DESIGN OF ALL DIGITAL PHASE-LOCKED LOOP IN SERIAL LINK COMMUNICATION

BY

YUBO LIU

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Adviser:

Professor José Schutt-Ainé

ABSTRACT

The speed of wireline and wireless communication systems has been increasing aggressively over the past decade. Multi-GHz clocks are in demand more than ever. In particular, wireline inter-IC communications systems such as broadband Internet, multi-core CPU and system-on-chip have fueled the research on faster on-chip clock synthesizers. In addition, mobile products such as cell-phones and tablets have permeated the consumer market. Since these devices are battery-powered, it is necessary to minimize the battery consumption of the communication system circuitry inside to extend the battery life. As a result, low-power inter-IC communication design is another topic that is gaining interest.

In high speed links, clocking circuitry is vital, and phase-locked loop (PLL) is at the heart of every on-chip clocking circuit. The clocking circuitry needs to be robust, low-power and fast in order to fulfill the increasing demand for high data rate links. The performance of the input/output (I/O) communication channel needs to scale proportionally with the semiconductor fabrication technology (SFT). However, conventional analog PLLs are often incompatible from one technology node to the next and require entirely new designs. In recent years, with the increased performance of digital circuits, all digital PLL (ADPLL) has achieved speed performance similar to that of analog PLL. Since digital logic is more robust, portable, and power efficient, ADPLL is gaining traction in research.

This thesis presents the fundamentals and an in-depth analysis of the conventional analog PLL in Chapters 2 and 3. Then the discussion dives into ADPLL. Chapter 4 presents the building blocks and loop analysis of the ADPLL. Chapter 5 presents jitter sources and jitter analysis inside the ADPLL. Chapter 6 presents an ADPLL in model and transistor design. It has center frequency of 1.6GHz and operates from 1.2GHz to 2.0GHz. Chapter 7 concludes the thesis and discusses future work.

To my parents, for their love and support.

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my advisor Professor José Schutt-Ainé for the continuous support of my graduate study and research. His support is always there whenever I need it. His guidance helped me to overcome numerous difficulties encountered during the research. I could not have imagined having a better advisor and mentor for my graduate study.

Besides my advisor, I thank Professor Pavan Kumar Hanumolu for his patience, motivation, enthusiasm, and immense knowledge. He answered all my questions patiently and thoughtfully despite his busy schedule. It is because of his help that this thesis is possible.

I would also like to express my sincere gratitude to my wonderful colleagues, mentors and friends in graduate school: Rishi Ratan, Da Wei, Woo-Seok Choi, Xu Chen, Drew Newell, Xinying Wang, Drew Handler, Colin Madigan, Yi Ren, Ishita Bisht, Rushabh Mehta, Romesh Nandwana, Mrunmay Talegaonkar, Ahmed Elkholy, Saurabh Saxena, Tejasvi Anand and Guanghua Shu for the stimulating discussions during the sleepless nights we were working together and for all the fun we have had together.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Communication systems are essential in modern day life. These systems allow for transfer of information via wireline and wireless systems. For wireline systems, input/output (I/O) links are omnipresent in today's electronics. They provide the communication interface for backplane channels such as network switches and memory interfaces. As data rates reach multi-GHz range, parallel links become problematic due to channel phase offset and intern-channel coupling. In addition, as devices shrink, the number of pins is limited, and numerous parallel links cannot be supported. As a result, serial link communication is the better choice in modern link design. Phase locked loops (PLL) are vital components in high speed and low-power I/O link design. Designing a PLL for such purposes is extremely challenging, for the following reason. In recent years, consumer products such as personal computers, tablets, and mobile phones have permeated consumers' everyday lives. Processing power and power efficiency have become two vital aspects in defining the success of a modern consumer product. To satisfy the constant thirst for more powerful processors and faster data rates, there have been constant advancements in semiconductor fabrication technology (SFT). The aggressive scaling of semiconductor technology has driven transistors to smaller size and increased the performance of integrated circuits (IC). Along with the desire for instant information, the desire for power efficiency is also a driving force pushing the limits of IC technology. In response, novel designs and new forms of technologies have kept up with the demand. As the performance of individual IC chips constantly increases, the overall system performance is expected to also increase proportionally. However, that is not the case. Inter-IC interface design is the bottleneck of a robust, high-speed, and low-power

portable device. While intra-IC (I/O) links only travel very short distances relative to the size of the IC, inter-IC I/O links have to travel much longer distances. This is extremely problematic at high frequencies (multi-GHz) as long transmission lines are susceptible to loss, inter-symbol interference (ISI), and external distortion that cause degradation in the transmitted data. In addition, lower power exacerbates the situation. As a result of these difficulties, the demand for on-chip clock synthesis that can produce high clock frequencies has pushed the need for PLLs.

Over the last 50 years, advances in SFT coupled with innovations in IC technology scaling have fueled an unparalleled growth in computing. This aggressive scaling has revolutionized every aspect of modern society and triggered an insatiable demand for faster data rates and higher processing power, resulting in clock frequencies and corresponding data rates approaching multi-GHz and multi-Gbps ranges in everyday computing devices like personal computers, mobile devices, entertainment consoles and other such devices. Access to information promptly and efficiently in terms of power and portability/ease of use is the major driver pushing the limits of IC technology. Thus, the need for robust, high-speed, low-power and highly integrable compact systems-on-chip (SOCs) is paramount for inter-IC communication interfaces such as network switches and processor/memory interfaces across backplane channels. In order to meet this growing demand for wideband systems, clocking circuitry also has to keep increasing its performance.

In on-chip, fast clock synthesizers resides the PLL. The conventional analog PLL has been around for a few decades and has been studied very extensively. However, analog PLL suffers from large power consumption, large area, susceptibility to noise, and inability to transition from one technology node to the next. Digital PLLs have been drawing interest in recent years. However, the speed of the digital circuits has often been the bottleneck to achieving higher performance. With the scaling of SFT, the speed performance of digital circuits has been scaling up. This allows for the possibility of all digital PLL (ADPLL) to achieve speed performance similar to that of the analog counterpart. In addition, ADPLL brings advantages that alleviate the problems presented earlier, allowing the PLL to be low-power, compact, less susceptible to noise, and more portable. This thesis presents an ADPLL from block level to transistor level. An ADPLL in TSMC-65nm technology is implemented and achieves the same speed as an analog counterpart.

CHAPTER 2

PLL IN LINK COMMUNICATION

Input/output (I/O) links are omnipresent in today's electronics. They provide the communication interface for backplane channels such as network switches and memory interfaces. While parallel I/Os are still predominantly used for intra-IC communication, they are losing popularity in inter-IC communication because of timing inaccuracy and limited area on board since the channel is much longer. Thus serial I/Os have been gaining traction as the design for the link communication. Figure 2.1 [1] shows a generalized model of a high-speed serial link. The sender serializes the parallel on-chip data and sends out the data via the Transmitter (TX); on the other side, the Receiver (RX) receives the incoming information and recovers the data and the clock, and then the deserializer converts the data back to parallel form. A clock signal, generated by the PLL, is used to provide the timing so that the link can accurately serialize, transmit, receive, and recover the information. Any variation in the timing can result in the degradation or even loss of information. Therefore, the PLL design is very critical and challenging.

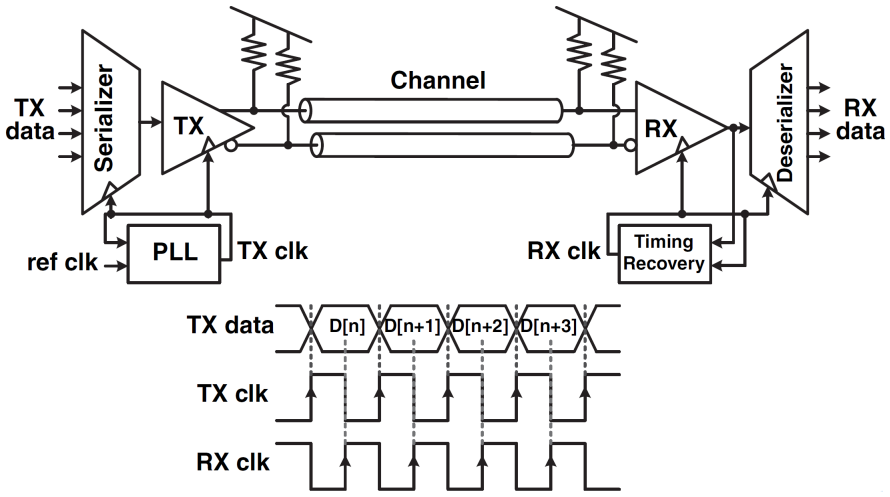

Figure 2.1: Basic High-Speed Electrical Link System

A well designed PLLs considerations of many tradeoffs, which are often application-specific. The main tradeoffs usually revolve around speed/data rates, power, noise, and area.

As mentioned in the Introduction, data rate scales proportionally with SFT; however, the inter-IC channel bandwidth is still very limited by non-linear effects at high clock and data rate. An example backplane channel interface is shown in Figure 2.2 [1]. This is a generic interface that exists inside a device such as personal computer. Note that each component in the interface has a non-ideal effect that causes degradation in the transmitted information.

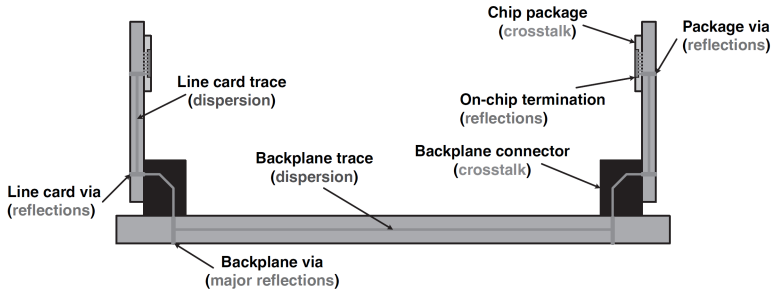

Figure 2.2: Example Backplane Channel Interface

The channel here consists of print circuit board (PCB) traces, vias, and connectors. At high data rates, the physical electrical path becomes distorted and lossy. One method of determining how a channel performs is to use an eye diagram. A model channel is built using Ansys HFSS and simulated at two different data rates. Figure 2.3 and Figure 2.4 [2] show the eye diagrams of two different simulations. The one-Gbps eye diagram shows a clean, open eye. The ten-Gbps eye diagram shows a closed and distorted eye in which it is difficult to identify the data. Once the channel is defined, it is important to design a robust PLL that can provide a clean and low-jitter clock. As shown in Figure 2.5, jitter is a time domain indeterminacy and results from variation of a clock transition due to many factors such as sensitivity to process, voltage, and temperature (PVT) variations. High jitter will cause poor eye diagrams. Therefore, a well designed PLL is critical to minimize the jitter in both TX and RX clock signals.

