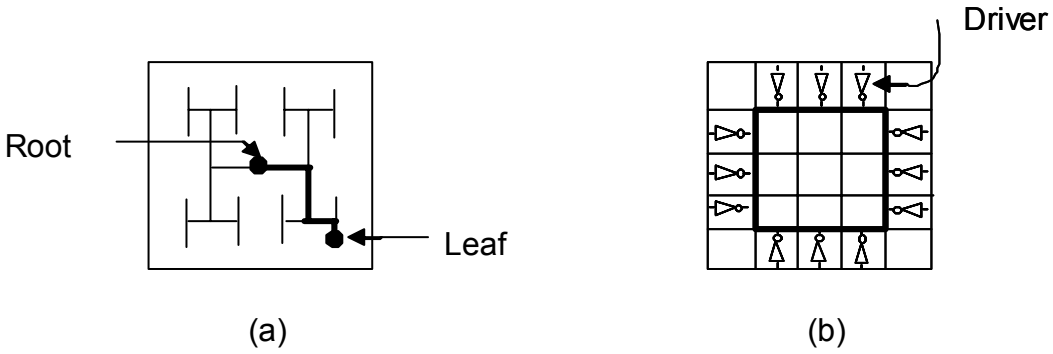


Figure 1.3 Clock distribution networks: (a) trees, (b) grids

In the distributed synchronous clocking, independent PLLs generate the clock signal at multiple nodes across the chip (Figure 1.4). Phase detectors (PDs) at boundaries produce error signals to adjust frequency of the node PLL. Within the tree, the clocks will be driven as sinusoidal signals without intermediate buffering; thus, the clocks at each terminal have a small swing due to resistive losses. With locally generated clocks, there are no full swing clock lines to couple in jitter. Also, since the clock is generated at each node, jitter does not accumulate with distance from the clock source.

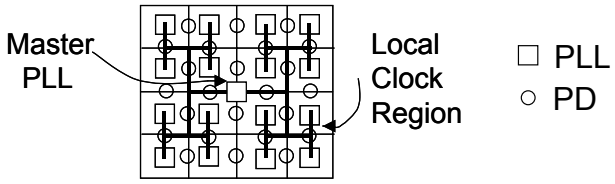


Figure 1.4 Distributed synchronous clocking with multiple PLLs

Since many of these phase-locking systems are required to be integrated within a single chip, the overall power and area overhead of a single phase-locking circuit are key constraints. A phase-locking system is not necessarily a PLL, however, it composes of similar components as a PLL. The power and area constraints make the design of a low-jitter PLL even more challenging due to the trade-off between low-jitter and low-power (and low-area) design techniques. This research presents new filtering techniques in the design of PLL components and loop parameters to overcome the low-power and low-area constraints. The proposed filtering techniques minimize the clock timing uncertainty while introducing minimum power and area overhead.

1.2 Organization

This thesis is composed of seven chapters. The functioning and components of a phase-locked loop (PLL) are described in Chapter 2. Then, the two common PLL architectures, delay-line based PLL (DLL) and oscillator-based PLL (PLL), are discussed and compared. The noise and power constraints associated with the design of a PLL are the next subject of the chapter. Noise minimization techniques at both system and circuit levels are the main subjects of the next four chapters.

At the system level, the timing jitter of the PLL output clock is minimized by proper design of PLL loop parameters, such as bandwidth and peaking in the frequency response. The jitter minimization relies on the fact that a PLL is a closed-loop system and filters each noise source in the PLL based on the transfer function from the correspondent noise source to the PLL output. For instance, a high-bandwidth PLL can track the phase of

a low-noise input clock and filter out voltage-controlled oscillator (VCO) noise. Conversely, a low-bandwidth PLL filters a noisy input clock. The goal is to explore an intuition for selecting design parameters to obtain the minimum output jitter based on the dominant noise source. Chapter 3 reviews jitter definitions and major timing jitter sources in a PLL. The relationship between the jitter, the power spectral density of each noise source and the correspondent PLL noise transfer function is extracted next. Based on the extracted equations, the sensitivity of jitter to PLL bandwidth and peaking in loop frequency response is derived. Finally, a PLL with tunable loop parameters is used to experimentally minimize jitter and verify the jitter analysis.

The proper design of PLL loop parameters for minimum output jitter performance requires knowledge of the dominant noise source in the PLL. For many systems, the magnitude of the noise sources are not well known which makes the design of loop parameters complicated. Chapter 4 develops a methodology for on-chip adaptive jitter minimization in PLLs. The algorithm functions during system operation and minimizes jitter as noise source conditions vary. The chapter shows that since the total jitter has only one minimum that is global, a gradient-descent algorithm suffices to converge to the minimum. The chapter, then, describes the circuit components necessary that dynamically measure and minimize jitter.

In addition to jitter minimization technique at the system level, this research explores designs of low-noise PLL components. Although both device noise and supply/ substrate noise are present, supply/substrate noise is the dominant noise source in digital systems which perturbs the most sensitive blocks such as voltage-controlled oscillators

(VCOs) and clock buffers. To achieve a high-noise performance requires design of VCOs and clock buffers with high immunity to supply/substrate noise.

Design of the PLL components are discussed in Chapter 5, starting with the design of a VCO. The new noise filtering technique is presented that achieves similar noise performance with improved power and area performance comparing with state-of-the-art designs. The chapter presents a self-biased charge-pump current and loop filter, next, that allows the PLL to operate over a wide frequency range with an adaptive bandwidth in a constant phase margin. The design of a high-performance phase-frequency detector is introduced next that has lower power consumption and larger lock-in range than conventional PFDs.

Clock buffers with improved supply sensitivity of buffer elements are introduced in Chapter 6. The design goal is to compensate the supply-induced delay variation with an improved dynamic behavior while introducing minimum power, area and delay overhead. The noise performance of the compensated buffer is verified with experimental results.

Chapter 2

Phase-Locked Loop Fundamentals

Phase-locked loops (PLLs) generate well-timed on-chip clocks for various applications such as clock-and-data recovery, microprocessor clock generation and frequency synthesizer. The basic concept of phase locking has remained the same since its invention in the 1930s [20]. However, design and implementation of PLLs continue to be challenging as design requirements of a PLL such as clock timing uncertainty, power consumption and area become more stringent. A large part of this research focuses on the design of a PLL for high-performance digital systems. In order to understand the challenges and trade-off behind the design of such a PLL, this chapter provides a brief study of phase-locked loops.

Section 2.1 provides an overview of a PLL system and briefly discusses the basic concept of phase locking. PLL components for charge-pump PLLs are discussed in Section 2.2. Section 2.3 discusses and compares the two possible PLL architectures: (1) delay-line based PLL and (2) oscillator-based PLL. Study of loop characteristics and loop

parameters is the subject of Section 2.4. This section provides a simple analysis of the PLL loop dynamics as a function of the loop parameters.

The noise sources present in digital systems are discussed in Section 2.5. The chapter concludes with a summary of design goals and issues involved in the design of PLLs for high-performance digital systems.

2.1 PLL Definition

The basic block diagram of a PLL is shown in Figure 2.1. A PLL is a closed-loop feedback system that sets fixed phase relationship between its output clock phase and the phase of a reference clock. A PLL tracks the phase changes that are within the bandwidth of the PLL. A PLL also multiplies a low-frequency reference clock, CK_{ref}, to produce a high-frequency clock, CK_{out}.

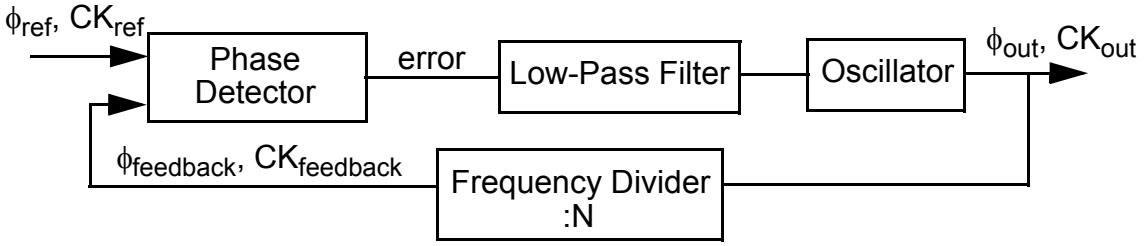


Figure 2.1: Basic block diagram of a PLL

The basic operation of a PLL is as follows. The phase detector (comparator) produces an error output signal based on the phase difference between the phase of the feedback clock and the phase of the reference clock. Over time, small frequency differences accumulate as an increasing phase error. The difference or error signal is low-

pass filtered and drives the oscillator. The filtered error signal acts as a control signal (voltage or current) of the oscillator and adjusts the frequency of oscillation to align $\phi_{feedback}$ with ϕ_{ref}. The frequency of oscillation is divided down to the feedback clock by a frequency divider. The phase is locked when the feedback clock has a constant phase error and the same frequency as the reference clock. Because the feedback clock is a divided version of the oscillator's clock frequency, the frequency of oscillation is N times the reference clock.

2.2 PLL Components

The block diagram of a charge-pump PLL is shown in Figure 2.2. A PLL comprises of several components: (1) phase or phase-frequency detector, (2) charge-pump current, (3) loop filter, (4) voltage-controlled oscillator, and (5) frequency divider. The functioning of each block is briefly described below.

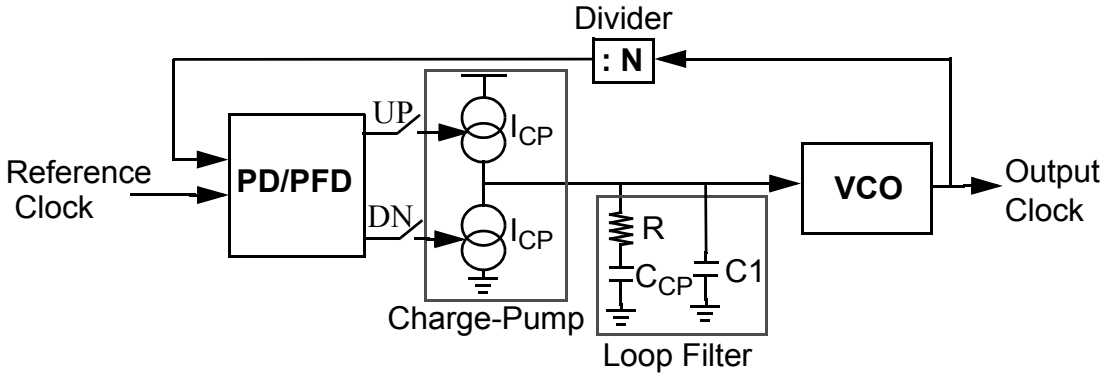


Figure 2.2: Individual blocks in a PLL

2.2.1 Voltage-Controlled Oscillator (VCO)

An oscillator is an autonomous system that generates a periodic output without any input. A CMOS ring oscillator shown in Figure 2.3 is an example of an oscillator. So that

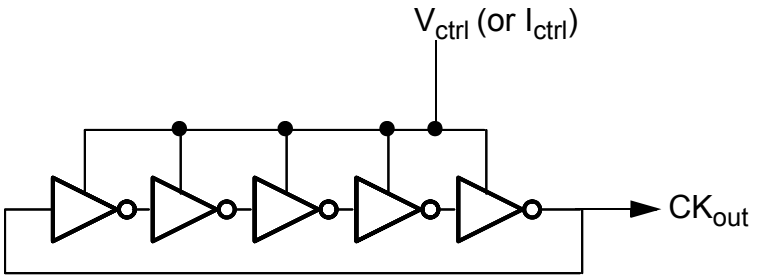


Figure 2.3: A five-stage ring oscillator

the phase of a PLL is adjustable, the frequency of oscillation must be tunable. In the example of an inverter ring oscillator, the frequency could easily be adjusted with controlling the supply (voltage or current) of inverters. The slope of frequency versus control signal curve at the oscillation frequency is called voltage-to-frequency (or current-to-frequency) conversion gain, K_{VCO}; $K_{VCO}=df_{VCO}/dV_{ctrl}$ evaluated at f_{VCO}. Since phase is the integral of frequency, the output phase of the oscillator is equal to $\phi_{VCO} = \int K_{VCO} \cdot V_{ctrl} \cdot dt$. In other words, the VCO in the frequency domain (s-domain), is modeled as $\frac{\phi_{VCO}}{V_{ctrl}}(s) = \frac{K_{VCO}}{s}$. Ideally, for the linear analysis to apply over a large frequency range, K_{VCO}, needs to be relatively constant.

2.2.2 Frequency Divider

The PLL reference clock is generated from a crystal. The crystals typically operate from tens to a few hundreds of MHz. On the other hand, VCOs for clocking and parallel link applications operate at a few GHz or even ten GHz. For proper functioning of the

phase detector or phase-frequency detector, discussed in the next section, a frequency divider divides down the VCO frequency to the frequency of the reference clock.

2.2.3 Phase Detector or Phase-Frequency Detector

The phase detector (PD) compares the phase difference between two input signals and produces an error signal that is proportional to the phase difference. In the presence of a large frequency difference, a pure phase detector does not always generate the correct direction of phase error. Phase error accumulates rapidly and can oscillate between phase error of $>180^{o}$and $<180^{o}$ from cycle to cycle. The average phase detector output contains little frequency information and no valuable phase information. Since the phase detector is insensitive to frequency difference at the input, upon start-up when the oscillator's frequency divided by N[1] is far from the reference frequency, the PLL may fail to lock. The problem is known as an inadequate *acquisition range* of the PLL. To remedy the problem, a phase-frequency detector (PFD) is used that can detect both phase and frequency differences. Figure 2.4 conceptually demonstrates the operation of a PFD for two cases: (a) the two input signals have the same frequency, and (b) one input has higher frequency than another input. In both cases, the DC contents of PFD's outputs, *UP* and *DN*, provide information about phase or frequency difference.

1. Loop divide ratio

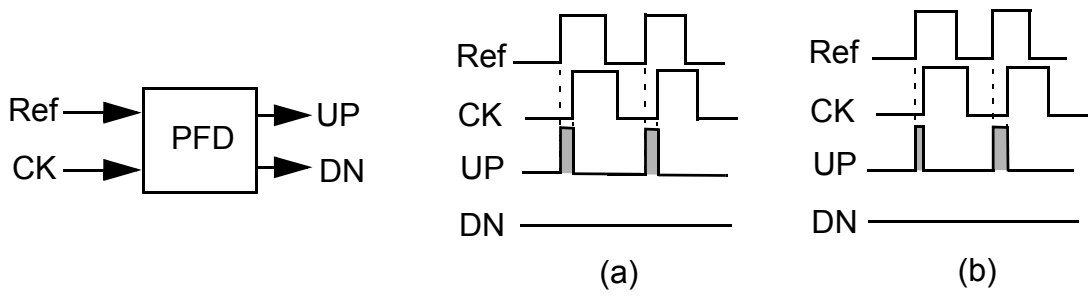


Figure 2.4: Operation of a PFD: (a) $f_{ref}=f_{CK}$, $\phi_{ref}\#\phi_{Ck}$ and (b) $f_{ref}>f_{CK}$

2.2.4 Charge-Pump and Loop Filter

The charge-pump circuit comprises of two switches that are driven with *UP* and *DN* outputs of PFD as shown in Figure 2.2. The charge-pump injects the charge into or out of the loop filter capacitor (C_{CP}). The combination of charge-pump and C_{CP} is an integrator that generates the average of *UP* (or *DN*) pulses. This average voltage adjusts the frequency of the subsequent oscillator circuit. Since the VCO introduces another integrator, the loop gain of a charge-pump PLL has two poles at origin; thus, the closed-loop system is unstable. To stabilize the system, a zero, $\omega_z = 1/RC_{CP}$, is introduced in the loop gain by adding a resistor, R, in series with C_{CP}.

The PFD, charge pump and filter are often modeled with a linear continuous-time model. In reality, the PFD acts as a pulse modulator system and drives the charge-pump for the duration of pulse width which is equal to PFD input phase difference, $\Delta\phi$. The actual phase response is not linear because phase is cyclical. Furthermore, the phase information is discrete, sampled at the clock reference frequency.

However, a linear continuous-time approximation is often used to model the stability of an operating point. The error due to approximation is negligible if the PLL bandwidth is 1/10th or smaller than the reference clock frequency [79]. The reference frequency determines the rate that PFD output is refreshed. With a linear approximation, V_{ctrl} is equal to: $\frac{V_{ctrl}}{\Delta\phi}(s) = \frac{I_{CP}}{2\pi} \cdot F(s)$ where F(s) is the transfer function of the loop filter and is equal to: $F(s) = \frac{1}{C_{CP}s} \cdot (1 + RC_{CP}s)$, ignoring C_1 in Figure 2.2.

2.3 Delay-locked Loops

In the previous section, the PLL components for an oscillator-based PLL architecture are discussed. An alternative to an oscillator-based PLL is a delay-line-based PLL or a delay-locked loop (DLL). A DLL is similar to a PLL except that a variable delay line replaces the oscillator [21]. Thus, phase is the only state variable in a DLL while both phase and frequency are the state variables in a PLL. The basic DLL building blocks are shown in Figure 2.5, similar to that of a PLL. A phase detector (PD) measures the phase

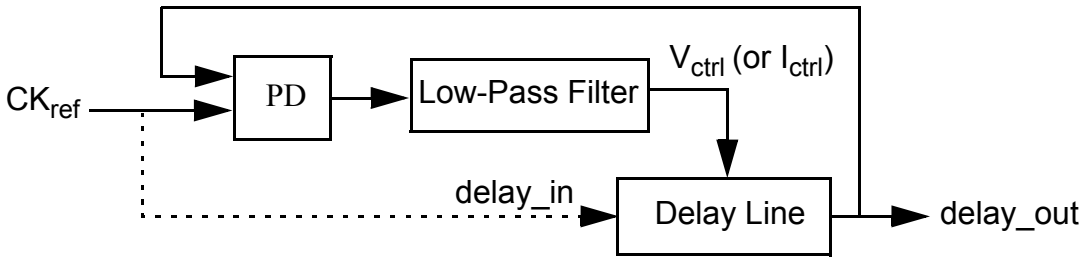


Figure 2.5: Block diagram of a DLL

difference between the reference clock and the delay-line output. The error signal is low-

pass filtered to produce the control signal that adjusts the delay of the delay line. Note that the delay-line input can be a separate external clock instead of the CK_{ref}.

To eliminate the phase offset in a DLL, the filter is an integrator. DLL with only a single pole is unconditionally stable. Only at loop bandwidths close to the reference frequency, where the loop delay and the sampling nature of the PD degrade phase margin, is the stability a concern. In response to a noise perturbation, a PLL accumulates phase error before correcting the error because the output phase is an integration of the frequency change. In contrast, a DLL does not accumulate the phase error and corrects the error by the time constant of the loop.

Although, a simple loop characteristic of a DLL is desirable, a DLL has its own limitations. First, for clock generation, only one input clock is available so the clock is used as the input to the delay line as well as the phase detector. Therefore, any high-frequency jitter at the reference clock directly passes through the delay line to the DLL output. Low-frequency jitter is tracked. This configuration results in an all-pass response to any phase variations in a reference clock. Secondly, it is not as easy to multiply the reference frequency [65]-[66] as a PLL. Third, delay lines usually have a finite delay range. The limited delay range causes the loop to not lock properly. In contrast, a PLL can filter out a noisy reference clock by lowering the PLL bandwidth. A PLL can achieve a wide frequency range, provided that the VCO is designed to operate over a wide range. The output frequency can be any frequency different from the reference clock frequency. The advantages of a PLL over a DLL motivates us to focus on a design of a PLL in this

research. Nevertheless, the circuits and jitter reduction techniques discussed in following chapters are applicable to DLLs because PLL and DLL architectures share many similar components and loop characteristics.

2.4 Loop Characteristics

This section describes the dynamic behavior of the entire PLL. The s-domain presentation of each loop element, discussed in Section 2.2, is depicted within each block in Figure 2.6. The open-loop transfer function can be written as

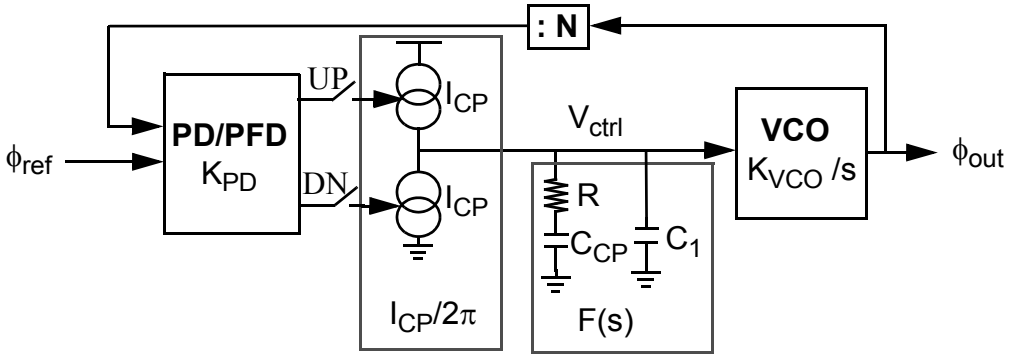


Figure 2.6: Representation of PLL individual blocks in s-domain

$H_{open}(s) = K_{PFD} \cdot I_{CP}/2\pi \cdot F(s) \cdot K_{VCO}/s$ where K_{PFD} is phase-frequency detector gain, F(s) is the loop filter transfer function and K_{VCO} is the conversion gain of the VCO. The open-loop transfer function for a second-order PLL (ignoring C_1 in the loop filter) is equal to:

$$H_{open}(s) = K_{PFD} \cdot \frac{I_{CP}}{2\pi \cdot C_{CP}} \cdot (1 + RC_{CP}s) \cdot \frac{K_{VCO}}{s^2} \qquad \text{‹2.1›}$$

This transfer function has two poles at origin and one compensating zero that guarantees the closed-loop stability. Including the third pole, the open-loop transfer function is equal to:

$$H_{open}(s) = K_{PFD} \cdot \frac{I_{CP}}{2\pi \cdot (C_{CP} + C_1)} \cdot (1 + RC_{CP}s) \cdot \frac{K_{VCO}}{s^2 \cdot [1 + R(C_{CP} \| C_1)s]} \qquad \langle 2.2 \rangle$$

The magnitude and phase of the open-loop transfer functions for a second and third-order PLL are shown in Figure 2.7. $\omega_z = \frac{1}{RC_{CP}}$ and $\omega_{p3} = \frac{1}{R(C_{CP} \| C_1)}$ indicate

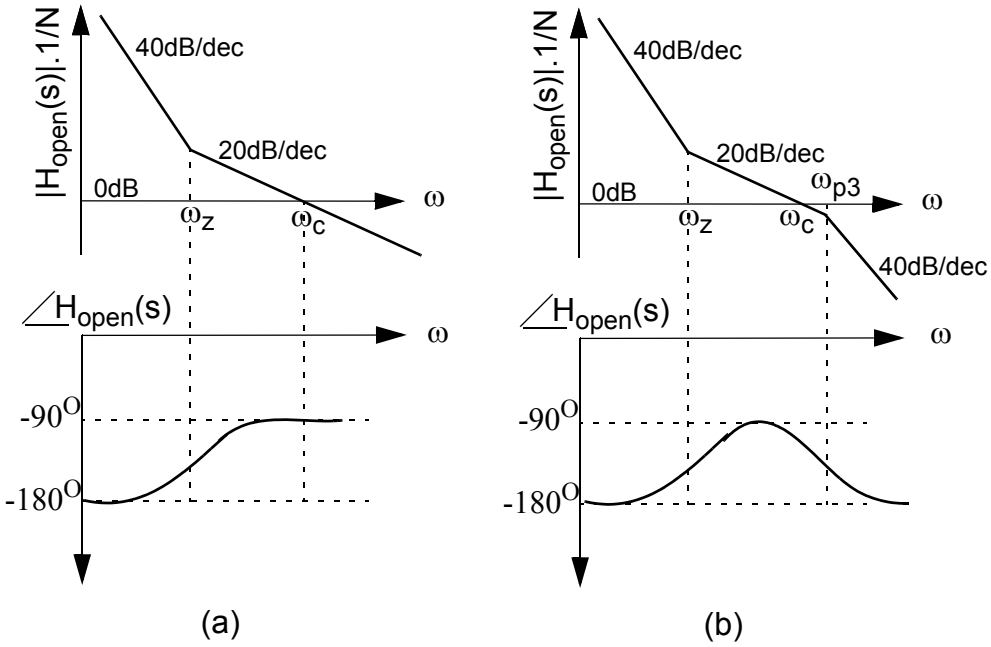


Figure 2.7: Magnitude and phase of the open-loop transfer function for (a) a second-order PLL, (b) a third-order PLL

the zero and third pole frequency, respectively. ω_c is the open-loop unity gain frequency. Without a compensating zero, neither a closed-loop second-order nor a closed-loop third-

order PLL is stable. The zero locus for an ideal second-order loop is not critical for stability, in contrast to a third-order (or higher order) PLL.

To understand the effect of the zero and other PLL parameters on the closed-loop behavior of the PLL, the closed-loop transfer function of a PLL from input phase to output phase is calculated:

$$\frac{\phi_{out}}{\phi_{in}}(s) = H_{closed}(s) = \frac{H_{open}(s)}{1 + H_{open}(s) \cdot 1/N} \quad \text{‹2.3›}$$

For a second-order PLL, the closed-loop transfer function is equal to:

$$\frac{\phi_{out}}{\phi_{in}}(s) = \frac{K_{Loop} \cdot (1 + RC_{CP}s)}{s^2 + (K_{Loop}/N)RC_{CP}s + K_{Loop}/N} \quad \text{‹2.4›}$$

K_{Loop} is the *loop gain* and is equal to:

$$K_{Loop} = K_{PFD} \cdot K_{VCO} \cdot I_{CP}/(2\pi C_{CP}) \quad \text{‹2.5›}$$

The closed-loop transfer function from the input phase to the output phase (Equation 2.4) is a low-pass filter. This low-pass behavior of a PLL is desirable because it rejects input noise frequencies higher than the PLL bandwidth. Similarly, the closed-loop transfer function from the VCO control voltage, V_{ctrl}, to the output phase is calculated:

$$\frac{\phi_{out}}{V_{ctrl}}(s) = \frac{K_{VCO} \cdot s}{s^2 + (K_{Loop}/N)RC_{CP}s + K_{Loop}/N} \quad \text{‹2.6›}$$

This closed-loop transfer function is a band-pass filter. This band-pass filter rejects internal noise coupled into V_{ctrl} within the PLL bandwidth.

Filtering out noise sources by the closed-loop behavior of the PLL forms the baseline for jitter analysis discussed in Chapter 3. Noise of the PLL's output clock can be

optimally filtered by adjusting the loop bandwidth and peaking in frequency response based on the dominant noise source. The loop bandwidth and peaking are adjustable by varying loop parameters.

The natural frequency, ω_n, and damping factor, ζ[1], are equal to $\omega_n = \sqrt{\frac{K_{Loop}}{N}}$ and $\zeta = \frac{\omega_n}{2 \cdot \omega_z}$, respectively. Natural frequency is proportional to square-root of the loop gain. Since K_{PFD}, K_{VCO} and C_{CP} are typically design constant parameters, the natural frequency is proportional to square-root of the charge-pump current (Equation 2.5). Damping factor is inversely proportional to zero frequency. By adjusting the zero frequency (typically through the loop filter resistor, R) and charge-pump current, ζ and ω_n can be adjusted. In other words, the bandwidth and peaking in frequency response are adjustable by varying ω_z and I_{CP}. The closed-loop frequency response for different values of ω_z in constant I_{CP} are shown in Figure 2.8-(a). As ω_z decreases the loop bandwidth increases while the peaking in frequency response decreases.

For a third-order PLL with sampling/feedback delay, decreasing the zero frequency increases the bandwidth. However, the peaking in frequency response increases because of the phase margin degradation due to the third pole and delay. The phase margin (PM) for a third-order PLL with loop delay of t_{delay} can be approximated with [79]:

$$PM = \text{atan}\left(\frac{\omega_c}{\omega_z}\right) - \text{atan}\left(\frac{\omega_c}{\omega_{p3}}\right) - \frac{360^o}{2\pi} \cdot \omega_c \cdot t_{delay} \qquad \text{‹2.7›}$$

1. ω_n and ζ (for a second-order PLL) are calculated from

$$s^2 + 2\zeta\omega_n s + \omega_n^2 \equiv s^2 + \frac{K_{Loop}}{N} RCs + \frac{K_{Loop}}{N} \qquad \text{‹2.8›}$$

The closed-loop frequency response of a third-order PLL for different values of ω_z in constant I_{CP} are shown in Figure 2.8-(b).

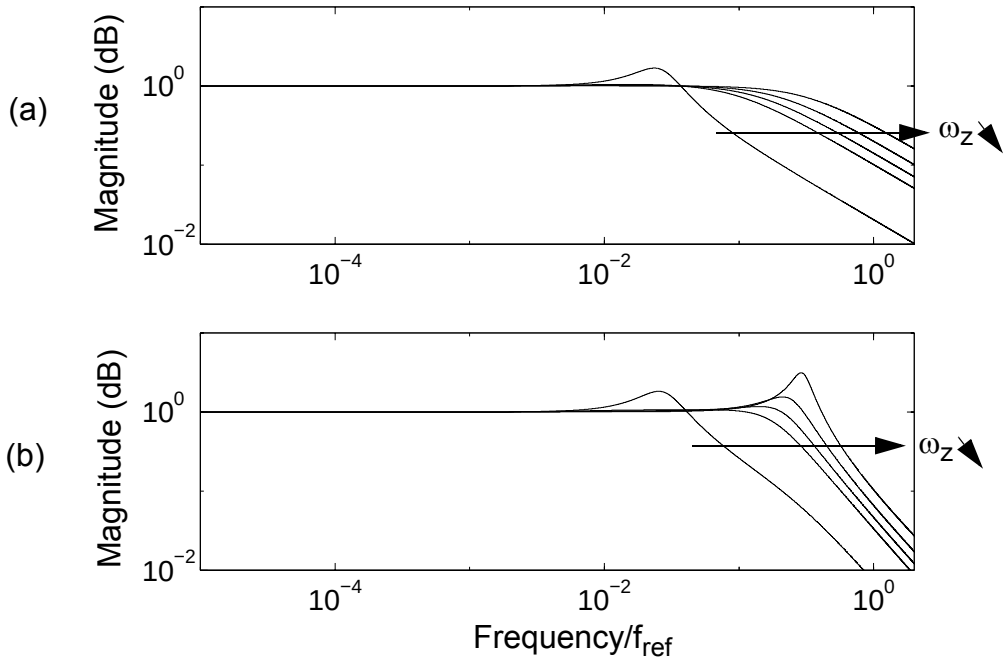


Figure 2.8: Closed-loop frequency response of: (a) an ideal second-order PLL, (b) a sampling third-order PLL

2.5 Noise and Power Considerations

The primary goal to design a PLL for high-performance digital systems is to generate an output clock with minimum timing uncertainty. The timing uncertainty arises from mismatches in devices and noise sources present in the system.

Device mismatches causes a static phase shift (or skew) in the PLL output clock from its desired phase. Skew can be minimized with a careful layout and increasing the device size [11]-[12]. Skew is generally less critical than jitter because, due to its static

nature, the system can compensate for the static errors [13]-[14]. Dynamic noise causes a random phase shift (or jitter) in the PLL output clock. The noise sources in a PLL are (1) device electronic noise such as thermal noise or flicker noise and (2) power-supply or substrate noise.

2.5.1 Device Electronic Noise

The device electronic noise at any individual blocks in a PLL perturbs the output clock timing. Numerous studies provide models that predict the jitter due to device noise. Most of these studies ([22]-[33]) focus on the modeling and prediction of jitter (or phase noise) due to VCOs. A few studies discuss the effect of noise in other PLL blocks such as PDs ([34]-[35]) and frequency dividers ([36]-[38]) on the PLL output jitter.

The previous studies also provide some guidance to reduce jitter. Some architectures demonstrate an improved jitter performance over the others. For example, resonant circuit-based VCOs (or harmonic oscillators) exhibit less jitter than relaxation oscillators (such as ring oscillators) [24]-[25]. The jitter due to device electronic noise generally demonstrates an inverse dependence upon power consumptions of PLL components ([22], [26] and [30]-[32]). Therefore, there is a trade-off between power consumption and jitter performance. For instance, Hajimiri in [26] demonstrates that the jitter of a ring oscillator with a constant frequency decreases as the number of stages and power increase.

2.5.2 Supply or Substrate Noise

Switching activities in digital systems introduces supply or substrate noise. The supply or substrate noise perturbs the sensitive blocks in a PLL such as VCO and clock buffer and leads to increased jitter.

Variation in supply or substrate voltage is coupled into the control voltage of a VCO which changes the VCO operating frequency. The change in the oscillation frequency of a VCO appears as a phase step in the input of the phase detector. The phase error accumulates jitter until it is corrected by the PLL. Therefore, supply or substrate noise causes jitter in a VCO which is persistent for the time duration equal to the time constant of the PLL.

For a clock buffer[1], supply or substrate noise varies the delay and introduces a phase shift at the output clock of the buffer. The impact of the supply voltage step for a clock buffer is considerably shorter lived. However, clock buffers are designed for power and area efficient capacitance driving and not supply rejection. The long chain of buffers needed in modern processors causes a significant transient phase shift at the output.

2.5.3 Noise Sensitivity Metric

The noise performance of VCOs and clock buffers are traditionally characterized with noise sensitivity metric. Noise sensitivity for a VCO is defined as a percentage of VCO clock frequency (or period) variation per percentage of supply voltage (or substrate)

1. Conventional clock buffers are composed of chain of CMOS inverters

variation; %-f_{VCO}/%-V_{DD}. Similarly, noise sensitivity for a clock buffer is defined as a percentage of the inverter's delay variation per percentage of supply voltage (or substrate) variation; %-delay/%-V_{DD}. One of the primary considerations in design of VCO and clock buffer is to minimize the noise sensitivity of these circuits to supply or substrate noise. For most digital systems, the supply or substrate noise does not exceed ±10-15% [49].

2.6 Summary

This chapter discussed the basic concept behind phase locking and in particular, a PLL. The operation of each PLL component is briefly explained which provides a framework to understand the design of a PLL as discussed in the following chapters. Two main architectures to design a PLL were discussed. A DLL has a simpler loop characteristic than a PLL and does not suffer from jitter accumulation presented in a PLL. However, a DLL passes input clock noise while a PLL low-pass filters the input noise. The frequency multiplication is easier in a PLL than a DLL. These two reasons motivate us to focus on the design of a PLL in this research.

The primary goal to design a PLL is to generate a low-jitter clock due to noise and mismatches. This chapter discussed sources of noise. It also showed that there is a trade-off between jitter, power consumption, and area.

To reduce noise, this research first studies the effect of loop parameters in filtering out noise sources in a PLL. Chapter 3 develops a simple yet accurate model that predicts the output jitter and provides an intuition toward optimum loop parameter design for

minimum jitter. To further adaptively minimize the jitter, Chapter 4 discusses a methodology for on-chip adaptive jitter optimization.

Supply or substrate noise is a dominant noise source in large digital systems. This research presents innovative filtering techniques at circuit level that achieve the noise performance comparable to prior work but with lower power and area. The design of such a high-performance PLL components is the subject of Chapter 5. The design of low-jitter clock buffer with minimum power, area and delay overhead is discussed in Chapter 6.