In addition to signal integrity, power consumption is another important figure of merit in serial link design and is often dominated by the PLL. As mentioned before, high data rates require high clock frequency, which

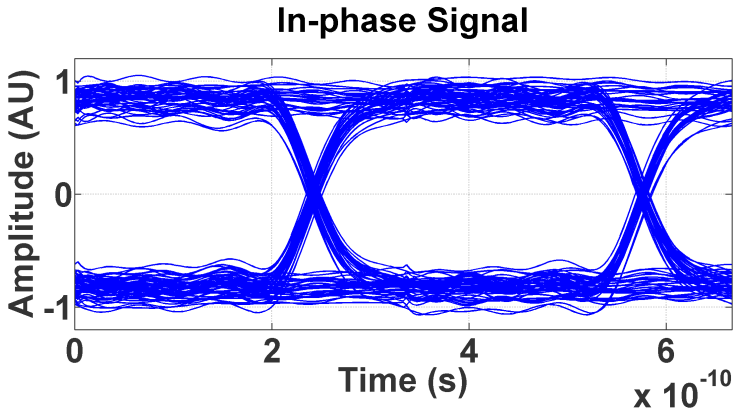

Figure 2.3: 1 Gbps Backplane Link Eye Diagram

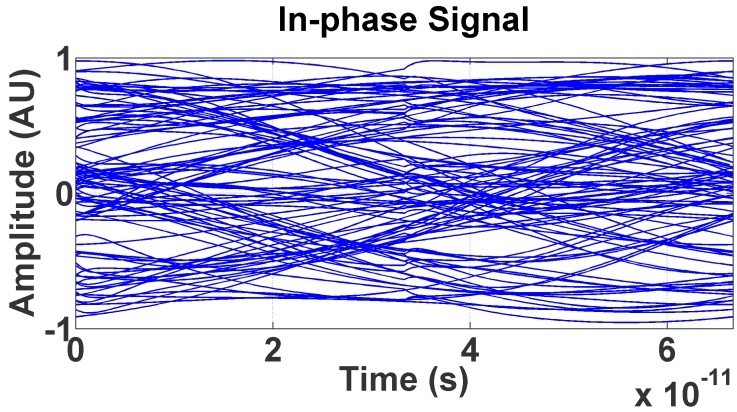

Figure 2.4: 10 Gbps Backplane Link Eye Diagram

the scaling SFT enables; however, increasing clock speed also scales with power consumption. Too much power consumption means decreased battery performance, which is especially undesirable for mobile devices. Figure 2.6 illustrates the power breakdown of a 4.8 Gbps serial link designed for a fully buffered DIMM system [3]. Note that more than 80% of total power is consumed by the clock system and 50% of this power is claimed by the PLL (TX and RX PLL). Therefore, designing a lower-powered PLL will directly impact the power consumption of the entire serial link.

As a serial link designer, ensuring integrity of the transmitted signal and lowering power consumption are two main goals. Therefore, it is critical to focus on the PLL design in order for the serial link to achieve its necessary requirements.

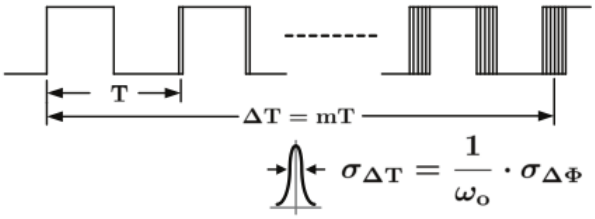

Figure 2.5: Time Domain Jitter

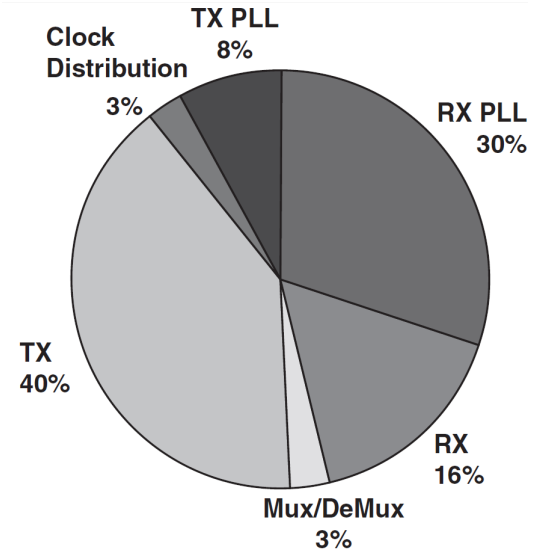

Figure 2.6: Power Breakdown for a Fully Buffered DIMM System

CHAPTER 3

PHASE LOCKED LOOP FUNDAMENTALS

This chapter will introduce the fundamentals of the PLL. The building blocks will be presented, and analysis will follow.

3.1 PLL Applications

Phase locked loops are widely adopted by wireless and wireline communication systems for a variety of applications. The main applications of PLLs include:

- Clock synthesizing - generating a higher frequency clock signal from a lower frequency one,
- Frequency tracking - ensuring a clock signal does not vary due to external effects
- Clock and data recovery (CDR) - important block in link receiver design
- Compensation for clock skew - useful in distributed clocking systems

CMOS technology has enabled cheap, fast, and accurate implementation of PLL in digital systems.

3.2 PLL Fundamental Blocks

As technology advances, the individual components of PLLs become more sophisticated. However, PLL has been around for a long time and its theory has been studied extensively. Figure 3.1 illustrates the basic building blocks of a conventional PLL. The basic blocks of a PLL include:

- Phase detector (PD)
- Charge pump (CP)
- Loop filter (LF)
- Voltage controlled oscillator (VCO)
- Divider

As illustrated in Figure 3.1, a PLL is a closed-loop negative feedback system that reacts to the phase relationship between the incoming reference clock and the output clock. When the two phases drift apart, the PLL will be able to track the phase changes that are within the PLL bandwidth. Note that the divider indicates that output frequency is higher than that of the reference, which means the PLL can effectively serve as a frequency multiplier (to higher frequency).

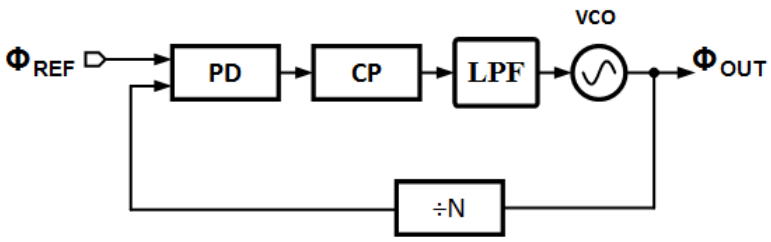

Figure 3.1: PLL Block Diagram

3.2.1 Phase Detector

The PD function is to detect the difference between the incoming reference clock and the divided clock from the feedback path. The relative phase information is then passed onto the next stage. The most basic relationship is described as follows:

$$V_e = K_{PD} \times \phi_e \tag{3.1}$$

K_{PD} is the gain of the PD. Note that the output is a voltage and the inputs are phase.

Unfortunately, this formulation is impractical because as ϕ_e increases continuously, the output voltage will increase continuously. Thus we need to

include another limitation to avoid unreasonable voltage:

$$\phi_e| < 2\pi \tag{3.2}$$

This condition indicates that the phase will roll over and repeat itself as the input difference increases.

There are various PD implementations, ranging from analog to digital. A simple analog filter's behavior is shown in Figure 3.2. The output is not ideal for the PLL loop dynamic because the gain is not linear. Non-linear PD gain will lead to undesired variations. As a result, digital PDs are the more common choice.

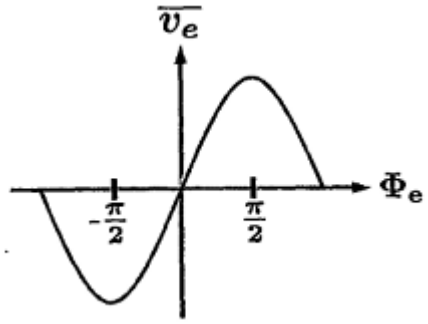

Figure 3.2: Analog PD Operation

Digital Phase Detector As CMOS technology is the norm in digital circuitry, it makes sense for CMOS PLL to include digital CMOS PD. An EXOR gate is the most simple PD. A logic HIGH from either input translates to a logic HIGH at the output. As shown in Figure 3.3, the gain of the PD is linear across a region. Outside the largest difference, the phase rolls over and starts again. This has the following implications:

$$LinearRange = \pm\frac{\pi}{2} \tag{3.3}$$

$$K_{PD} = \frac{2V_e}{\pi} \tag{3.4}$$

Another advantage of EXOR PD is the ability to hold its gain linearity in case of any duty cycle offset. When a duty cycle offset occurs, the linearity range shrinks while the gain of the PD remains linear. However, the problem

of an EXOR PD is that its range is small and can become even smaller when duty cycle offset occurs. The range can be extended by a state PD, also known as phase frequency detector (PFD).

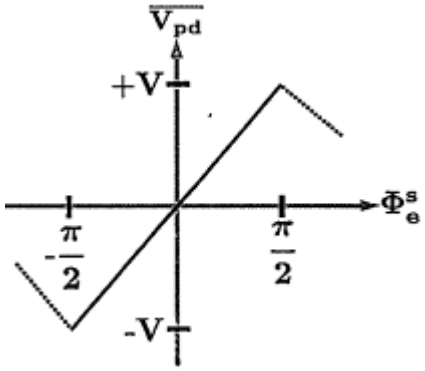

Figure 3.3: EXOR PD Operation

Phase Frequency Detector Figures 3.4 and 3.5 show the PFD and its operation. Rather than one output, the PFD has three states: UP, DN, and RESET. Its UP and DN output are from the feedback path's point of view with respect to the reference signal. From the operation graph, two observations can be made. The first is that the linear range is extended to $\pm\ 2\pi$. This is much larger than that of the EXOR PD case. The second observation is that the positive and negative phase differences are only on the respective side, and both cannot happen within one cycle. Since the plot represents phase information, the area under the plot is the integration of the phase, which is frequency. The RESET path is added to indicate which incoming frequency is faster. With this extra piece of information, one can see that the PFD proves to be much more robust. On the other hand, since the PFD adopts flip flops, timing can be problematic as its uncertainty contributes to the uncertainty of the PLL loop in the form of jitter. Jitter will be discussed more in detail in a later section.