Chapter 3

Jitter Optimization Based on PLL Design Loop Parameters

Timing jitter has been the subject of numerous studies ([22]-[39]) which provide many models to predict the jitter of individual blocks in a PLL, in particular, different types of voltage controlled oscillators (VCOs) due to device noise and supply/substrate noise. While most of previous work focuses on jitter study of individual blocks, there has been done less work on modeling the overal jitter at PLL output clock ([22] and [43]-[46]). This research extends the previous work by investigating the effect of PLL parameters such as bandwidth and damping factor toward minimizing output clock jitter for various noise sources.

The common design practice for systems with low-noise input clock is to critically-damp or overdamp a PLL to minimize peaking in jitter transfer function and to design the loop with the highest possible bandwidth to eliminate the effects of noise

sources at the output. Very low bandwidth and high damping factor are commonly used to filter a noisy input clock with a clean oscillator within the PLL. By understanding the sensitivity of jitter to loop parameters, we can refine these common practices in designing low-jitter PLLs. Section 3.1 reviews the definitions of timing jitter. The brief study of the previous work on jitter optimization is discussed in Section 3.2. The noise sources in a PLL are the subject of the next section. Section 3.4 extracts the relationship between the overall rms jitter at the PLL output clock, the power spectral density of each noise source and the correspondent PLL noise transfer function. In Section 3.5, the sensitivity of jitter to PLL damping factor and bandwidth is first derived for second-order loops and then extended to third-order loops. The sensitivity of jitter to loop parameters is studied for all primary noise sources in a PLL. Section 3.6 describes the design of a tunable PLL that is used to minimize jitter and to verify our analysis. Finally, the experimental methods and results that verify the jitter analysis are given in Section 3.7.

3.1 Definitions of Jitter

Phase jitter is defined as the standard deviation, $\sigma_{\Delta\phi}$, of the phase difference between the first cycle and m^{th} cycle of the clock (Figure 3.1). Timing jitter can be

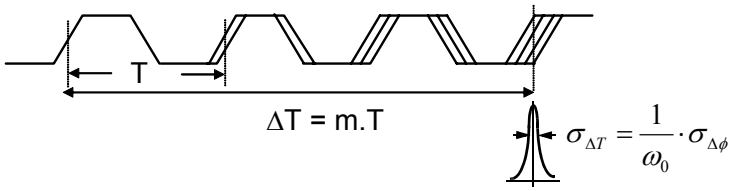


Figure 3.1: Timing jitter

expressed in terms of phase jitter by $\sigma_{\Delta T} = (T/2\pi) \cdot \sigma_{\Delta\phi} = (1/\omega_0)\sigma_{\Delta\phi}$ where the clock period, T, is $2\pi/\omega_0$. Timing jitter is called short-term jitter for small ΔT and long-term jitter as ΔT goes to infinity. The tracking jitter, σ_{tr}, is a commonly used metric for a PLL output clock. It is measured as the phase difference between a clean reference clock and the PLL output clock as shown in Figure 3.2. The tracking jitter is related to timing jitter by $\sigma_{tr} = \frac{\sigma_{\Delta T \to \infty}}{\sqrt{2}}$ at very large ΔT as shown in [22].

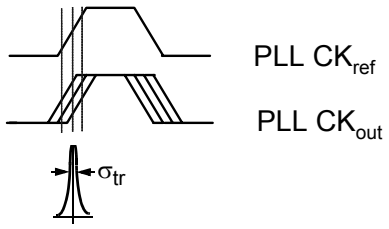


Figure 3.2: Tracking jitter at PLL output clock

Before starting with our jitter analysis in a PLL, a background on jitter optimization is discussed in the next section.

3.2 Previous Work

Prior research in [22] has shown that for an open loop VCO, jitter from random noise sources is proportional to the square root of measurement interval (ΔT), $\sigma_{\Delta T} \approx \kappa\sqrt{\Delta T}$, where the proportionality constant, κ, is a time-domain figure of merit which depends on the VCO design. For the case of a first-order PLL with bandwidth of f_{-3dB}, the long-term jitter of the output clock due to VCO noise is calculated in [22] as $\sigma_{\Delta T \to \infty} = \sigma_T = \kappa\sqrt{\frac{1}{2\pi f_{-3dB}}}$. The first-order loop roughly approximates an overdamped

second-order PLL. The short-term jitter of the first-order PLL is calculated in [40]. Although, [40] conceptually discusses jitter in higher-order loops and for different noise sources, it does not elaborate the impact of loop parameters on the output jitter. The previous work in [42] investigates the effect of only loop bandwidth on jitter due to VCO noise. Recently, the impact of the loop parameters on long-term jitter in an ideal second-order PLL is studied [41]. While this con-current work achieves similar closed-form equations for jitter as our analysis, it does not include higher-order effects of a PLL on jitter.

In this work, we extend the jitter analysis to different noise sources and to any second-order and third-order PLL loop parameters by including the delay and sampling nature of the loop in the analysis. The main goal of this analysis is to provide a simple, yet accurate model, to predict the short-term jitter as well as long-term jitter. The model should also provide designers with some guidance for proper design of the loop parameters for minimum jitter performance. First, we explain the primary noise sources in a PLL and then, we discuss the jitter analysis.

3.3 Noise Sources in a PLL

This research includes the three primary noise sources in a PLL: input clock noise (Vn_{in}), VCO noise (Vn_{VCO}), and clock buffer noise (Vn_{buf}) as shown in Figure 3.3. Open loop noise psd of a clock source is equal to $S_{\phi n_{in}}(f) = \frac{N_{Clk-in}}{f^2}$. $N_{in\text{-}CLK}$ is $K_0^2 \cdot \frac{e_n^2}{2}$ [22] where K_0 (Hz/V) represents the gain of the clock source oscillator and e_n ($V/\sqrt{Hz}$) is a white noise source. $N_{Clk\text{-}in}$ is related to κ with $\kappa = \frac{\sqrt{N_{Clk-in}}}{\omega_{in}/2\pi}$ [22]. Being a clock source

as well, the VCO has a similar noise that can be characterized using N_{vco} to represent the noise sources in the VCO[1]. For the buffer, open-loop noise psd is calculated by $S_{\phi n_{Buf}}(f) = \frac{N_{buf}}{f^2/f_{buf}^2 + 1}$ where f_{Buf} is the buffer 3-dB bandwidth (typically much larger than PLL loop bandwidth) and $N_{buf} = (K_{delay} \cdot 2\pi \cdot f_{VCO})^2 \cdot \frac{e_n^2}{2}$. K_{delay} (s/V) represents buffer delay variation to voltage noise. Multiplying K_{delay} by clock frequency (f_{VCO}) converts delay to phase variation due to noise.

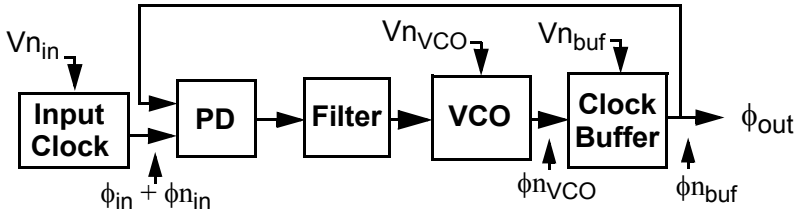


Figure 3.3: Noise sources in a PLL

The transfer functions from each noise source to the output of the PLL shape the noise. For example, the loop transfer function from the input phase to the output phase is a low-pass filter as seen from Equation 2.2. The lower the PLL loop bandwidth, the more strongly the PLL rejects the input clock noise. Next section discusses and extracts the relationship between the timing jitter at PLL output, each noise source and PLL loop parameters.

1. VCO noise spectrum falls as $1/f^2$ for a bounded frequency range. At lower frequencies, it falls as $1/f^3$, and at higher frequencies, it flattens out. Since low-frequency noise is suppressed by the PLL, and high-frequency noise is inconsequential to jitter (because it is so small), the $1/f^2$ approximation is a reasonable assumption.

3.4 Jitter Calculation Model

The goal is to relate the timing jitter at the PLL output clock to each noise source. As shown in Appendex A.1, the relationship between the timing jitter, $\sigma_{\Delta T}$ and noise power spectral density (psd), $S_\phi(f)$, is:

$$\sigma_{\Delta T}^2 = \frac{8}{\omega_0^2}\int_0^\infty S_\phi(f)\sin^2(\pi f \Delta T)df \qquad \langle 3.1 \rangle$$

At long delays $(\Delta T \to \infty)$, the expression is simplified as:

$$\sigma_T^2 = \frac{2}{\omega_0^2}R_\phi(0) = \frac{4}{\omega_0^2}\int_0^\infty S_\phi(f)df \qquad \langle 3.2 \rangle$$

Figure 3.4 graphically depicts Equation 3.1 and as shown, reducing the area under the phase noise psd lowers jitter at the output. The phase noise psd associated with each noise source is shaped as each noise is filtered out by the loop transfer function of the PLL from the correspondent noise source to the output.

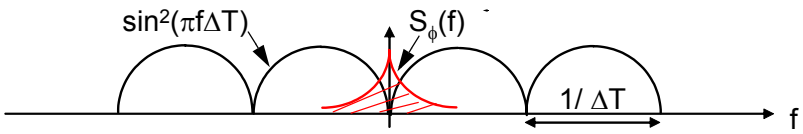


Figure 3.4: Timing jitter as a function of noise psd, $S_\phi(f)$

The filtering of the PLL on each input noise is included in the timing jitter by replacing the noise psd in Equation 3.1 (or Equation 3.2) with closed-loop noise psd. Under closed-loop condition, the total noise psd is calculated by

$$S_{\phi}(f) = S_{\phi n-closed}(f) = \sum_{i} S_{\phi n_i-open}(f) \cdot \left|Hn_i(j2\pi f)\right|^2 \quad \text{‹3.3›}$$

$\left|Hn_i(j2\pi f)\right|^2$ is the square magnitude of noise transfer function (NTF) from each input phase noise to PLL output phase, i.e. $\frac{\phi_{out}}{\phi n_i}(f) = Hn_i(j2\pi f)$. $S_{\phi ni-open}(f)$ indicates the open-loop phase noise of each noise source as calculated in Section 3.3.

Replacing the open-loop phase noise of each noise source, the total noise psd at the output is given by:

$$S_{\phi closed}(s) = \frac{N_{in-CLK}}{f^2} \cdot \left|Hn_{in}(j2\pi f)\right|^2 + \frac{N_{VCO}}{f^2} \cdot \left|Hn_{VCO}(j2\pi f)\right|^2 + \frac{N_{buf}}{f^2/f^2_{buf}+1} \cdot \left|Hn_{buf}(j2\pi f)\right|^2 \quad \text{‹3.4›}$$

Note that this analysis assumes white noise sources. The same analysis can be done for colored noise sources (such as supply and substrate noise) by replacing $\frac{e_n^2}{2}$ by $\frac{e_n^2}{2} \cdot \frac{1}{f^2/f^2_{noise}+1}$ where f_{noise} is the 3-dB bandwidth of the noise.

3.4.1 PLL Noise Transfer Function (NTF)

The second-order block diagram of a charge-pump PLL is shown in Figure 3.5. The loop transfer function from the input phase to the output phase was calculated in

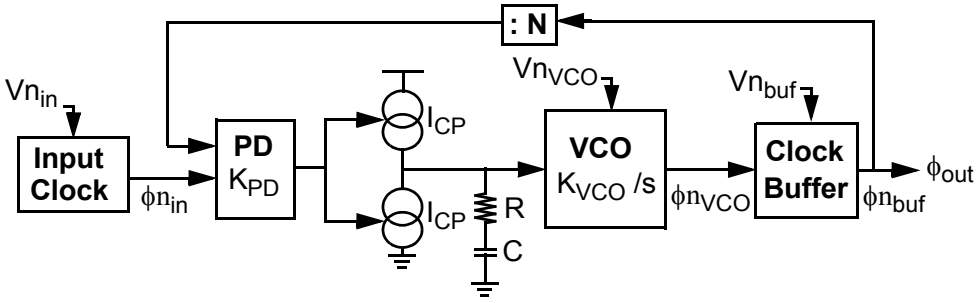


Figure 3.5: Block diagram of a second-order PLL

Section 2.4 (Equation 2.4). Similarly, the noise transfer functions from VCO[1] and clock buffer phase noise are calculated. The NTFs for three noise sources are[2]:

$$Hn_{In}(s) = \frac{\phi_{out}}{\phi n_{In}} = \frac{K_{Loop}RCs + K_{Loop}}{s^2 + K_{Loop}RCs + K_{Loop}} = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$
$$Hn_{VCO}(s) = Hn_{buf}(s) = \frac{\phi_{out}}{\phi n_{VCO,\,buf}} = \frac{s^2}{s^2 + K_{Loop}RCs + K_{Loop}} = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad \text{‹3.5›}$$

where $K_{Loop} = \frac{I_{CP}}{2\pi C}K_{PD}K_{VCO}$, $\omega_n = \sqrt{K_{loop}}$, and $\zeta = \sqrt{K_{loop}}RC/2$.

The NTFs for VCO and clock buffer noise are high-pass filters while the NTF for input clock noise is a low-pass filter. Multiplying each noise source's NTF with the transfer function of the correspondent block provides the overall transfer function from any voltage (or current) noise to the PLL output:

$$Tn_{In}(s) = \frac{\phi_{out}}{Vn_{In}} = (K_0 \cdot K_{Loop}) \cdot \frac{RCs + 1}{s \cdot (s^2 + K_{Loop}RCs + K_{Loop})}$$
$$Tn_{VCO}(s) = \frac{\phi_{out}}{Vn_{VCO}} = K_{VCO} \cdot \frac{s}{s^2 + K_{Loop}RCs + K_{Loop}} \quad \text{‹3.6›}$$
$$Tn_{buf}(s) = \frac{\phi_{out}}{Vn_{buf}} = \frac{1}{s/\omega_{buf} + 1} \cdot \frac{s^2}{s^2 + K_{Loop}RCs + K_{Loop}}$$

As seen from Equation 3.6, the overal loop transfer functions are low-pass filter, band-pass filter and high-pass filter for input clock noise, VCO noise and clock buffer noise, respectively. The overall transfer function for a clock buffer can be approximated as

1. For the VCO control voltage noise, the gain from the noise source to the VCO output phase is K_{VCO}. For power-supply noise, K_{VCO} is substituted with the gain from supply noise to VCO output phase.

2. The loop multiplication factor is one.

a high-pass filter because the buffer 3-dB bandwidth, $f_{buf} = \frac{\omega_{buf}}{2\pi}$, is typically much larger than the PLL bandwidth. Figure 3.6 demonstrates the overall transfer functions for three noise sources:

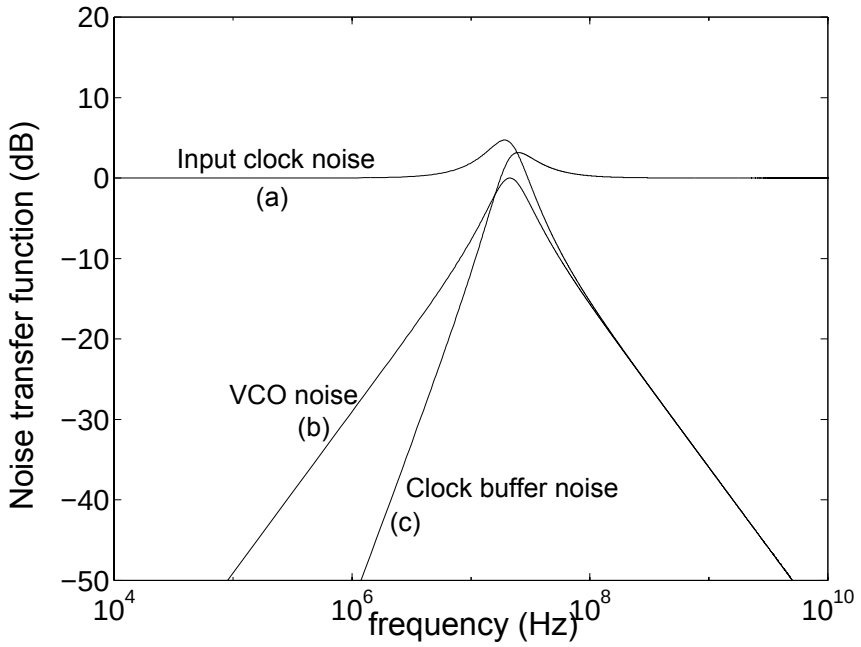


Figure 3.6: Loop transfer function from each noise source to PLL output

3.5 Output Jitter of PLL

The total jitter at the PLL output clock is calculated by substituting Equation 3.4 in Equation 3.1. The noise transfer functions in Equation 3.4 are substituted from Equation 3.5.

3.5.1 Jitter due to VCO Noise

To study the effect of each noise source on jitter, we first consider the VCO noise term in overal jitter equation:

$$\sigma_{\Delta T}^2 = \frac{8}{\omega_0^2}\int_0^{\infty}\left(\frac{N_{VCO}}{f^2}\cdot\left|Hn_{VCO}(j2\pi f)\right|^2\right)\sin^2(\pi f\Delta T)df \qquad \langle 3.7\rangle$$

We first study the jitter due to VCO noise in an ideal second-order PLL.

Jitter due to VCO Noise in an Ideal Second-Order PLL

By substituting the VCO NTF from Equation 3.5 into Equation 3.7:

$$\sigma_{\Delta T}^2 = \frac{4N_{VCO}}{\omega_0^2}\int_{-\infty}^{\infty}\left|\frac{s^2}{s^2+2\zeta\omega_n s+\omega_n^2}\right|_{s=j\omega}^2\frac{\sin^2(\pi f\Delta T)}{f^2}df \qquad \langle 3.8\rangle$$

The equation is simplified as follows (Appendex A.2):

$$\sigma_{\Delta T}^2 = \frac{4\pi^2 N_{VCO}}{\omega_0^2}\int_{-\infty}^{\infty}\left[x\left(t+\frac{\Delta T}{2}\right)-x\left(t-\frac{\Delta T}{2}\right)\right]^2 dt \qquad \langle 3.9\rangle$$

where x(t) is inverse Fourier transform of $\left.\frac{s}{s^2+2\zeta\omega_n s+\omega_n^2}\right|_{s=j\omega}$. For damping factors smaller and larger than one, the jitter expression is as follows (Appendex A.3):

$$\sigma_{\Delta T}^2 = \underbrace{\frac{4\pi^2 N_{VCO}}{\omega_0^2}}_{\kappa^2}\cdot\begin{cases}\dfrac{1}{2\zeta\omega_n}+\dfrac{e^{-\zeta\omega_n\Delta T}}{2(1-\zeta^2)}\cdot\left(\dfrac{\sin(\omega_d\Delta T+\theta)}{\omega_n}-\dfrac{\cos(\omega_d\Delta T)}{\zeta\omega_n}\right) & \zeta<1\\ \dfrac{1}{2\zeta\omega_n}-e^{-a\Delta T}\left(\dfrac{2\alpha\beta}{a+b}+\dfrac{\alpha^2}{a}\right)-e^{-b\Delta T}\left(\dfrac{2\alpha\beta}{a+b}+\dfrac{\beta^2}{b}\right) & \zeta\geq 1\end{cases} \qquad \langle 3.10\rangle$$

where $\omega_d = \omega_n\cdot\sqrt{1-\zeta^2}$, $\cos\theta = \sqrt{1-\zeta^2}$, $a,b = \zeta\omega_n \mp \omega_n\cdot\sqrt{\zeta^2-1}$, $\alpha = \frac{-a}{b-a}$ and $\beta = \frac{b}{b-a}$.

Figure 3.7-(a) shows the short-term jitter behavior for different damping factors. The details of Figure 3.7-(b) is discussed in the next section. For ΔT of within a few cycles, jitter accumulates as with an open-loop VCO. As ΔT increases, jitter behaves similarly to the time-domain step response of the PLL output phase with similar dependence on the damping factor and bandwidth. The lower damping factor appears as more peaking in short-term jitter. For small short-term jitter, damping factor should be designed to be equal to or greater than one to avoid ringing in the jitter response.

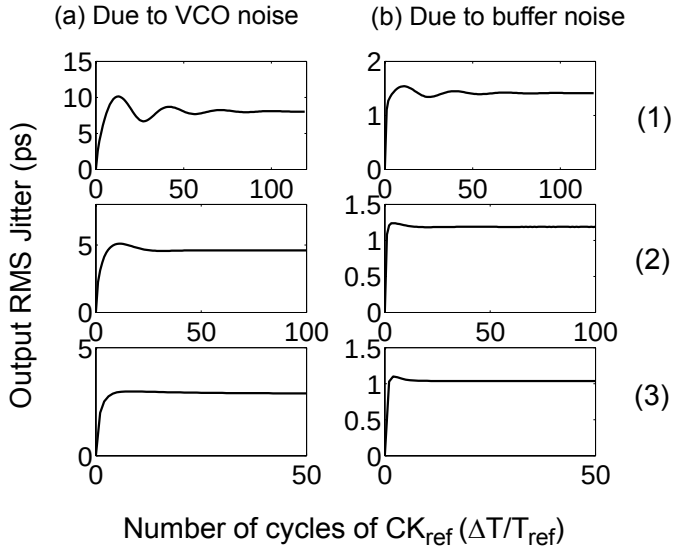


Figure 3.7: Short-term jitter behavior with different f_{-3dB} and ζ due to (a) VCO and (b) clock buffering noise. ((1) f_{-3dB} = 5.5% f_{ref}, ζ = 0.2 (2) f_{-3dB} = 6.4% f_{ref}, ζ = 0.65 (3) f_{-3dB} = 11.4%f_{ref}, ζ = 1.63)

At large ΔT, long-term jitter converges to final value of $\kappa \cdot \sqrt{\frac{1}{2\zeta\omega_n}}$. Note that this result is similar to the result derived in [41]. The sensitivity of jitter to loop parameters can be illustrated graphically. Sweeping loop bandwidth (f_{-3dB}) (or equivalently $f_n = \omega_n / 2\pi$)

while ζ is constant results in Figure 3.8-(a) in which jitter is reduced proportional to $\frac{1}{\sqrt{f_{-3dB}}}$. Figure 3.8-(b) illustrates the effects of varying ζ (or peaking in the frequency response) with constant f_{-3dB}. In the plot, f_n is adjusted to maintain the same f_{-3dB} while sweeping ζ. For ζ less than one (or greater peaking in frequency response), long-term jitter is proportional to $\frac{1}{\sqrt{\zeta}}$, but the sensitivity reduces as ζ increases. For ζ greater than 2 with constant loop bandwidth, long-term jitter is relatively constant, independent of ζ value.

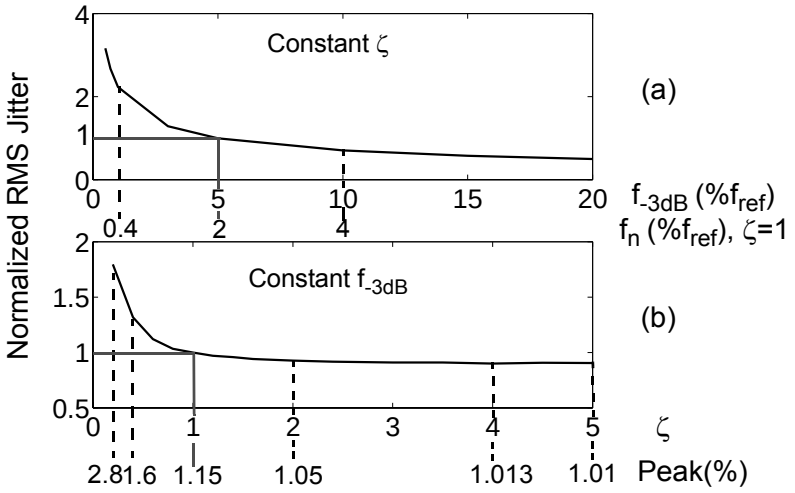


Figure 3.8: Long-term jitter (due to VCO noise) as a function of: (a) loop bandwidth, (b) loop damping factor

Jitter due to VCO Noise in a Third-Order Sampled PLL:

So far we investigated the effect of VCO noise using an ideal second-order PLL without considering the effects of the third-order pole or the inherent loop delay in a sampled system. In many PLLs, a 3rd-order pole is often included to filter control voltage ripple. For high loop bandwidths, this pole degrades the phase margin and causes peaking

in the frequency response. A similar frequency response peaking occurs when accounting for the delay in the feedback loop and the sampled-nature of the loop. These non-idealities can be taken into account using Equation 3.2 with a more accurate NTF.

We included these non-idealities into a MATLAB analysis. Figure 3.9 compares the output long-term jitter as bandwidth is increased for a second-order loop (curve-a), third-order loop without loop delay (curve-b), and third-order loop with loop delay (curve-c). In the plot, the 3rd-order pole is kept constant while the zero frequency is decreased which simultaneously increases the open-loop cross-over frequency, ω_c, and the damping factor. The plots on the right illustrate the loop frequency responses for a 2nd-order, 3rd-order PLL without and with loop delay as zero frequency (ω_z) is decreased. Curve-a shows the anticipated decrease in jitter due to the higher bandwidth and damping factor. In curve-b, as the loop bandwidth nears the 3rd-order pole, the peaking in frequency response increases due to phase margin degradation. Thus jitter is roughly flattened at bandwidths higher than 3rd pole due to the opposing effect of peaking and bandwidth on jitter. Accounting for loop delay (curve-c), the jitter increases at high bandwidth due to the additional peaking in the NTF from more phase margin degradation[1]. A minimum exists and is modestly flat over a significant range of loop parameter variations. This implies that a loop designed near this minimum has an output jitter that is relatively insensitive to the parameter variations that may be due to process, voltage and temperature (PVT).

1. To the first order, using the loop delay accounts for the effect of the sampled system. The measurement results of Section 3.7 matches the simulated results from this model better than that from a z-domain model using impulse invariant transformation [80].

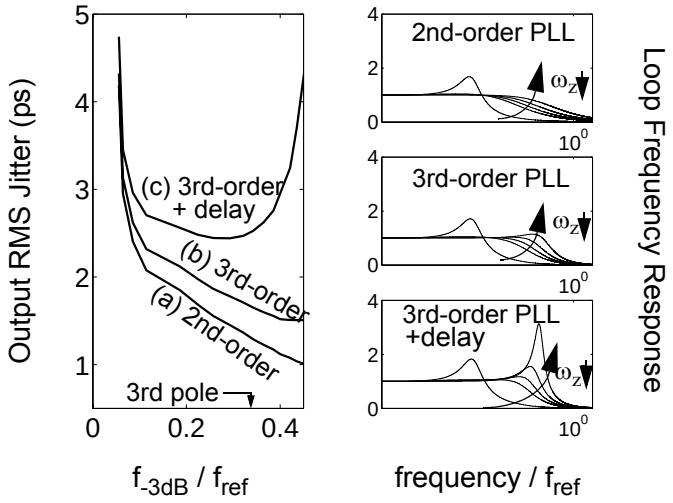


Figure 3.9: Comparison of long-term jitter (due to VCO noise) in: (a) 2nd, 3rd order loop (b) without loop delay and c) with loop delay

Analysis of the minimum indicates that it depends on all four variables (loop gain, zero frequency, 3rd-order pole frequency, and loop delay) because each contribute to phase margin degradation (Equation 2.7). The analytical results show that jitter is minimum with PM between 30^o and 45^o. Consequently, the PLL bandwidth at minimum jitter reduces as 3rd-pole frequency decreases or loop delay increases as shown in Figure 3.10. This result counters common practice of designing with large phase margins and damping factor of $\sqrt{1/2}$.

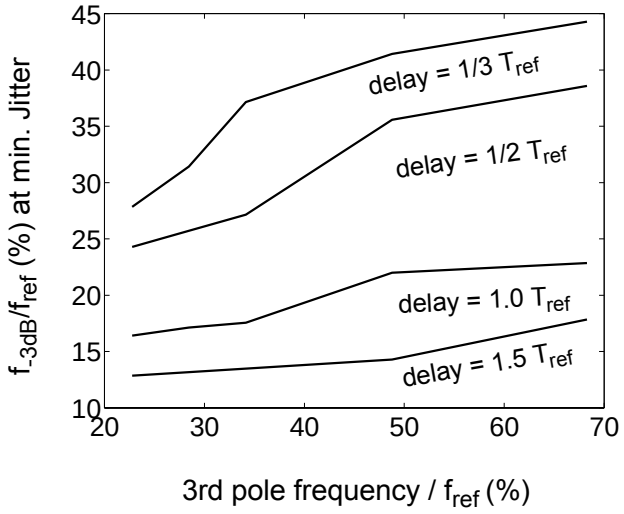


Figure 3.10: PLL bandwidth (at minimum jitter) as a function of 3rd pole frequency and PLL loop delay

Noise from the buffering and the input clock can be similarly analyzed using the corresponding closed-loop noise psds. Similar to the VCO noise, we first analyze the jitter behavior in an ideal second-order PLL. The final equations are summarized in Appendex A.3 and Appendex A.4, respectively. Then, the jitter analysis is extended to a third-order PLL taking into account the delay and sampling nature of the loop.

3.5.2 Jitter due to Clock Buffer Noise

Jitter behavior due to buffer noise over different time intervals has similar behavior to VCO noise except for small ΔT where jitter is increased sharply due to the high-pass filtering of the buffer NTF. Figure 3.7-(b) illustrates the output jitter for different ΔT with different damping factors.

To compare buffer noise magnitude with VCO noise, the jitter values are extracted from Equation 3.10 and Equation A.8 (Appendex A.4) for $\Delta T \to \infty$. The ratio of the buffer noise variance with VCO noise variance is:

$$\frac{\sigma_{Buf}^2}{\sigma_{VCO}^2} \approx \frac{(N_{Buf}/\omega_0^2)\cdot\omega_{Buf}}{(4\pi^2\cdot N_{VCO}/\omega_0^2)\cdot(1/2\zeta\omega_n)} = \frac{m\cdot K_{delay}^2\cdot\omega_0^2\cdot\omega_{Buf}\cdot e_{n_{buf}}^2/2}{4\pi^2\cdot K_{VCO}^2\cdot(1/2\zeta\omega_n)\cdot e_{n_{VCO}}^2/2} \qquad \text{‹3.11›}$$

where m is the number of buffer stages. For a ring oscillator with the same delay elements as the buffering, the K_{VCO} can be expressed in terms of K_{delay}, $K_{VCO} = K_{delay}\cdot\frac{-1}{2n\cdot t_d^2}$ where n is the number of stages in ring oscillator VCO and t_d is the delay of each stage. This simplifies Equation 3.11 to:

$$\frac{\sigma_{Buf}^2}{\sigma_{VCO}^2} \approx \frac{m\zeta\omega_n}{nf_{osc}} \qquad \text{‹3.12›}$$

With $\omega_n=0.2f_{osc}$ and $\zeta=1$, in order for the noise contribution of the buffer to be less than that of the VCO, either m<5n or the VCO element must have 5x lower noise sensitivity than the buffer elements. With lower loop bandwidths, buffer noise contribution decreases proportionally.

3.5.3 Jitter due to Input Clock Noise

Jitter due to Input Clock Noise in an Ideal Second-Order PLL:

When accounting for the effect of the PLL filtering on a noisy input clock, the analytical results[1] for a 2nd-order PLL show that the output clock timing jitter is

1. Equation A.18 in Appendex A.5

suppressed at small ΔT and asymptotically approaches a value, $\kappa\sqrt{1/(2\zeta\omega_n)}$, greater than the input jitter at large ΔT. The shape and final value depend on the bandwidth and the damping factor. Figure 3.11 illustrates the behavior of output clock jitter for different damping factors with constant bandwidth. The figure also includes the behavior of input

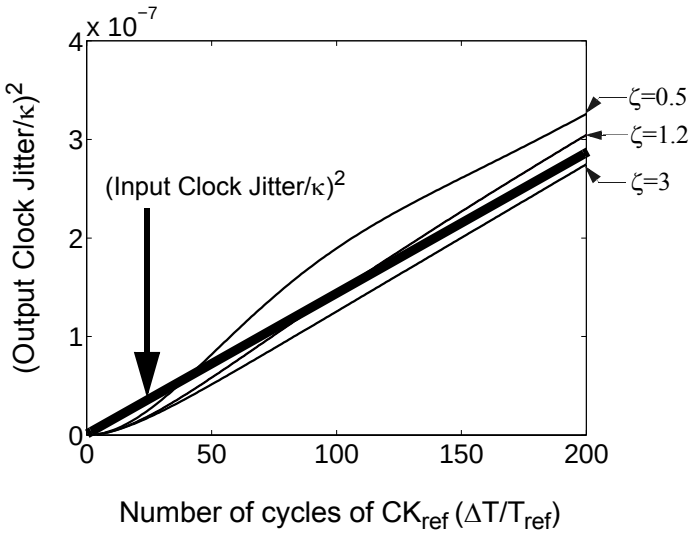


Figure 3.11: Output clock jitter (due to input clock noise) behavior vs. input clock jitter behavior

clock jitter. The ΔT at which the jitter exceeds the input jitter (the crossover time, ΔT_{cr}) is larger for higher damping factors and lower bandwidths. For most clock source PLLs, jitter of the overall system is suppressed as long as ΔT_{cr} is longer than the response time of any subsequent PLLs locking to the output clock. The jitter analysis due to noisy input clock not only confirms common practice but also elaborates the roles of bandwidth and damping factor on the output jitter. Figure 3.12-(a) shows how the output jitter (at ΔT=100 cycles) is reduced as bandwidth is decreased. Equation 3.12-(b) demonstrates that the output jitter (at ΔT=100 cycles) is reduced as damping factor is increased for two different

bandwidths. Similar to VCO noise analysis, output jitter is roughly constant for damping factor greater than 2. For instance, for output jitter to be less than 0.1 input jitter at $\Delta T >$ 100 cycles, the PLL should be designed with a damping factor greater than 2 and bandwidth less than 0.002% of operating frequency.

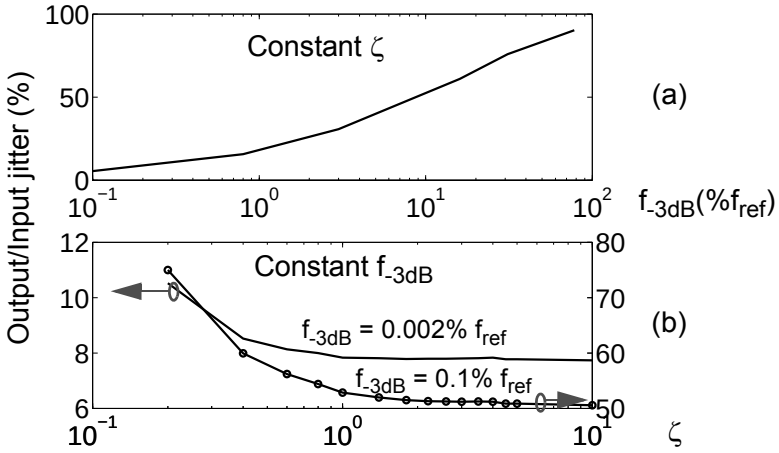


Figure 3.12: Output to input jitter ratio behavior of a 2nd-order loop as a function of: (a) loop bandwidth, (b) loop damping factor

Jitter due to Input Clock Noise in a Third-Order Sampled PLL:

To investigate the effects of the loop non-idealities, the jitter (due to input clock noise) of an ideal 2nd-order loop is compared to that of a 3rd-order PLL with loop delay. To better show the comparison, we assume white noise at PLL input phase instead of $1/f^2$ noise (of a noisy input clock). Figure 3.13 illustrates the output long-term jitter while the zero frequency is decreased which simultaneously increases the loop cross-over frequency and the damping factor. Jitter decreases initially for all three curves due to the lower frequency-response peaking where the bandwidth changes only slightly. As the zero

frequency decreases further, the bandwidth increases causing jitter to increase. At bandwidths close to 3rd pole, the peaking is increased due to phase margin degradation which results in more jitter increase in curve-b compared with curve-a. When accounting for loop delay (curve-c), additional peaking in the NTF from more phase margin degradation manifests the sharp jitter increase.