3.2.2 Charge Pump

Since the PFD translates phase information into digital UP and DN signals, these logic ONEs and ZEROs cannot be directly fed into the VCO. The CP is used to charge or discharge from the PLL loop. Since moving charges will

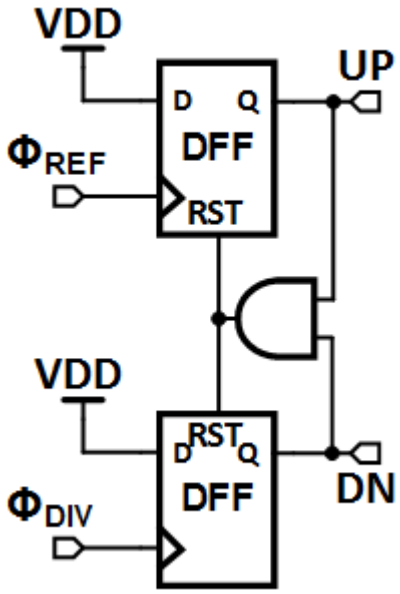

Figure 3.4: PFD Schematic

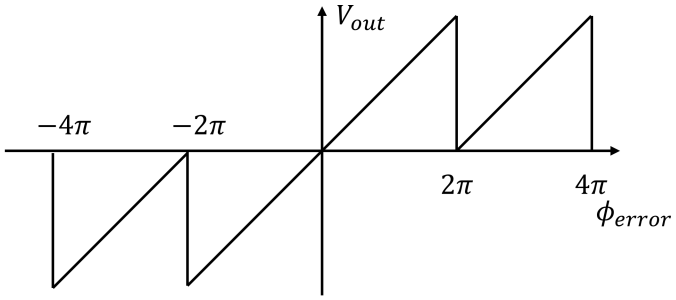

Figure 3.5: PFD Operation

induce current, the CP consists of two current sources, as shown in Figure 3.6. Under ideal conditions, the UP and DN current utilize equal amounts of currents. Thus, the average gain of the CP is:

$$K_CP = \frac{I_CP}{2\pi} \tag{3.5}$$

Usually the PFD and the CP are combined and modeled together as a single block because the output of the PFD directly controls the CP. In that case, the gain of the combined block is still the same as Equation (3.5). A major concern of the CP is the equality of the current provided by the UP and DN current sources. This is due to the fact that the sourcing current requires PMOS while the sinking current requires NMOS and matching them

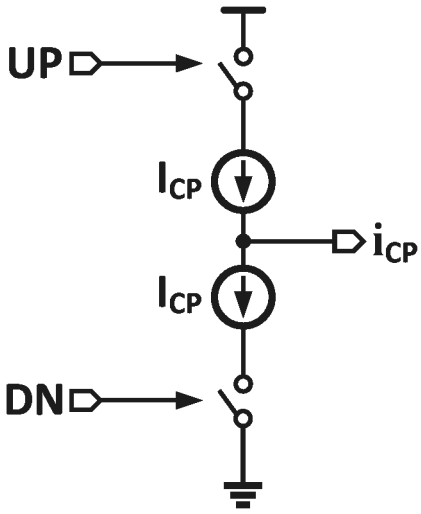

Figure 3.6: Charge Pump Schematic

is difficult. It is problematic because different currents will cause unequal amounts of charge flow; over a period of time, the control voltage will saturate to either rail and the PLL loop will break down. While complete matching of the current is impossible due to process variations, approaches can be taken to minimize the effect. The approaches are out of scope for this thesis; readers can refer to [4] for charge pump current matching.

3.2.3 Loop Filter

The loop filter is usually in the form of a low pass filter (LPF). Its purpose is twofold. First, it suppresses the high frequency switching noise from the digital blocks in the PFD. Second, it holds the charges so the sourcing/sinking current from the CP can be stored in the form of voltage, which is outputted to control the VCO. The simplest filter is a capacitor. The capacitor achieves the two tasks above. However, the simple capacitor adds an additional order to the PLL loop and creates potential instability. From basic control theory, we need to introduce a zero to stabilize. Thus, a resistor is needed as shown in Figure 3.7.

The RC LPF alleviates the stability problem. However, the CP has current fluctuation, which will induce voltage fluctuation on the LF output, and the VCO's output frequency will vary accordingly. The RC LPF can further be expanded with another capacitor in parallel as shown in Figure 3.8. This extra capacitor is used to limit the ripple of the control voltage into the VCO

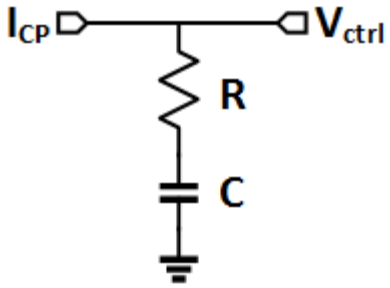

Figure 3.7: RC Loop Filter

[5]. The transfer function is shown in Equation (3.6). So why not use a very large capacitor so the ripple can be as small as possible? Because if the extra capacitor is too large, then the area and power will increase drastically.

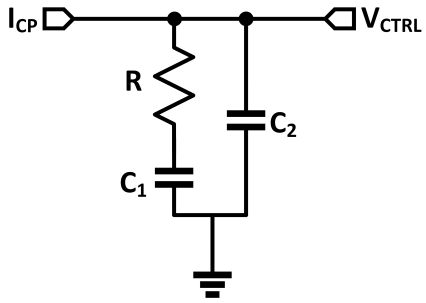

Figure 3.8: Loop Filter With Stabilizing Capacitor

$$LF(s) = \frac{V_{ctrl}(s)}{i_{Charge\ Pump}} = \frac{s + \frac{1}{RC_1}}{C_2 s(s + \frac{C_1+C_2}{RC_1C_2})} \tag{3.6}$$

3.2.4 Voltage Controlled Oscillator

The most important component in the PLL is the VCO. It generates the output clock frequency according to the voltage provided from the LF. It is also a subject of extensive studies of various design topologies. LC-tank, delay lines, and ring oscillators are some common choices, and each has its advantages and disadvantages for different applications. The transfer function is best to be analyzed in Laplace transform. It is formulated as follows:

$$\omega_{out}(t) = K_{VCO} v_{ctrl}(t) \tag{3.7}$$

$$\mathcal{L}[\omega_{out}(t)] = \omega_{out}(s) = K_{VCO}v_{ctrl}(s) \tag{3.8}$$

$$\phi_{out}(t) = \int_0^t \omega_{out}(\tau)d\tau = \int_0^t K_{VCO}v_{ctrl}(\tau)d\tau \tag{3.9}$$

$$\mathcal{L}[\phi_{out}(t)] = \phi_{out}(s) = \frac{\omega_{out}(s)}{s} = \frac{K_{VCO}v_{ctrl}(s)}{s} \tag{3.10}$$

Therefore, from the above, the Laplace transform function for VCO is derived to be Equation (3.11). The K_{VCO} is defined as the VCO gain.

$$H_{VCO}(s) = \frac{\phi_{out}(s)}{v_{ctrl}(s)} = \frac{K_{VCO}}{s} \tag{3.11}$$

3.2.5 Divider

The output frequency of the PLL is an integer-multiple times greater than the reference frequency. In order to feed back the output clock signal to the PFD to compare with the incoming reference frequency, the frequency has to be lowered to that of the reference. In the case of tracking the incoming frequency, the output can be directly fed back to the PFD. The divider is usually a power of 2, which makes the design simpler as it can be achieved by cascading flip-flops. If the PLL is used to track the reference clock, then the dividing ratio is simply 1 and a divider is not necessary.

3.2.6 Loop Concept

As mentioned above, the PLL is a negative feedback system. In a feedback loop system, it is important to know two concepts before analyzing the loop dynamics. These two concepts usually provide enough information to analyze the loop dynamics of the PLL [6]. The two concepts are:

1. The order of the PLL is the number of poles in the loop
2. The type of the PLL is the number of integrators in the loop

The VCO will automatically increase 1-order to the PLL. From 3.11, the transfer function in Laplace transform indicates that the VCO has a pole at origin. In addition, the transfer function also indicates that the VCO acts as an integrator. Thus, any VCO based PLL has to be at least 1st order

and type 1. The additional order and type will increase as the number of LF poles increases. From an acquisition time point of view, increasing the order will decrease the lock time; however, a feedback loop generally cannot handle too many poles because of decreased phase margin, which impacts the stability of the loop. As a result, the PLL is generally 3rd order to strike the trade-off.

Since the PLL is a feedback loop, its stability is particularly important. A robust design should be stable under external and internal disturbances. Stability is closely related to phase margin (PM). It is defined to be the phase difference between 180deg and the phase when the magnitude of the system's gain is 1. The loop becomes more unstable as PM decreases. As a result, a first order PLL is always stable since it only has one pole, which only contributes to 90deg phase shift. Practical PLLs, however, are usually type 2, and order 3. Figure 3.9 [2] shows the points where ideal PM should exist.

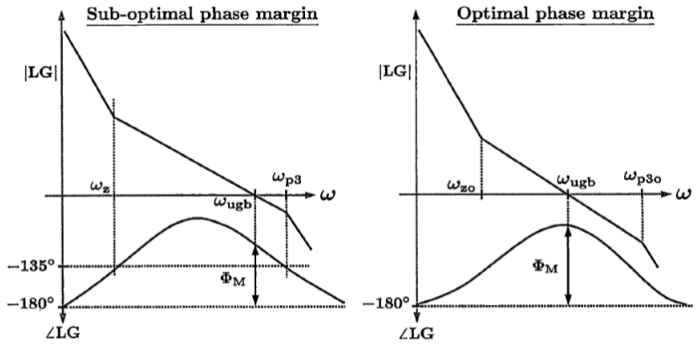

Figure 3.9: Loop Gain Response

3.2.7 Loop Analysis

From the previous sections, transfer functions in Laplace domain can be combined to produce the open-loop transfer function of the PLL.

$$LG_O(s) = K_{PD} \cdot F(s) \cdot \frac{K_{VCO}}{s} \tag{3.12a}$$

$$= K_{PD} \cdot K_{VCO} \cdot \frac{s + \frac{1}{RC_1}}{C_2 s^2 (s + \frac{C_1 + C_2}{RC_1C_2})} \tag{3.12b}$$

The poles and zeros can be determined as

$$\omega_z = \frac{1}{RC_1};\ \omega_{p1} = \omega_{p2} = 0; \omega_{p3} = \frac{C_1 + C_2}{RC_1C_2} \tag{3.13}$$

The phase margin can be determined as follows:

$$\phi_M = \arctan(\frac{\omega_{ugb}}{\omega_z}) - \arctan(\frac{\omega_{ugb}}{\omega_p 3}) \tag{3.14}$$

$$\omega_{ugb} = \omega_z \sqrt{\frac{C_1}{C_2} + 1} \tag{3.15}$$

where ω_{ugb} is the open-loop unity gain bandwidth and $\omega_z < \omega_{ugb}$. To achieve maximum stability, a high phase margin has to be chosen. Thus, the values of C_1 and C_2 have to be determined carefully. Equation (3.16) shows the maximum ϕ_M value from a specific ratio pair. Calculating the expression of ϕ_{M_max} needs the first-order derivative of Equation (3.14) with respect to ω_{ugb} and to equate the result to zero; this will yield the maximum value that ϕ_{M_max} can take. Note that the ϕ_M depends on the ratio of the poles, and thus the ratio of C_1 and C_2.