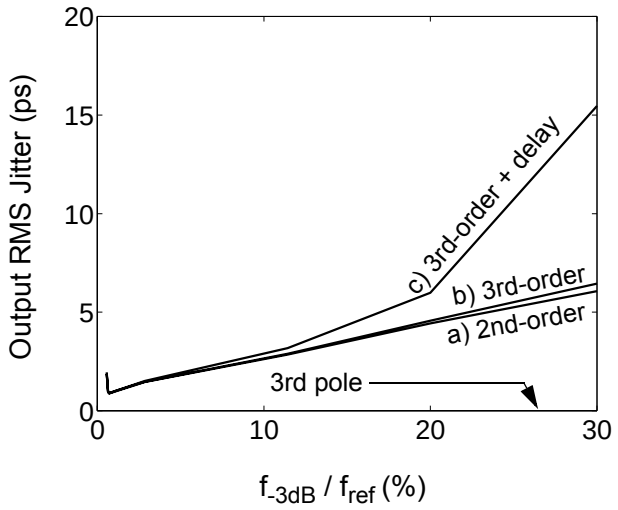


Figure 3.13: Comparison of long-term jitter (due to white noise at PLL input) in: (a) 2nd, 3rd order loop (b) without loop delay and (c) with loop delay

3.6 PLL Design with Adjustable Loop Parameters

As discussed in the previous section, a trade-off is present between input noise and the noise from within the loop. A high-bandwidth PLL can track the phase of a low-noise input clock and filter out VCO and clock buffer noise. Conversely, a low-bandwidth PLL filters a noisy input clock while it is transparent to VCO and clock buffer noise. We design a PLL with adjustable loop bandwidth and peaking in frequency response to verify the

results in the previous section. The parameters can be adjusted by varying the loop stabilizing zero and the open loop gain.

One possible architecture [52] is shown in Figure 3.14. This PLL has an adaptive

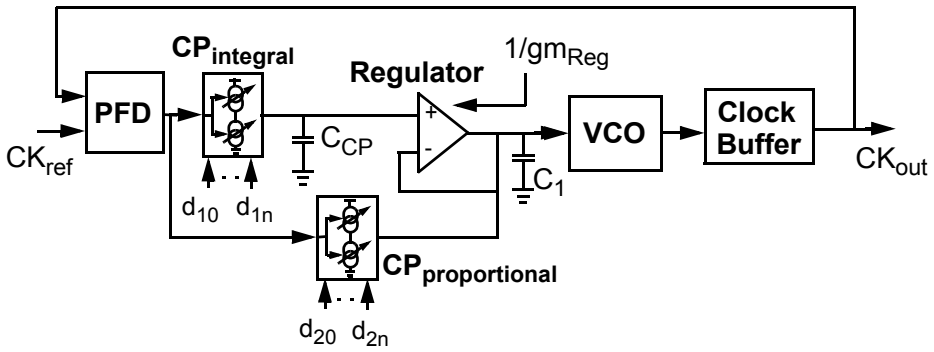


Figure 3.14: An adaptive bandwidth PLL with tunable loop parameters

bandwidth with tunable loop parameters. The design employs two digitally controllable charge pump currents in the proportional and integral paths to adjust ω_z and K_{loop}:

$$\begin{cases} \omega_z = 1/\left(\left(\frac{1}{gm_{Reg}} \cdot \frac{I_{CPproportional}}{I_{CPintegral}}\right) \cdot C_{CP} + \frac{1}{gm_{Reg}} \cdot C_2\right) \\ K_{Loop} = \frac{K_{VCO} \cdot I_{CPintegral}}{N \cdot C_{CP}} \end{cases} \quad \text{‹3.13›}$$

While the proportional charge-pump current varies the zero locus only, sweeping the integral charge-pump current changes both the zero and the open loop gain. Varying any of the two charge-pump currents does not vary the position of the PLL third-order pole.

3.7 Experimental Methods and Results

The adaptive bandwidth PLL clock generator with tunable loop parameters (shown in Figure 3.14) is designed and fabricated in 0.25-µm CMOS technology. The PLL die photograph is shown in Figure 3.15 where the area overhead due to digital controller logic is approximately 15% of PLL core area.

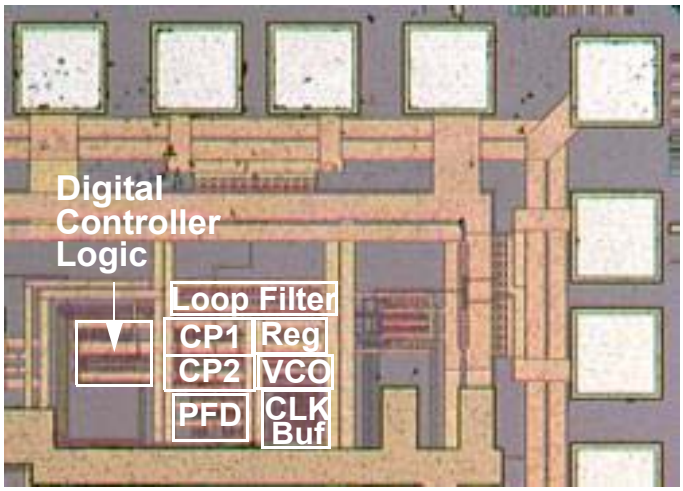


Figure 3.15: Die photograph of the PLL

3.7.1 Verification of Jitter Analysis due to VCO Noise

To observe only VCO noise, a clean signal generator (with rms jitter of less than 1 ps) produces the reference clock and the design uses only a few buffer stages in the feedback so that the buffer noise is small compared to VCO noise.

Tracking Jitter due to VCO Noise:

To verify the presence of minimum tracking jitter due to VCO noise, the integral charge pump current is kept constant (i.e. K_{Loop} = constant) while the proportional charge

pump current is swept (i.e. ω_z is decreased). For each value of $I_{CPproportional}$, the rms tracking jitter of PLL output clock is measured based on the configuration of Figure 3.16. The same measurement is repeated when I_{CP1} is varied.

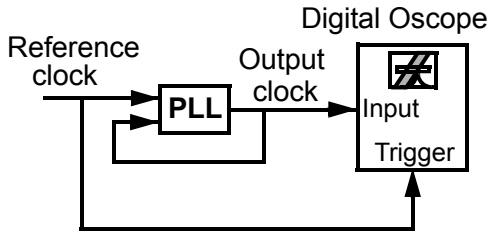


Figure 3.16: Measurement technique in time domain, referenced to reference clock

Table 3.1 summarizes some of the results at reference clock frequency of 700MHz where I_1 and I_2 are constant currents.

Figure 3.17-(a) and (b) show the measured and calculated jitter for one set of measurements repeated for two reference clock frequencies. As seen in the figure, the measured jitter corresponds closely with the analytical results and there is a minimum jitter with a low sensitivity to loop parameter variations. For example, ±20% of bandwidth variation increases jitter by less than 5%. In each set of measurements, jitter initially decreases because the peaking decreases (or ζ grows linearly) with $I_{CPproportional}$ and the f_{-3dB} increases with the decreasing zero frequency (f_n is held constant). As I_{CP2} increases, the cross-over frequency approaches the third-order pole and degrades the phase margin. Jitter reaches a relatively flat minimum before increasing due to the loop delay (approximately 0.47ns).

Increasing reference clock frequency from 700MHz to 1.1GHz in our adaptive bandwidth PLL, effectively measures the result of changing the loop's feedback delay from 1/3 to 1/2 of the reference clock period. The bandwidth at minimum jitter is reduced from 26% to 12% of reference clock (Figure 3.17-(c)).

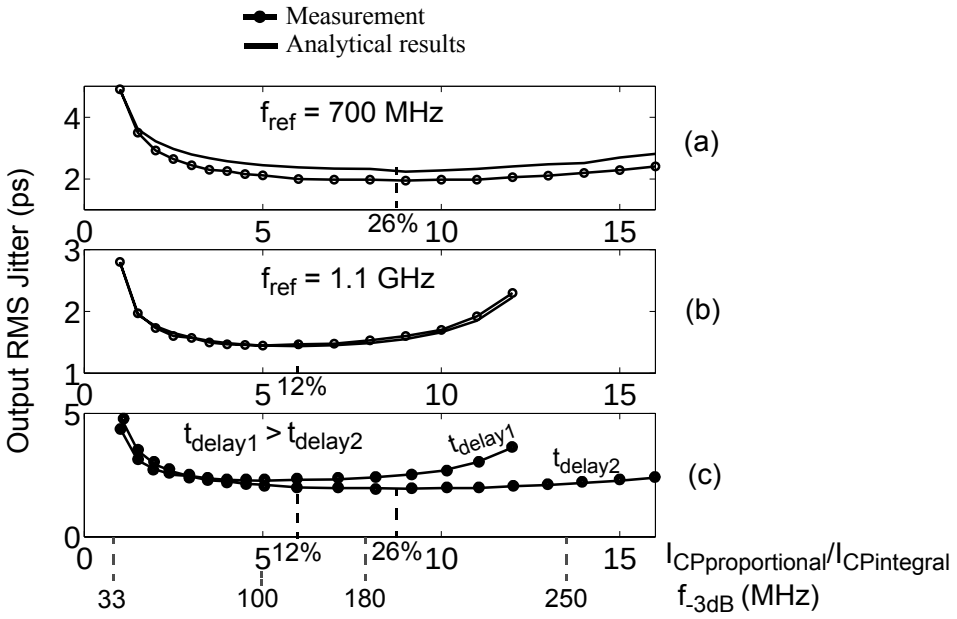


Figure 3.17: Measured and calculated tracking jitter as ω_z is reduced in constant

Table 3.1: Tracking jitter (in ps) for different loop parameters (f_{ref} = 700MHz)

$I_{CPintegral}$ / $I_{CPproportional}$	$2.I_1$ rms jitter	$3.I_1$ rms jitter	$4.I_1$ rms jitter	$5.I_1$ rms jitter	$6.I_1$ rms jitter
$2.I_2$	4.49	4.67	5	5.57	6.8
$3.I_2$	3.4	3.41	3.45	3.57	3.76
$4.I_2$	2.8	2.81	2.96	2.87	2.99
$5.I_2$	2.58	2.54	2.6	2.52	2.55
$6.I_2$	2.37	2.35	2.3	2.35	2.37
$7.I_2$	2.24	2.2	2.17	2.23	2.18
$8.I_2$	2.14	2.1	2.08	2.1	2.1
$9.I_2$	2.04	2	2.03	1.99	2.03
$10.I_2$	2.01	1.97	1.9	1.99	1.93
$16.I_2$	1.88	1.87	1.87	1.8	1.85
$17.I_2$	1.9	1.8	1.72	1.73	1.84
$18.I_2$	1.91	1.88	1.73	1.71	1.85
$19.I_2$	1.89	1.89	1.77	1.73	1.88
$21.I_2$	1.94	1.86	1.73	1.72	1.83
$24.I_2$	2.03	1.99	1.8	1.77	2
$32.I_2$	2.4	2.1	2.16	2.21	2.32

Short-Term Jitter due to VCO Noise:

The short-term jitter sensitivity to PLL loop parameters is also verified. The short-term jitter is calculated with the analytical model. The time domain figure of merit of the VCO is equal to $\kappa \approx 5.4e-8\sqrt{s}$ at 700MHz oscillating frequency. The 3-dB bandwidth and peaking used for the model are first calculated through circuit simulations and then verified with direct measurements. The test setup that measures the loop parameters is

shown in Figure 3.18. A radio frequency (RF) signal is added to the input clock. The

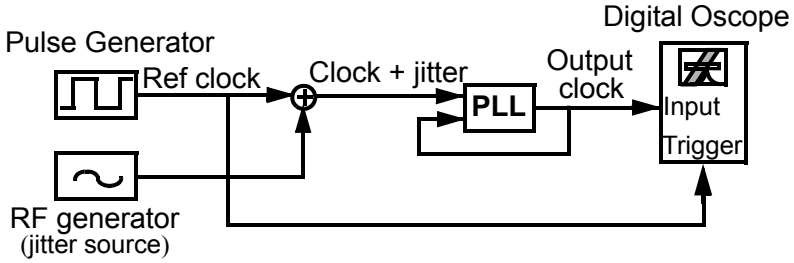


Figure 3.18: Measurement technique for calculating PLL loop transfer function

output clock jitter is measured over different RF frequencies. The measured PLL loop transfer functions with their *effective* f_{-3dB} and *effective* peaking (Appendex A.6) are shown in Figure 3.19 for four different values of $I_{CPproportional}$ with constant $I_{CPintegral}$.

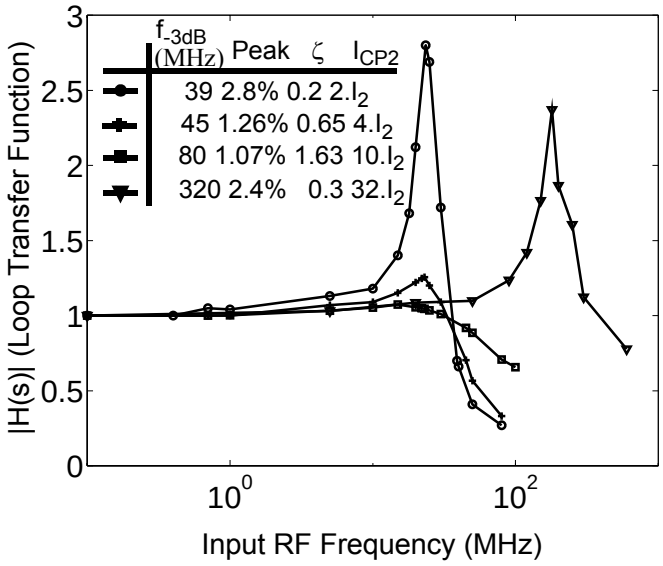


Figure 3.19: Measured PLL loop transfer function (@ 700MHz reference clock) at a constant $I_{CPintegral}$ (constant K_{Loop})

The rms jitter is measured over different time interval (ΔT) for each of the four different settings of loop parameters. The measurement uses a self-referenced technique shown in Figure 3.20. The dummy delay in the test setup is critical to compensate for the

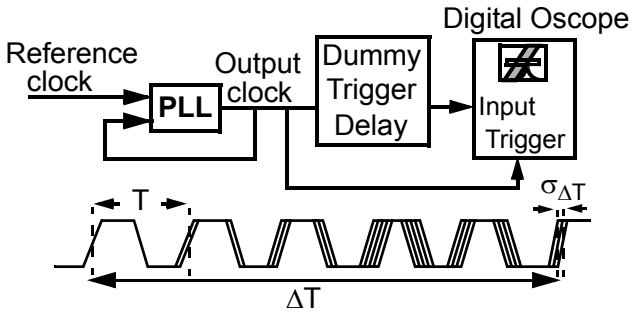


Figure 3.20: Measurement technique in time domain, referenced to output clock

triggering delay of an oscilloscope. Figure 3.21 shows the measured and calculated short-

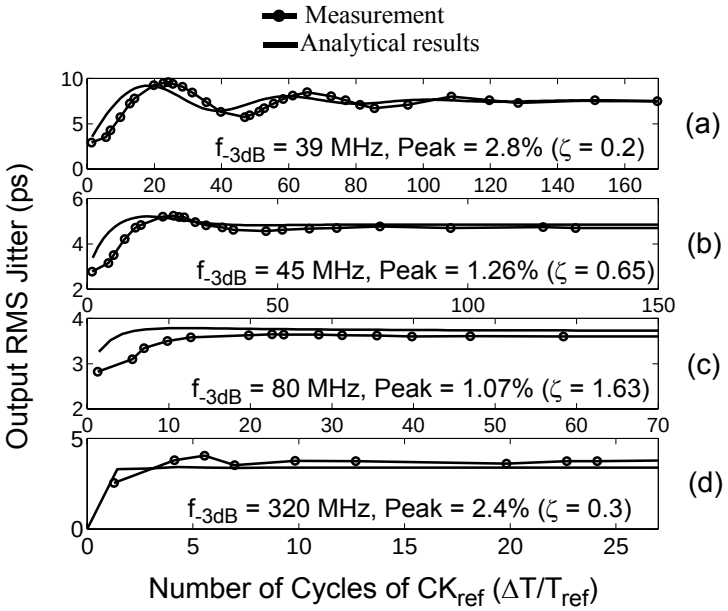


Figure 3.21: Measured and calculated short-term jitter (@ 700MHz reference clock) for four different loop parameters

term jitter. A slight timing shift between predicted and measured jitter is present because of time uncertainty due to the delay of input trigger and dummy trigger delay at the input of oscilloscope.

3.7.2 Verification of Jitter Analysis due to Input Clock Noise

To verify the jitter analysis due to input clock noise, we apply a free running VCO at 700MHz as a reference clock of the PLL. A white noise source is injected to the control voltage of the free running VCO so that the input clock noise is the dominant noise source.

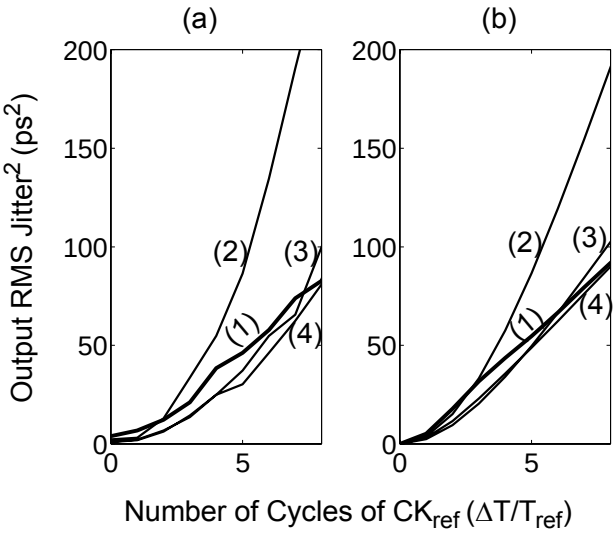


Figure 3.22: Output jitter (due to input clock noise) behavior for three different PLL loop parameters: (a) measurement results, (b) analytical results ((1) Input jitter (2) ζ = 0.2, f_{-3dB} = 39MHz (3) ζ = 0.65, f_{-3dB} = 45MHz (4) ζ = 1.63, f_{-3dB} = 80MHz)

As the baseline measurement, we measure the rms jitter of this reference input over different time interval (ΔT) based on the self-referenced technique (Figure 3.20). We also

measure the PLL output rms jitter while varying ΔT for three different loop parameters. The measurement results in Figure 3.22-(a) demonstrate the same behavior to the analytical results (Figure 3.22-(b)) with approximately the same ΔT_{cr}.

3.8 Summary

This chapter investigates the role of PLL loop parameters on timing jitter. Several common noise sources have been included in the analysis. We develop an intuition for designing low-jitter PLLs both by deriving a closed-form solution for a second-order loop and by plotting the jitter sensitivity to various loop parameters for higher-order loops. One possible PLL architecture with digitally-controllable loop parameters is designed that can optimize jitter performance. Furthermore, the loop serves as a test bench to verify our analysis.

The analysis shows a simple expression for long-term jitter due to VCO and buffering noise to the damping factor and natural frequency. We derive an expression that relates the jitter contribution of clock buffering (in the feedback) and VCO to the same parameters. We validate the common design practice of using high loop bandwidth to reduce VCO-induced jitter. However, to minimize jitter, we find that accounting for the loop delay in the phase margin is critical. Interestingly, this minimum is very insensitive to PVT and parameter variations making such a design robust. For applications that require small short-term jitter (i.e. short distance links and block to block interconnect), an underdamped loop can result in much higher short-term rms jitter. For applications that filters input jitter, our modeling shows that very low bandwidths (0.002% f_{osc}) are

necessary to reduce noise by a factor of 10 while a damping factor greater than 2 is sufficient.

The result of jitter analysis extracted in this chapter can be applied to the optimum design of PLL loop parameters to minimize the PLL output jitter. The jitter optimization requires a well-known knowledge about the noise sources in a PLL. Since the noise sources are not predetermined, the preliminary design of loop parameters does not neccessarily result in minimum jitter performance. To further improve the noise performance, the loop parameters of a PLL should be tuned for a minimum output jitter in real system noise conditions. The next chapter presents a methodology for on-chip jitter minimization and verifies the accuracy of the method in converging to the minimum jitter at PLL output clock.

Chapter 4

Methodology for On-Chip Adaptive Jitter Minimization in PLLs

The previous chapter shows that the output jitter of a PLL depends strongly on the magnitude and frequency response of the noise sources and the loop parameters. For many systems, the loop design is complicated because the magnitude of the noise sources is not well known; a noisier clock reference may be used or larger on-chip switching noise may be present. Jitter can still be minimized under various noise conditions if jitter can be dynamically measured with an on-chip noise measuring circuit and the loop parameters can be adapted with a programmable loop filter. This chapter investigates the methodology and accuracy of jitter minimization that occurs during system operation and not just during calibration or system startup.

Section 4.1 reviews the relationship between the minimum jitter and the loop parameters for two noise sources, input clock noise and internal VCO noise, as extracted

in Chapter 3. It is observed that the total jitter due to the two noise sources has only one minimum that is global for a range of loop parameters that the PLL is stable. This result leads to the gradient-descent algorithm described in Section 4.3. The circuit components needed to dynamically minimize jitter is described in Section 4.2. Several of the existing circuits that can measure jitter both for clocking and for data recovery are discussed. Section 4.3 discusses the algorithms that converge to the minimum jitter during active system operation. Because jitter is a stochastic process, any on-chip measurements are subject to errors depending on the amount of averaging. The section illustrates the performance of the convergence as related to the amount of jitter information. The chapter concludes with some guidance on design of on-chip jitter minimization.

4.1 Overview

Previous chapter discussed the relationship between minimum jitter due to each noise source and PLL loop parameters. The block diagram of a PLL with two primary noise sources, input clock and internal VCO noise, is shown in Figure 4.1. Although this

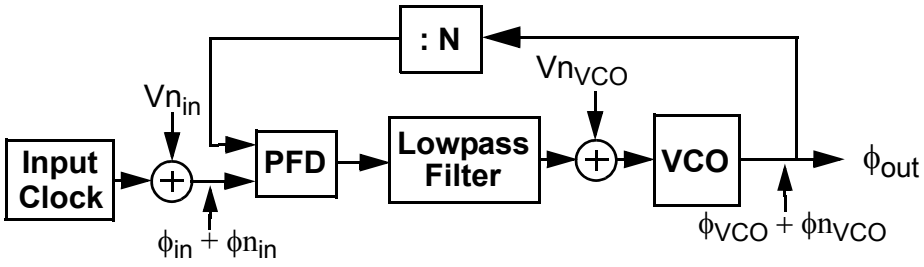


Figure 4.1: The PLL block diagram with VCO and input noise

work only considers these two noise sources, the results can be extended to other noise sources in a PLL.

Each of the noise sources is shaped by the loop transfer function from the corresponding noise voltage source to the output phase. Figure 4.2 illustrates the filter response for each of the noise sources.

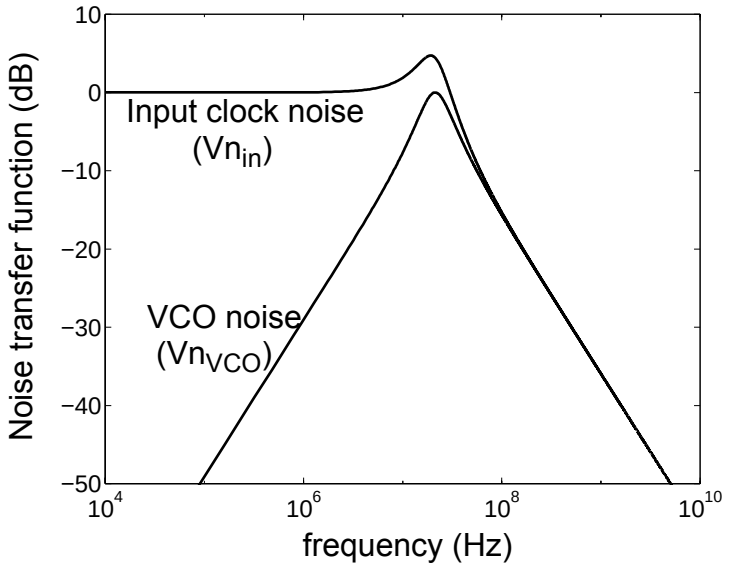


Figure 4.2: Loop transfer functions from VCO and input clock noise to the PLL output

As seen in Figure 4.2, the loop transfer function for the VCO noise (Vn_{VCO}) is a band-pass filter that suppresses the VCO noise within the PLL bandwidth. In a second-order PLL, the long-term rms jitter due to VCO noise[1] is calculated as:

1. VCO phase noise is assumed to fall as $1/f^2$. The long-term jitter is calculated from Equation 3.10 when ΔT goes to infinity

$$\sigma_{rms} = \frac{\sqrt{N_{VCO}}}{f_0} \cdot \sqrt{\frac{1}{2\zeta\omega_n}} \qquad \langle 4.1 \rangle$$

where f_0 is the VCO frequency, ζ is the PLL damping factor and ω_n is the PLL natural frequency. The two loop parameters that can be easily tuned in a charge-pump PLL are the PLL zero frequency, ω_z, and the PLL loop gain, K_{loop}. Sweeping ω_z and K_{loop} effectively changes the PLL bandwidth and peaking in the PLL frequency response. Substituting ζ and ω_n with ω_z and K_{loop} in Equation 4.1 results in:

$$\sigma_{rms} = \frac{\sqrt{N_{VCO}}}{f_0} \cdot \sqrt{\frac{1}{(\omega_z)^{-1} \cdot K_{Loop}}} \qquad \langle 4.2 \rangle$$

Based on Equation 4.2, the relationship in a second-order PLL between the VCO-induced jitter and the loop parameters, $(\omega_z)^{-1}$ and K_{loop}, can be shown to be convex and hence has only a global minimum without local minima. The jitter behavior as a function of $(\omega_z)^{-1}$ and K_{loop}, in a third-order sampling PLL is graphically shown in Figure 4.3. The plot includes the higher-order pole and sampling/feedback delay and still maintains the convexity. The minimum jitter, as shown with the contours in Figure 4.3-(a), occurs at a high loop bandwidth with low peaking in the PLL frequency response. As the bandwidth is further increased, the phase margin degrades which increases the peaking and eventually increases jitter.

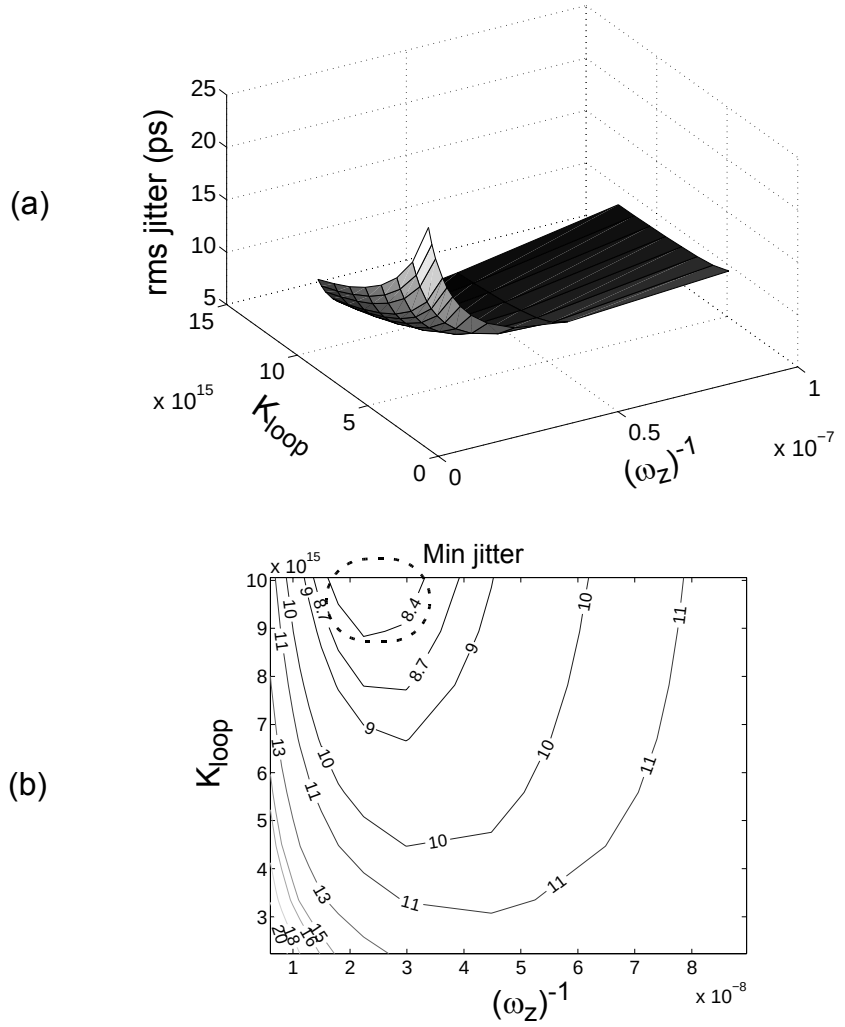


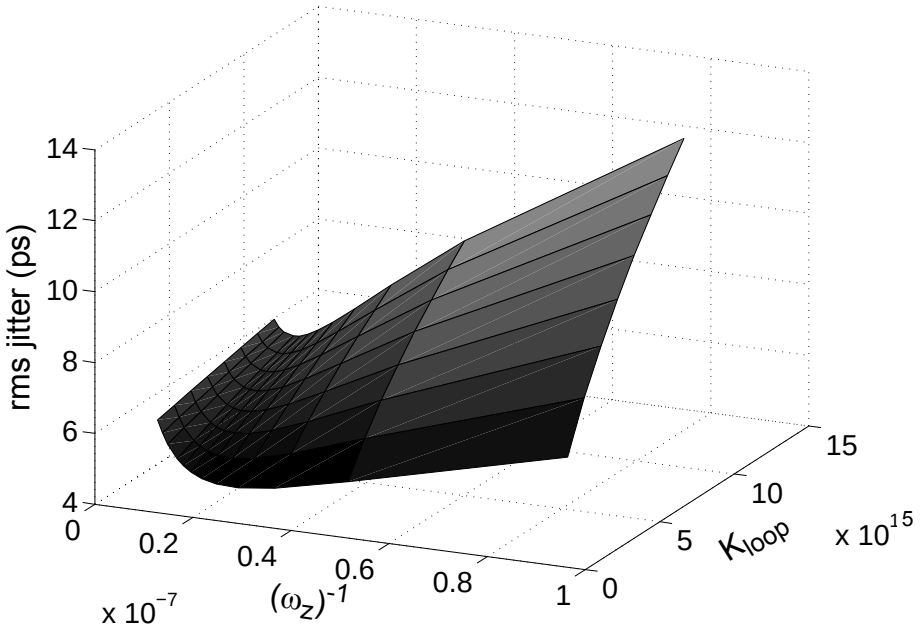
Figure 4.3: Behavior of output clock jitter due to VCO noise for various loop parameters: (a) 3-D, (b) contour

In contrast with the VCO noise, the loop transfer function for the input clock noise is a low-pass filter that suppresses the input noise outside the loop bandwidth. The long-term jitter due to input clock noise[1] in a second-order PLL is equal to:

1. The input clock noise is assumed to be white, i.e. $S_{\phi n-in}(f) = N_{Clk-in}$. The detail of long-term jitter calculation is given in Appendix A.5.2.

$$\sigma_{rms} = \frac{\sqrt{N_{Clk-in}}}{2\pi f_0} \cdot \sqrt{K_{loop} \cdot (\omega_z)^{-1} + \frac{1}{(\omega_z)^{-1}}} \qquad \text{‹4.3›}$$

(a)



(b)

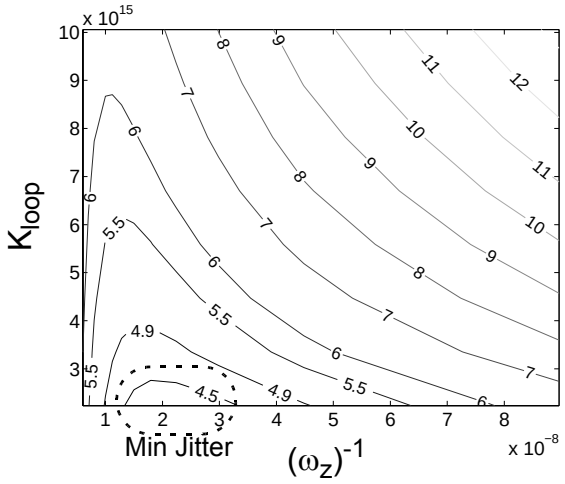


Figure 4.4: Behavior of output clock jitter due to input noise for various loop parameters: (a) 3-D, (b) contour

It can be shown that the relationship in Equation 4.3 is not convex[1]. Including the higher-order pole and sampling/feedback delay, Figure 4.4 plots the output jitter due to only input-clock noise for a third-order sampling PLL. As seen in the figure, no local minimums exist. The concavity of the surface is not very apparent and only occurs when the phase margin is small ($<30^o$). Such small phase margin is an unlikely operating point due to possible loop instability. Similar to VCO noise, a single minimum exists except that it is at a low loop gain as shown by the contours in Figure 4.4-(b).

The total jitter is the sum of the jitter variances due to both VCO and input noise sources. Figure 4.5-(a) shows the total jitter when two noise sources are comparable. As one noise source becomes dominant, the minimum point of the contour shown in Figure 4.5-(b) moves toward the minimum for that particular noise source (Figure 4.3-(b) or Figure 4.4-(b)).

Although, the jitter function due to the input noise is not entirely convex, it is shown in Appendix A.8 that the total jitter in a second-order PLL has one global minimum without any local minima. Simulation results for various ratios of VCO and input noise sources show that the single minimum holds even when including a higher-order pole and sampling/feedback delay. This important result motivates the proposed gradient-descent algorithm of Section 4.3 when the loop parameters are dynamically adjusted to achieve minimum jitter.

1. Please see Appendix A.7.

(a)

(b)

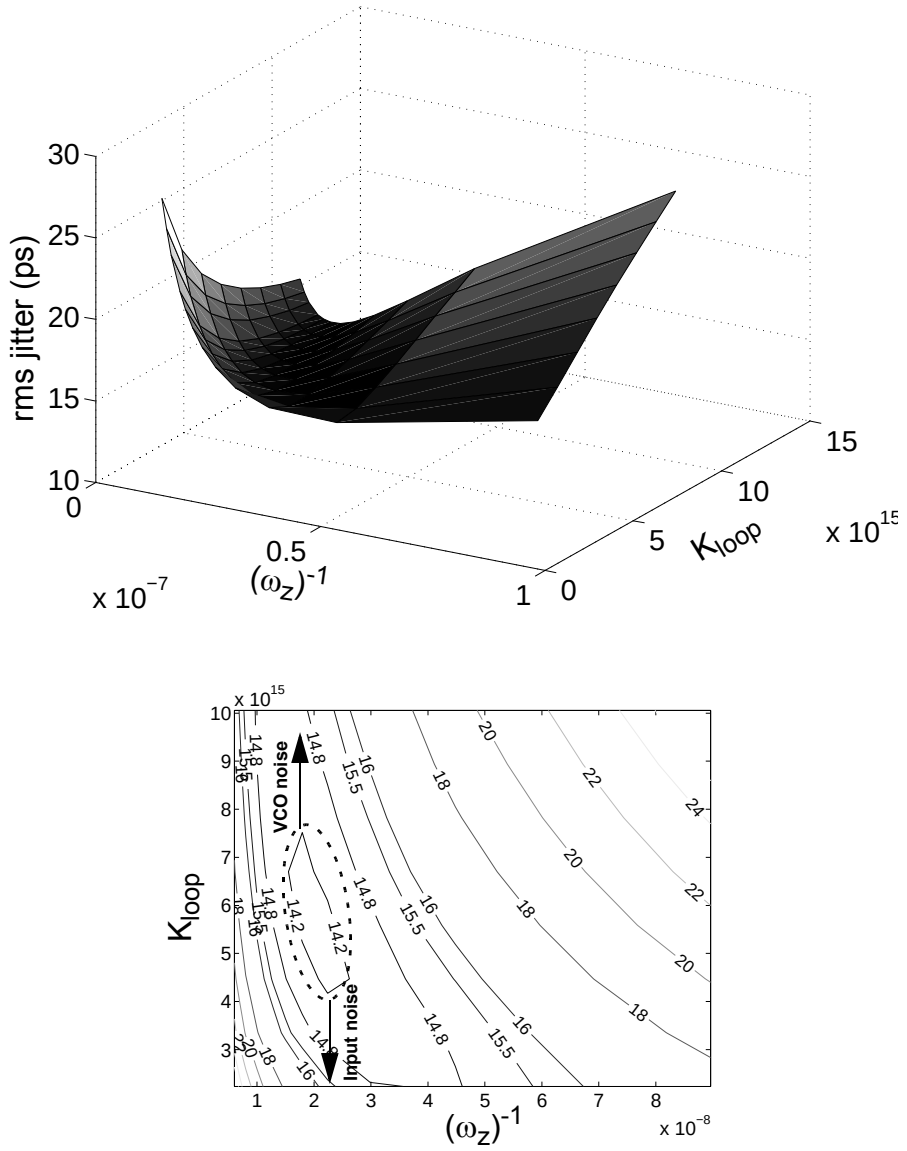


Figure 4.5: Behavior of output clock jitter due to both VCO and input noise for various loop parameters: (a) 3-D, (b) contour

4.2 Jitter Detection Circuits and Architectures

To dynamically minimize jitter at the PLL output during system-operation, the design requires three elements: 1) a PLL that has appropriately adjustable loop parameters, 2) an on-chip jitter measuring that can compare the jitter between measurements, and 3) an algorithm that adjusts the loop parameters to minimize jitter based on the on-chip measurements. The first two are discussed in this section.

4.2.1 PLL Design with Adjustable Loop Parameters

As discussed in the previous chapter, the two loop parameters of a PLL that significantly impact jitter are the loop bandwidth and peaking in the frequency response. They can be adjusted by varying the loop stabilizing zero and the open loop gain. One possible PLL architecture is the one used in Chapter 3 to verify the jitter analysis (Figure 3.14). While the proportional charge-pump current varies the zero locus only, sweeping the integral charge-pump current changes both the zero and the open loop gain.

In the second configuration shown in Figure 4.6, ω_z and K_{loop}, are independently adjustable by varying the loop stabilizing resistor (R) and charge pump current (I_{CP}), respectively:

$$\begin{cases} \omega_z = 1/(R \cdot C_{CP}) \\ K_{loop} \cong \dfrac{K_{VCO} \cdot I_{CP}}{N \cdot C_{CP}} \end{cases} \quad \text{‹4.4›}$$

In this configuration, third-pole does not move as I_{CP} or R are changed.

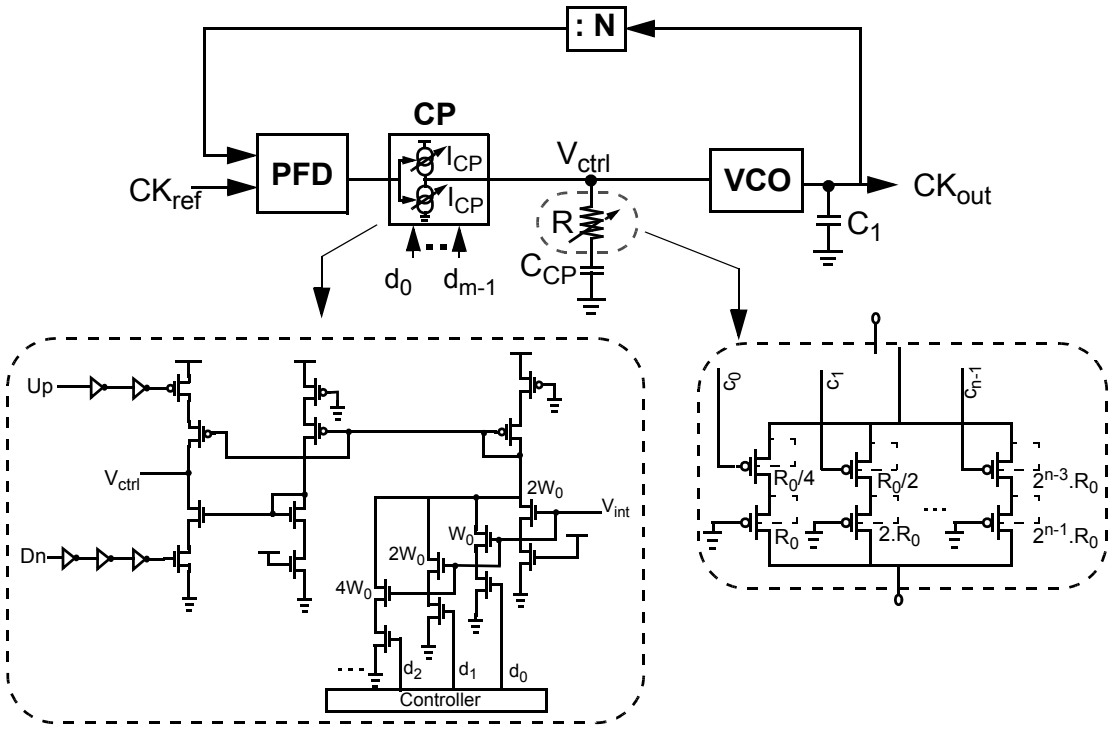


Figure 4.6: A PLL architecture with adjustable loop parameters using adjustable R and I_{CP}

The configuration shown in Figure 4.6 is used in both simulations and measurements. The design permits 4-bits of digital adjustment for resistor that varies the zero position by more than 10x (from 0.1 to 1.6 rad/sec). In this implementation of the adjustable resistor, the resistance steps with non-linear digital quantization levels[1]. The design also permits 3-bits of digital adjustment for charge-pump current that varies the loop gain by 5x (from 2e15 to 10e15 $(rad/sec)^2$) with a linear quantization level.

1. [..., $5/7R_0$, $5/6R_0$, $5/4\ R_0$,..., $5/2R_0$, $5R_0$]

4.2.2 On-chip Jitter Measurement Architectures

The on-chip jitter measurement circuit depends on the application. This section first describes the approach for a data recovery system. Possible approaches for on-chip clock generation is addressed next.

1) Data-Recovery Applications:

In data-recovery applications, clocks sample not only the center of the data eye to recover the data pattern but also the data transitions to determine phase information. The goal of the PLL is to track the data jitter while rejecting the noise from the VCO. By correlating the sampling clock with the data transitions, the loop minimizes the phase error between the sampling clock and the data input.

Several previously published techniques demonstrate on-chip jitter measurement. In [67], a flash time-to-digital converter (TDC) measures the data jitter with the sampling clock. This technique requires significant number of arbiters and on-chip buffering of the data and clock as shown in Figure 4.7.

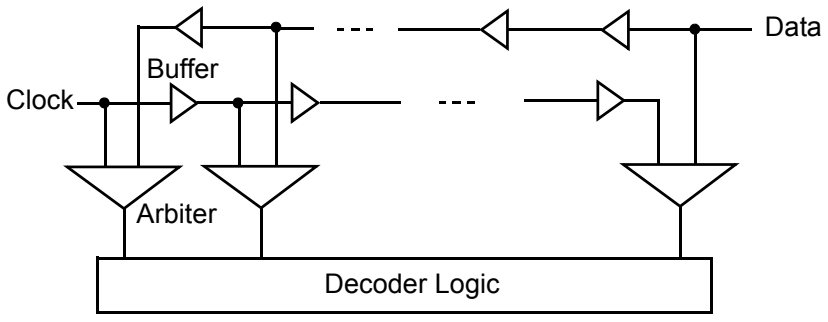


Figure 4.7: Jitter measurement with a flash TDC architecture

Another technique demonstrated by [68]-[70] uses a dead-zone phase detector to measure the jitter. Figure 4.8 illustrates the basic concept of the jitter measurement. A

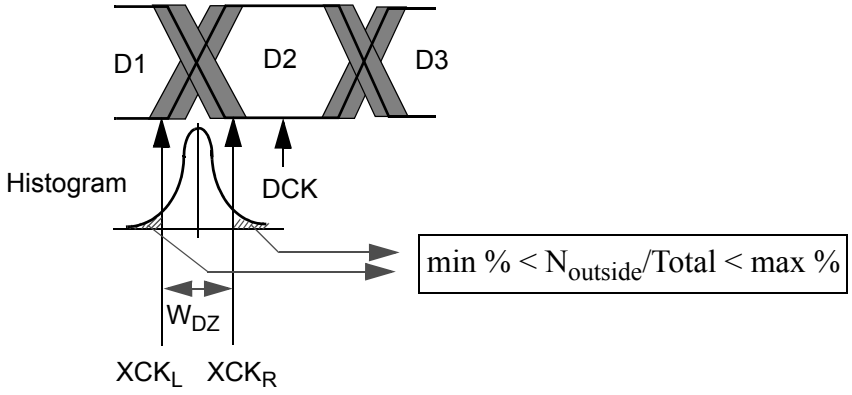


Figure 4.8: Jitter measurement with a dead-zone window establishment

dead-zone window is constructed by using two data-transition samplers in addition to sampling the data in the middle of the eye. The transition sampling clocks, XCK_L and XCK_R, are programmed to track the left and right edges of the data eye and adjust the dead-zone width, W_{DZ}, accordingly. The design in [70] uses only one data-transition sampler to construct the dead-zone window (W_{DZ}) by alternating the edge sampling clock position. The data transition outside the window is detected when the value of data sampled by the transition sampling clock is equal to that sampled by the data sampling clock. The magnitude of jitter is estimated by comparing the number of data transitions outside the dead-zone for a given total count of data transitions. The window size is adjusted when the number of transitions (measured hits) outside the zone is greater or less than predetermined bounds to avoid saturating the counters. A similar method can adjust

the width of the dead-zone window until the number of measured hits is roughly a fixed percentage of the total hits. This effectively directly measures the width of the jitter histogram. The dead-zone technique is the measuring jitter circuit that is mimicked in the next section.

2) On-Chip Clock Generation Applications:

The design is considerably different for an application that minimizes the jitter of a large digital system's clock. A similar architecture to [67] has been shown in [71] where an array of phase detectors compares consecutive clock edges and measures the *cycle-to-cycle* jitter. However, it is important to note that cycle-to-cycle jitter can not be minimized through adapting the loop parameters. As shown in Section 3.5, cycle-to-cycle jitter is primarily determined by the noise characteristics of the VCO alone and not the PLL loop parameters. Adjusting loop parameters may result in large long-term jitter or an unstable loop.

In the event that the long-term tracking jitter is important, a circuit that accumulates phase over multiple cycles is necessary. The design of the accumulation circuit is very challenging because it must strongly reject supply and substrate noise. A simple delay line that spans multiple cycles is not adequate because a multi-cycle on-chip delay line would likely introduce a significant noise floor to the measurement. Integrator techniques similar to that used by Wavecrest SIA-3000 would suffer similar issues on-chip. The design of this challenging circuit is left as future work and not addressed in this work.

4.3 Jitter Minimization Algorithms and Measurements

Due to the stochastic nature of jitter, the measurement accuracy is a function of the number of the samples. In addition, the jitter measurement circuit itself introduces some noise. After describing the measurement setup, this section discusses the sensitivity of the measurement to the total number of samples. Next, two jitter minimization algorithms are described and their effectiveness is verified with measurement results.

4.3.1 Measurement Setup

The PLL in Figure 4.6 with adjustable loop parameters has been fabricated in a 0.25-μm CMOS technology. The chip die photograph is shown in Figure 4.9. To demonstrate the adaptive jitter minimization, a sub-sampling digital scope is used as a proxy of the on-chip dead-zone phase detector circuit to measure the jitter of the output clock. None of the features of the scope such as rms or p2p jitter information is used. Instead, only the histogram data is downloaded to the computer through the GPIB port. By counting the number of transitions (measured hits) outside a dead-zone window as a percentage of the total number of transitions (total hits), the measurement replicates that of a dead-zone phase detector. The number of measured hits (or percentage) is an indication of the jitter magnitude[1]. The dead-zone width is adjusted when the number of measured hits (outside the dead-zone) exceeds 1-10% of the total hits. The histogram can also model the behavior of other jitter-measuring circuits. As an example, mentioned in Section 4.3.3,

1. The total jitter for a data-recovery system is the sum of the data jitter and sampling clock jitter.

the histogram can directly determine the width of dead-zone window such that the number of hits outside the window is a fixed percentage, i.e. 4%.

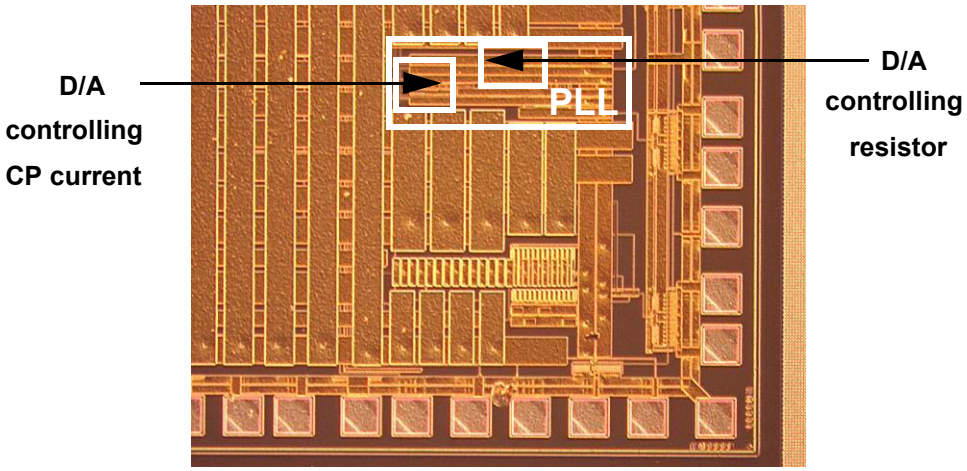


Figure 4.9: PLL die photograph

Figure 4.10 shows the measurement setup. The PLL loop parameters (I_{CP} and R) are changed by D/A converters controllable by a data-acquisition board. The digital scope is controlled by a C-program through a GPIB interface with the PC computer.

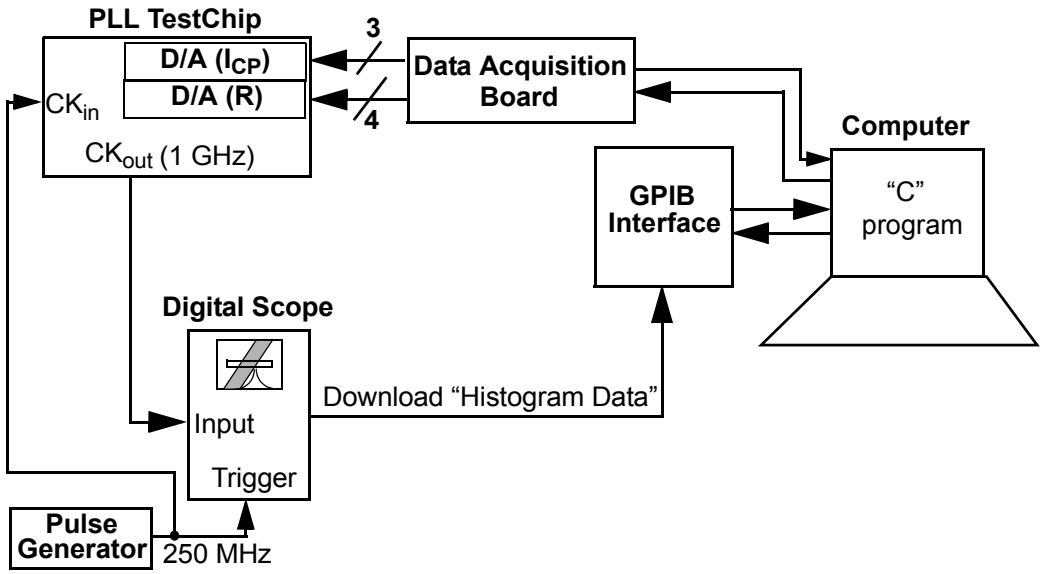


Figure 4.10: Test setup for the jitter measurement and optimization

Before jitter minimization algorithms being discussed, we first discuss the sensitivity of the measurement to the total number of hits. The inherent randomness of jitter results in some measurement error that will degrade the performance of the jitter minimization.

4.3.2 Measurement Uncertainty

With limited number of total hits, the percentage of hits that is outside the dead-zone window varies between measurements. The percentage forms a distribution where the standard deviation of the distribution is inversely proportional to the total hits, N, in the histogram. Figure 4.11-(a) illustrates four distributions of the percentage of hits outside the dead-zone. The curves represent two values of total hits (N=500 and N=5000)

and two dead-zone positions ($W_{DZ}=4\sigma$ and $W_{DZ}=5\sigma$ where σ is the jitter standard deviation). The additional shaded lines illustrate the impact on the measurement when

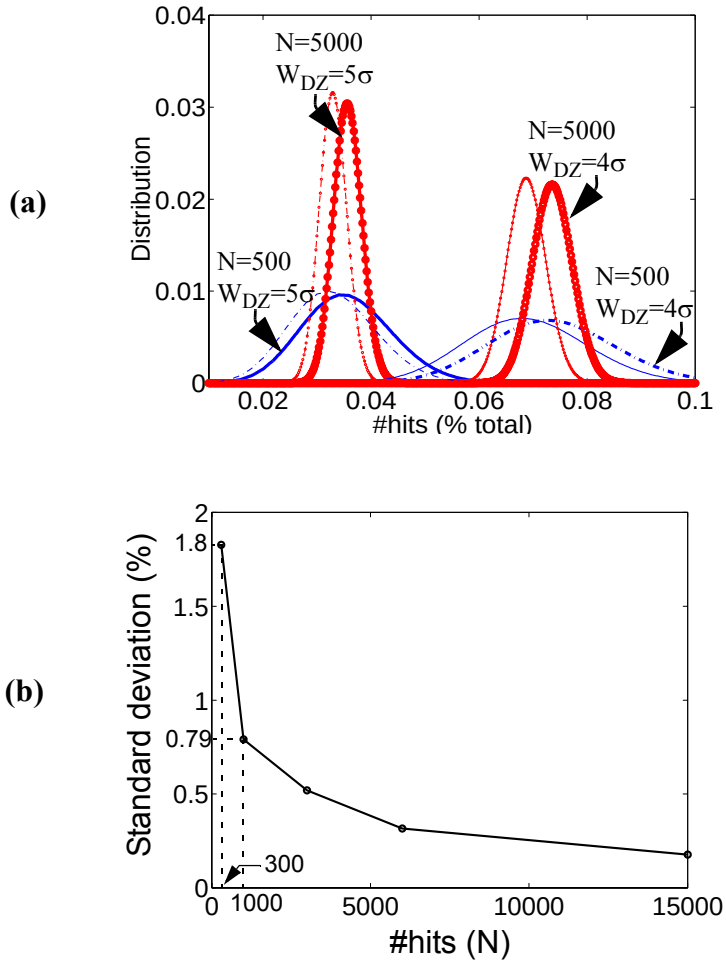


Figure 4.11: (a) Measured percentage hits distribution for one set of PLL loop parameters for N=500 and N=5000, (b) standard deviation of measured percentage hits

there is a ΔW_{DZ} of 0.1σ. Figure 4.11-(b) shows the measured standard deviation of the measured hits as a function of N. Increasing the number of hits from 300 to 1000 reduces the standard deviation of the measured percentage from 1.8% to 0.79%. With very large

number of hits and at the cost of more hardware and time, the jitter measurement uncertainty can be reduced such that noise of the jitter measurement circuit dominates the uncertainty.

4.3.3 Jitter Minimization Algorithms

The simulation results in Section 4.1 shows that the total jitter due to the combined VCO and input noise has only one global minimum without any local minima for a range of loop parameters that the PLL is stable. Jitter can be dynamically minimized by an algorithm that descends the gradient. However, the jitter measurement uncertainty can degrade the performance of the descent algorithm or cause the algorithm to fail. To understand how the uncertainty affects the algorithm, a table-lookup method is first discussed. Then, a descent algorithm with proper initialization is described.

1) Table Comparison Method:

The simplest jitter minimization method is to use a brute force table lookup. By measuring the jitter for all values of I_{CP} and R during a system calibration, the results in the table can be compared to find the global minimum. The method adapts to the jitter environment only during explicit calibration periods. Figure 4.12 shows a table-lookup measurement only due to VCO noise as input clock is supplied from a clean signal generator with long-term rms jitter less than 1ps. Figure 4.12-(a) on the left illustrates a contour of the measured hits (as a percentage of total hits) for each loop parameter setting with the large total hits, N=30khits. The minimum jitter, shown in the figure, is in agreement with the absolute minimum from simulation. Reducing the number of total hits

results in greater measurement uncertainty. The minimum value, from measuring all table values once, may deviate from the absolute minimum. The contours, overlaid in the figure on the right, indicate the range of possible minima for smaller number of hits (N=300 and N=3000). As expected the contour for N=300 is larger than N=3000. Figure 4.12-(b)

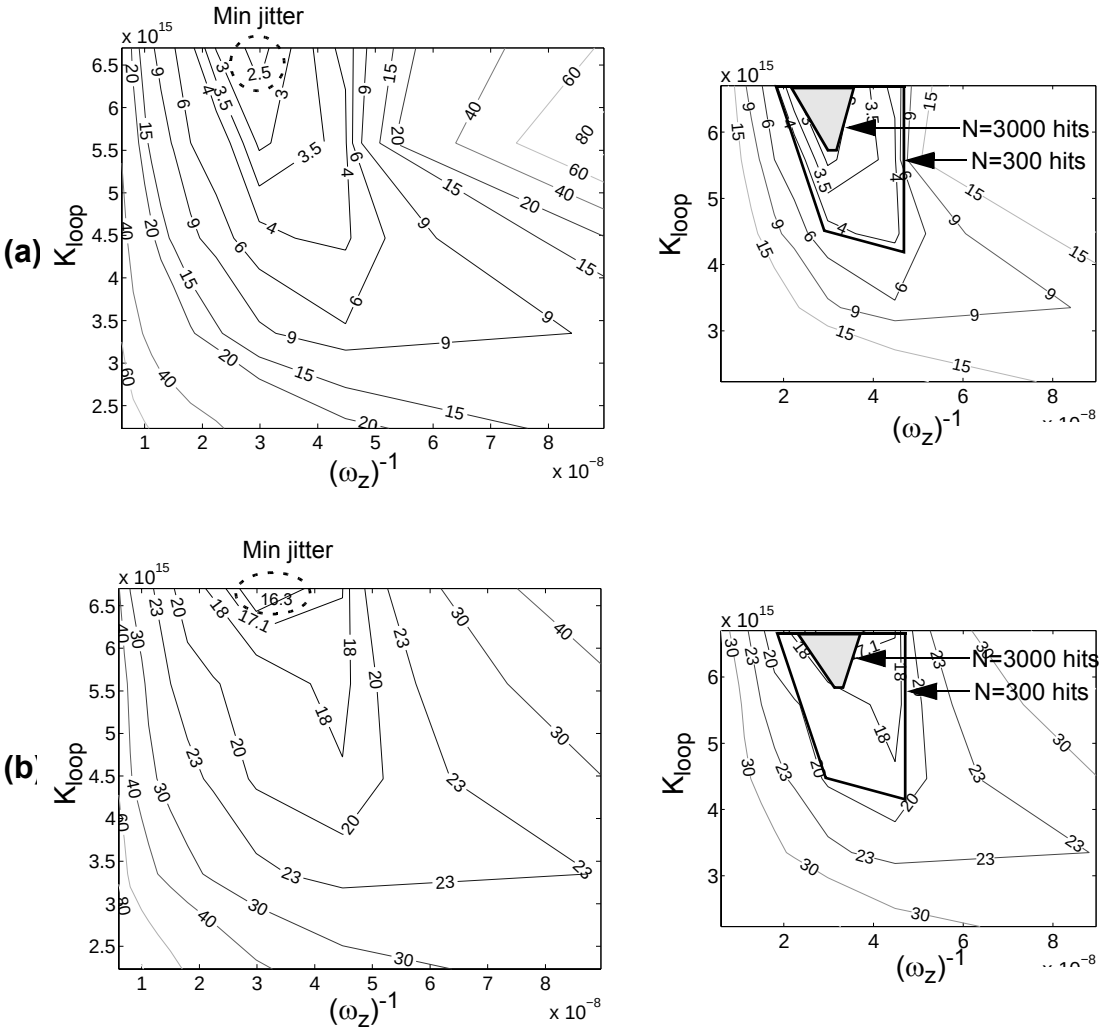


Figure 4.12: Jitter measurement contours (due to VCO noise) for all loop parameters with (a) constant dead-zone width and measuring hits (percentage), (b) constant 4% measured hits and measuring dead-zone width

illustrates the same measurement by finding the width of the dead-zone with 4% of the hits outside the zone. The contours represent actual measured jitter in picoseconds. The impact of N is also overlaid in the figure on the right. Notice that the methods yield essentially the same results. The added uncertainty for using an N>3000hits gives reasonably small uncertainty of <2ps or <10% of the minimum jitter.

Similar measurements are made to show the impact of the combined input and VCO noise for all measurements. In the test setup, the VCO noise is mainly due to the thermal noise whereas the input noise is adjustable. Figure 4.13 illustrates the contours (in ps) for a large number of hits (N=30khits). The figure illustrates the case that the input jitter is dominant. As the input noise is reduced, the minimum jitter moves upward toward the minimum jitter point shown in Figure 4.12, in agreement with the simulation results. The contours, overlaid in the figure on the right, illustrates the range of possible minimums for smaller number of hits (N=3000 and N=300). The added uncertainty for using N=300 is 9ps (20% minimum jitter). For N>3000, the added uncertainty is <5ps (<10% minimum jitter). It should be noted that the local minimum seen in Figure 4.13 is due to the errors of measuring dead-zone width. The local minima appears where the long-term jitter difference between neighboring loop parameters is less than the measurement error. The measured long-term rms jitter does not show any local minima.

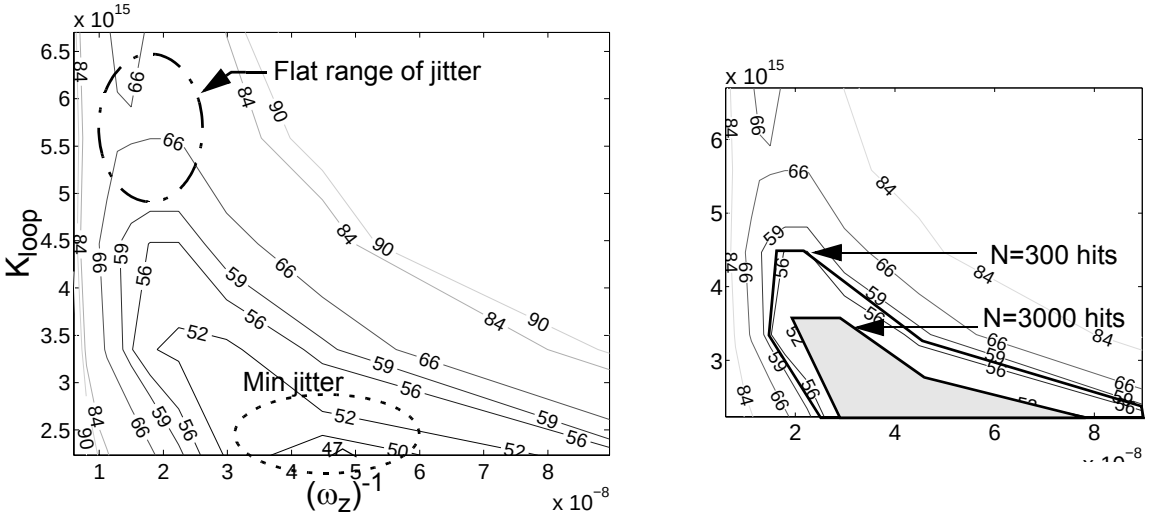


Figure 4.13: Jitter measurement contours (due to input noise) for all loop parameters with constant 4% measured hits and measuring dead-zone width

It is important to note that the variable range of the loop parameters is bounded. With excessively large adjustment range, the loop may become unstable and lose lock at the extreme values. If the loss of lock occurs during calibration, it can be detected by observing the PLL control voltage and the PLL can be forced to reset. However, the range must be bounded if a digital step is taken while the system is active.

2) Gradient Descent Method:

Without any local minima, jitter can be dynamically minimized by an algorithm that descends the gradient. This allows dynamic jitter minimization during run-time with significantly fewer measurements. However, as shown in Figure 4.13, measurement uncertainty at relatively flat regions of the jitter surface causes difficulty for the algorithm

to converge to the minimum. Proper initialization can improve the performance. One viable choice is to use the table-lookup results from system calibration as the starting point. In another option, the algorithm could be initialized with lower loop gains (K_{loop} = 2.5e15 in Figure 4.13). If the loop is dominated by input noise, the initialized value is close to the optimum. If instead VCO noise dominates, the steeper slope of the surface at lower loop gains (as shown in Figure 4.12) allows a rapid descent to the correct minimum jitter (at a high loop-gain setting).

Figure 4.14 shows the flow chart for a descent jitter minimization algorithm. First, the PLL is initialized to the starting values of the loop parameters (R[n], I_{CP}[m]) and the output clock jitter is measured. The width of the dead-zone is also initialized. Based on the nearest neighbor measurements, the algorithm chooses the direction of descent for the first loop parameter (R). The first loop parameter (R) is swept until the minimum jitter is found while keeping the second parameter (I_{CP}) constant. Then the algorithm chooses the direction of descent for the second loop parameter (I_{CP}) starting from (R[k], I_{CP}[m]). The minimum jitter for the second parameter is found for a fixed first parameter. The algorithm repeats alternating between the two loop parameters. Several flags are used to keep track of the neighbors to check.

Since the loop-parameter adjustments are digitally quantized, the curvature of the jitter function, in particular as a function $(\omega_z)^{-1}$, may be large enough such that the descent gradient needs to be diagonal. The algorithm is designed to check diagonal neighbors once

a minimum is reached. An alternative is to add two more digital bits to reduce the non-linear quantization levels at larger R values (Figure 4.6).

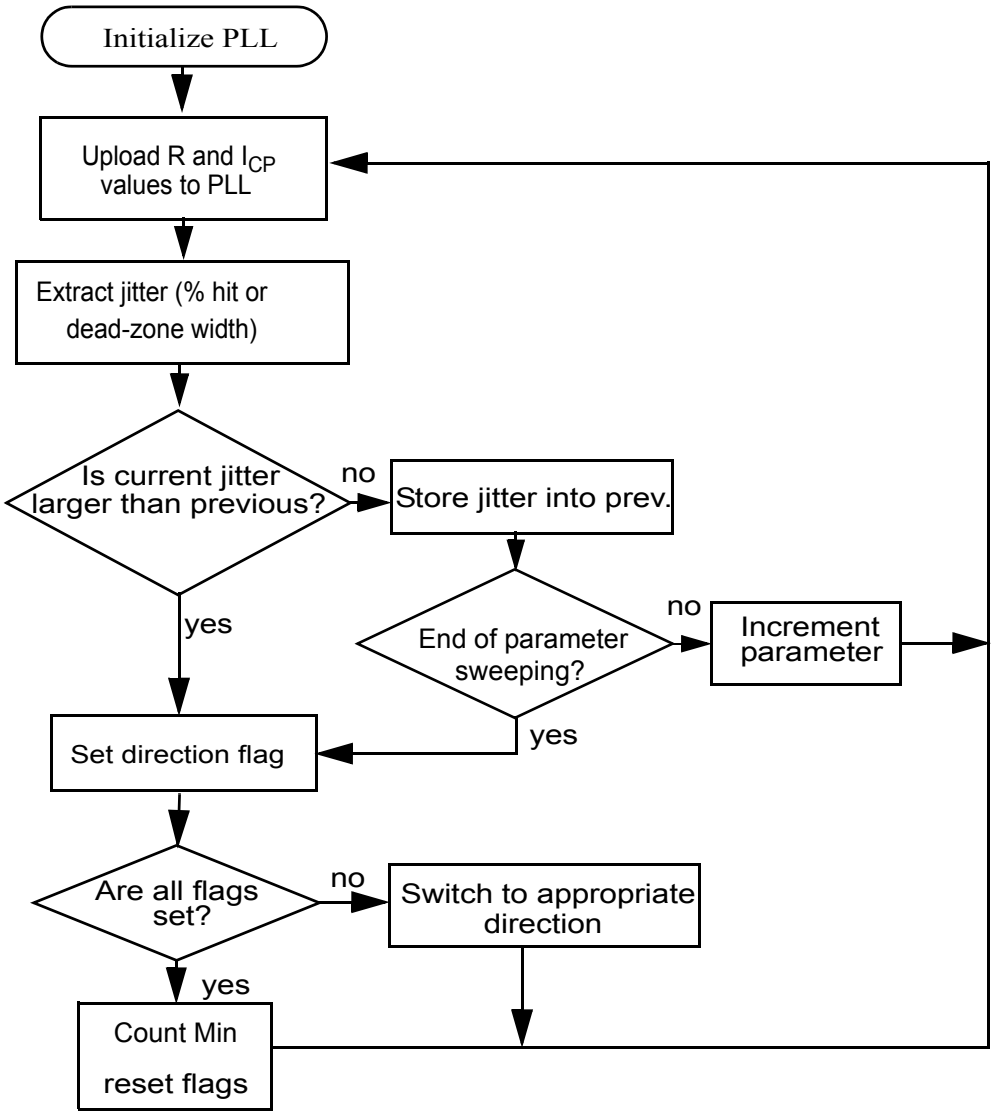


Figure 4.14: Flow chart of jitter minimization algorithm

Similar to the jitter optimization with a table method, the algorithm converges to a range of possible loop-parameter settings. Figure 4.15 shows the histogram of the

movement of the algorithm. The z-axis indicates the number of times the algorithm lands at each loop setting. For N=3000 hits, the minimum jitter mostly occurs at the global minimum while for N=1000 the minimum jitter moves over several loop settings as the algorithm runs. The method converges to the minimum jitter that is higher than the absolute jitter by <10% for N=3000 (or <20% results for N=300).