$$\phi_{M\_max} = \arctan(\sqrt{\frac{C_1}{C_2} + 1}) - \arctan(\frac{1}{\sqrt{\frac{C_1}{C_2} + 1}}) \tag{3.16}$$

The bandwidth ω_{ugb}, phase margin ϕ_M, and resistance R need to be first determined according to specifications. These values are determined from noise, stability, and power requirements. Then the ratio of $\frac{C_1}{C_2}$, or K_c, can be determined as follows:

$$K_c = \frac{C_1}{C_2} = 2(\tan^2(\phi_M) + \tan(\phi_M)\sqrt{\tan^2(\phi_M) + 1}) \tag{3.17}$$

Obviously only knowing the K_c is not enough to design the loop filter with proper component values. From Equation (3.15) and Equation (3.13), we can solve the equations for C_1 and C_2:

$$C_1 = \frac{1}{\omega_z R}; C_2 = \frac{C_1}{K_c}; \tag{3.18}$$

Finally, from Equations (3.5), (3.12b) and (3.13) we can determine the

value for the charge pump current, I_CP:

$$I_{CP} = \frac{2\pi C_2}{K_{VCO}} \cdot \omega_{ugb}^2 \cdot \sqrt{\frac{\omega_{p3}^2 + \omega_{ugb}^2}{\omega_z^2 + \omega_{ugb}^2}} \tag{3.19}$$

Once the parameters are determined, it is vital to analytically confirm that the PLL will lock when a frequency step is applied at the input. This is necessary as an external condition may cause this sudden change. Without loss of generality, assume an input frequency step $\omega_{in} = \frac{\Delta\omega}{s}$, then $\Phi_{in}(s) = \frac{\Delta\omega}{s^2}$. We can first obtain the PLL's closed loop transfer function:

$$H_{PLL}(s) = \frac{LG(s)}{1 + LG(s)} \tag{3.20}$$

Then, we can define the steady state error transfer function in Equation (3.21), which denotes the relationship between the error phase and the input phase.

$$\frac{\Phi_{error}(s)}{\Phi_{in}(s)} = H_e(s) = 1 - H_{PLL}(s) = \frac{1}{1 + LG(s)} \tag{3.21}$$

LG is from (3.12b). Note that in order to ensure that the PLL can sustain a step change, the ratio H_e has to disappear after some time, meaning the loop eventually settled and the difference has been corrected. By applying the final value theorem, we can derive the steady state error to be:

$$\Phi_{ss\_error}^{F_{step}} = \lim_{s\to 0} s \cdot H_e(s) \cdot \Phi_{in}(s) \tag{3.22a}$$

$$= \lim_{s\to 0} s \cdot \frac{1}{1 + LG(s)} \cdot \frac{\Delta\omega}{s^2} \tag{3.22b}$$

$$= \lim_{s\to 0} \frac{[RC_1C_2s^2 + (C_1 + C_2)s]\Delta\omega}{RC_1C_2s^3 + (C_1 + C_2)s^2 + K_{VCO}K_{PD}s + 1} \tag{3.22c}$$

$$= \frac{0}{1} \tag{3.22d}$$

$$= 0 \tag{3.22e}$$

Equation (3.22e) indicates that the phase error will eventually converge to 0, meaning the type 2, 3rd order PLL can eliminate any phase error and adjust according to frequency step changes. As a result of Equation (3.12b) and (3.22e), this is a stable and robust PLL loop.

3.2.8 PLL Noise Analysis

Noise has a direct impact on the performance of the PLL. Poor noise isolation will result in variation of the output clock frequency, which is an undesired effect as a clocking circuitry. In order to understand how noise affects the loop, a more direct and mathematical analysis is needed.

Figure 3.10 [2] shows the PLL loop with potential noise sources added. Note that it is impossible to eliminate all noise since every component can contribute noise. The correct approach is to understand the noise sources and design addition circuitry to counter it.

The noise sources can also be characterized in their mathematical domain, and thus included into the transfer function of the PLL loop. Thus the output referred noise and the noise transfer function (NTF) can be derived according to Figure 3.10.

$$S_{\Phi_{OUT}}^{\Phi_{IN}} = S_{\Phi_{IN}} |NTF_{IN}(s)|^2 \tag{3.23a}$$

$$S_{\Phi_{OUT}}^{i_{CP}} = S_{i_{CP}} |NTF_{CP}(s)|^2 \tag{3.23b}$$

$$S_{\Phi_{OUT}}^{v_R} = S_{v_R} |NTF_R(s)|^2 \tag{3.23c}$$

$$S_{\Phi_{OUT}}^{\Phi_{VCO}} = S_{\Phi_{VCO}} |NTF_{VCO}(s)|^2 \tag{3.23d}$$

where

$$NTF_{IN}(s) = \frac{\Phi_{OUT}(s)}{\Phi_{IN}(s)} = \frac{N \cdot LG(s)}{1 + LG(s)} \tag{3.24a}$$

$$NTF_{DIV}(s) = NTF_{IN}(s) \tag{3.24b}$$

$$NTF_{CP}(s) = \frac{\Phi_{OUT}(s)}{i_{CP}(s)} = \frac{2\pi}{I_{CP}} \cdot NTF_{IN}(s) \tag{3.24c}$$

$$NTF_R(s) = \frac{\Phi_{OUT}(s)}{v_R(s)} = \frac{\frac{K_{VCO}}{s}}{1 + LG(s)} \tag{3.24d}$$

These relations can be used to accurately model the noise behavior. Figure 3.11 [2] shows the noise simulation of a typical PLL. Note that some noise sources show low-pass and some show high-pass characteristics, but the total noise will always follow the dominant source.

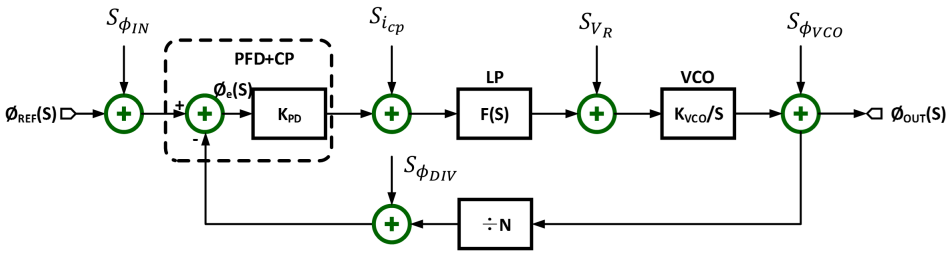

Figure 3.10: PLL Feedback Loop with Possible Noise Sources

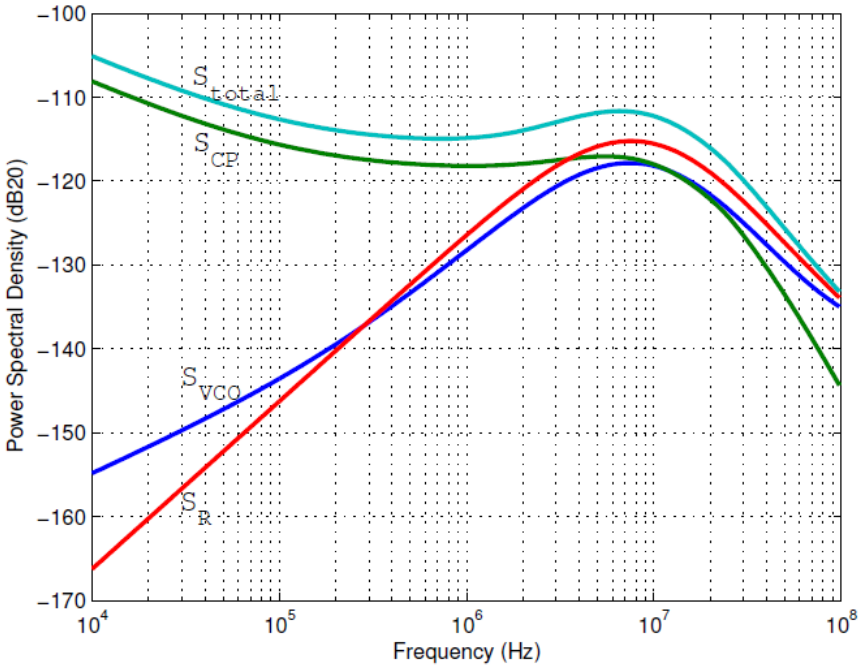

Figure 3.11: Typical PLL Output Referred Noise Simulation

CHAPTER 4

ALL DIGITAL PLL

While analog PLL has been the convention for the past few decades, with the increasing performance and decreasing cost of digital VLSI design, all-digital PLL (ADPLL) has been gaining more popularity. While the digital counterpart still lacks capability in some areas, it proves to be more advantageous in some areas such as lock-in time, stability, scalability, and testability over different processes [7], [8], [9], [10], [11]. This chapter will introduce and discuss the ADPLL's building blocks. Then it will provide analysis of ADPLL.

4.1 Analog and Digital PLL Trade-offs

The principal of analog PLL and ADPLL is the same. The loop consists of similar structures. Figure 4.1 shows the generic PLL loop.

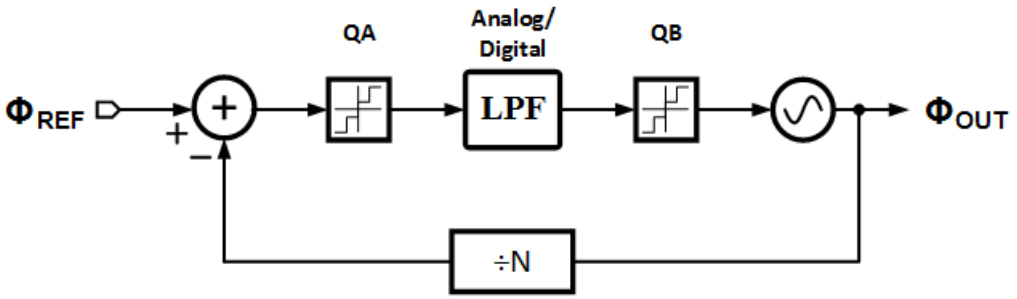

Figure 4.1: Generic PLL Loop

The two quantizers can be replaced with the corresponding component depending the type of PLL. For QA in the analog PLL, a discrete-time detector is used to quantize the phase error and is sampled once per reference cycle; the QA also has infinitely small quantization step. For QA in the ADPLL, the phase error is quantized by finite steps, and thus the phase error is represented discretely. Since the signal coming out of the LPF is digital, the

oscillator has to be digitally controlled. Therefore, the quantization effect is also present in the digital controlled oscillator (DCO), which is often the limitation on the whole PLL system.