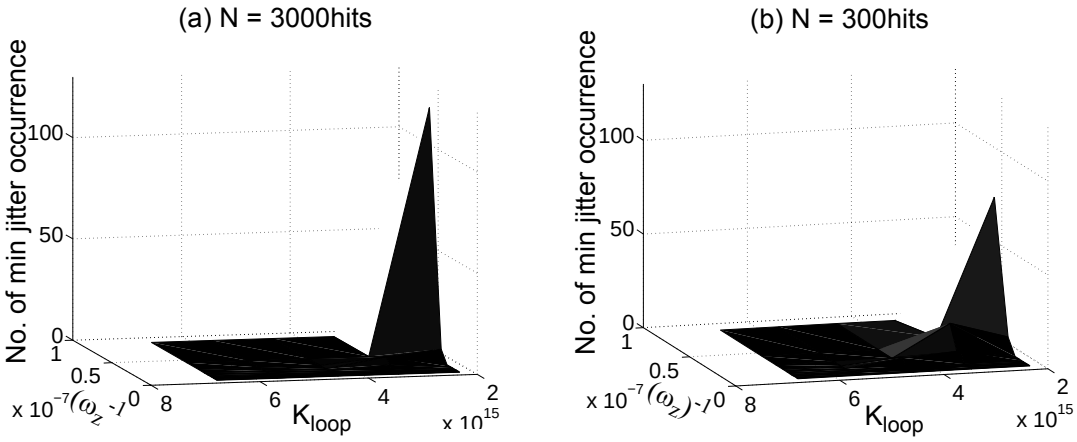


Figure 4.15: Measured minimum jitter due to the sum of VCO and input noise for (a) 3000hits, (b) 300hits

4.4 Design Considerations

The two previous sections discussed the algorithms that optimize the operating parameters of the PLL for minimum jitter. The jitter analysis and measurements reveal several key considerations when implementing the algorithm. First, for a rapid convergence, the starting point of the algorithm is important. Calibrating the system upon startup with a table of measurements would produce a near minimum initial point. As long as the noise conditions change slowly, the algorithm will safely adapt. Second, the long-

term jitter magnitude must be measured and not cycle-to-cycle jitter. Adapting loop parameters based on short-term jitter may result in an unstable loop. The paper shows that a dead-zone phase detection circuit suffices as a measuring circuit for data-recovery applications. However, the measurement uncertainty limits the performance. Due to the uncertainty, an algorithm would result in an operating point that wanders over a region. Choosing a total number of hits >3000 produces reasonable results of <5ps of added jitter from the uncertainty. The implication of a large number of hits is that ≥12-bit accumulator and long measurement intervals are needed.

A third issue in implementation is that the PLL needs to have adaptable loop parameters and careful implementation is needed. Varying the loop parameters could cause static phase offsets which would shift the dead-zone window and cause measurement errors. In particular, with programmable charge-pump currents, dynamic current mismatches due to output impedance variation must be accurately compensated. Switching of digital-parameter settings will inevitably inject charge that is often proportional to the step-size. The injected charge must be sufficiently small so that the loop can track and not lose phase lock. To ensure that the jitter measurements circuits collect steady-state jitter information, a delay is needed between changing the loop parameters and collecting hits. The waiting period corresponds to the time needed for the loop to settle well within the measurement uncertainty at the worst-case parameter settings. Since the tracking depends on the loop bandwidth, an intelligent implementation would adapt the waiting period based on the digital loop-parameter settings.

An implementation using digitally-programmable loop parameters gives the greatest flexibility in the design of the algorithm. A fourth issue to consider is the range and the digital quantization of the loop parameter. The range must be bounded by the ability of the loop to remain in lock especially if the algorithm operates when the system is active. The quantization or resolution of the parameter adjustment has a similar constraint. Typically, large quantization steps results in long waiting period and the risk of losing lock. The resolution of 4-bits and 3-bits of the design shown in this paper, that provided 10x and 3x range for resistor and charge-pump current, is sufficient. However, the 4-bit resolution for resistor is not fine enough to avoid being trapped in a false minima. Although the jitter function is relatively flat over the minimum, the algorithm may become stuck in the descent. The previously shown algorithm checks diagonals to alleviate the problem. A more robust and less complex solution is to increase the resolution by two bits.

4.5 Summary

This chapter demonstrated a run-time technique that minimizes jitter at a PLL output clock. This work addresses the considerations in the design of the PLL and the on-chip jitter measuring circuit. Based on design considerations and jitter analysis results, an algorithm is implemented and experimentally verified that optimizes the operating parameters of the PLL to accommodate a changing noise environment. Without adapting the loop parameters and not knowing noise conditions a priori, jitter can considerably be higher than the minimum. This work shows that jitter of a PLL can be minimized to within 10% of the minimum jitter.

This chapter concludes the jitter minimization method based on the PLL loop parameters. To design a low-jitter PLL, individual blocks in a PLL should also be designed with high immunity to noise. Due to switching activities in large digital systems, power-supply or substrate noise, in particular, are of concern. The next two chapters presents innovative circuit techniques in implementing PLL components and clock buffers with high-noise performance.

Chapter 5

Design of PLL Components

Meeting the jitter requirement in high-performance digital systems requires design of low-noise PLL components in the presence of power-supply or substrate noise. Supply/ substrate noise perturb the most sensitive blocks in a PLL such as voltage-controlled oscillators (VCOs) which can significantly degrade the jitter performance of the PLL. Prior state-of-the-art designs implement VCOs with high immunity to supply or substrate noise with the cost of power and area. This research focuses on a new filtering technique in the design of a VCO. The primary goal is to achieve similar noise performance as prior designs but with less power and area overhead.

To accommodate further power optimization [15]-[16] and testability[1], this work focuses on the design of PLL that operates over a wide frequency range with adaptive bandwidth. To accomplish the adaptive bandwidth, this research employs self-biased

1. Wide operating frequency range allows to test a microprocessor or implement multi-rate links

techniques in the design of the loop filter. The loop filter is also designed with digitally controllable loop parameters to allow further jitter optimization as discussed in Chapter 3 and Chapter 4. This work also addresses the limitation of the conventional PFD. It proposes new circuit techniques for design of high-performance PFDs that achieve larger lock-in range with lower power consumption.

Section 5.1 demonstrates the proposed charge-pump PLL architecture. Section 5.2 discusses the design of a low-power VCO with high immunity to noise. The design of a self-biased loop filter will be discussed in Section 5.3. The design of high-performance phase-frequency detectors (PFDs) are introduced in Section 5.4. The measurement results are discussed in Section 5.5. The chapter concludes with summary performance of the proposed PLL as it is compared to prior state-of-the-art designs.

5.1 Proposed PLL Block Diagram

Figure 5.1 illustrates the block diagram of the proposed charge-pump PLL. A three-state phase-frequency detector (PFD) is followed by a charge pump filter which produces the VCO control voltage. The VCO is composed of a voltage to current (V-I) converter, a current-controlled oscillator (CCO) and a noise-canceling circuit. The output signal of the VCO passes through a low-to-full swing (L-F) amplifier and feeds back to the PFD through a frequency divider.

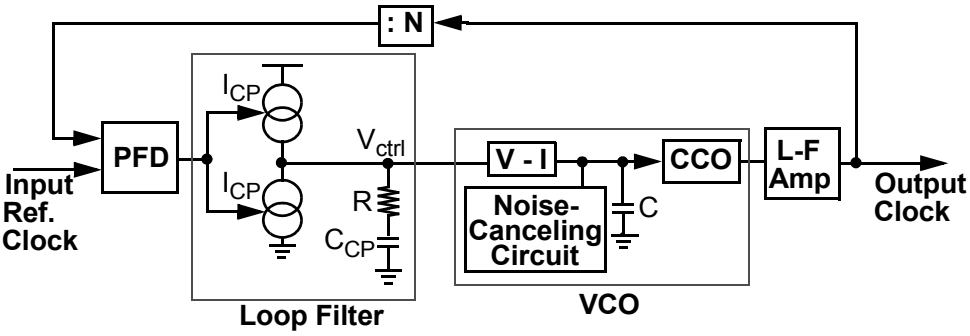


Figure 5.1: The proposed PLL architecture

The primary design goals for the proposed PLL are: 1) to achieve high supply/ substrate noise rejection with adding a noise-canceling circuit to the VCO, 2) low power and low area, and 3) to operate over a wide frequency range with an adaptive bandwidth. The design of each PLL component is discussed in the following sections.

5.2 Design of a Voltage-Controlled Oscillator

Among all PLL components, the design of a low-jitter VCO is the most critical one because any noise coupled into the VCO control voltage is directly translated to the change in the oscillation frequency. The change in frequency appears as a phase error which is persistent for the time duration equal to the time constant of the PLL. The jitter accumulation issue becomes more sever for lower loop bandwidths or higher loop frequency multiplications. To remedy the problem, design of a VCO with high immunity to supply/substrate noise is required.

In high-performance digital systems, CMOS delay buffers are typically used to implement voltage-controlled oscillators (VCOs) due to their wide tuning range, portable design and relaxed supply headroom requirement. However, they have high noise sensitivity to their control voltage (or V_{DD}); 1%-delay/1%-V_{DD}. The next two sections discuss several techniques that improve the noise performance of CMOS buffers. First, the advantages and drawbacks of prior design techniques are discussed. The design of the VCO with a new filtering technique will be explained next.

5.2.1 Previous State-of-the-Art VCO Designs

Two common techniques improve supply noise rejection. The first technique is to filter the supply voltage using either a passive or active filter [51]-[55]. Designs in [51]-[52] employ voltage regulators to filter out supply noise (Figure 5.2). Filtering a high-

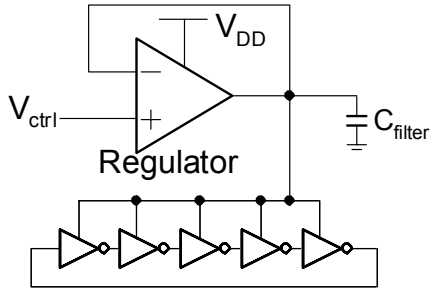


Figure 5.2: Power-supply regulated VCO

frequency supply noise requires a supply coupling capacitor (C_{filter}) [51] that shunts the noise. The capacitor can occupy large area. Alternatively, a high-bandwidth regulator [52] can compensate noise. In addition to regulating supply-voltage, [53] employs a cascode configuration that boosts the output resistance and rejects noise. Similarly, [54]-[55] use a

feedback cascode to boost the output resistance of the V-I converter circuit. Figure 5.3 shows the VCO schematic with a feedback cascode, using an operational transconductance amplifier (OTA) [54]. Although supply regulation and feedback cascode techniques rejects the supply noise significantly, they typically consumes significant amount of power to supply the VCO and clock buffer. A second technique is through

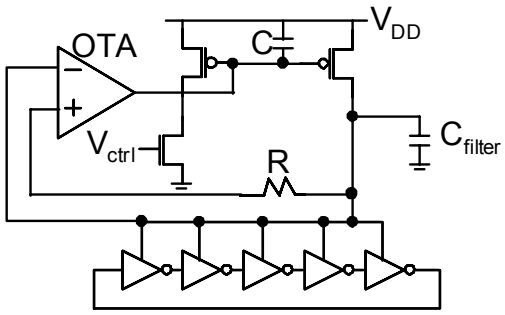


Figure 5.3: VCO with a feedback cascode using OTA

improving the supply sensitivity of VCO elements. A common design strategy employs differential topologies. Differential VCOs and clock buffers ([39], [56]-[57]) demonstrate improved noise performance with respect to single-ended topologies. However, similar to filtering techniques, the differential elements consume significant power, especially in the case of clock buffers.

The next section discusses the design of the VCO with a new filtering technique that reduces supply/substrate noise with less power consumption and area than prior designs.

5.2.2 Proposed VCO Design

The four primary goals in design of the VCO are: 1) high static and dynamic power-supply noise rejection ratio (PSRR), 2) low power and low area, 3) wide operating frequency range, and 4) linear gain for the entire range of the control voltage (V_{ctrl}).

Figure 5.4 shows the proposed VCO design. To achieve a wide operating

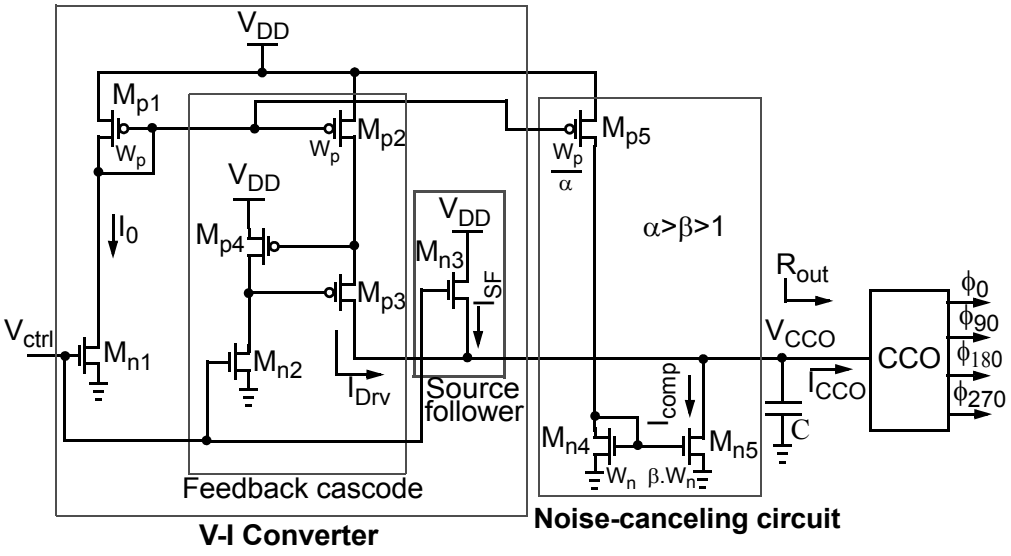


Figure 5.4: Voltage-controlled oscillator with a noise-canceling circuit

frequency range, the design uses a CMOS inverter ring oscillator with controllable supply. Figure 5.5 shows the current-controlled oscillator (CCO) circuit composing of four stages of pseudo-differential CMOS inverters [59]. The design employs negative-skew delay elements to enable the VCO to run faster at a given V_{ctrl}. The CCO produces quadrature

clock phases, making the design suitable for applications such as clock/data recovery circuits and multi-phase systems.

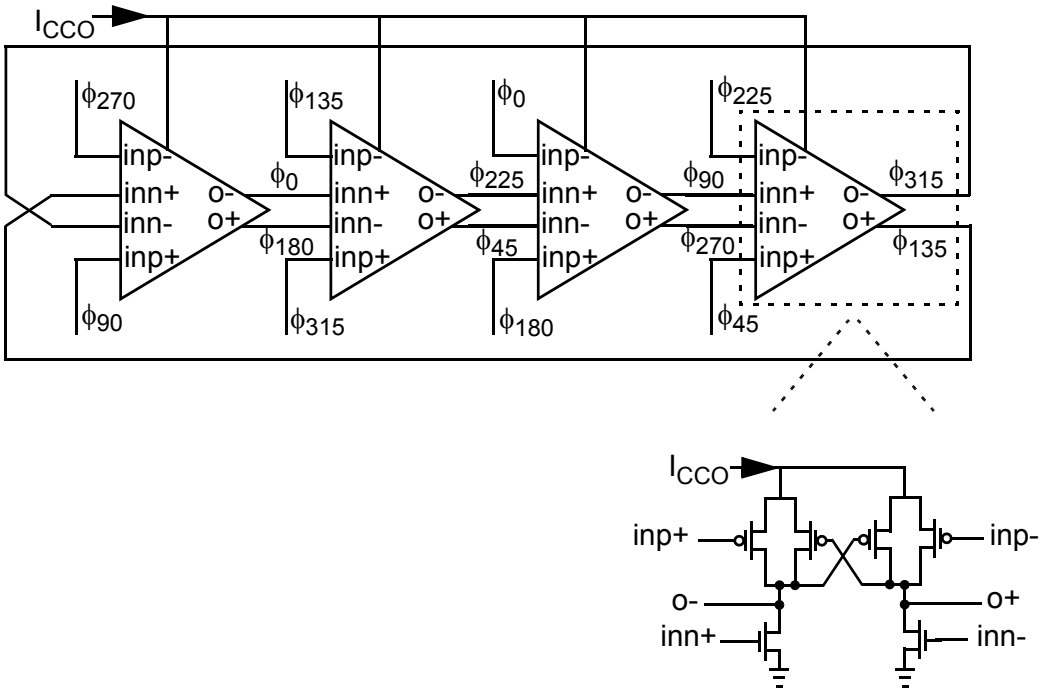


Figure 5.5: Quadrature pseudo-differential current-controlled oscillator (CCO)

The V-I converter circuit, transistors M_{n1}, M_{p1}-M_{p3}, converts the control voltage to current (I_{Drv}) that drives the CCO and controls the frequency of CCO output signal. To maintain linear VCO conversion gain (K_{VCO}), M_{p1}-M_{p3} are designed with large widths for minimum overdrive voltage. The minimum overdrive voltage of PMOS transistors guarantees the linear K_{VCO} due to the fact that M_{n1} stays in saturation for almost the entire range of control voltage VCO, $V_{Tn1} \leq V_{ctrl} \leq V_{DD}$, where V_{Tn1} is the threshold voltage of M_{n1}. However, at a V_{ctrl} that is near V_{DD}, M_{n1} enters triode region which reduces the conversion gain and saturates K_{VCO}. To compensate for the gain drop at high V_{ctrl}, the

circuit uses a source follower transistor (M_{n3}). Source follower is *off* for V_{ctrl} - V_{CCO} < V_{Tn3} and gradually turns *on* at high V_{ctrl} which injects current (I_{SF}) and compensates for I_{Drv} drop.

Figure 5.6 shows the simulated V-I converter gain characteristics for different process corners. The proposed V-I converter achieves the linear gain that varies only by a

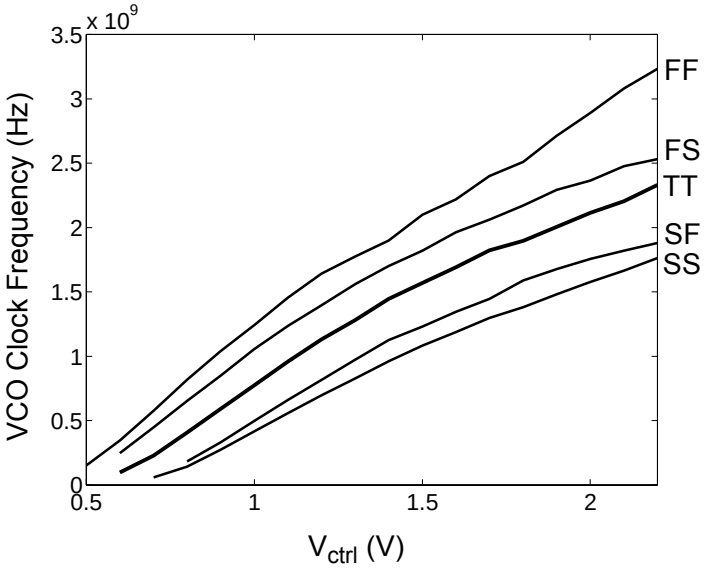


Figure 5.6: Simulated V-I converter gain characteristic across process corners

factor of less than 1.5 for almost the entire range of the control voltage ($V_{Tn1} \leq V_{ctrl} \leq V_{DD}$). For instance, the K_{VCO} varies between 1.15 and 1.7GHz/V at typical corner for $V_{Tn1} \leq V_{ctrl} \leq V_{DD}$. The slight variation of K_{VCO} modestly impacts the loop dynamics. If low-V_T devices were available in the process technology, using one for the follower would further improve the gain linearity at high VCO frequencies. The V-I converter in [60] achieves a linear gain for the entire range of V_{ctrl} ($0 \leq V_{ctrl} \leq V_{DD}$), slightly larger

than the range of this proposed V-I converter. However, the V-I converter in [60] suffers from high power-supply noise sensitivity due to the coupling of V_{ctrl} to both ground and V_{DD}. The gain linearity improvement technique proposed in this work resolves the problem by coupling V_{ctrl} only to the ground reference.

Further supply rejection is achieved by capacitively coupling V_{CCO} to ground. The capacitor and output resistor (R_{out}) at V_{CCO} forms the third pole of the PLL and filters the high-frequency noise. The cascode current source that supplies I_{Drv} uses a feedback circuit (M_{p4} and M_{n2}) to boost the output impedance [55]. The resulting supply noise sensitivity is 0.2%-VCO frequency/1%-V_{DD} because the finite output resistance of M_{n1} causes I_{Drv} to vary with supply. An auxiliary noise-canceling circuit (M_{p5}, M_{n4} and M_{n5}) is added to compensate the residual variation of the output current (I_{Drv}) due to supply noise. This circuit generates a compensator current, I_{comp}, by mirroring a fraction of I_0. I_{comp} is then subtracted from I_{Drv}. The current to the CCO is $I_{CCO} = I_{Drv} - I_{comp}$ for $V_{ctrl} - V_{CCO} < V_{Tn3}$. The ideal supply noise cancellation occurs when I_{Drv} variation is equal to I_{comp} variation due to V_{DD} noise, i.e. $\Delta I_{Drv} = \Delta I_{comp}$. In other words, when there is no supply-induced variation in I_{CCO}, $\Delta I_{CCO} = 0$. The noise-canceling circuit is designed to have a much worse supply sensitivity than the feedback cascode circuit that generates I_{Drv}. The noise-canceling circuit uses a single device without the feedback cascode and with minimum channel length. The simulation result shows that I_{comp} is 4 times more sensitive to V_{DD} variation than I_{Drv}, i.e. $\frac{\partial I_{comp}/\partial V_{DD}}{\partial I_{Drv}/\partial V_{DD}} = 4$. By setting the ratio of the mirroring, β/α, to the ratio of the supply sensitivity of the currents ($\frac{\beta}{\alpha} = \frac{1}{4}$),

ΔI_{Drv} will be equal to ΔI_{comp}[1]. The power penalty to source the same I_{CCO} for a given V_{ctrl} is 40%. The proposed VCO consumes 2mW at 1GHz.

To verify the noise performance of the proposed V/I converter, the dynamic response of V_{CCO} to supply noise is simulated. The curves (1) and (2) shown in Figure 5.7 demonstrate the V_{CCO} response for the V/I converter without and with the noise-canceling circuit, when a -10% V_{DD} step with 100ps slew rate inserted at t=2ns. Adding the noise-canceling circuit to the V-I converter improves the PSRR by 6dB for very high frequency

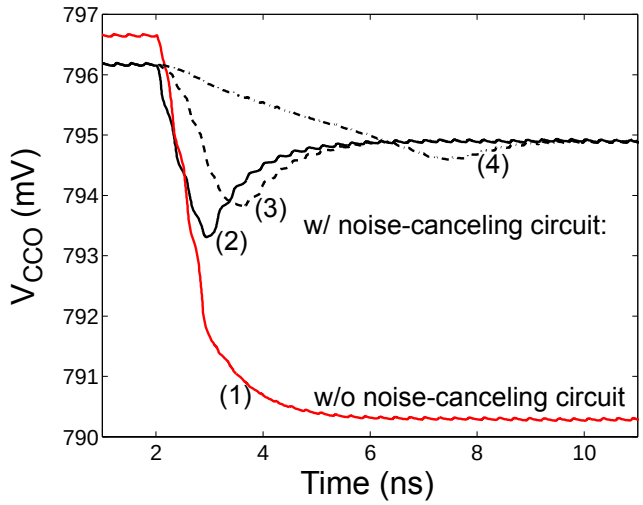


Figure 5.7: V_{CCO} response of V-I converter to -10% V_{DD} step inserted at t=2ns

noise. Increasing the slew rate of V_{DD} step from 100ps to 1ns and 5ns (curves (3) and (4)) improves the dynamic PSRR of the V-I converter with noise-canceling circuit to 8dB and

1. Adjusting β/α alleviates any output impedance variation over the process corners. The simulation results indicate that the proposed VCO maintains its noise rejection performance at the process corners by adjusting the value of β/α, $3/16 \leq \beta/\alpha \leq 1/4$.

12dB, respectively. Also, the bandwidth of the feedback cascode current source of this design is sufficiently high to correct the high-frequency supply-induced noise in V_{CCO}. This bandwidth is larger than 20x the loop bandwidth of the PLL. For DC supply noise, the PSRR is improved by more than 15dB. Equivalently, the supply sensitivity of VCO frequency is improved from 0.2%-f_{VCO}/1%-V_{DD} (for the V-I converter without the noise-canceling circuit) to ≤ 0.035%-f_{VCO}/1%-V_{DD} (for the V-I converter with noise-canceling circuit).

At very high frequencies, M_{n1} enters triode region, which increases ΔI_{Drv} beyond the available ΔI_{comp}. Therefore, the supply sensitivity of the VCO degrades at high control voltages similar to regulated VCOs and differential VCOs. At very low control voltages, the supply sensitivity also degrades due to greater susceptibility of the CCO to noise. While the VCO has an operating range of 200-2300MHz, the simulation results indicate that the VCO achieves the supply noise rejection of ≤ 0.035%-f_{VCO}/%-V_{DD} over a smaller range of 400-2000MHz in the typical corner.

5.3 Loop Filter

Loop filter for a charge-pump PLL composes of a capacitor, C, that the charge-pump injects the charge into or out of it. To stabilize the system, as discussed in Chapter 2, a zero should be introduced by adding a resistor, R, in series with the loop filter capacitor. Figure 5.8 shows the conventional loop filter, implemented with constant and linear RC. To guarantee the loop stability under varying process or operating frequency, PLLs with the conventional loop filter achieve a constant and relatively low bandwidth. The low

bandwidth results in a poor tracking jitter performance due to VCO noise as discussed in Chapter 3.

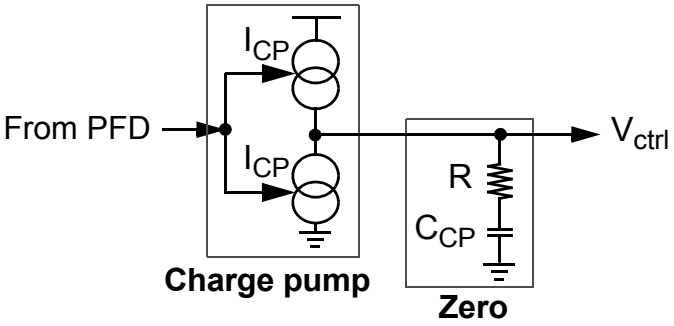


Figure 5.8: Conventional loop filter

To maximize the loop bandwidth over the operating frequency range requires that the loop gain tracks the operating frequency. In order to maintain the loop stability, the zero should also track the operating frequency such that the loop bandwidth scales with the operating frequency in a constant phase margin. For a second-order PLL, damping factor, ζ, and natural frequency, ω_n, are calculated from Equation 2.4 and Equation 2.8:

$$\begin{aligned} \zeta &= 0.5 \cdot R \cdot \sqrt{K_{loop}/N} \\ \omega_n &= \frac{2 \cdot \zeta}{R \cdot C_{CP}} \end{aligned} \qquad [4.1]$$

where $K_{loop} = K_{PFD} \cdot K_{VCO} \cdot I_{CP}/(2\pi C_{CP})$. Equation 4.1 suggests that $R \cdot \sqrt{I_{CP}}$ should be kept constant over the operating frequency range for a constant ζ (or equivalently constant phase margin). With a constant ζ, ω_n (or equivalently the loop bandwidth) varies inversely with R.

The designs proposed in [52], [55] and [57] employ self-biased techniques to achieve an adaptive bandwidth PLL with a constant phase margin. These designs

implement the resistor through active components. Figure 5.9 shows an adaptive loop filter [52] that uses two charge-pump currents to implement the resistor: $R = \frac{I_{CP-proportional}}{I_{CP-integral}} \cdot \frac{1}{gm_{Reg}}$

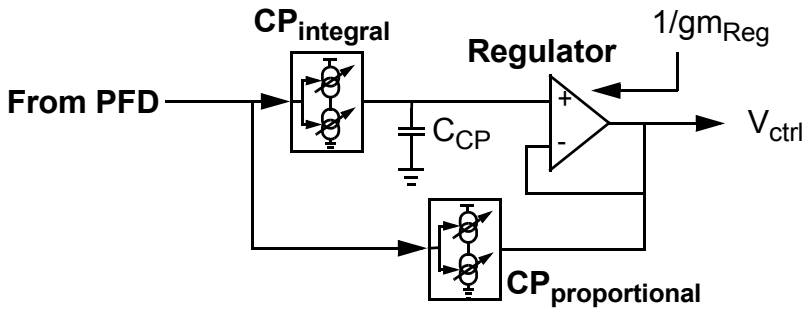


Figure 5.9: Implementing the PLL stabilizing zero with two charge-pump currents and a regulator

5.3.1 Proposed Loop Filter Design

Our proposed loop filter, shown in Figure 5.10, is composed of: 1) charge pump circuit, 2) loop stabilizing zero and 3) a third pole. The design is similar to [52], [57], and [55] in that the loop characteristics track the VCO operating frequency such that the loop bandwidth scales with operating frequency in a constant phase margin.

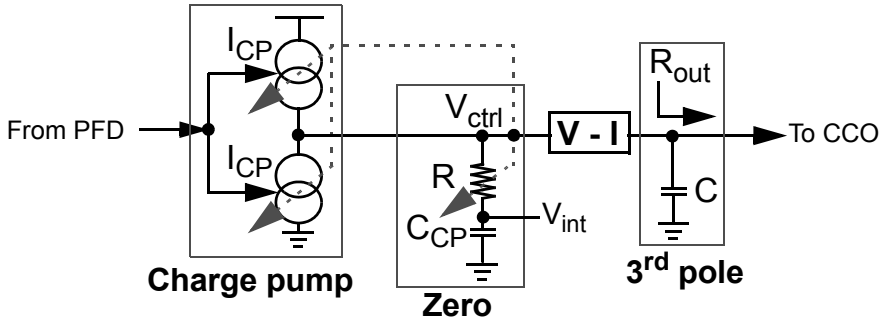


Figure 5.10: Proposed loop filter architecture

The charge pump uses a similar structure as [52] where it is self-biased with the VCO control voltage (Figure 5.11). Therefore, the charge-pump current scales with the

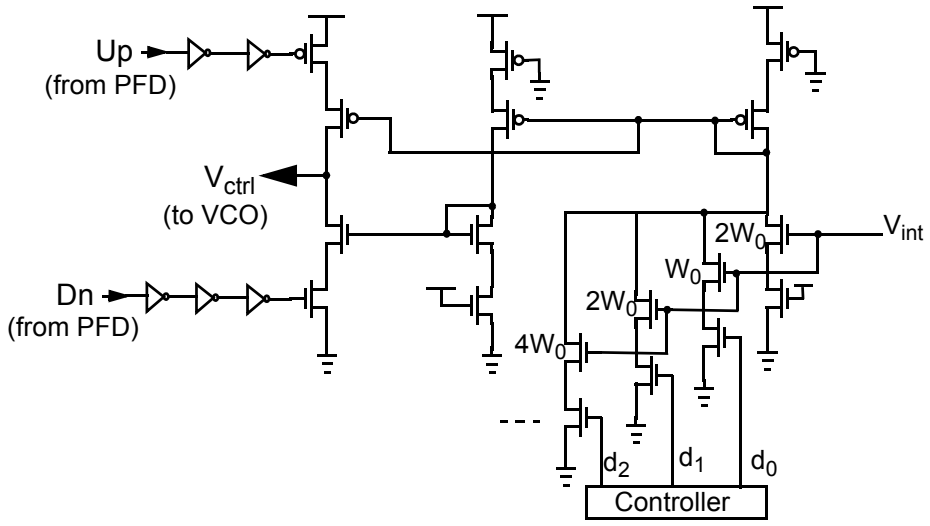


Figure 5.11: Charge-pump current circuit

PLL operating frequency. The series of a resistor and a capacitor forms the loop stabilizing zero. The design implements the resistor and capacitor with a MOS channel resistance [62] and a MOS capacitor, respectively, as shown in Figure 5.12. The MOS resistor is

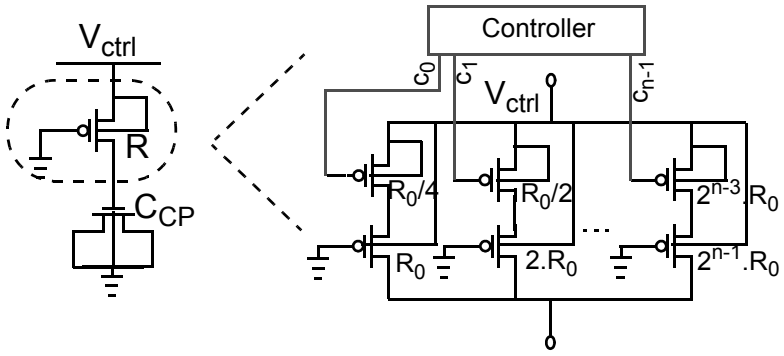


Figure 5.12: Loop stabilizing zero with a 4-bit controller (n=4)

biased by the VCO control voltage so that the loop zero scales with the PLL's operating frequency. The proposed circuit achieves the scalable zero with a modest improvement in power and area upon the previous designs ([52] and [57]) that use an additional charge-pump to inject current in a feed-forward path.

D/A converters in Figure 5.11 and Figure 5.12 adjust the charge-pump current and MOS resistor to allow further loop-parameter adjustments to optimize jitter at the output clock as discussed in Chapter 3 and Chapter 4. The area overhead due to a 3-bit controller for the charge-pump current and a 4-bit controller for the loop filter resistor is negligible in comparison with the overall charge-pump area and loop filter capacitor. The tunability of the MOS resistor also provides an additional tuning to adjust the zero position for any process variation of the MOS capacitor.

The switching activity of PFD produces ripple on the VCO control voltage at the same rate as the reference clock frequency. The ripple modulates the VCO frequency resulting in jitter at the output clock. This effect worsens with higher frequency multiplication by the loop. The loop's third pole (formed at the CCO input) filters out the ripple. The third pole also tracks the PLL operating frequency because the output resistor (R_{out}) scales with the oscillator's frequency. With all primary loop parameters adapting to the oscillator frequency, the loop operates with a wide frequency range with a constant phase margin.

5.4 Phase-Frequency Detector

A common architecture for clock generation uses a phase-frequency detector (PFD) for simultaneous phase and frequency acquisition. Generating high frequency clock increases the difficulty of the design of the PFDs particularly for systems with a high input clock frequency and minimum frequency multiplication. As will be described in Section 5.4.1, the speed of the conventional NAND D-flip-flop phase-frequency detectors (PFDs) limits the operating frequency and slows the frequency acquisition. This research proposes two improved PFD designs.

5.4.1 Conventional PFD Design

Figure 5.13 illustrates a common linear PFD architecture using resettable D-flip-

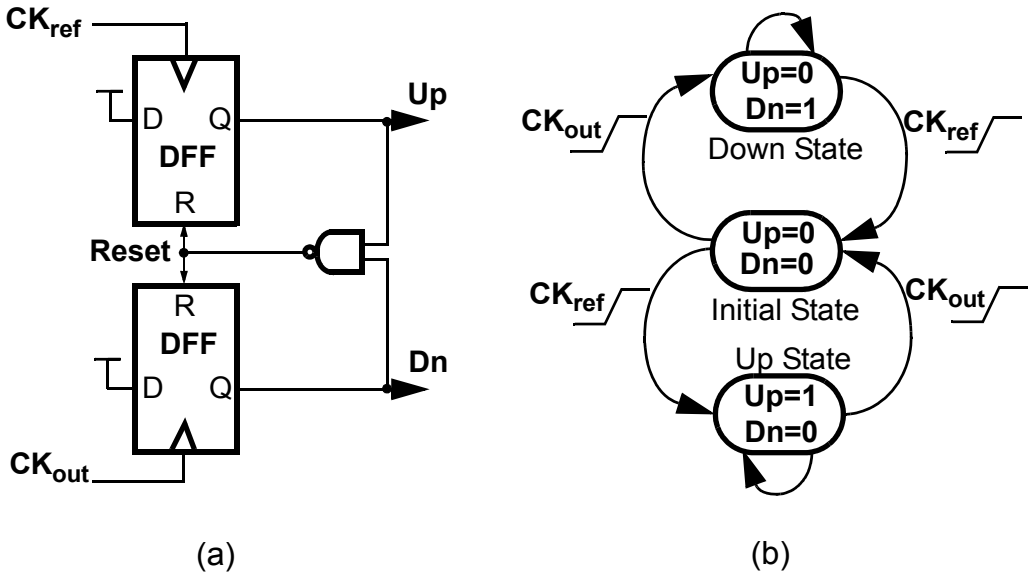


Figure 5.13: (a) Linear PFD architecture, (b) PFD state diagram

flops (DFFs) and its state diagram. This PFD generates an Up and a Dn signal that

switches the current of a charge pump. The DFFs are triggered by the inputs to the PFD. Initially, both outputs are low. When one of the PFD inputs rises, the corresponding output becomes HIGH. The state of FSM moves from an initial state to an Up or Down state. The state is held until the second input goes high which in turn resets the circuit and returns the FSM to the initial state.

The PFD's characteristic is ideally linear for the entire range of input phase differences from -2π to 2π (Figure 5.14-(a)). When the inputs differ in frequency, the phase difference changes each cycle by $2\pi \cdot \frac{(T_{CK_{ref}} - T_{CK_{out}})}{max(T_{CK_{out}}, T_{CK_{ref}})}$ [1]. On every clock cycle during frequency acquisition, the phase difference steps across the PFD transfer curve from 0 to +/-2π and repeats as the output clock cycle slips. The control voltage of voltage-controlled oscillator (VCO) is pumped monotonically toward that of the desired frequency. As the frequency error decreases, the sweep slows until the frequency difference is within the lock-in range. Note that because phase roughly sweeps linearly and that the voltage is integrated, the voltage accumulates quadratically between each slip of the clock cycle. Once within the lock-in range, the cycle slipping stops and the phase is acquired, behaving as a linear system.

However due to the delay of the reset path, the linear range is less than 4π (Figure 5.14-(b)). Figure 5.14-(c) illustrates the non-ideal behavior with the reference clock (CK_{ref}) leading the output clock (CK_{out}) causing an Up output. As the input phase difference nears 2π, the next leading edge (CK_{ref}) arrives before the DFFs are reset due to

1. [1]Phase difference in radians referring to the slower clock frequency

the finite reset delay. The reset overrides the new CK_{ref} edge and does not activate the Up signal. The subsequent CK_{out} edge causes a Dn signal. The effect appears as a negative output for phase differences higher than $2\pi - \Delta$ where $\Delta = 2\pi \cdot t_{reset}/T_{cyc}$ which depends on the reset path delay (t_{reset}) and the reference clock period (T_{cyc}). Note that t_{reset} is determined by the delay of logic gates in the reset path and is not a function of input frequency.

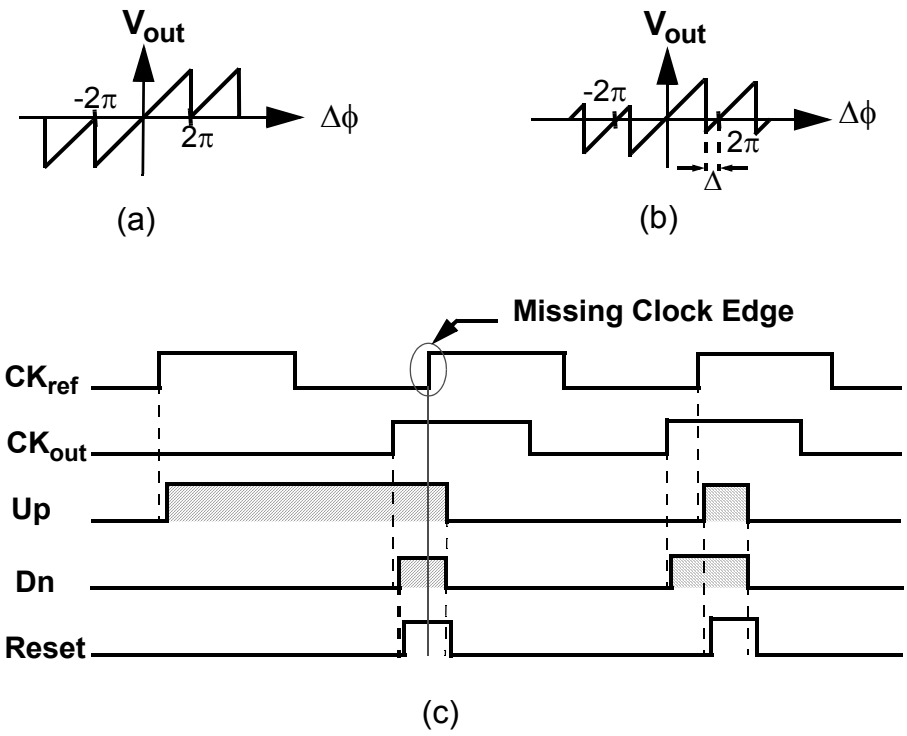


Figure 5.14: (a) Ideal PFD characteristic. (b) Nonideal linear PFD characteristic. (c) PFD nonideal behavior due to nonzero reset delay

During acquisition, the frequency will not monotonically approach lock-in range because the non-ideal PFD gives the wrong information periodically. The acquisition slows by how often the wrong information occurs which depends on Δ. At an input

frequency ($T_{CK_{ref}} = 2 \cdot t_{reset}$) where Δ equals π, the PFD outputs the wrong information half the time and thereby fails to acquire frequency lock unconditionally. The maximum operating frequency can be expressed as $f_{ref} \leq \frac{1}{2 \cdot t_{reset}}$.

A commonly used PFD design is one used in [72] using NAND-based latches to build the D-flip-flops. The reset path includes one 2-input NAND, one 4-input NAND and two 3-input NANDs. We characterize the reset delay by normalizing it with the delay of a fan-out of 4 inverter to remove process/voltage/temperature dependence. The design measures a delay of 5.3 FO-4 thereby limiting the maximum clock period to 10.6 FO-4.

The next two sections describe two proposed designs that significantly improve the maximum operating frequency of the PFD.

5.4.2 Pass-Transistor PFD Design

The first proposed design is shown in Figure 5.15. The PFD is similar to a dynamic two-phase master-slave pass-transistor flip-flop. Only single-edge clocks are used to minimize clock skew. As both outputs become HIGH, the slave is reset asynchronously while the master is reset synchronously i.e., the reset is allowed only when the slave latch is transparent. Synchronously resetting the master increases the operating range and also reduces the power consumption. If the master latch is reset while it is transparent, then there will be significant short-circuit current, resulting in more power. The synchronized reset transistors (N1 and N4), must be at the bottom of the stack because “RST” is the late arriving signal when the nodes “out” and “ref” are reset. The reset circuit shown in Figure 3 includes one pass transistor, one inverter and one NAND gate. In order to properly reset

the slave, the pass-transistor output should become HIGH before the master becomes transparent. Hence, the NAND gate delay is counted twice in the delay path. The smaller gates in the reset path as compared to NAND FF PFD reduces t_{reset} to 4.4 FO-4 and T_{ref} by 17% to 8.8 FO-4.

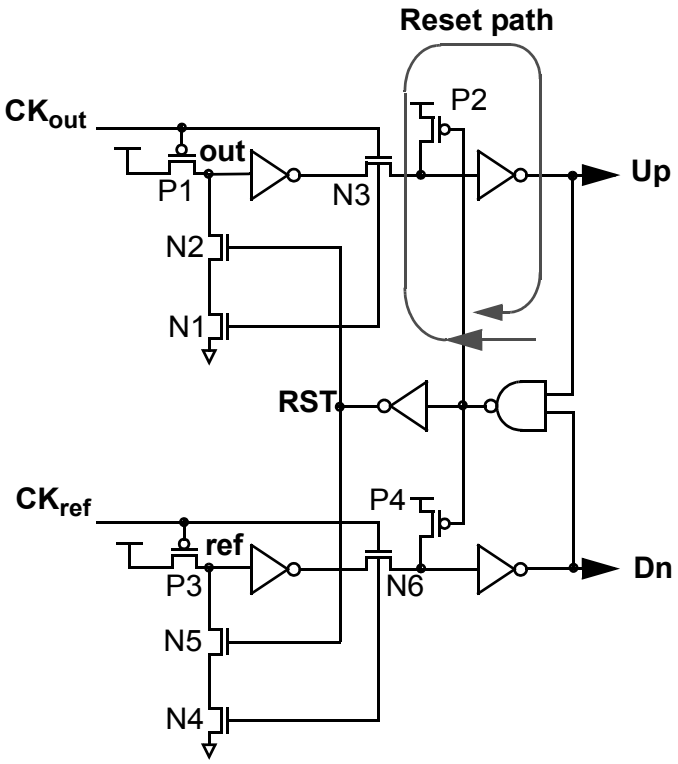


Figure 5.15: Pass-transistor DFF PFD architecture

5.4.3 Latch-Based PFD Design

In the second proposed design, pulsed latches [73] are used instead of flip-flops which fundamentally changes the dependence on the reset delay. This is illustrated in Figure 5.16-(a) with the same case as before. When CK_{ref} arrives during the reset, the edge information propagates to the output as long as CK_{ref} pulse ($Pulse_{ref}$) is still HIGH (level-sensitive) when the reset period ends. The PFD no longer loses the edge that arrives

during reset and does not output the wrong direction. However, since the PFD output becomes active HIGH at the end of the reset (Δ), the output pulse width would be constant (2π-Δ) for phase differences greater than 2π−Δ. The characteristic is shown in Figure 5.16-(b). The input clock pulse widths (W_{in}) should be designed to be slightly smaller than

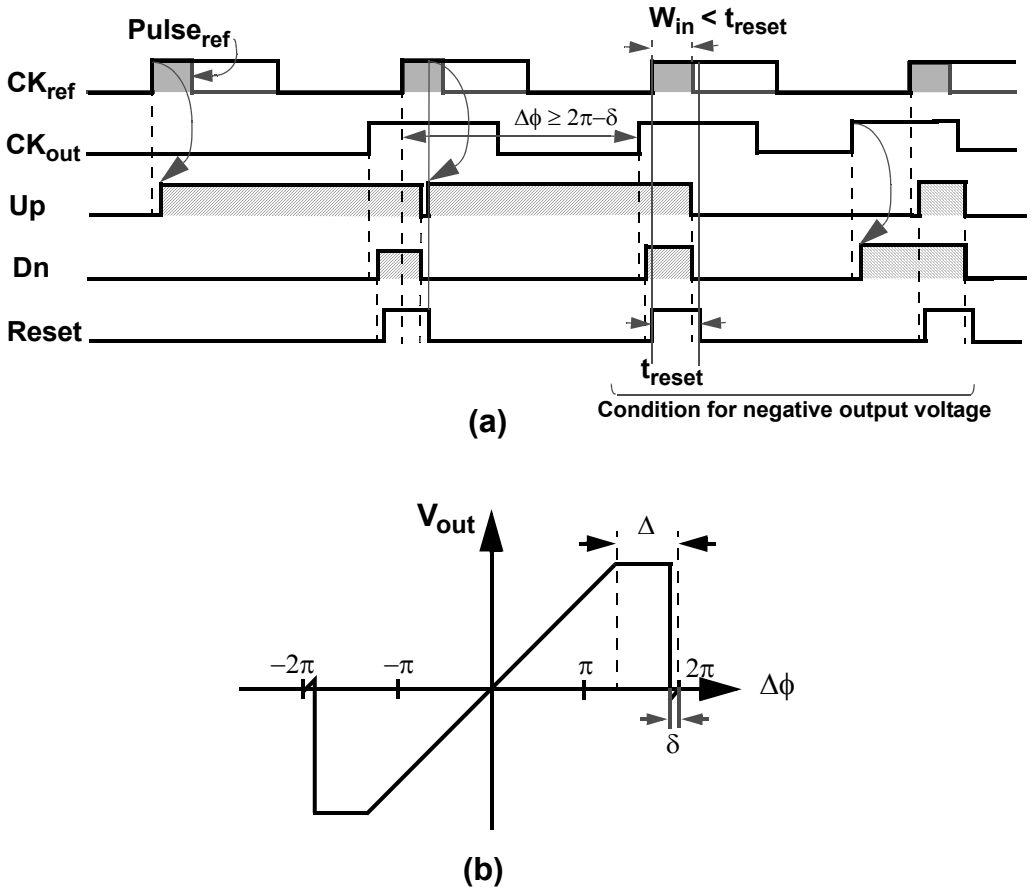


Figure 5.16: (a) Behavior of a latch-based PFD, including the description of the nonideal behavior origin. (b) characteristic of a latch-based PFD

t_{reset}, otherwise the PFD would fail to lock at zero input phase difference. The PFD failure is due to the fact that the input clock pulse that triggers the reset would activate the output after the reset pulse ends for $W_{in} \geq t_{reset}$. This design criteria results in a negative output

voltage for $|\Delta\phi| \geq 2\pi - \delta$ as illustrated in Figure 5.16-(a) and (b). Note that this PFD has faster acquisition rate compared to the first type (with the same operating frequency) because it outputs less incorrect phase information. However, the PFD has a gain that saturates when the input difference is larger than $2\pi-\Delta$.

Figure 5.17 illustrates design of the latch-based PFD [74], using glitch latches. The

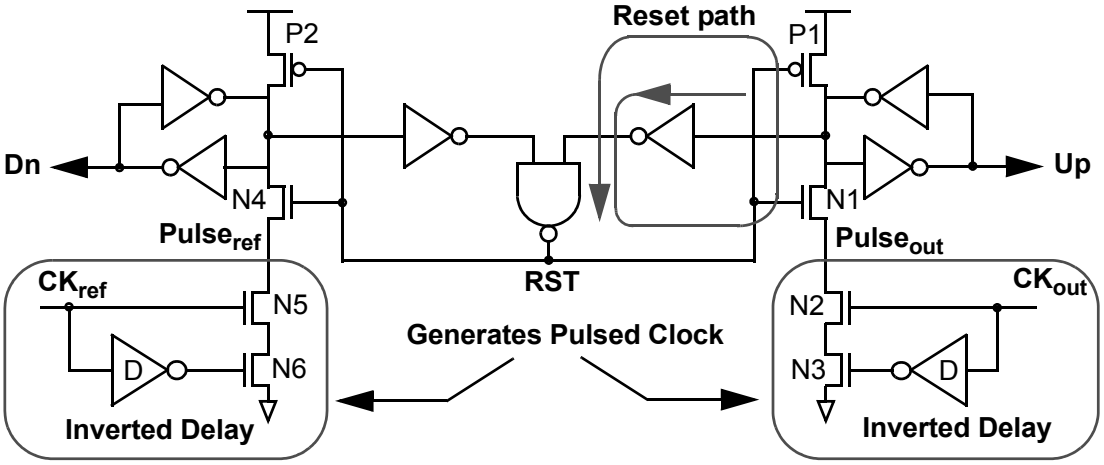


Figure 5.17: Latch-based PFD architecture

delay elements control the pulse width of the clocks. As shown in Figure 5.17, the reset circuit includes two inverters and one NAND. The reset also traverses the circuit twice because the reset should return HIGH. Therefore, t_{reset} delay is roughly 5.5 FO-4 and contains three inverters and two NANDs. As the clock period is less than twice the pulse width, the clock pulses from N2 (N5) and N3 (N6) are no longer constant width but reduce with the period. Therefore δ is no longer constant and grows with increasing frequency. The PFD fails as frequency approaches $\frac{1}{t_{reset}}$ which is potentially twice that of the previously proposed PFD for the same t_{reset}. Consequently the maximum frequency is

higher than the DFF-based designs despite longer t_{reset}. The higher performance is at a cost of 3x the power as compared to the first proposed circuit due to DC current and extra power consumption in the delay circuit. When the reset node and clock inputs are simultaneously LOW and HIGH respectively, the DC current flows through N1, N2, N3 and P1 or (N4,N5, N6 and P2).

It should be noted that the first PFD design (Figure 5.15) can also be converted to latch-based type PFD by adding a delay cell to the gate inputs of P1 and P3 transistors. The delay allows P1 and N3 (P3 and N6) to both conduct briefly, behaving like a glitch latch. This new design has the similar functionality as PFD in Figure 5.17 in terms of frequency acquisition, maximum operating frequency and power.

5.4.4 Simulated Transfer Curve of PFDs

Figure 5.18 illustrates the simulated transfer curve of NAND DFF PFD and two proposed designs (Figure 5.15 and Figure 5.17) for reference clock of 435 MHz ($= 1/(10 \bullet FO-4)$). Figure 5.19 compares the simulated frequency acquisition for three PFDs, starting the VCO at 375 MHz and locking at 800 MHz. As expected, the PLL with latch-based PFD has the fastest frequency acquisition among the three PFDs.

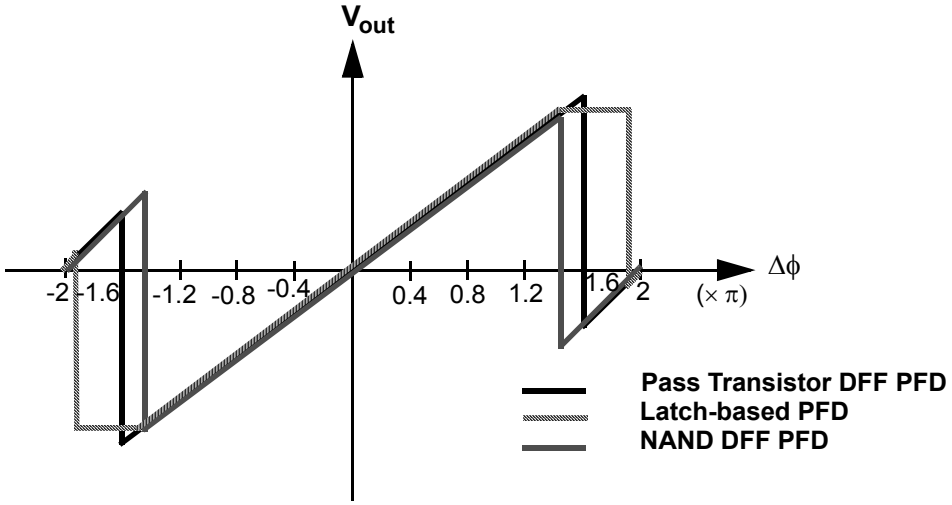


Figure 5.18: Characteristics of three PFDs at 435MHz

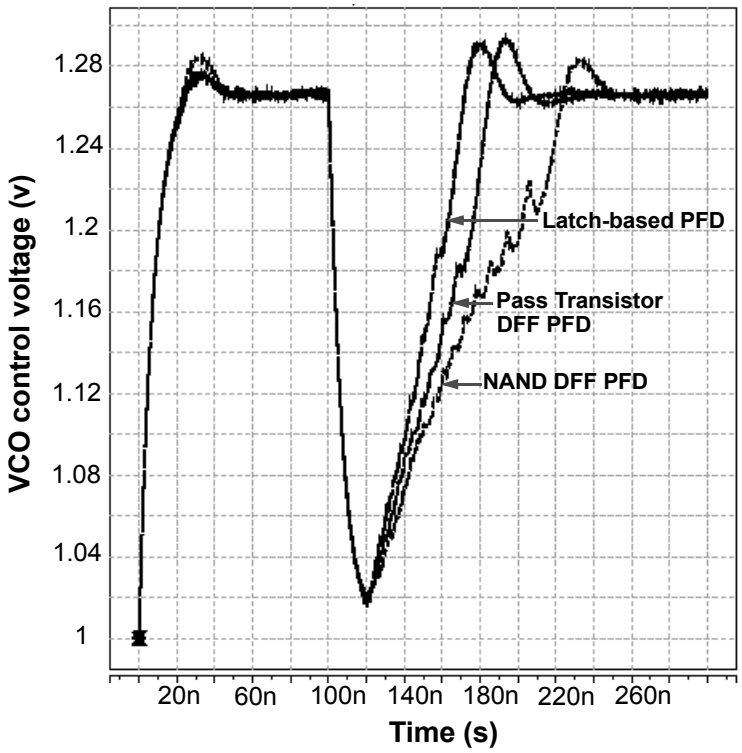


Figure 5.19: Simulated frequency acquisition

5.5 Measurement Results

The PLL and clock buffer[1] have been designed and fabricated in a 0.25-µm CMOS technology. As shown in the chip micrograph, Figure 5.20, the PLL core area is 0.028mm^2 (120µm x 230µm).

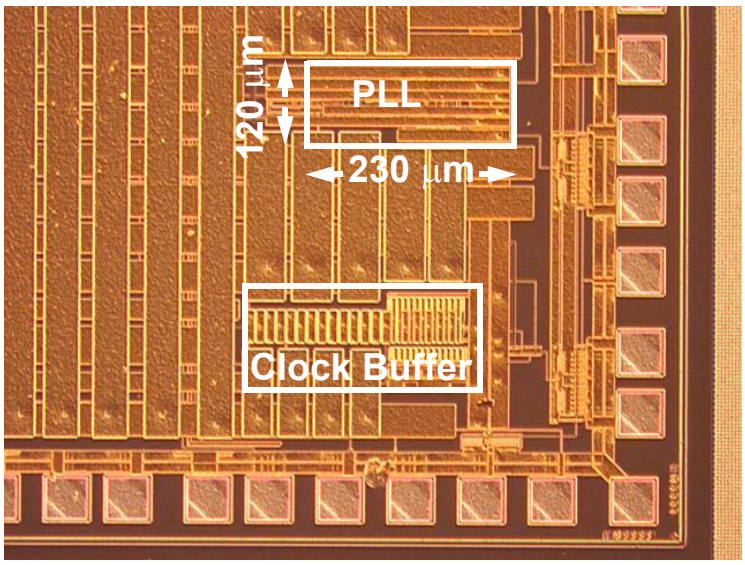


Figure 5.20: PLL and clock buffer die photograph

The measured VCO operating frequency is 130-1600 MHz. Figure 5.21 depicts the measured VCO gain indicating that the gain varies only between 0.9-1.35 GHz/V for the entire range of control voltage.

1. The design of noise-compensated clock buffer is discussed in Chapter 6

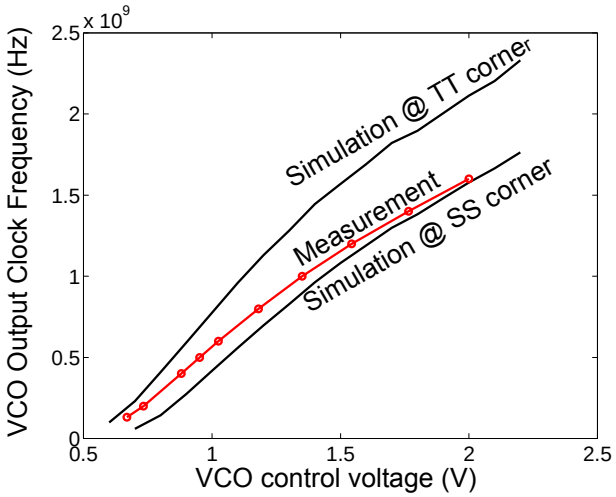


Figure 5.21: Measured and simulated VCO gain

The input reference frequency generated by a signal generator is set to 250 MHz and the loop multiplication factor is four. The long-term jitter performance of the PLL output at 1 GHz is demonstrated in Figure 5.22. The jitter histogram measures the rms

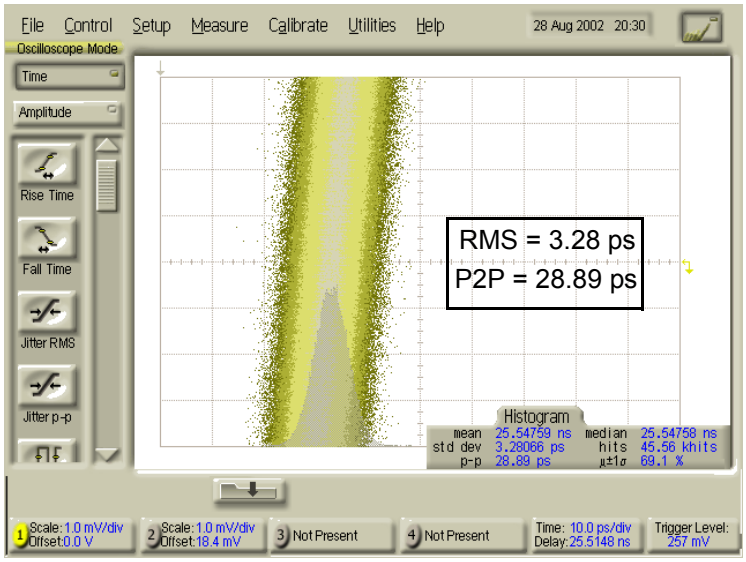


Figure 5.22: PLL output jitter histogram at 1GHz

jitter at 3.28 ps and P2P jitter at 28.89 ps (> 45 Khits) without the supply noise. The measured power consumption is 10mW at 2.5-V supply and 1-GHz output clock frequency.

To characterize the sensitivity of the VCO frequency to supply noise, both static and dynamic VCO supply sensitivity measurements are performed. For static measurement, the DC value of the supply is varied by ±10% and the frequency variation of free-running VCO is measured. Figure 5.23 demonstrates the measured sensitivity results expressed in %-f_{VCO}/%-V_{DD}. The measurement results indicate that the VCO achieves ≤ 0.03%-f_{VCO}/1%-V_{DD} at low frequency supply noise for $0.8 \leq V_{ctrl} \leq 1.7$ (in terms of frequency, 300 MHz ≤ f_{VCO} ≤ 1.4 GHz). At V_{ctrl} greater than 1.7V, where the noise-canceling circuit becomes less effective, the noise sensitivity increases to 0.25%-f_{VCO}/1%-V_{DD}. The dynamic sensitivity of the VCO is characterized by measuring the overall jitter performance of the PLL to high frequency noise. A ±10% supply step with 1-ns slew rate (the fastest possible on-chip frequency) is injected to the VCO supply and the P2P jitter at PLL output clock is measured. Figure 5.23 demonstrates the measured long-term P2P jitter expressed in terms of the percentage of the PLL output clock period, %-T_{PLL}. The measurement results indicate that the PLL achieves the jitter performance of ≤ 0.1%−T_{PLL}/1%−V_{DD} step, with the VCO frequency varying from 800 MHz to 1.4 GHz. The PLL bandwidth is set to roughly 1/40th of VCO frequency[1].

1. The loop multiplication factor is four.

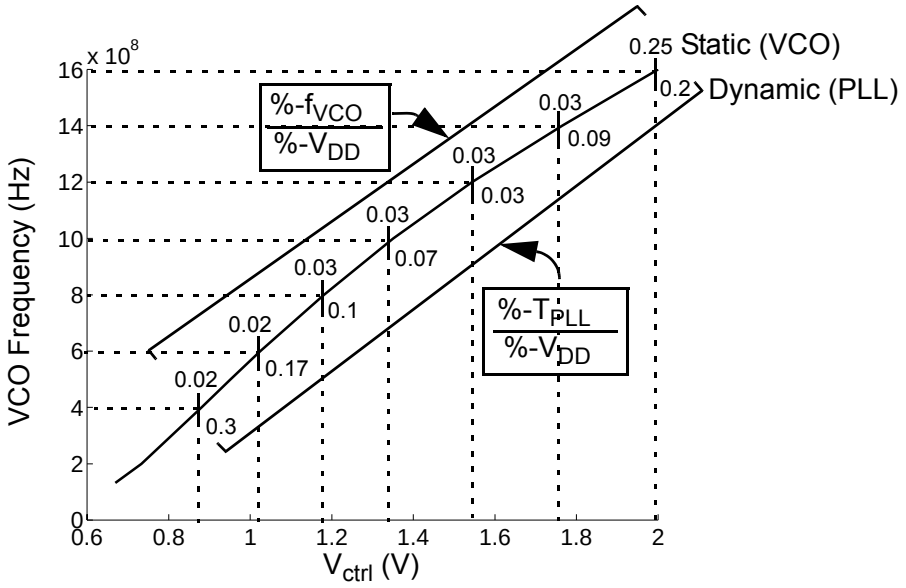


Figure 5.23: Measured sensitivity of VCO output clock frequency to static and dynamic supply noise

To verify the performance of the proposed PFDs, the three PFDs and PLL proposed in [52] are fabricated in a 0.25-μm CMOS technology. The die photogragh is shown in Figure 5.24. The first and second circuits show 18.5% and 41.7% improvements in maximum locking frequency compared to NAND DFF PFD, respectively. The measurement results match the simulated FO-4 results.

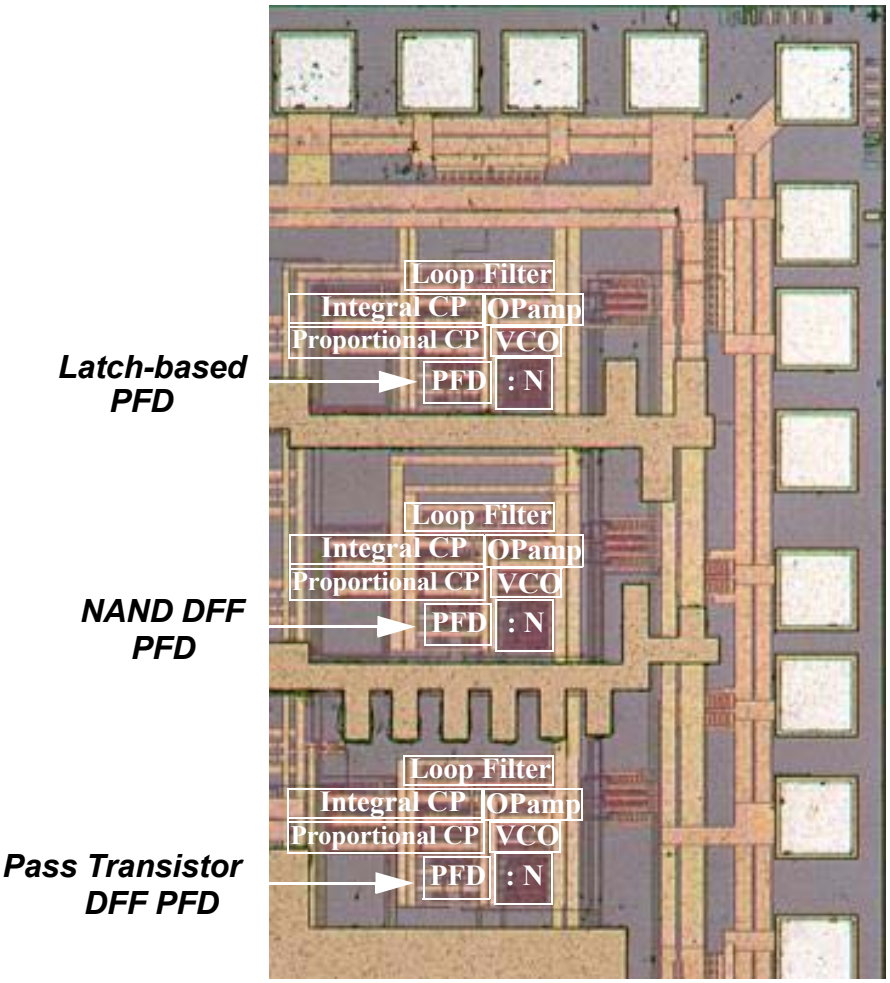


Figure 5.24: Die photograph of three different PFDs implemented in a PLL

The measured frequency acquisition time of PLLs are depicted in Figure 5.25 for all three PFDs. To analyze the frequency acquisition, a reference clock of 1 GHz is supplied while the VCO frequency is initially reset to 200 MHz. Sampling circuits monitor the VCO control voltage as the PLL's reset is disabled. The loop acquires lock with a slightly underdamped behavior. The latch-based PFD has a 1.7x faster acquisition

rate than the NAND DFF PFD and is 1.4x faster than the pass transistor DFF PFD. Note that the PFD with fast acquisition has larger lock-in range.

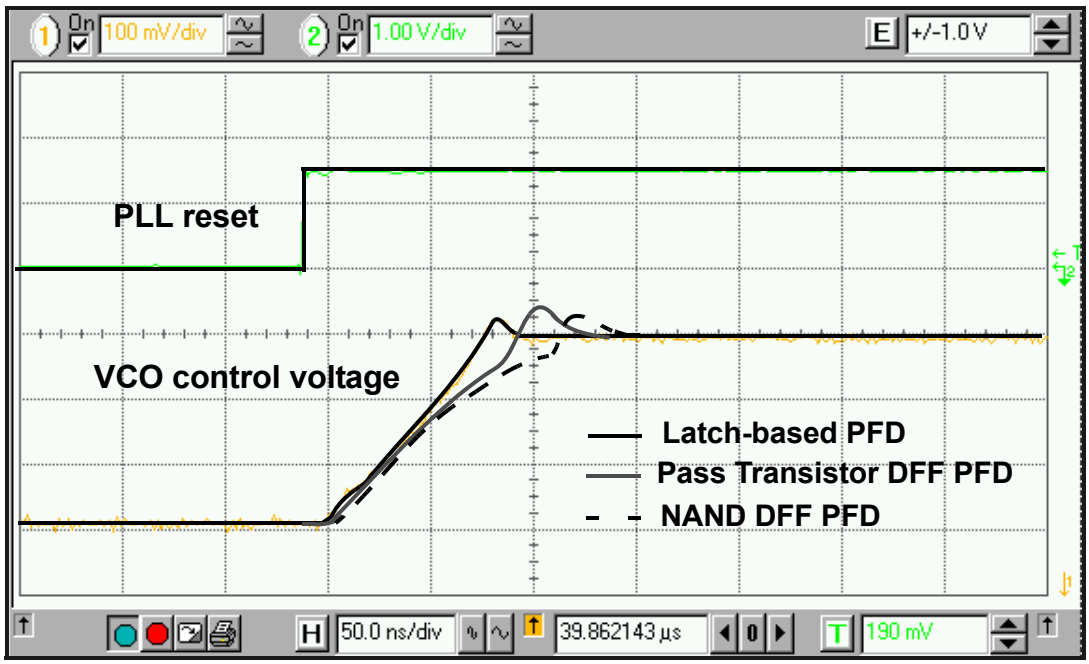


Figure 5.25: Measured frequency acquisition

Table 5.1 summarizes the measured and simulated power consumption and speed performance of each PFD. The power consumption is calculated for PFDs in the lock mode for the reference clock of 500 MHz. The pass-transistor DFF PFD consumes the least power as predicted.

Table 5.1: PFDs performance summary

	NAND DFF PFD (Type I)	Pass Transistor DFF PFD (Type I)	Latch-based PFD (Type II)
Maximum operating frequency *(Measurement)*	1.08 GHz	1.28 GHz	1.53 GHz
Maximum operating frequency *(Simulation)*	0.945 GHz	1.14 GHz	1.5 GHz
Lock-in range @ 1 GHz *(Measurement)*	133 MHz	160 MHz	200 MHz
Power	1.65 mW	0.62 mW	1.4 mW

5.6 PLL Performance Comparison

Table 5.2 and Table 5.3 compare the performance of the proposed PLL (and VCO) with prior state-of-the-art designs. The first two designs by Sidiropoulos [52] and Ingino [51] are examples of the regulated VCOs whereas the designs by Kaenel [54] and Ahn [55] are examples of V-I converters with cascode current sources. The design by Maneatis [57] is an example of the differential VCO and finally, the design by Minami [60] is an example of a V-I converter with linear gain for the entire range of the VCO control voltage. For a fair comparison, all designs are normalized to 0.25-μm technology with 2.5V supply by the use of scaling equations for the short-channel and long-channel devices. The proposed design achieves the lowest power and area among all designs while achieving comparable noise performance with the regulated VCO proposed in [52]. The regulated VCO proposed by Ingino achieves an excellent dynamic noise rejection of 0.007%/1% by coupling the V_{DD} to ground with a large capacitor of 1.2nF and with higher power consumption. While the area and noise rejection performance of the proposed PLL is comparable with [52], it consumes 43% less power than the design in

[52]. With a comparable power consumption, the proposed PLL achieves better dynamic noise rejection than the VCO in [54].