Note that while digital components themselves are immune to noise such as PVT variation, quantization noise from discrete sampling of the data is often a bottleneck. However, as the speed of digital circuits scales down with SFT, digitizing of the analog PLL is drawing increasing interest.

4.2 ADPLL Operations

Figure 4.2 outlines the general building blocks of an ADPLL. The main components that have replaced the analog counterparts are time-to-digital converter (TDC), digital loop filter (DLF), digital controlled oscillator (DCO).

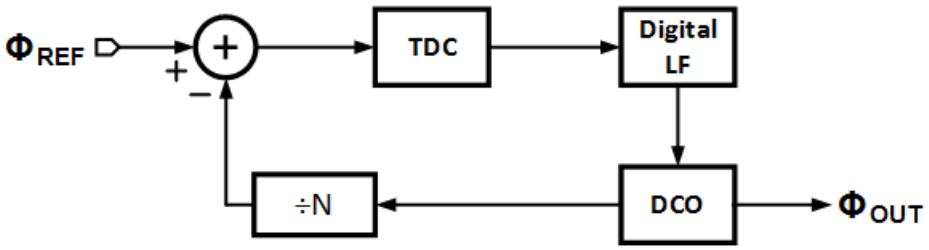

Figure 4.2: Block Diagram of ADPLL

The basic operation is as follows. The TDC translates the phase error between Φ_{REF} and Φ_{DIV} to digital domain that indicates how much faster/slower the Φ_{OUT} should react. The information is fed into the DLF, which can output digital control words based on the information the DLF receives (i.e. faster or slower code). The digital control words are then used to control the DCO's output frequency. Note that since the control is not a continuous voltage, the output frequency's change cannot be continuous. As a result, Φ_{OUT} is susceptible to quantization noise that degrades the jitter performance compared to analog counterpart.

Even though digital components are often nonlinear, the ADPLL can be approximated to be linear as long as the phase error is large enough (better than the resolution of the digital circuitry). Under this assumption, the S-domain analysis of the ADPLL loop is similar to that of the analog PLL. Figure 4.3 shows the ADPLL loop in S-domain.

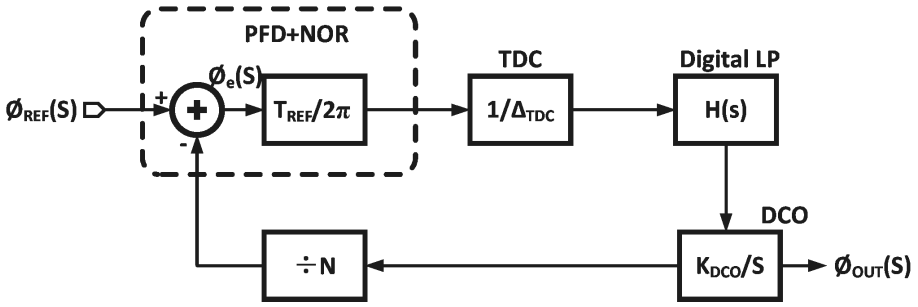

Figure 4.3: S-Domain Approximation of ADPLL

4.2.1 Time to Digital Converter

The TDC translates the difference between the reference frequency and the divided feedback frequency to the digital domain that can indicate which input frequency is faster/slower. Recall that in Chapter 3, the PFD can also output UP and DN signals. However, these signals, though they appear to be digital $HIGH$ and LOW, are analog because they are often short pulses. Thus a simple PFD is not enough to provide the digital signal.

Figure 4.4 shows the typical design of a phase to digital (P2D) [12]. In addition to the PFD, the TDC is needed to translate the UP and DN pulses to the corresponding digital word. Another sampling flip-flop is inserted to indicate the sign. Figure 4.5 shows a typical design of the TDC. The digital

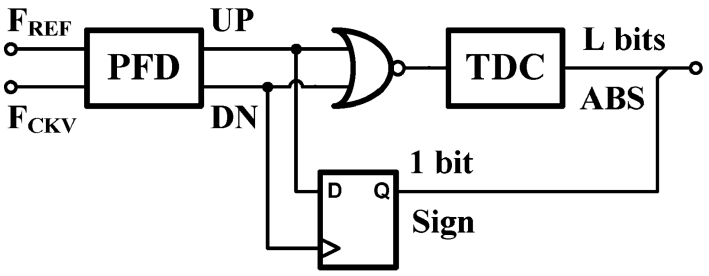

Figure 4.4: Typical P2D Design

word is constructed as the Φ_{REF} is sampled across several times during one clock cycle. The advantage of such a design is that the output digital word has inherent information regarding how much faster/slower the Φ_{DIV} is compared to the sampled reference. This mimics the analog PLL's UP and DN pulse behavior that controls the charge pump. As a result of discretizing the phase difference, the TDC's phase resolution is critical to the P2D and the ADPLL. Equation (4.1) specifies the resolution of the P2D. Δ_{TDC} represents the time

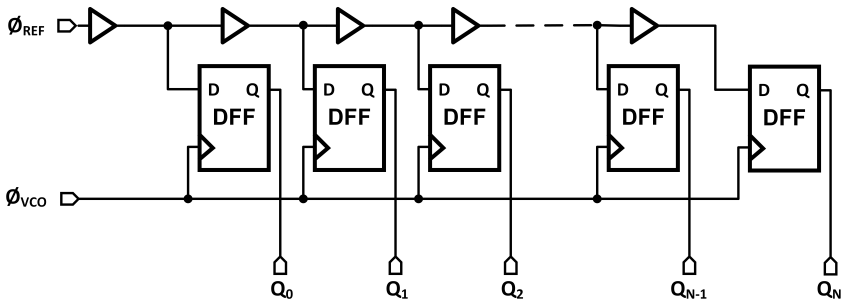

Figure 4.5: Typical TDC Design

resolution of the TDC unit. Thus, Φ_{P2D} defines the minimum phase error that the ADPLL can resolve. Phase error less than Φ_{P2D} will be ignored or represented as the Φ_{P2D}; when this happens, the P2D becomes a non-linear bang-bang phase detector (BBPD) that can only signify the direction. It is critical to keep Φ_{P2D} as minimal as a certain technology allows, which often is limited by the technology node. Therefore, SFT scaling can drive the increasing performance of P2D and the ADPLL.

$$\Delta\Phi_{P2D} = \frac{2\pi\Delta_{TDC}}{T_{REF}} \tag{4.1}$$

4.2.2 Digital Loop Filter

A main disadvantage of analog PLL is the capacitor and the charge pump, which increases the power and size of the entire PLL. In an ADPLL, the digital loop filter (DLF) mimics the effect of the charge pump and the capacitor. In order to translate from analog to digital, recall the S-domain analysis from Chapter 2. Here Z-domain is suitable to digital circuitry, and a bilinear transform is needed to translate between the two domains. A simplification can be made in the complexity of the analog loop filter since the necessity to limit the control voltage ripple will not be needed in the digital domain, thus the capacitor C_2 can be bypassed. As a result, the analysis can be carried out assuming a first order RC low pass filter.

Now the simple RC low pass filter's transfer function can be transformed from S-domain to Z-domain. A relationship exists between the discrete time operator $z = e^{j\omega T_s}$ and the continuous time operator $s = j\omega$. T_s denotes the sampling clock's period. Since the model is assumed to be linear, then the

translation can be approximated via the relation in Equation (4.2b). The assumption is valid if $\omega \ll 1/T_s$; otherwise, this is susceptible to frequency wrapping which degrades the frequency response near the Nyquist rate. This is alleviated because the bandwidth of the PLL is much smaller than that of the reference clock. We can use the mathematical relation to block representations. Since the filter is RC, a proportional path is used to mimic the resistor and the integral path is used to mimic the integration effect of the capacitor. Figure 4.6 shows the block diagram. The α is the proportional path and can be realized using a multiplier; the β is the integral path and can be realized using an accumulator.

$$z = e^{j\omega T_s} \approx 1 + j\omega T_s = 1 + sT_s \tag{4.2a}$$

$$s = \frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \tag{4.2b}$$

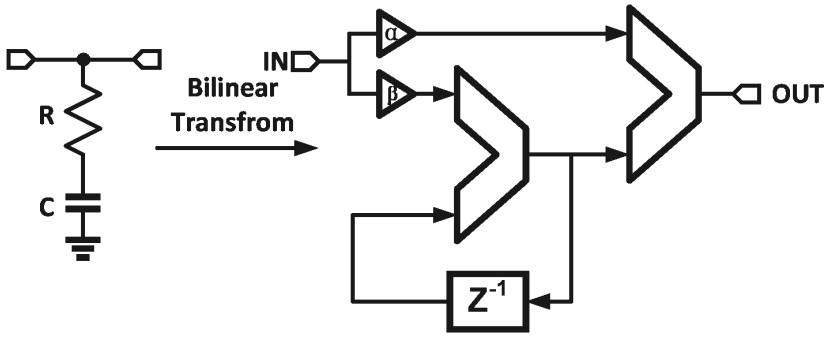

Figure 4.6: Bilinear Transform for Low Pass Filter

At the output, the DLF outputs an N-bit digital control word based on the input digital command from the P2D. The N-bit digital control word should have enough bits for a LSB to resolve.

4.2.3 Digital Controlled Oscillator

The N-bit digital control word is used to manipulate the DCO's output frequency. The concept of DCO is similar that of VCO; however, instead of a voltage controller, a digital code of $HIGH$s and LOWs is used to tune the frequency. As a result, the conventional architecture of VCO can still

be applied. The two most common architectures are LC-tank based and ring-oscillator based.

LC-Tank DCO In the VCO, a varactor is used to tune the frequency of oscillation based on the controlling voltage. There are two ways to approach the control and tuning mechanism in the digital version. The first is to use a digital-to-analog converter (DAC) that converts the digital control word to a corresponding analog voltage. Figure 4.7 shows the block diagram. Note that the varactor can still be used since the control is still an analog voltage. The disadvantage is that the DAC requires very high resolution and thus is very power-consuming. In addition, the inductor and varactor occupy too much area.

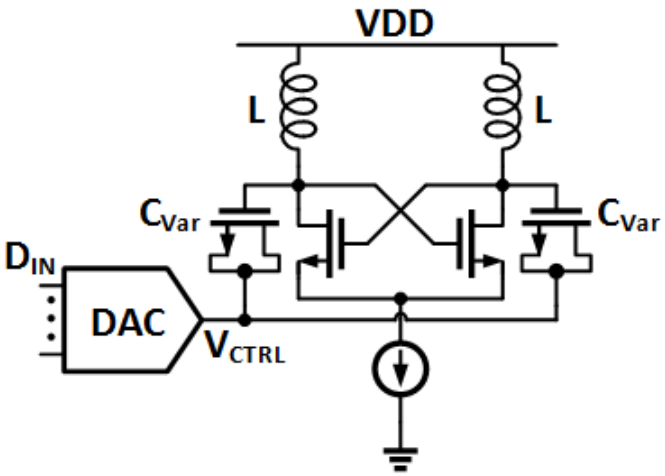

Figure 4.7: LC-Tank DCO Using DAC

The second approach is to use an array of capacitor banks to mimic the effect of the varactor, as shown in Figure 4.8. This alleviates the need for a power-hungry high resolution DAC. However, an inductor is still needed and area is a concern.

Ring Oscillator DCO Similar to the VCO mentioned in Chapter 2, ring oscillators are the better choice for low-power and minimal area designs. Two approaches can be used to tune the frequency of the ring oscillator based DCO and can often be used together to achieve a more robust design.

Figure 4.9 shows the methods of tuning the frequency. Since loading each delay cell can adjust the speed of each stage, an array of capacitor banks

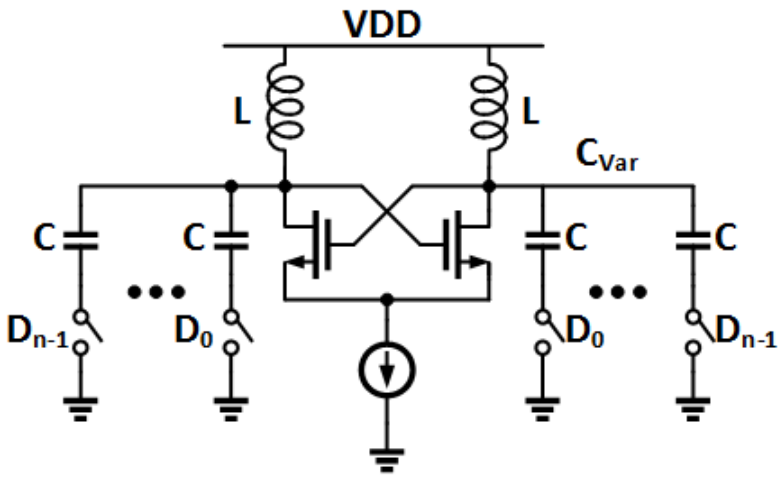

Figure 4.8: LC-Tank DCO Using Capacitor Banks

can be used. However, in order to achieve high resolution, the number of capacitors needed drastically increases, which results in increase in area and power. Since the capacitors are all in forms of MOSFET capacitors (MOS-cap), this tuning method is considered to be fine-tuning, as indicated by "Fine Control" in the figure.

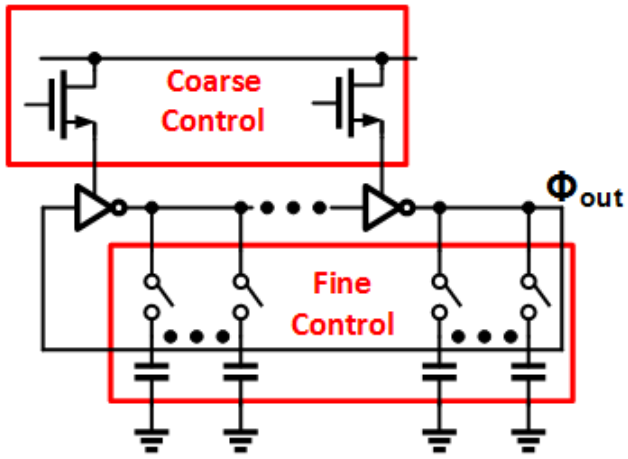

Figure 4.9: Ring Oscillator DCO Tuning

The other method of tuning the frequency is by adjusting the drive strength, or the active current, of the delay cell. "Coarse Control" in Figure 4.9 indicates that amount of current driving the delay cell is adjusted by a variable resistor, which can be easily achieved by changing the gate voltage of a MOS transistor. As a result, the frequency can be tuned over a wider range compared to capacitor banks.

In practice, a combination of the two is used to achieve a wide tuning range while maintaining a low resolution frequency step.

4.2.4 Divider

Since the divider in analog VCO is also a digital block, it can be simply the same block in ADPLL.

4.3 ADPLL Loop Analysis

Like the analog PLL, the ADPLL also has numerous noise sources. In order to quantify the noise contributions, a frequency domain analysis of a close-loop PLL is needed.

4.3.1 ADPLL Noise Analysis

As Figure 4.3 is only the transfer function of each block [9], Figure 4.10 shows the linear model with potential noise sources added. The TDC quantization error $S_{Q_{TDC}}$, DCO quantization error $S_{Q_{dco}}$ (can also be part of the DLF), and loop delay z^{-M} are the main DJ sources. $S_{Q_{TDC}}$ and $S_{Q_{dco}}$ are caused by limited resolution in the circuit. They can be shown by the relationship in Equation (4.3b), where $\Delta\Phi$ and ΔF are TDC and DCO resolution. z^{-M} is usually circuit implementation dependent and M is the number of cycles.

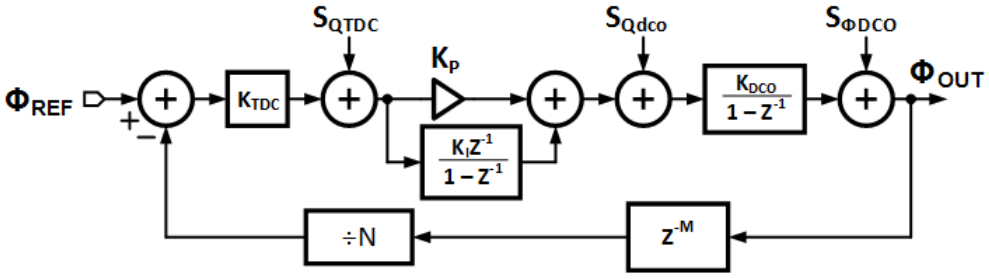

Figure 4.10: ADPLL Linear Model with Noise Contributions

$$S_{Q_{TDC}} = \frac{\Delta\Phi^2}{12F_{REF}} \tag{4.3a}$$

$$S_{Q_{dco}} = \frac{\Delta F^2}{12F_{REF}} \tag{4.3b}$$

There are also intrinsic noise sources that usually come in the form of thermal and flicker noise. The dominant contributor is the DCO, and its noise contribution is $S_{\Phi_{DCO}}$ as shown in Figure 4.10.

Assuming a linear model, the open loop gain (LG) can be derived as:

$$LG(z^{-1}) = K_{TDC} \cdot (K_P + \frac{K_I \cdot z^{-1}}{1 - z^{-1}}) \cdot \frac{K_{DCO}}{1 - z^{-1}} \cdot \frac{z^{-M}}{N} \tag{4.4}$$

Note that this is similar to the analog PLL, and thus the noise analysis can also be done using the transfer function. Let $S_{\Phi_{OUT}}$ be the total phase noise of the ADPLL, then the TDC noise transfer function is:

$$H_{TDC}(z^{-1}) = \sqrt{\frac{S_{\Phi_{OUT}}}{S_{Q_{TDC}}}} = \frac{N}{K_{TDC}} \cdot \frac{LG(z^{-1})}{1 + LG(z^{-1})} \tag{4.5}$$

This transfer function shows low-pass characteristics, meaning that the noise of TDC is suppressed at high frequency. The DCO quantization error exists at the DLF output, so the DLF's noise transfer function is:

$$H_{DLF}(z^{-1}) = \sqrt{\frac{S_{\Phi_{OUT}}}{S_{Q_{dco}}}} = \frac{K_{DCO}}{1 - z^{-1}} \cdot \frac{1}{1 + LG(z^{-1})} \tag{4.6}$$

Note the absence of LG in the numerator, showing a band-pass characteristic. Since it is also proportional to K_{DCO}, reducing the DCO gain can reduce the effect of the DLF noise. Then the DCO noise transfer function is:

$$H_{DCO}(z^{-1}) = \sqrt{\frac{S_{\Phi_{OUT}}}{S_{\Phi_{DCO}}}} = \frac{1}{1 + LG(z^{-1})} \tag{4.7}$$

This shows a high-pass characteristic. As a result, increasing loop bandwidth can filter out the low frequency noise, but still pass through the high frequency components. Combining Equations (4.5), (4.6), and (4.7), we ar-

rive at the total output phase noise:

$$S_{\Phi_{OUT}} = H_{TDC}^2 \cdot S_{Q_{TDC}} + H_{DLF}^2 \cdot S_{Q_f} + H_{DCO}^2 \cdot S_{\Phi_{DCO}} \tag{4.8}$$

A typical plot of the noise transfer function is shown in Figure 4.11, which clearly demonstrates the characteristics described by the equations.

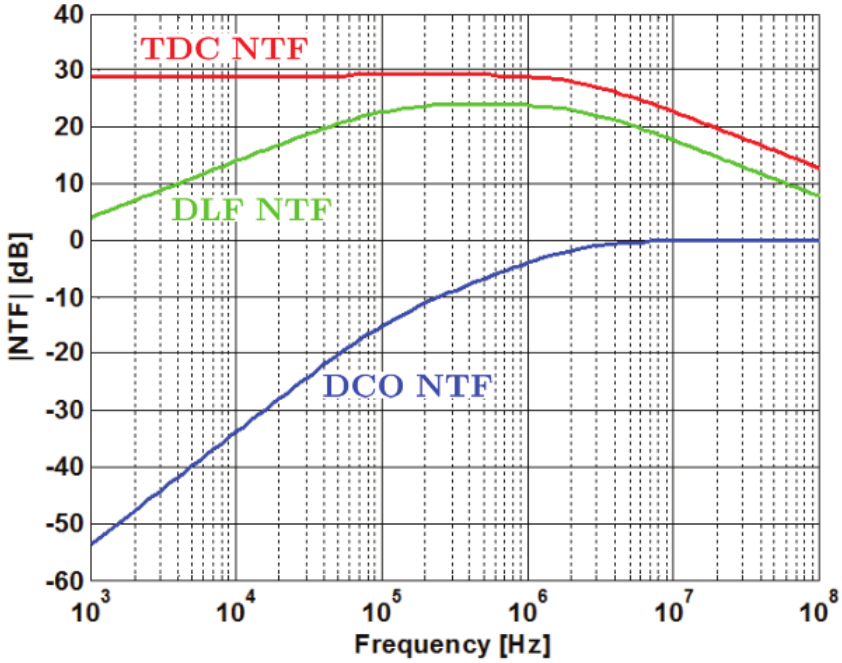

Figure 4.11: Noise Transfer Function of TDC, DLF, and DCO

4.4 Design Parameters

As is the case with the analog VCO, the loop filter parameters need to be designed. In this case, α and β are to be determined based on the previous analysis [13]. Figure 4.3 indicates that the Z domain transfer function of the digital loop filter is

$$H(z) = \alpha + \beta \frac{1}{1 - z^{-1}} = \frac{(\alpha + \beta - \alpha z^{-1})}{1 - z^{-1}} \tag{4.9}$$