Table 5.2: PLL performance comparison (1)

	S. Sidiropoulos VLSI '00 0.35μm, 3.3V	J.M. Ingino ISSCC '01 0.15μm, 3.3 & 1.5V	V. Kaenel JSSC '96 0.35μm, 1.35V	H. Ahn JSSC '00 0.25μm, 1.9V
Power	21.5mW @ 500MHz	132mW @ 4GHz	1.2mW @ 320MHz	25mW @ 360MHz
Normalized (f, λ, V)[a] power	**17.5mW**	**> 31mW**	**9.2mW**	**60mW**
Area	0.047mm^2	1.48mm^2	0.21mm^2	0.087mm^2
Normalized (λ) area	**0.024mm^2**	**4.1mm^2**	**0.107mm^2**	**0.087mm^2**
VCO frequency range	30-650MHz	0.6-4GHz	176-574MHz	17-1320MHz
Jitter (long-term) w/o V_{DD} noise: P2P rms		44ps (@ 700MHz)		
Normalized (f) Jitter (long-term): P2P rms		**30.8ps**		
Static sensitivity: (%-f_{VCO}/%-V_{DD})			**0.008%/1%** (1ps/52mV)	**0.074%/1%** (±2.6ps/100mV)
Dynamic sensitivity: (%-f_{VCO}/%-V_{DD})	0.06%/1% (PSRR ~ 24dB)	**0.007%/1%** (PSRR>43dB)	**0.18%/1%** (1ps/2.5mV)	**0.32%/1%** (±12ps/100mV)

a. f, λ and V indicates the normalization with frequency, technology and voltage, respectively.

Table 5.3: PLL performance comparison (2)

	J. Maneatis JSSC '96 0.5μm, 2.1V	K. Minami CICC '01 0.1μm, 1.2V	**This work 0.25μm, 2.5V**
Power	9.24mW @ 250MHz	30mW @ 2GHz	10mW @ 1GHz
Normalized (f, λ, V)[a] power	**26mW**	**163mW**	**10mW**
Area	1.91mm^2	0.15mm^2	0.028mm^2
Normalized (λ) area	**0.48mm^2**	**0.94mm^2**	**0.028mm^2**
VCO frequency range	0.0025-550MHz	0.5-2.35GHz	0.2-2GHz
Jitter (long-term) w/o V_{DD} noise: P2P rms		21ps 2.8ps	28.89ps 3.28ps
Normalized (f) Jitter (long-term): P2P rms		**42ps** **5.6ps**	**28.89ps** **3.28ps**
Static sensitivity: (%-f_{VCO}/%-V_{DD})	**0.005%/1%** (0.25%/V)		0.03%/1%
Dynamic sensitivity: (%-f_{VCO}/%-V_{DD})	**0.37%/1%** (704ps/V)		0.07%/1%

a. f, λ and V indicates the normalization with frequency, technology and voltage, respectively.

5.7 Summary

To produce low-jitter clocks in noisy supply environments, we demonstrated an effective supply rejection technique for the VCO. The proposed VCO achieves high supply noise rejection comparable to that of a regulated supply VCO with lower power consumption. The VCO operates over a wide operating frequency range and has a linear

voltage-to-frequency gain. The PLL design demonstrates scaling loop parameters with the oscillator's frequency that tracks over a 10x frequency range. The self-biased design allows the PLL to operate over a wide frequency range with an adaptive loop bandwidth and a constant phase margin. The proposed PFDs acquire frequency lock faster and consume less power than the traditional PFD.

The generated on-chip clock by the PLL should be distributed through clock buffers to the entire system with small uncertainty. Next chapter discusses the design of a noise-compensated clock buffer.

Chapter 6

Design of Clock Buffer

One of the challenges in digital systems is the distribution of the generated on-chip clock with a small uncertainty. Static CMOS inverters are traditionally used for clock buffering due to their simplicity and drive capability with low power consumption. However, CMOS inverters have poor supply-induced delay sensitivity of approximately 1%-delay/1%-V_{DD}. With long chains, this poor supply noise rejection of the inverter could result in significant jitter. For example, in IBM S/390 microprocessor [76], the generated on-chip clock is distributed through clock buffers to all latches in three levels of H-like tree hierarchy. The total simulated delay is about 750ps. The amount of jitter with 10% supply noise is roughly 75ps which is 3% of clock period of 400MHz. As technology scales down[1] and clock frequency increases, jitter becomes a larger fraction of the clock period and may cause system to fail. To reduce jitter, the delay sensitivity of the clock

1. The delay of interconnect is not scaling down as fast as technology does [77].

buffers should be improved. One method of improving the supply noise rejection is by filtering, using a regulator or RC filter. These techniques require large capacitors and hence area. This chapter introduces a compensator circuit added to the inverter that offsets any supply-induced delay variation. This circuit technique supplements other methods of reducing supply noise.

6.1 Concept of Noise Compensation

Ideally, the task of the noise compensator circuit is to introduce an inverse and equal delay sensitivity to the supply noise as an inverter (Figure 6.1-(a)). This noise compensation can be accomplished by using a variable capacitor at the output of each inverter such that as V_{DD} drops, the capacitor value decreases to compensate for inverter delay increase and vice versa. A simple circuit capable of the delay compensation is a MOS resistor in series with a capacitor. Figure 6.1-(b) shows the clock buffer with the compensator circuit, where the capacitor and resistor are implemented by PMOS transistors[1]. The gate voltage of the PMOS resistor, V_{gap}, is set to a constant voltage with respect to ground. As V_{DD} drops, the source-gate voltage (V_{SG}) of the PMOS resistor is decreased, increasing the resistance, R. Thus, the capacitor, C, appears as a lower capacitive loading, which compensates for the increase in the inverter's resistance.

1. If the PMOS resistor and capacitor are switched in the compensator circuit, the circuit is similar to delay elements commonly used for variable-delay lines [63]. However, this configuration is undesirable for the noise compensation because the V_{SG} is decoupled from V_{DD} by PMOS capacitor whereas the V_{SG} in Figure 6.1-(b) experiences the V_{DD} noise directly.

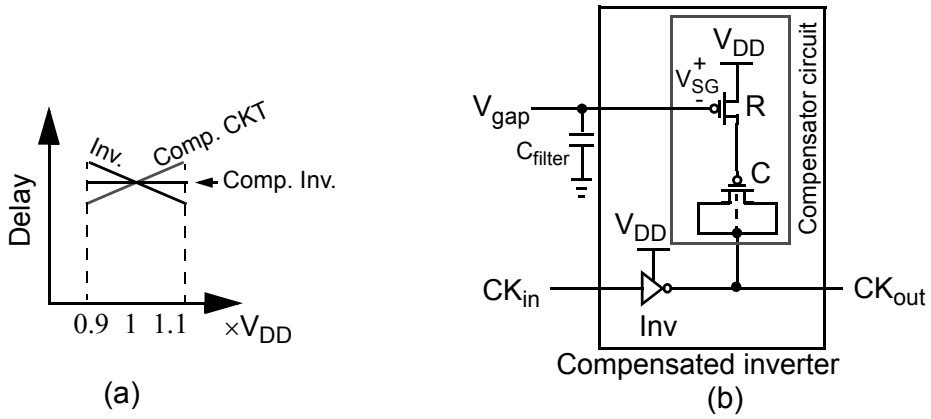


Figure 6.1 (a) Ideal compensation of supply-induced inverter delay variation, (b) proposed compensator inverter

One of the main advantages of this compensation technique is the circuit's excellent dynamic behavior due to a very small time constant of the compensator circuit. The compensated inverter can have a high power-supply rejection ratio (PSRR) for both low and high supply noise frequencies. A regulator in comparison would have a larger time constant with a large filter capacitor. Also, for most applications, the supply noise does not exceed ±15% of supply voltage [49]. Thus, the extra capacitive loading introduced by the compensator circuit would be within 10-15% of the inverter's load and does not change the fanout of the inverter significantly. Due to the small loading effect, the delay and power overhead added by the compensator circuit are a small fraction of the inverter's total delay and power.

6.2 Design Implications

To achieve the high PSRR in the compensated inverter, the compensator circuit should provide an inverse and equal delay variation (from V_{DD} noise) as the delay variation of an uncompensated inverter. While the delay variation of an inverter is roughly proportional to V_{DD}, the delay variation of the compensator circuit varies non-linearly with V_{DD}. Figure 6.2-(a) shows the non-linear behavior of the delay variation of the compensator circuit as a function of V_{SG}. For minimum power, delay and area overhead, the compensator circuit should be used over the range where it achieves the maximum delay sensitivity; $Sensitivity = \left(\frac{\partial}{\partial V_{SG}} delay\right) \cdot \frac{V_{SG}}{delay}$. The maximum delay sensitivity occurs over the V_{SG} range where the resistance of the PMOS device, R, is the most sensitive to the V_{SG} variation. $V_{SG}\big|_{max}$ in Figure 6.2-(a) indicates the middle of the region with maximum delay sensitivity. Therefore, V_{gap} should be set to $V_{gap} = V_{DD}\big|_{nom} - V_{SG}\big|_{max}$. The delay variation of the inverter is well compensated over the range of ΔV_{DD} ($= V_{DD} - V_{DD}\big|_{nom}$) where the delay sensitivity of the compensator circuit approximates the delay sensitivity of the inverter. For V_{DD} noise exceeding this range, the delay compensation performance of the compensator circuit degrades. Figure 6.2-(b) shows the *desired* delay sensitivity of the compensator circuit (normalized to the supply-induced delay sensitivity of the uncompensated inverter) for a proper delay compensation. The voltage range where the normalized sensitivity curve crosses *one* is when the compensator circuit compensates for the delay variation of the uncompensated

inverter due to V_{DD} noise. The delay sensitivity and the compensating range of the compensator circuit are adjusted through sizing the devices in the compensator circuit.

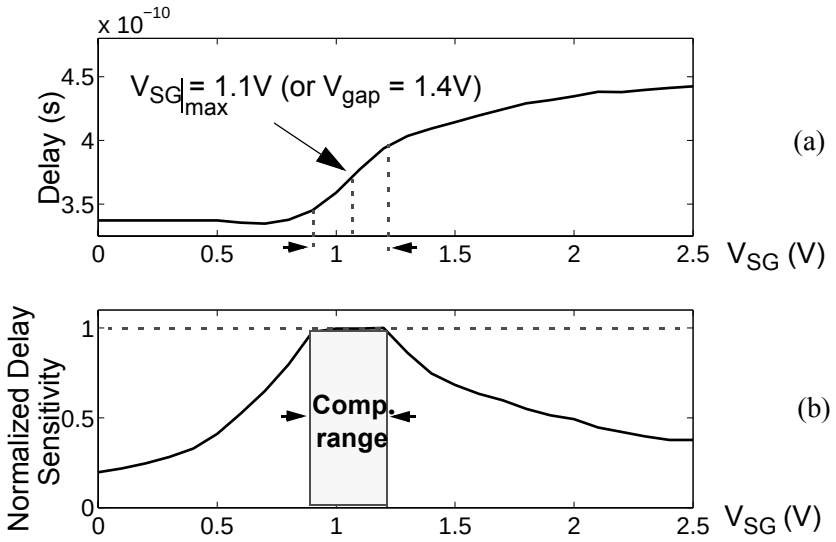


Figure 6.2 (a) Delay variation of compensated inverter due to V_{SG} variation, (b) delay sensitivity of compensator circuit, normalized to delay sensitivity of an inverter

6.2.1 Design of the Compensator Circuit

Figure 6.3 illustrates the behavior of the delay sensitivity of the compensator circuit (normalized to the delay sensitivity of the uncompensated inverter) as the PMOS resistor and capacitor vary. Figure 6.3-(a) shows the delay sensitivity behavior as a function of the capacitor while keeping the width of PMOS resistor constant. Using a larger compensating capacitor as a fraction of the total capacitive load of the inverter increases the delay variation and, hence, the delay sensitivity of the compensator circuit. However, increasing the capacitor reduces the compensating range. Figure 6.3-(b) shows similar curves, varying the PMOS resistor value in a constant capacitor value. Decreasing

the resistor value increases the delay sensitivity by introducing larger capacitive loading to the inverter while reducing the compensating range. By proper adjustment of the resistor and capacitor, both the maximum normalized delay sensitivity and the compensating range can be set to *one* and peak-to-peak supply noise, respectively. Curve (2) in Figure 6.3-(a) or (b) is an example of the proper sizing that roughly results in the same delay sensitivity as an inverter within the operating range of ±10% V_{DD} noise.

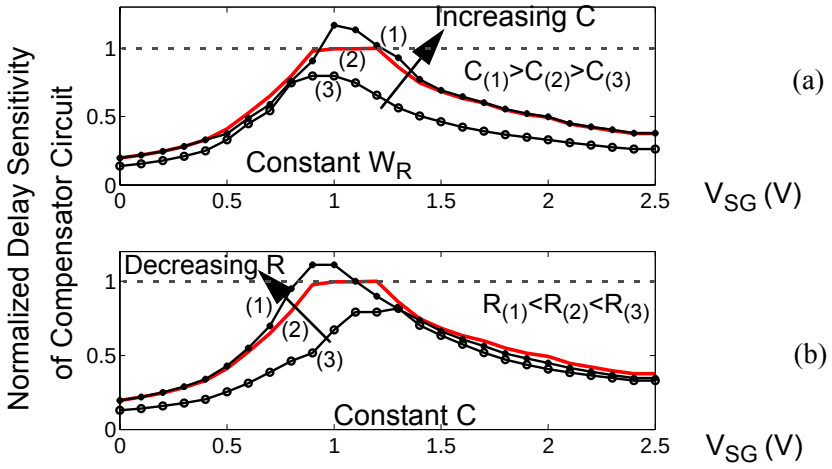


Figure 6.3 Behavior of normalized delay sensitivity of compensator circuit due to V_{SG} (V_{DD}) variation as a function of: (a) PMOS capacitor, (b) PMOS resistor

Figure 6.4 illustrates the simulated delay compensation characteristics of the compensated inverter (with R and C values of curve (2) in Figure 6.3) when V_{DD} varies by ±10%. Curve(1) illustrates the supply-induced delay variation of the compensated inverter while keeping V_{DD} of the compensator circuit constant. This curve represents the delay variation of an uncompensated inverter. Curve(2) illustrates the supply-induced delay variation while keeping V_{DD} of the inverter constant. This curve represents the delay variation solely due to the compensator circuit. Curve(3) shows the overall delay variation

of the compensated inverter to V_{DD} noise. Curve(3) is effectively an average of the first two curves. The overall delay sensitivity of the compensated inverter is approximately 0.1%-delay/%-V_{DD} for V_{DD} variation of $\leq \pm 10\%$.

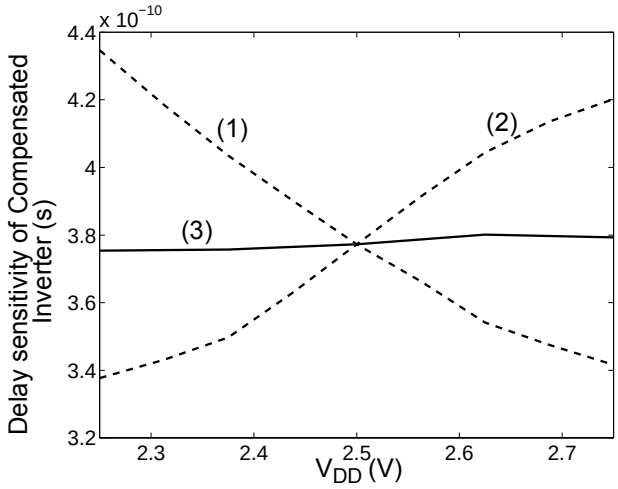


Figure 6.4 Supply-induced delay variation of: (1) uncompensated inverter, (2) compensated inverter with inverter's V_{DD} held constant and (3) compensated

Although the delay sensitivity metric has been traditionally used to illustrate the circuit noise performance, the overall delay variation of curve(3) in Figure 6.4 suggests another useful metric. Since the delay may not change linearly with V_{DD} variation, the alternate metric is defined as the maximum percentage delay variation from its nominal value ($\frac{\Delta delay_{max}}{delay_{nominal}} \cdot 100$) within the V_{DD} noise range ($\frac{\Delta V_{DD}}{V_{DD}} \cdot 100$). The maximum delay variation for curve(3) is 1.2% within $\pm 10\%$ V_{DD} noise.

6.2.2 Bias Circuit for V_{gap}

The previous discussion of the delay compensation indicates that the bias circuit for V_{gap} must be constant with respect to ground. Also, the optimum biasing point for the V_{gap} (the middle of the voltage range with the maximum delay sensitivity) varies across process corners, as PMOS devices become faster or slower. To maintain the high PSRR across the corners, the biasing circuit for V_{gap} should track the variation of the PMOS threshold such that V_{gap} is set to the middle of the compensating range. Therefore, the desired V_{gap} should compose of a voltage that is independent of supply and PVT (a bandgap reference) and a voltage that depends on the PMOS threshold voltage. Figure 6.5 shows a realization of the bias circuit. A diode connected PMOS transistor is biased with a small current such that $V_{SG} \sim |V_{Tp}|$. To generate V_{gap}, the $|V_{Tp}|$ is subtracted from an amplified bandgap voltage.

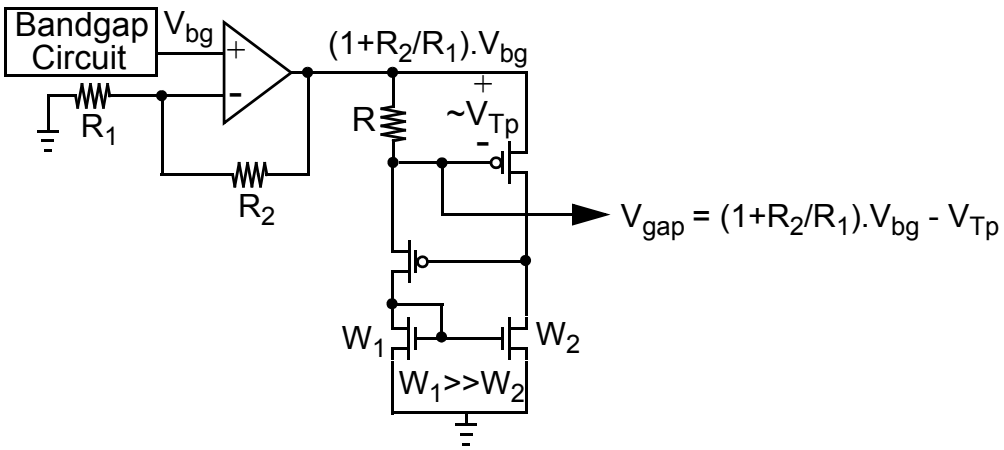


Figure 6.5 Bias circuit generating V_{gap}

The generated V_{gap} bias is distributed to the entire clock buffers. Due to the coupling noise into V_{gap}, there is uncertainty in the V_{gap} voltage from buffer-to-buffer. The deviation of V_{gap} from the middle of the compensating range decreases the effective compensating range. Figure 6.6 shows the simulated delay variation of the compensated inverter for the optimum V_{gap}, and ±100mV deviation from the optimum V_{gap}. The maximum delay variation increases from 1.2% (within ±10% V_{DD} noise) at the optimum V_{gap} to 2% and 2.7% at 50mV and 100mV of the offset in the V_{gap} value, respectively. The uncertainty in the V_{gap} can be reduced by minimizing the coupling capacitors with a careful layout design. Also, the V_{gap} uncertainty can be significantly suppressed by supplying the clock buffers with their own local V_{gap} bias generator with the cost of power and area overhead added by each V_{gap} bias circuit.

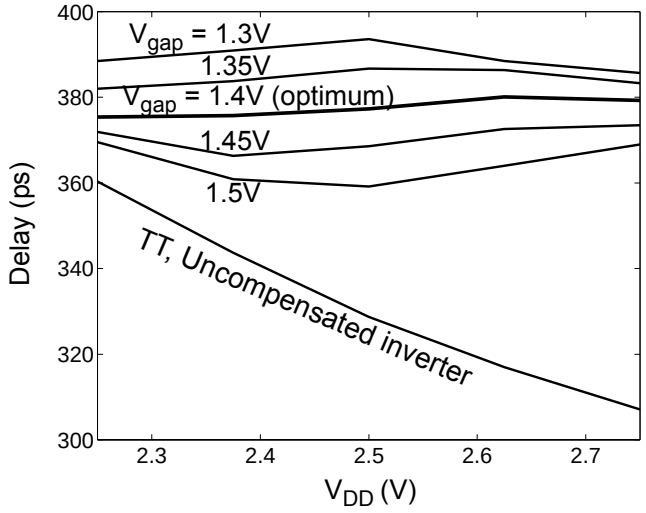


Figure 6.6 Sensitivity of supply-induced delay variation of compensated inverter due to V_{gap} offset

6.2.3 Performance Sensitivity to PVT[1]

To characterize the performance of the delay compensating technique, the compensated clock buffer is simulated over temperature and process variations. As the temperature increases, the V_{gap} increases due to the negative sensitivity of V_{Tp} to the temperature (~ -1mV/K). Figure 6.7 demonstrates the supply-induced delay variation of the compensated clock buffer as the temperature varies between 0^o to 125^o. Increasing the temperature from 25^o to 125^o increases the maximum delay variation from 1.2% to 2.4% (within ±10% V_{DD} noise).

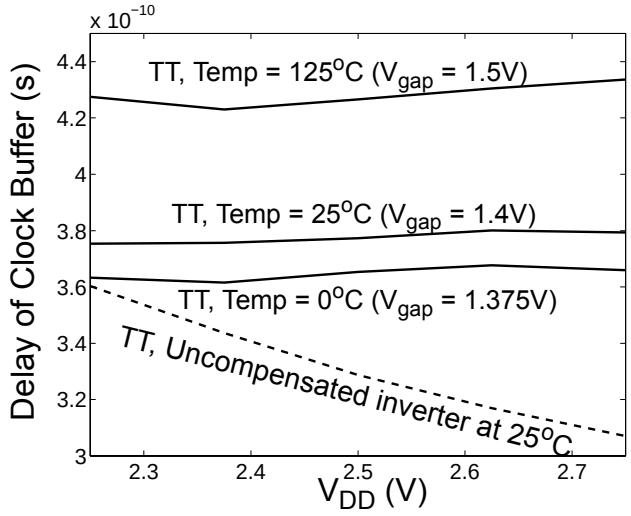


Figure 6.7 Delay variation of compensated clock buffer over temperature as V_{DD} varies ≤ ±10%

Figure 6.8 shows the supply-induced delay variation across the process corners where V_{gap} tracks the PMOS threshold variation. The maximum delay variation increases to

1. Process, voltage or temperature variations

2.5% (within ±10% V_{DD} noise) at fast NMOS corners in the worst case. The PSRR degradation at fast NMOS corners is due to the fact that neither the compensated circuit nor the V_{gap} voltage tracks the NMOS corner variation. To further improve the PSRR, a series of an NMOS capacitor and resistor can be added to the compensator circuit.

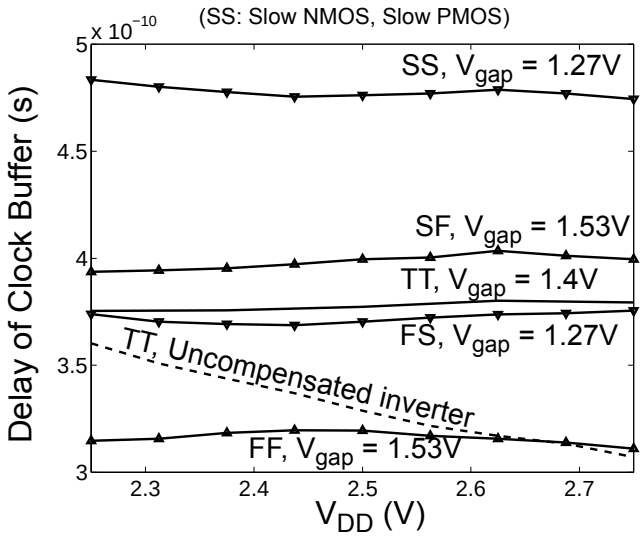


Figure 6.8 Delay variation of compensated clock buffer across the corners as V_{DD} varies ≤ ±10%

Five stages of fanout-4 compensated inverters as shown in Figure 6.9 are used for simulation. The optimum sizes of the PMOS resistor and capacitor are 0.5x and 3x the PMOS transistor width size in the preceding inverter stage. The simulated power and delay increase due to the compensator circuit (V_{gap} bias circuit is not included) are 25% and 12% of the conventional clock buffer (uncompensated inverter), respectively.

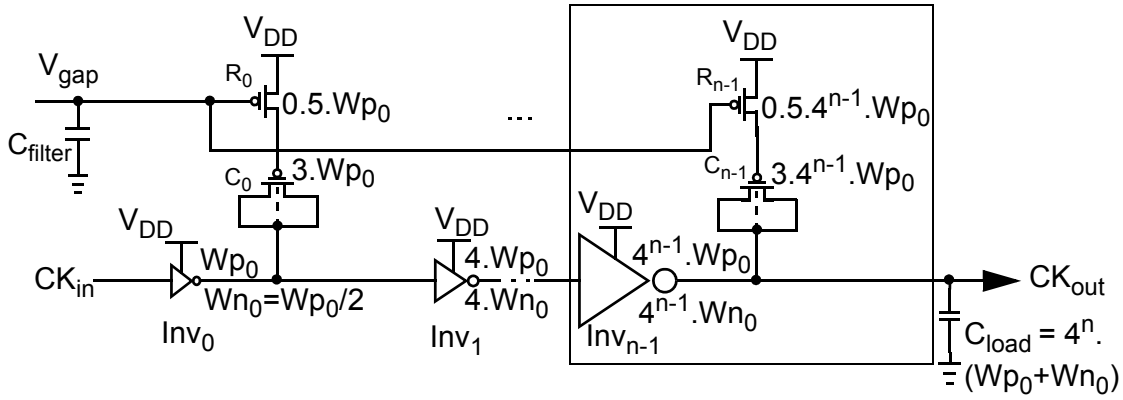


Figure 6.9 Five stages of fanout of four (FO-4) compensated inverters (n=5)

6.3 Measurement Results

To characterize the delay sensitivity of the clock buffer, both static and dynamic V_{DD} variations are measured. Five stages of FO-4 inverters and compensated inverters are fabricated in 0.25-µm CMOS technology[1]. The compensator inverters includes the PMOS compensator circuit only. For the measurement purpose, a separate power supply is used to supply the V_{gap} instead of the bias generator shown in Figure 6.5. V_{gap} is held constant as V_{DD} noise is injected. The measurement results shown in Figure 6.10 indicates that the compensated clock buffer at optimum V_{gap}=1.45V has a maximum delay variation of 3.8% within ±10% V_{DD} noise for a slow corner device, which is 5x less than the uncompensated inverter. The measured maximum delay variation of the compensated

1. Clock buffer die photo is shown in Figure 5.20

clock buffer is greater than the simulation results in a typical corner (1.2% within ±10% V_{DD}) due to not tracking the NMOS process variation and also the parasitic capacitances. The supply noise rejection performance can be improved by adding an NMOS compensator circuit.

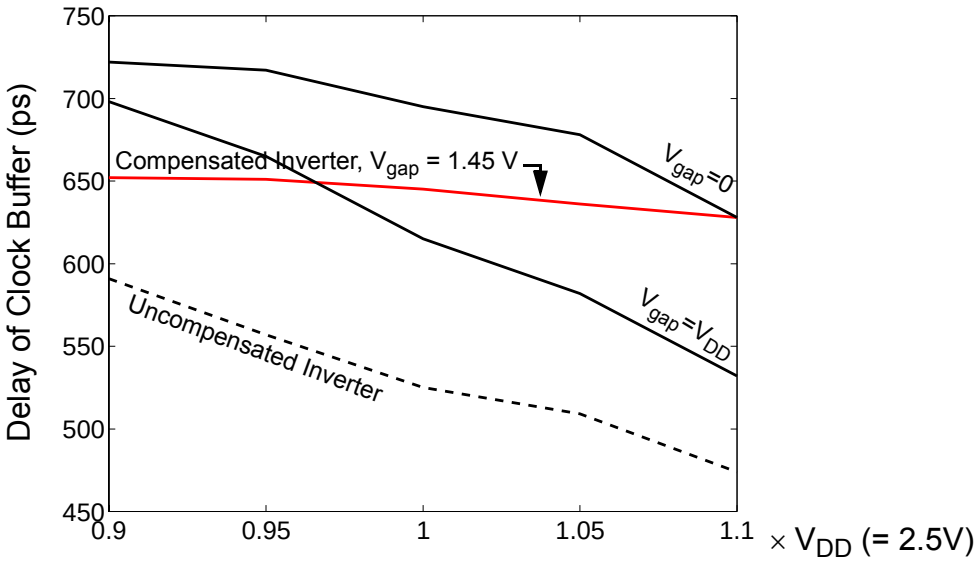


Figure 6.10 Measured supply-induced delay variation of uncompensated (--) and compensated clock buffer

For comparison, Figure 6.10 also demonstrates the performance of the compensated clock buffer for V_{gap} values far from the optimum V_{gap}=1.45V. The measured result at V_{gap}=0 shows an increased maximum delay variation to 5.7% (within ±10% V_{DD} noise) and for V_{gap}=V_{DD}, where the PMOS resistor is off, the maximum delay variation becomes 22%, which is roughly the same as that of an inverter. The measured power and delay overhead are 30% and 18%, slightly greater than simulation results due to the parasitic capacitances. The area overhead, excluding decoupling capacitors, is 50%

as compared to inverters alone. The overhead numbers do not include the overhead due to the V_{gap} bias generator.

6.4 Summary

To distribute low-jitter generated on-chip clocks in noisy supply-noise environments, an effective supply noise compensation technique has been demonstrated for the clock buffer. The proposed clock buffer achieves high supply-noise rejection with an excellent dynamic behavior and with small area and power overhead. This technique can supplement existing supply filtering using decoupling capacitors and supply-voltage regulation. The design dissipates low power for its jitter performance and has low area overhead.

Chapter 7

Conclusion

This dissertation has shown the generation and distribution of low-jitter on-chip clocks for low-power applications in noisy supply environment. The major noise sources in a PLL were discussed: VCO internal noise, clock buffer noise and input (reference) clock noise. The performance of circuits to supply noise is characterized with noise sensitivity metric; %-delay/%-V_{DD}. This is a useful metric that reports the delay variation of buffer element in percentage per percentage of supply variation rather than absolute value. Therefore, it can conveniently be used to compare the noise performance of various designs in different technologies.

To generate and distribute a clock with small uncertainty requires to reduce the noise sensitivity of the most sensitive blocks in a PLL, i.e., VCO and clock buffer. Priors state-of-the-art designs regulate and filter the supply-voltage or use differential delay elements to significantly improve the noise performance. However, they typically consume large power to supply delay elements and occupy large area due to decoupling

capacitors. To overcome power and area issues associated with the prior designs, this research proposes two new filtering techniques that effectively improves the noise sensitivity of VCO and clock buffer with small power and area overhead. Furthermore, both techniques demonstrate an excellent dynamic behavior with a faster response time than the time constant of the PLL. The faster response time enables the VCO (or clock buffer) to correct for errors introduced by high-frequency noise much faster than the loop response of the PLL.

While the proposed filtering techniques are proved to reduce the jitter at PLL output clock significantly, further improvement is achieved through appropriate filtering of various noise sources with PLL closed-loop feedback system. Investigation to the impact of PLL loop parameters on output jitter reveals that the loop parameter settings at which minimum jitter occurs depends on the dominant noise source in a PLL. Therefore, to achieve the minimum jitter performance based on the loop parameters requires knowledge of the dominant source.

For most systems, the dominant noise source is not well-known. To minimize jitter under different noise conditions, a run-time methodology measures the jitter on-chip and adjusts the PLL loop parameters toward minimum jitter performance based on gradient-descent algorithm. A dead-zone phase detection circuit suffices as a measuring circuit for data-recovery applications. However, the measurement uncertainty limits the performance. The implication of a large number of hits is that ≥12-bit accumulator and long measurement intervals are needed. The range of loop parameters must be bounded by

the ability of the loop to remain in lock especially if the algorithm operates when the system is active. The quantization or resolution of the parameter adjustment is determined such that there is no risk of losing lock.

To accommodate testability and further power optimization, a PLL with wide frequency range and adaptive bandwidth is designed. To accomplish the adaptive bandwidth, we used self-biased techniques in the design of the loop filter. Also, this research addresses the drawback of the conventional PFD and proposes new circuit techniques to design PFDs. The proposed PFDs consume lower power and achieve wider acquisition range than conventional PFDs.

The PLL and clock buffer were fabricated in 0.25-μm CMOS technology. Experimental results indicate that both VCO and clock buffer demonstrate the delay sensitivity of ≤ 0.1%-delay/1%-V_{DD}. VCO consumes only 2mW at 1GHz from 2.5-V supply. The total power consumption of PLL is 10mW at 1GHz. Using the run-time adaptive method of minimizing jitter for the PLL minimizes jitter to within 5ps of the minimum peak-to-peak jitter as noise conditions are changed. The PLL demonstrates scaling loop parameters with the oscillator's frequency that tracks over a 10x frequency range.

This research points to several areas of potential future work. Although the performance of the circuits proposed in this thesis has continued to scale down with improving fabrication technology[1], it still needs to be validated for deep-submicron technologies less than 90nm. Understanding and overcoming scaling limitations can be an

interesting area for future work. Design of PLL circuits might require innovative design techniques or architectures to maintain close to desired performance anticipated with scaling.

In this work, we proposed a new filtering technique that effectively compensates for noise. We have successfully shown the successfulness of this technique by implementing a compensated clock buffer. One might benefit form this technique due its lower overhead and excellent dynamic behavior. Thus, another interesting area of research is to extend the noise compensation concept to develop new circuit techniques with better noise performance while introducing lower overhead.

Noise sources in digital systems are not well known and also the noise conditions could be changed. We developed a run-time algorithm that dynamically minimizes jitter. However, this algorithm has its own implications such as error in on-chip jitter measurement, required resolution and range for loop parameters that impacts the performance of algorithm. Understanding and developing new algorithms with shorter convergence time are interesting subjects. Although a dead-zone phase detector suffices for clock/data recovery applications as shown in this work, an appropriate on-chip circuit to measure the jitter of a clock's system is a challenge.

Finally, supply or substrate noise are the most dominant noise sources in digital systems. To achieve a low-jitter performance in such systems requires a good

1. From starting point of this research, the design has been fabricated in three process technologies: 0.35μm [41], 0.25μm and 0.18μm

understanding of these noise sources. There are a few studies ([39] and [48]-[50]) that focus on understanding and modeling of supply or substrate noise. However, their impact on jitter distribution is not well understood, because supply or substrate noise are deterministic rather than probabilistic and they vary from a system to another system. Proper modeling of supply noise may further enhance jitter performance.