On the other hand, the analog filter s domain transform and its bilinear transform are given by:

$$Z(s) = \frac{V(s)}{I(s)} = R + \frac{1}{sC} \tag{4.10}$$

$$H(z) = \frac{(\frac{T_s}{2C}) + R + z^{-1}(\frac{T_s}{2C} - R)}{1 - z^{-1}} \tag{4.11}$$

By comparing Equations (4.9) and (4.11) we can conclude that

$$\alpha = R - \frac{T_s}{2C} \tag{4.12}$$

$$\beta = \frac{T_s}{C} \tag{4.13}$$

4.4.1 α and β Ratio

The phase margin is a very important specification of a PLL circuit since it defines the stability of the PLL. The frequency of the reference clock, bandwidth of the PLL and α-to-β ratio together determine the phase margin of an ADPLL. The derivation is as follows. The zero frequency is given by

$$\omega_z = \frac{1}{RC} \tag{4.14}$$

The phase margin is given by

$$PM = \arctan(\frac{\omega_{ugb}}{\omega_z}) \tag{4.15}$$

Then we can have

$$\omega_z = \frac{\omega_{ugb}}{\tan(PM)} \tag{4.16}$$

From Equations (4.12) and (4.13)

$$\frac{\alpha}{\beta} = \frac{RC}{T_s} - \frac{1}{2} \tag{4.17a}$$

$$= \frac{1}{T_s \omega_z} - \frac{1}{2} \tag{4.17b}$$

$$= \frac{1}{T_s} \frac{\tan(PM)}{\omega_{ugb}} - \frac{1}{2} \tag{4.17c}$$

$$= \frac{F_{REF}}{F_{ugb}} \frac{\tan(PM)}{2\pi} - \frac{1}{2} \tag{4.17d}$$

CHAPTER 5

JITTER ANALYSIS

The ADPLL has been shown to benefit from PVT insensitivity, low-supply, and process scalability compared to the conventional analog PLL. As SFT scales and the speed of digital technology increases with novel designs, ADPLL has proven to achieve performance that is equal to or better than that of analog PLL. However, there are still bottlenecks in ADPLL that limits the ADPLL and possibly the overall performance of the system [9]. These limitations mainly take the form of jitter, which is the uncertainty and variation in precise timing events. This chapter will discuss the different types of jitter and provide analysis of jitter inside the ADPLL.

5.1 Jitter Definition

Jitter is the uncertainty and variation in timing events. It is unavoidable in practical systems, especially electrical systems that use voltage, current, or phase to represent timing. Thus, timing jitter $\sigma_{\Delta T}$ is an important metric for PLL output jitter. It can also be represented in phase domain, which is related by the equation

$$\sigma_{\Delta T} = \sigma_{\Delta \Phi} \cdot \frac{T}{2\pi} = \sigma_{\Delta \Phi} \cdot \frac{1}{\omega_0} \tag{5.1}$$

Clock period $T = \frac{2\pi}{\omega_0}$. When ΔT is large, it is long-term jitter; when it is small, it is short-term jitter. Figure 5.1 [9] illustrates the idea of timing jitter, which is defined as the standard deviation of the time difference between the first 1^{st} cycle and the m_{th} cycle of the clock.

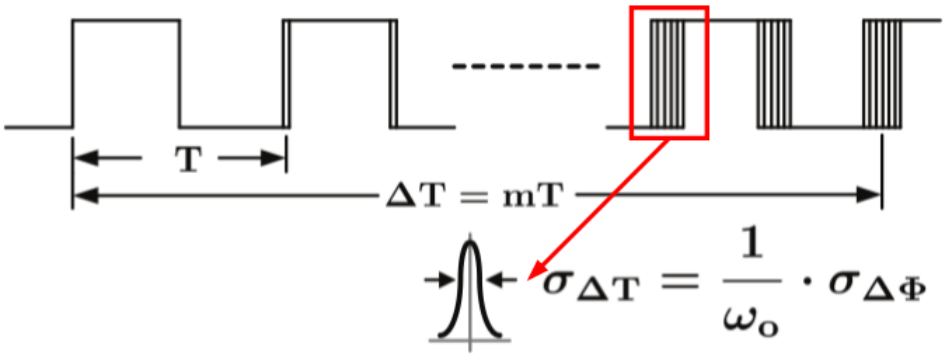

Figure 5.1: Timing Jitter

5.2 ADPLL Clock Jitter

PLL typically has two types of jitter.

- Deterministic jitter (DJ): jitter with known probability distribution and can be predicted
- Random jitter (RJ): non-deterministic and unpredictable, usually follows the form of a Gaussian distribution

Deterministic Jitter In ADPLL, DJ typically results from quantization error and loop delay. Quantization is usually the dominant DJ contributor, and two main sources are TDC quantization error and DCO quantization error.

Figure 5.2 again shows the typical, basic TDC. It consists of delay-elements such as buffer, registers (D flip-flops), and possibly a thermometer-to-binary code converter (not shown).

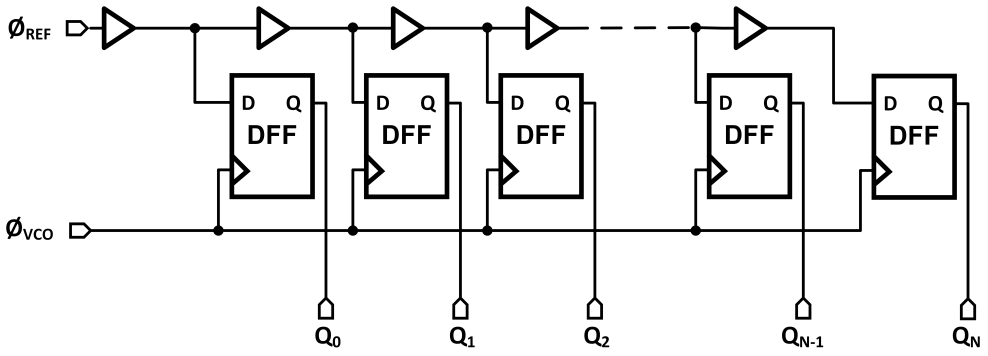

Figure 5.2: Delay Line Flash TDC

The input clock F_{REF} samples the feedback signal over the delay line and outputs the thermometer code. Assuming each delay element has delay time T_D, then the resolution of N-stage TDC can be derived as

$$T_D = T_{REF}/2^N \tag{5.2}$$

The resolution of TDC directly relates to the phase quantization error and translates to DJ at the output of the ADPLL. This is visualized in Figure 5.3. The input phase error $\Delta\Phi$ is dependent on and limited by the resolution of the delay cell. Since this is an intrinsic property, the manifested DJ is large. As a result, the quantization error in TDC is a main contributor; usually the error can be minimized at the expense of addition power and area, which constitutes a tradeoff.

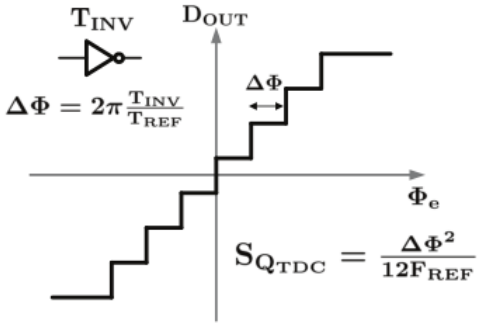

Figure 5.3: TDC Transfer Characteristics

The other source of DJ stems from DCO. Quantization error of the DCO manifests itself as frequency quantization error. Since some form of a DAC is involved designing a DCO, the finite resolution of the DAC is often the limiting factor. The design of a high-resolution DAC is very challenging and power hungry. In addition, there exists a tradeoff between the DCO's resolution and tuning range. This is critical because a wide tuning range is needed in order to operate the DCO under PVT and external variations. As a result, reducing the DJ from DCO is also very challenging.

The last type of DJ results from loop delay. This is due to the discrete-time nature of ADPLL. The variable delays of each component can add latency to the delay of the loop. Different designs can reduce the effect of the loop delay, but the effect cannot be avoided since delays are inherent in digital components.

Random Jitter The other form of jitter is random jitter (RJ). In contrast to DJ, RJ results from external and internal noise sources that are unpredictable and exhibit a Gaussian distribution. These noises sources are similar to those of analog PLL. Common dominant external contributors include supply noise, substrate noise, and coupling noise; common dominant internal contributors include thermal and flicker noise. The external sources are typically beyond the designer's control, whereas the internal sources can be examined and understood better. The main source of the internal error comes from the phase noise of the oscillator. Figure 5.4 illustrates the typical plot of the phase noise versus frequency. At low frequency (<1kHz), the phase noise is dominated by flicker noise. At higher frequencies (>5kHz), the phase noise is dominated by the oscillator's thermal noise.

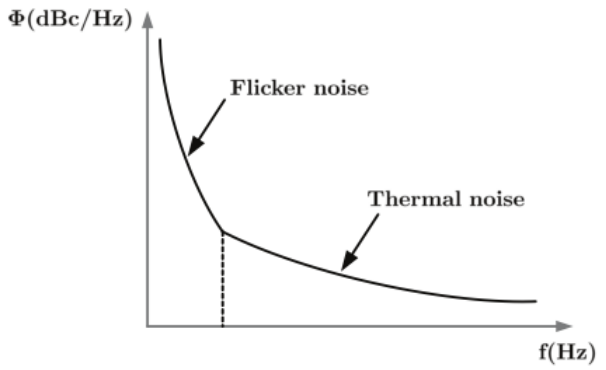

Figure 5.4: Open-Loop Oscillator Phase Noise

CHAPTER 6

PLL DESIGN

This chapter presents an ADPLL that is modeled and designed using TSMC-65nm RF PDK.

6.1 Behavior Modeling

Since the PLL is a mixed-signal and complex system, it is difficult to simply design each block individually and then put them together. The better approach is a top-down methodology. Figure 6.1 [14] illustrates the general steps.

Once the system's requirements are determined, the entire system is modeled to ensure the specification is achievable. After model validation, which is only an approximation of the system, each block component can be designed carefully at the transistor level. Since each block has a model place holder, transistor design can be used to replace the models block by block. This reduces simulation time drastically as well as allows the designer to easily verify the transistor design since each transistor block can be tested with behavior models of other blocks. Lastly, there should be an iterative process of tuning the transistor level design and behavior model to achieve the best design possible.

In mixed-signal design such as PLL, behavior modeling is typically done using Cadence's Verilog-AMS (Analog/Mix-Signal) or MatLab's SimuLink. For the design here, Verilog-AMS is used for behavior modeling since it is very compatible with the Verilog, which is the industry standard for digital circuitry. The transistor level design is done in Cadence's Virtuoso and simulated in Spectre.