Appendices

A.1 Relationship Between Timing Jitter and Noise Power Spectral Density

Using timing jitter definition given in Section 3.1, the jitter is equal to[1]:

$$\sigma_{\Delta T}^2 = \frac{1}{\omega_0^2} \cdot E\{[\phi(t+\Delta T) - \phi(t)]^2\}$$
$$= \frac{1}{\omega_0^2} \cdot \{E[\phi^2(t)] + E[\phi^2(t+\Delta T)] - 2 \cdot E[\phi(t) \cdot \phi(t+\Delta T)]\} \quad \text{‹A.1›}$$

Since $E[\phi(t) \cdot \phi(t+\Delta T)]$ is equal to autocorrelation of $\phi(t)$, $R_\phi(\Delta T)$, the timing jitter in Equation A.1 can be written as:

$$\sigma_{\Delta T}^2 = \frac{2}{\omega_0^2}[R_\phi(0) - R_\phi(\Delta T)] \quad \text{‹A.2›}$$

Replacing autocorrelation with power spectral density (given by Khinchin theorem [82]), $R_\phi(t) = \int_{-\infty}^{\infty} S_\phi(f) e^{j2\pi ft} df$ in Equation A.2, results in:

1. Equations A.1-A.4 have been extracted from [26]

$$\sigma_{\Delta T}^2 = \frac{4}{\omega_0^2}\int_{-\infty}^{\infty} S_\phi(f)\sin^2(\pi f \Delta T)df \quad \text{‹A.3›}$$

Equation A.3 describes the relationship between the timing jitter and noise power spectral density (psd), $S_\phi(f)$. As ΔT goes to infinity, timing jitter is calculated from Equation A.2:

$$\sigma_{\Delta T}^2 = \frac{2}{\omega_0^2}R_\phi(0) \quad \text{‹A.4›}$$

or,

$$\sigma_{\Delta T \to \infty}^2 = \frac{2}{\omega_0^2}\int_{-\infty}^{\infty} S_\phi(f)df \quad \text{‹A.5›}$$

A.2 Relationship Between Jitter and PLL Noise Transfer Function (NTF)

Noise psd, $S_\phi(f)$, in Equation A.3 (or Equation A.5) is calculated by multiplying the open-loop phase noise of each noise source, $S_{\phi ni\text{-}open}(f)$, with the square magnitude of of PLL NTF from the correspondent phase noise to the PLL output phase[1], $Hn_i(j2\pi f)$:

$$\sigma_{\Delta T}^2 = \frac{4}{\omega_0^2}\int_{-\infty}^{\infty} S_{\phi ni-open}(f)\left|Hn_i(j2\pi f)\right|^2\sin^2(\pi f \Delta T)df \quad \text{‹A.6›}$$

To simplify the equation, Parseval's relation is used, $\frac{1}{2\pi}\int_{-\infty}^{\infty}|Z(\omega)|^2 d\omega = \int_{-\infty}^{\infty}|z(t)|^2 dt$. To do so, $Z(\omega)$ is expressed as:

$$Z(\omega) = X(\omega)\cdot Y(\omega) = \underbrace{H_{open}(j\omega)\cdot Hn_i(j\omega)}_{X(\omega)} \cdot \underbrace{j\omega\cdot\sin\left(\omega\frac{\Delta T}{2}\right)/\omega}_{Y(\omega)} \quad \text{‹A.7›}$$

1. Please see Equation 3.3

where $S_{\phi ni-open}(f) = \left|H_{open}(j\omega)\right|^2$. z(t) is equal to convolution of x(t) and y(t). Since $y(t) = \frac{1}{2}\delta\left(t+\frac{\Delta T}{2}\right) - \frac{1}{2}\delta\left(t-\frac{\Delta T}{2}\right)$ where δ(t) represents dirac's delta function, $z(t) = \frac{1}{2}x\left(t+\frac{\Delta T}{2}\right) - \frac{1}{2}x\left(t-\frac{\Delta T}{2}\right)$ where x(t) is the inverse fourier of ($H_{open}(j\omega).Hn_i(j\omega)$).

Therefore, timing jitter equation is simplified as:

$$\sigma_{\Delta T}^2 = \frac{4\cdot 2\pi}{\omega_0^{\,2}}\int_{-\infty}^{\infty}\frac{1}{4}\left|x\left(t+\frac{\Delta T}{2}\right) - x\left(t-\frac{\Delta T}{2}\right)\right|^2 dt \qquad \text{‹A.8›}$$

A.3 Relationship Between Output Jitter and VCO Noise

For the VCO noise, X(ω) is calculated from Equation 3.4 and Equation 3.5:

$$\begin{aligned} X(\omega) &= H_{open}(j\omega)\cdot Hn_{buf}(j\omega) \\ &= \left(\frac{\sqrt{N_{VCO}}}{s}\cdot\frac{s^2}{s^2+K_{Loop}RCs+K_{Loop}}\right)\Bigg|_{s=j\omega} \end{aligned} \qquad \text{‹A.9›}$$

x(t) is calculated by taking the inverse fourier of Equation A.12:

$$x(t) = \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\cdot\cos(\omega_d t+\theta)\cdot u(t) \qquad \text{‹A.10›}$$

where $\omega_d = \omega_n\cdot\sqrt{1-\zeta^2}$ and $\cos\theta = \sqrt{1-\zeta^2}$.

By substituting x(t) in Equation A.8, timing jitter is calculated:

$$\sigma_{\Delta T}^2 = \frac{4\pi^2 N_{VCO}}{\omega_0^{\,2}}\cdot\begin{cases}\frac{1}{2\zeta\omega_n}+\frac{e^{-\zeta\omega_n\Delta T}}{2(1-\zeta^2)}\cdot\left(\frac{\sin(\omega_d\Delta T+\theta)}{\omega_n}-\frac{\cos(\omega_d\Delta T)}{\zeta\omega_n}\right) & \zeta<1 \\ \frac{1}{2\zeta\omega_n}-e^{-a\Delta T}\left(\frac{2\alpha\beta}{a+b}+\frac{\alpha^2}{a}\right)-e^{-b\Delta T}\left(\frac{2\alpha\beta}{a+b}+\frac{\beta^2}{b}\right) & \zeta\geq 1\end{cases} \qquad \text{‹A.11›}$$

where $a, b = \zeta\omega_n \mp \omega_n\cdot\sqrt{\zeta^2-1}$, $\alpha = \frac{-a}{b-a}$ and $\beta = \frac{b}{b-a}$.

A.4 Relationship Between Output Jitter and Clock Buffer Noise

For the clock buffer noise, X(ω) is calculated from Equation 3.4 and Equation 3.5:

$$X(\omega) = H_{open}(j\omega) \cdot Hn_{buf}(j\omega) = \left(\frac{\sqrt{N_{buf}}}{s/\omega_{buf} + 1} \cdot \frac{s^2}{s^2 + K_{Loop}RCs + K_{Loop}} \right)\Bigg|_{s=j\omega} \quad \text{‹A.12›}$$

x(t) is calculated by taking the inverse fourier of Equation A.12:

$$x(t) = \left\{ \left(\frac{-\omega_n^2}{\omega_d} \cdot \sin(\omega_d t) - \frac{2\zeta\omega_n}{\sqrt{1-\zeta^2}} \cdot \cos(\omega_d t + \theta) \right) \cdot e^{-\zeta\omega_n t} + \omega_{buf} \cdot e^{-\omega_{buf} t} \right\} \cdot u(t) \quad \text{‹A.13›}$$

where $\omega_{Buf} = 2\pi f_{Buf}$, $\zeta<1$, $\omega_d = \omega_n \cdot \sqrt{1-\zeta^2}$ and $\cos\theta = \sqrt{1-\zeta^2}$.

By substituting x(t) in Equation A.8, timing jitter is calculated:

$$\sigma_{\Delta T}^2 = \frac{N_{Buf}}{\omega_0^2} \cdot \begin{cases} \left\{ \begin{array}{c} \omega_{Buf} + \omega_n \frac{1-12\zeta^2}{2\zeta} - e^{-\omega_{Buf}\Delta T}(\omega_{Buf} - 4\zeta\omega_n) \\ -e^{-\zeta\omega_n\Delta T}\left(-\frac{\omega_n \sin(\omega_d\Delta T + 3\theta - \pi)}{2(1-\zeta^2)} + \frac{\omega_n \cos(\omega_d\Delta T)}{2(1-\zeta^2)\zeta} - \frac{2\omega_n \sin(\omega_d\Delta T + 2\theta)}{\sqrt{1-\zeta^2}} \right) \end{array} \right\} & \zeta < 1 \\ \left\{ \begin{array}{c} \omega_{Buf} + \frac{\gamma^2}{a} + \frac{\upsilon^2}{b} + \frac{4\upsilon\gamma}{a+b} + \frac{4\omega_{Buf}\upsilon}{a+\omega_{Buf}} + \frac{4\omega_{Buf}\gamma}{b+\omega_{Buf}} - e^{-a\Delta T}\left(\frac{2\upsilon\omega_{Buf}}{a+\omega_{Buf}} + \frac{2\upsilon\gamma}{a+b} + \frac{\upsilon^2}{a} \right) \\ -e^{-b\Delta T}\left(\frac{2\gamma\omega_{Buf}}{b+\omega_{Buf}} + \frac{2\upsilon\gamma}{a+b} + \frac{\gamma^2}{b} \right) - e^{-\omega_{Buf}\Delta T}\left(\omega_{Buf} + \frac{2\upsilon\omega_{Buf}}{a+\omega_{Buf}} + \frac{2\gamma\omega_{Buf}}{b+\omega_{Buf}} \right) \end{array} \right\} & \zeta \geq 1 \end{cases} \quad \text{‹A.14›}$$

where $\upsilon = \frac{2\zeta\omega_n a - \omega_n^2}{b-a}$ and $\gamma = \frac{-2\zeta\omega_n b + \omega_n^2}{b-a}$.

A.5 Relationship Between Output Jitter and Input Clock Noise

For the input clock noise, X(ω) is calculated from Equation 3.4 and Equation 3.5:

$$X(\omega) = H_{open}(j\omega) \cdot Hn_{in}(j\omega)$$
$$= \left(H_{open}(s) \cdot \frac{2\zeta\omega_n s + {\omega_n}^2}{s^2 + K_{Loop}RCs + K_{Loop}} \right)\Bigg|_{s=j\omega} \quad \text{‹A.15›}$$

In the next two sections, we calculate the timing jitter for two different input noise psd.

A.5.1 1/f^2 noise, i.e. $S_{\phi n_{in}}(f) = \frac{N_{Clk-in}}{f^2}$

Using $S_{\phi n_{in}}(f) = \left|H_{open}(j\omega)\right|^2$, X(ω) is calculated:

$$X(\omega) = \left(\frac{2\pi \cdot \sqrt{N_{Clk-in}}}{s} \cdot \frac{2\zeta\omega_n s + {\omega_n}^2}{s^2 + K_{Loop}RCs + K_{Loop}} \right)\Bigg|_{s=j\omega} \quad \text{‹A.16›}$$

x(t) is calculated by taking the inverse fourier of Equation A.16:

$$x(t) = \left(1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \cdot \cos(\omega_d t + \theta) \right) \cdot u(t) \quad \text{‹A.17›}$$

By substituting x(t) in Equation A.8, timing jitter is calculated:

$$\sigma^2_{\Delta T} = \underbrace{\kappa^2 \cdot \Delta T}_{} \cdot \begin{cases} 1 + \frac{1}{2\zeta\omega_n \cdot \Delta T} + \frac{e^{-\zeta\omega_n \Delta T}}{\Delta T} \cdot \left(\frac{\sin(\omega_d \Delta T + \theta)}{2(1-\zeta^2)\omega_n} - \frac{\cos(\omega_d \Delta T)}{2(1-\zeta^2)\zeta\omega_n} - \frac{2\sin(\omega_d \Delta T)}{\omega_d} \right) & \zeta < 1 \\ 1 + \frac{1}{2\zeta\omega_n \cdot \Delta T} + \frac{e^{-a\Delta T}}{\Delta T}\left(\frac{2\alpha}{a} - \frac{2\alpha\beta}{a+b} - \frac{\alpha^2}{a} \right) + \frac{e^{-b\Delta T}}{\Delta T}\left(\frac{2\beta}{b} - \frac{2\alpha\beta}{a+b} - \frac{\beta^2}{b} \right) & \zeta \geq 1 \end{cases} \quad \text{‹A.18›}$$

where ω_d, θ, a, b, α and β are the same as Equation A.11.

A.5.2 White noise, i.e. $S_{\phi n_{in}}(f) = N_{Clk-in}$

Similarly, X(ω) is calculated:

$$X(\omega) = \left(\sqrt{N_{Clk-in}} \cdot \frac{2\zeta\omega_n s + {\omega_n}^2}{s^2 + K_{Loop}RCs + K_{Loop}} \right)\Bigg|_{s=j\omega} \quad \text{‹A.19›}$$

x(t) is calculated by taking the inverse fourier of Equation A.16:

$$x(t) = \sqrt{N_{Clk-in}} \cdot \left\{ 2\zeta\omega_n \cdot \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \cdot \cos(\omega_d t + \theta) + \frac{\omega_n^2 \cdot e^{-\zeta\omega_n t}}{\omega_d} \cdot \sin(\omega_d t) \right\} \cdot u(t) \quad \text{‹A.20›}$$

where ω_d and θ are the same as Equation 3.10. The long-term timing jitter is calculated:

$$\sigma^2_{\Delta T \to \infty} = \frac{2 \cdot N_{Clk-in}}{\omega_0^2} \int_{-\infty}^{\infty} |x(t)|^2 dt \quad \text{‹A.21›}$$

Simplifying Equation A.21 results in:

$$\sigma^2_{\Delta T \to \infty} = \frac{2}{\omega_0^2} \cdot N_{Clk-in} \cdot \left(\omega_n \cdot \frac{4\zeta^2 + 1}{4\zeta} \right) \quad \text{‹A.22›}$$

A.6 Jitter Estimation by Applying Effective 2nd-Order Model to any PLL

Although a complete 3rd-order model of a PLL is needed to understand the jitter contribution of different loop parameters, our analytical results and measurements have found that tracking jitter due to VCO noise for a particular design can be easily estimated by simply using the second-order equations. As shown in the jitter analysis of Section 3.4 and Section 3.5, tracking jitter (σ_{tr}) is the integral of the noise shaped by the frequency response. The critical parameters that determine the jitter are the f_{-3dB} and the peaking in the NTF.

In a higher-order loop, the parameters such as ζ and f_n cannot be directly applied to the equations for the second-order loop because the resulting frequency response can differ greatly. To still use the equation, for a given frequency response, we find an

effective f_n and effective ζ that result in the same bandwidth and peaking. Figure 3.8-(a) and (b) shows the corresponding f_{-3dB} for each value of f_n, and the corresponding peaking for each value of ζ. This method is verified by measuring the tracking jitter for the different loop bandwidths and frequency-response peaking. Jitter is calculated for the same parameters using $\sigma_{tr} = \frac{\kappa}{\sqrt{2}} \cdot \sqrt{\frac{1}{2\zeta\omega_n}}$. Table A.1 compares the measured and calculated jitter. By changing only one variable, we express the change in the jitter as a ratio. The ratio can be directly predicted from Figure 3.8-(a) or (b). The small error between measurement and predicted result is primarily due to the oscilloscope's inherent noise.

Table A.1: Comparison of estimated tracking jitter (by second-order PLL analysis) with measured tracking jitter at f_{ref} = 700MHz

f_{-3dB} (MHz)	Peak (%)	f_n (MHz)	ζ	ratio (Figure 3.8(a), (b))	estimated rms jitter (ps)	measured rms jitter (ps)
39	1.61	22.4	0.42	1	$\sigma = \kappa/(2.\sqrt{\zeta\omega_n}) = 3.51$	3.67
39	2.73		0.2	1.8/1.38 = 1.3	$\sigma = 1.3 \,.\, 3.51 = 4.56$	5
45	1.19	19.5	0.9	1	$\sigma = \kappa/(2.\sqrt{\zeta\omega_n}) = 2.57$	2.83
45	1.31		0.65	1.1/1.02 = 1.07	$\sigma = 1.07 \,.\, 2.57 = 2.75$	2.94
26	1.66	15.3	0.4	1	$\sigma = \kappa/(2.\sqrt{\zeta\omega_n}) = 4.35$	4.49
42	1.66		0.4	$\sqrt{26/42} = 0.79$	$\sigma = 0.79 \,.\, 4.35 = 3.42$	3.45

A.7 Is Jitter due to Input Clock Noise Convex?

The long-term jitter due to input clock noise is given by Equation 4.4 (or equivalently Equation A.22). To verify the convexity of jitter, the second derivite of jitter as a function of K_{loop} and ω_n should be positive.

$$\begin{cases} \dfrac{d^2\sigma_{rms}}{dK_{loop}^2} = \dfrac{-1}{4}\cdot\dfrac{\sqrt{N_{in}}}{\omega_0}\cdot\dfrac{(\omega_z)^{-2}}{\left[(\omega_z)^{-1}\cdot K_{loop}+\dfrac{1}{(\omega_z)^{-1}}\right]^{\frac{3}{2}}} < 0 \\ \dfrac{d^2\sigma_{rms}}{d(\omega^{-1}{}_z)^2} = \dfrac{-1}{4}\cdot\dfrac{\sqrt{N_{in}}}{\omega_0}\cdot\dfrac{\left[K_{loop}-\dfrac{1}{(\omega_z)^{-2}}\right]^2}{\left[(\omega_z)^{-1}\cdot K_{loop}+\dfrac{1}{(\omega_z)^{-1}}\right]^{\frac{3}{2}}}+\dfrac{\dfrac{1}{(\omega_z)^{-3}}}{\left[(\omega_z)^{-1}\cdot K_{loop}+\dfrac{1}{(\omega_z)^{-1}}\right]^{\frac{1}{2}}} \end{cases} \quad \text{‹A.23›}$$

As seen from Equation A.23, the second derivitive of jitter as a function of K_{loop} is always negative. Also, the second derivitive as a function of ω_n is conditionally positive. Therefore, the jitter is not a convex function.

A.8 Minima and Maxima of Total Jitter in a Second-Order PLL

The total jitter due to VCO and input clock noise is the sum of jitter variances:

$$\sigma_{tot}{}^2 = \sigma_{in}{}^2 + \sigma_{VCO}{}^2$$

or,

$$\sigma_{tot}{}^2 = \frac{N_{in}}{\omega_0{}^2}\cdot\left[(\omega_z)^{-1}\cdot K_{loop}+\frac{1}{\omega_z{}^{-1}}\right]+\frac{N_{VCO}}{f_0{}^2}\cdot\left[\frac{1}{\omega_z{}^{-1}\cdot K_{loop}}\right] \quad \text{‹A.24›}$$

We take the first derivative of total jitter as a function of K_{loop}:

$$\frac{d\sigma_{tot}}{dK_{loop}} = \frac{1}{2} \cdot \frac{\frac{N_{in} \cdot \omega_z^{-1}}{\omega_0^2} - \frac{N_{VCO}}{\omega_z^{-1} \cdot K_{loop}^2 \cdot f_0^2}}{\sigma_{tot}} \quad \text{‹A.25›}$$

σ_{tot} has only one minimum that occurs at:

$$K_{loop}\Big|_{\sigma_{tot} = min} = \frac{2\pi}{\omega_z^{-1}} \cdot \sqrt{\frac{N_{VCO}}{N_{in}}} \quad \text{‹A.26›}$$

Similarly, we take the first derivative of total jitter as a function of $(\omega_z)^{-1}$:

$$\frac{d\sigma_{tot}}{d(\omega_z^{-1})} = \frac{\frac{N_{in}}{\omega_0^2} \cdot \left[K_{loop} - \frac{1}{(\omega_z^{-1})^2}\right] \cdot -\frac{N_{VCO}}{(\omega_z^{-1})^2 \cdot K_{loop} \cdot f_0^2}}{\sigma_{tot}} \quad \text{‹A.27›}$$

σ_{tot} has only one minimum that occurs at:

$$\omega_z^{-1} = \sqrt{\frac{1 + \frac{N_{VCO}}{N_{in}} \cdot \frac{(2\pi)^2}{K_{loop}}}{K_{loop}}} \quad \text{‹A.28›}$$

Neither parameter yields any local minima.

Bibliography

[1] M. Saint-Laurent, et al., “A multi-PLL clock distribution architecture for gigascale integration,” *VLSI Proceedings on IEEE Computer Society Workshop*, pp. 30-35, April 2001

[2] G. A. Pratt, et al., “Distributed synchronous clocking,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 6, no. 3, pp. 314-328, March 1995

[3] H. Mizuno, et al., “A noise-immune GHz-clock distribution scheme using synchronous distributed oscillators”, *ISSCC Dig. Tech. Papers.*, pp. 404-405, Feb. 1998

[4] E. Fayneh and E. Knoll, “Clock generation and distribution for Intel Banias mobile microprocessor,” *Symposium on VLSI Circuits Dig. of Technical Papers*, pp. 17-20, June 2003

[5] V. Gutnik, et al., “Active GHz clock network using distributed PLLs”, *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1553-1560, November 2000

[6] F. O’Mahhony, et al., “10GHz clock distribution using coupled standing-wave oscillators,” *ISSCC Dig. Tech. Papers.*, pp. 428-429, Feb. 2003

[7] K.L. Wong, et al., “Cascaded PLL Design for a 90nm CMOS high performance microprocessor,” *ISSCC Dig. Tech. Papers.*, pp. 422-423, Feb. 2003

[8] K-Y.K. Chang, et al., “A 0.4-4Gb/s CMOS quad transceiver cell using on-chip regulated dual-loop PLLs,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 5, pp. 747-754, May 2003

[9] KL.J. Wong, M. Mansuri, H. Hatamkhani and CK.K. Yang, “A 27-mW 3.6-Gb/s I/O Transceiver,” *Symposium on VLSI Circuits Digest of Technical Papers*, pp. 99-102, June 2003

[10] E. Yeong and M. Horowitz, “A 2.4Gb/s/pin simultaneous bidirectional parallel link with per pin skew compensation,” *ISSCC Dig. Tech. Papers.*, pp. 255-256, Feb. 2000

[11] S. Sauter, et al., “Effect of parameter variation at chip and wafer level on clock skews,” IEEE Transactions on Semiconductor Manufacturing, vol. 13, issue 14, Nov. 2002

[12] P. Zarkesh-Ha, et al., “Characterization and modeling of clock skew with process variations,” IEEE Proceedings of Custom Integrated Circuits, pp. 441-444, May 1999

[13] H.-Y. Hsieh, et al., “Self-calibrating clock distribution with scheduled skews,” *IEEE International Symposium on Circuits and Systems*, pp. 470-473, 1998

[14] H. Sutoh, et al., “A clock distribution technique with an automatic skew compensation circuit,” *IEICE Transactions on Electronics*, pp. 277-283. Feb. 1998

[15] V. Gutnik, et al., “Embedded power supply for low-power DSP,” *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, vol. 5, no. 4, pp. 425-435, December 1997

[16] D. Duarte, et al., “A clock power model to evaluate impact of architectural and technology optimizations,” *IEEE Transactions on VLSI Systems*, Vol. 10, No. 6, December 2002

[17] D. Duarte, et al., “Impact of technology scaling in the clock system power,” *IEEE Computer Society Annual Symposium*, pp. 52-57, April 2002

[18] J. Tschanz, et al., “Dynamic-sleep transistor and body bias for active leakage power control of microprocessors,” *ISSCC Dig. Tech. Papers.*, pp. 102-103, Feb. 2003

[19] V. Gutnik and A. Chandrakasan, “An efficient controller for variable supply-voltage low power processing,” *Symposium on VLSI Circuits Dig. of Technical Papers*, pp. 158-159, 1996

[20] B. Razavi, *Design of Analog CMOS Integrated Circuits,* McGraw Hill, 2001

[21] B. Razavi, *Monolithic Phase-Locked Loops and Clock Recovery Circuits*, IEEE Press, 2003

[22] J.A. McNeill, “Jitter in ring oscillators,” *IEEE Journal of Solid-State Circuits*, vol. 32, no. 6, June 1997, pp. 870-879

[23] J.A.McNeill, “Jitter in ring oscillators,” Ph.D. dissertation, Boston University, 1994

[24] J. J. Rael and A. A. Abidi, “Physical processes of phase noise in differential LC oscillators,” *IEEE 2000 Custom Integrated Circuits Conference*, pp. 569-572

[25] A. Hajimiri, et al., “A general theory of phase noise in electrical oscillators,” *IEEE Journal of Solid-State Circuits*, vol. 33, no. 2, Feb. 1998, pp. 179-194

[26] A. Hajimiri, et al., “Jitter and phase noise in ring oscillators,” *IEEE Journal of Solid-State Circuits*, vol. 34, no. 6, June 1999, pp. 790-804

[27] R. Moore, “Phase noise and jitter”, *Agilent Technologies*, May 2001

[28] B. Razavi, “A study of phase noise in CMOS oscillators,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 3, March 1996, pp. 331-343

[29] A. Demir, A. Mehrotra and J. Roychowdhury “Phase noise in oscillators: a unifying theory and numerical methods for characterization,” *IEEE Trans. Circuits Syst. I*, vol. 47, no. 5, pp. 655-674, May 2000

[30] T. C. Weigandt, B. Kim, and P. R. Gray, “Analysis of timing jitter in ring oscillators,” *in Proc. ISCAS*, May 1994, vol. 4, pp. 27-30

[31] B. Kim, T. C. Weigandt, and P. R. Gray, “PLL/DLL system noise analysis for low jitter clock synthesizer design,” in Proc. ISCAS, May 1994, vol. 4, pp. 31-34

[32] M. J. E. Golay, “Monochromaticity and noise in regenerative electrical oscillator,” Proc. IRE, pp. 1473-1477, Aug. 1960

[33] X. Zhang, et al., “A theoretical and experimental study of the noise behavior of subharmonically injection locked local oscillators,” *IEEE Trans. on MTT*, vol. 40, no. 5, pp 895-902, May 1992

[34] J. A. Crawford, *Frequency Synthesizer Design Handbook*, Artech House, Inc. 1994

[35] V.F. Kroupa, “Noise properties of PLL systems,” IEEE Trans. Comm., vol. 30, no. 10, pp. 2244-52, Oct. 1982

[36] D. Huffman, “Extremely low noise frequency dividers,” *Microwave J.*, pp. 209-210, Nov. 1985

[37] M. M. Driscoll, “Phase noise performance of analog frequency dividers,” *IEEE Trans. Ultrasonic, Ferroelectrics and Freq. Cont.*, vol. 37, no. 4, pp. 295-304, July 1990

[38] A. A. L’vovich, “Design of noise-immune counter-type frequency dividers,” *Electrosvyaz, Translated in: Telecomm. and Radio Eng.*, part 1, vol. 29, no. 2, pp. 52-5, Feb. 1975

[39] F. Herzel, et al., “A study of oscillator jitter due to supply and substrate noise,” *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 46, no. 1, January 1999

[40] A. Hajimiri, “Noise in Phase-Locked Loops,” SSMSD, Southwest Symposium, pp. 1-6, February 2001

[41] K. Kishine, et al., “Loop-parameter optimization of a PLL for a low-jitter 2.5-Gb/s one-chip optical receiver IC with 1:8 DEMUX,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 1, January 2002, pp. 38-50

[42] K. Lim, et al., “A low-noise phase-locked loop design by loop bandwidth optimization,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 6, pp. 807-815, June 2002

[43] A. Mehrotra, “Noise analysis of phase-locked loops,” *IEEE Trans. Circuits Syst. I,* vol. 49, no. 9, pp. 1309-1316, Sep. 2002

[44] K. S. Kundert, “Modeling and simulation of jitter in PLL frequency synthesizers,” *Designer’s Guide www.designers-guide.com*, 2003

[45] D.C. Lee, “Analysis of jitter in phase-locked loops,” *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions*, vol. 49, issue 11, pp. 704-711, Nov. 2002

[46] M. Mansuri, et al., “Jitter Optimization Based on Phase-Locked Loop Design Parameters,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1375-1382, November 2002

[47] K.A. Jenkins and J.P. Eckhardt, “Measuring jitter and phase error in microprocessor phase-locked loops,” *IEEE Design and Test of Computers*, pp. 86-93, 2000

[48] P. Larsson, “Measurements and analysis of PLL jitter caused by digital switching noise,” *IEEE Journal of Solid-State Circuits*, vol. 36, no. 7, pp. 1113-1119, July 2001

[49] M. Saint-Laurent and M. Swaminathan, “Impact of power-supply noise on timing in high-frequency microprocessors,” *Electrical Performance of Electronic Packaging*, pp. 21-23, October 2002

[50] A. Demir, “Phase noise and timing jitter in oscillators with colored-noise sources,” *IEEE Transaction on Circuits and Systems-I*, vol. 49, no. 12, Dec. 2002

[51] J.M. Ingino, “A 4GHz 40dB PSRR PLL for an SOC Application,” *ISSCC Dig. Tech. Papers.*, pp. 392-393, Feb. 2001

[52] S. Sidiropoulos, et al., “Adaptive bandwidth DLL’s and PLL’s using regulated Supply CMOS buffers,” *Proceedings of 2000 IEEE Symposium on VLSI Circuits*, Dig. Tech. Papers, June 2000, Hawaii, pp. 124-127

[53] V. R. von Kaenel, et al., “A High-Speed, Low-Power Clock Generator for a Microprocessor Application,” *IEEE Journal of Solid-State Circuits*, vol. 33, no. 11, pp. 1634-1639, November 1998

[54] V. R. von Kaenel, et al., “A 320 MHz CMOS PLL for Microprocessor Clock Generation,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1715-1722, November 1996

[55] H. Ahn, et al., “A Low-Jitter 1.9-V CMOS PLL for UltraSPARC Microprocessor Applications,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 3, pp. 450-454, March 2000

[56] I.A. Young, et al., “A PLL clock generator with 5 to 110 MHz lock range for microprocessors,” *IEEE Journal of Solid-State Circuits*, Nov. 1992, vol.27, no.11, pp. 1599-607

[57] J.G. Maneatis, et al., “Low-jitter process-independent DLL and PLL based on self-biased techniques” *IEEE Journal of Solid-State Circuits*, Nov. 1996. vol.31, no.11, p. 1723-32

[58] S-J Lee, et al., “A novel high-speed ring oscillator for multiphase clock generation using negative skewed delay scheme,” *IEEE Journal of Solid-State Circuits*, vol. 32, no. 2, pp. 289-291, February 1997

[59] K.Y.K Chang, et al., “A 0.4-4Gb/s CMOS Quad Transceiver Cell using On-chip Regulated Dual-Loop PLLs,” *Proceedings of 2002 IEEE Symposium on VLSI Circuits*, Dig. Tech. Papers, pp. 88-91, June 2002

[60] K. Minami, et al., “A 0.10mm CMOS, 1.2V, 2GHz Phase-Locked Loop with Gain Compensation VCO,” *Proceedings of 2001 IEEE Custom Integrated Circuits Conference*, Dig. Tech. Papers, pp. 213-216, May 2001

[61] M. Mansuri and C.-K. Yang, “A low-power low-jitter adaptive bandwidth PLL and clock buffer,” *ISSCC Dig. Tech. Papers.*, pp. 430-431, Feb. 2003

[62] P. Larsson, “A 2-1600-MHz CMOS Clock Recovery PLL with Low-Vdd Capability,” *IEEE Journal of Solid-State Circuits*, vol. 34, no. 12, pp. 1951-1960, December 1999

[63] M.G. Johnson, et al., “A Variable Delay Line PLL for CPU-Coprocessor Synchronization,” *IEEE Journal of Solid-State Circuits*, vol. 23, no. 5, pp. 1218-1223, October 1988

[64] Y. Moon, et al., “An all-analog multiphase delay-locked loop using a replica delay line for wide-range operation and low-jitter performance,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 3, pp. 377-384, March 2000

[65] W.J. Dally, et al., “Clock multiplying delay-locked loop for data communication,” *US patent pending*

[66] R. Farjad-Rad, et al., “A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1804-1812, December 2002

[67] V. Gutnik and A. Chandrakasan, “On-chip picosecond time measurement,” *Proceedings of 2000 IEEE Symposium on VLSI Circuits*, Dig. Tech. Papers, pp. 52-53, June 2000

[68] B-J Lee et al., “A 2.5-10Gb/s CMOS transceiver with alternating edge sampling phase detection for loop characteristic stabilization,” *ISSCC Dig. Tech. Papers.*, pp. 76-77, Feb. 2003

[69] Y. Moon et al., “A 0.6 - 2.5GBaud CMOS tracked 3x oversampling transceiver with dead-zone phase detection for robust clock/data recovery,” *ISSCC Dig. Tech. Papers.*, pp. 212-213, Feb. 2001

[70] S-H Lee et al., “A 5Gb/s 0.25mm CMOS jitter-tolerant variable-interval oversampling clock/data recovery circuit,” *ISSCC Dig. Tech. Papers.*, pp. 256-257, Feb. 2002

[71] R. Kuppuswamy et al., “On-die clock jitter detector for high speed microprocessors,” *Proceedings of 2001 IEEE Symposium on VLSI Circuits*, Dig. Tech. Papers, pp. 187-190, June 2001

[72] H. Ebenhoech, “Make IC digital frequency comparators,” *Electron. Design*, vol. 15, no. 14, pp. 62-64, July 1967

[73] H. Partovi et al., “Flow-through latch and edge-triggered flip-flop hybrid elements,” in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, San Francisco, CA, Feb. 1996, pp. 138-139

[74] H. Partovi et al., “Phase frequency detector having reduced blind spot,” *US patent*, patent no. 5,963,059, Oct. 5, 1999

[75] M. Mansuri, et al., “Fast Frequency Acquisition Phase-Frequency Detectors for GSa/s Phase-Locked Loops,” *IEEE J. Solid-State Circuits*, vol. 37, no. 10, pp. 1331-1334, Oct. 2002

[76] C. Webb, et al., “A 400MHz S/390 microprocessor,” *IEEE J. Solid-State Circuits*, vol. 32, no. 11, pp. 1665-1675, Nov. 1997

[77] J.A. Davis, et al., “Interconnect limits on gigascale integration (GSI) in the 21st century,” Proceedings of IEEE, vol. 89, issue 3, pp. 305-324, March 2001

[78] R. Best, *Phase-Locked Loops*, 3rd ed, McGraw Hill, 1997

[79] F.M. Gardner, “Charge-pump phase-lock loops,” *IEEE Transactions on Communications,* Nov. 1980, vol.COM-28, no.11, p. 1849-58

[80] J. Hein, et al., “Z-domain model for discrete-time PLL’s,” IEEE Transactions on Circuits and Systems, Nov. 1988

[81] F. Gardner, *Phase Lock Techniques,* Wiley 1979

[82] F. Gardner, *Introduction to Random Processes: With Applications to Signals and Systems*, McGraw-Hill, 1989

[83] M. Rau, et al., “Clock/data recovery PLL using half-frequency clock,” *IEEE Journal of Solid-State Circuits*, Jul. 1997, vol.32, no.7, pp. 1156-9

[84] SS. Sidiropoulos, M.A. Horowitz, “A semidigital dual delay-locked loop,” *IEEE Journal of Solid-State Circuits*, Nov. 1997, vol.32, no.11, pp. 1683-92

[85] N.H.E. Weste, K. Eshraghian, *Principles of CMOS VLSI Design A Systems Perspective, 2nd Edition*, Addison Wesley, Menlo Park, CA. 1993