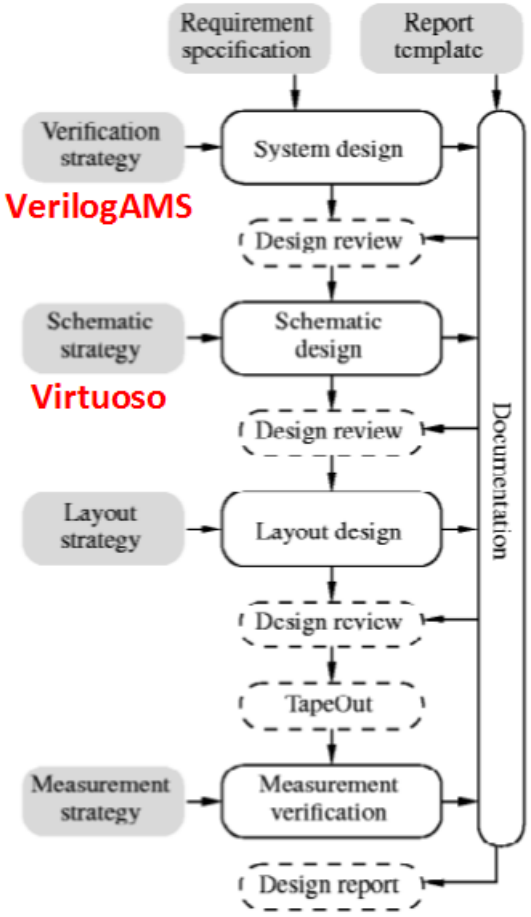

Figure 6.1: Top Down Design Flow

6.2 Analog PLL Design

A PLL is done in transistor design; however, transistor level simulations are often time-consuming. As a result, block level modeling is a critical step in analog and mix-signal design. Behavior modeling allows the designer to check and verify the functionality of each block and ensures the capability of the loop.

In [2], an analog PLL is designed in TSMC-180nm RF technology from a 200MHz reference input. It operates at nominal frequency of 1.6GHz. The procedure follows the top-down approach. The behavior modeling is done using Verilog-AMS and VerilogA. The transistor design is done using Cadence Virtuoso and simulated using Cadence Spectre. Its specifications will be used as the baseline for the ADPLL design in this thesis.

6.3 ADPLL Design

In this thesis, an ADPLL is designed also using the top-down approach. The primary goal is to utilize the behavior model to aid the transistor design. The design adopts TSMC-65nm RF technology. The behavior modeling uses Verilog-AMS, and the transistor design uses Cadence Virtuoso and uses Spectre for simulation. The final simulation has the digital loop filter as behavior modeling; the TDC, DCO, and divider are designed using the 65nm PDK's standard cells. The reason for using standard cells is that they can be synthesized, which is an advantage of ADPLL as mentioned previously. The reference is 200MHz, and the center frequency is at 1.6GHz. The supply is 1.2V. The DLF supplies a 12-bit control code; however, the 2 LSBs are dropped for dither jitter considerations. Thus, the DCO takes in a total of 10-bit control code, which is then divided into 5-bit coarse tuning and 5-bit fine tuning.

6.3.1 Phase to Digital Converter

Since the ADPLL does not have much requirement on the lock-in time, having a high resolution TDC is not necessary. In addition, only frequency locking is required, so phase offset is acceptable. As a result, the P2D thus degrades to a non-linear bang-bang PFD (BBPFD) that only indicates the UP and DN signal based the speed of Φ_{DIV} relative to Φ_{REF}. The design is shown in Figure 6.2. Note that each block is realized by standard cell as indicated by "S". This allows portability from one technology to another. Since the PFD's output is only short pulses that are modulated by the phase difference between Φ_{REF} and Φ_{DIV}, we need to convert these phase-modulated signals to digital domain. Thus, the first part of the latch is used to decide UP or DN, and the second part is used to hold the value of the UP and DN for the remainder of the clock cycle.

6.3.2 Digital Loop Filter

This block is done only using Verilog-AMS because it is quite challenging to design the loop parameters well to achieve a robust design. As mentioned in the previous chapter, the digital loop filter consists of a proportional path

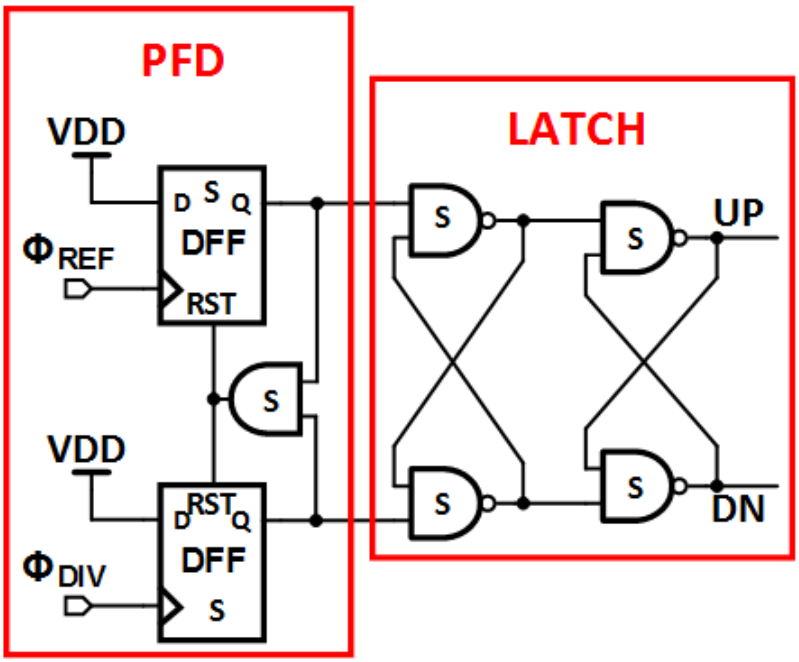

Figure 6.2: Transistor-Level P2D Design

(K_p or α) and an integral path (K_I or β). Determining the values impacts the loop dynamic. Also recall that these parameters can be derived from an analog equivalent, and the detailed derivation was in the previous chapter and is reiterated below.

$$\frac{\alpha}{\beta} = \frac{F_{REF}}{F_{ugb}} \frac{\tan(PM)}{2\pi} - \frac{1}{2} \tag{6.1}$$

α and *beta* are determined based on the tradeoff of noise specification and phase margin. After consideration, α is chosen to be 360 and β is chosen to be 10, and yielding the ratio of 36. This allows the loop to be stable and lock within reasonable time without sacrificing too much noise. The number of bits chosen is 12-bits from power, area, and noise considerations. This allows the system to achieve a reasonable tuning range.

6.3.3 Digital Controlled Oscillator

This block is first simulated using Verilog-AMS and realized in transistor design. The targeted frequency is 1.6GHz with tuning range from 1.3GHz to 1.9GHz. To achieve better noise specification, the design is chosen to be a 4-stage differential ring oscillator. Figure 6.3 presents the topology of the

DCO design.

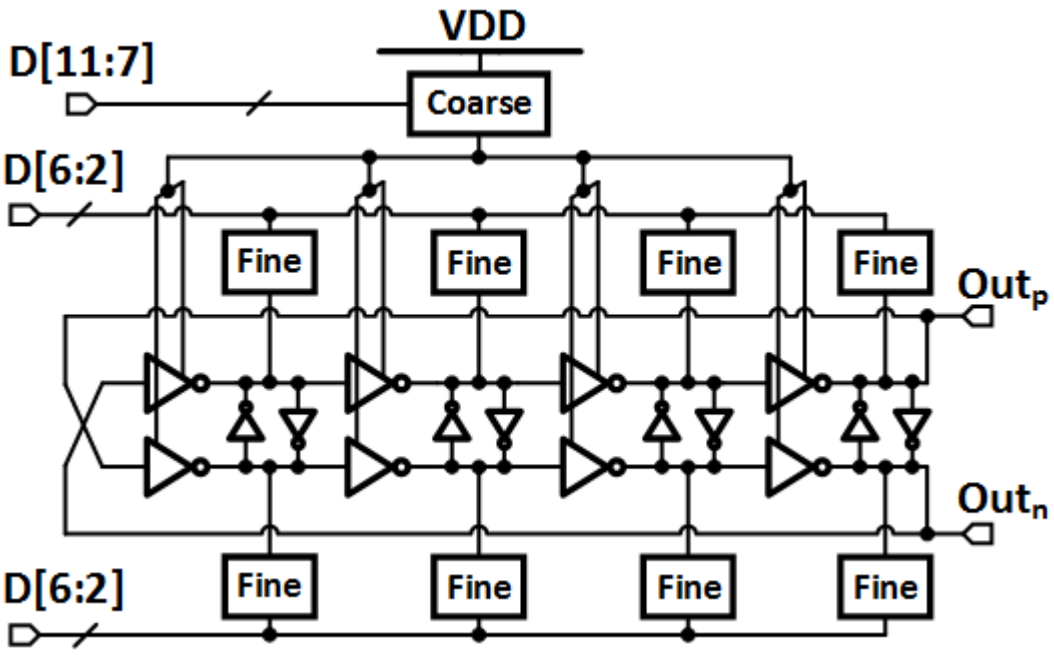

Figure 6.3: DCO Design

The designs for the coarse tuning and fine tuning are illustrated in Figure 6.4 and Figure 6.5, respectively. The coarse tuning is achieved by a shunt transistors that control the current flow through the delay cells. These binary-weighted transistors are sized to provide incremental change according to the thermometer code provided. The 5 most significant bits are used to control these. In essence, they act as resistors that control the current. PMOS was used because the transistors are closer to the supply; however, NMOS transistors are also valid. Note that an always-on transistor is needed to provide a path when all the current controlling transistors are off.

The fine tuning cells are realized using MOS-switches and MOS capacitors. The MOS capacitors are also binary weighted to enforce thermometer-code. The switches are simple NMOS transistors that will turn on and off according to the control codes. Bits 6 to 2 of the control code are used to control the switches. Note there is also a MOS capacitor that is always on, providing a static, nominal capacitance.

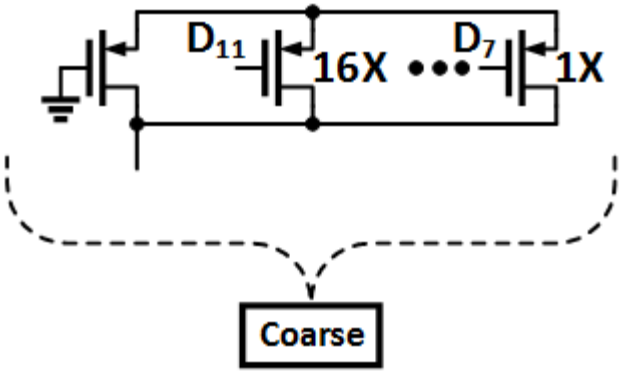

Figure 6.4: Coarse Tuning

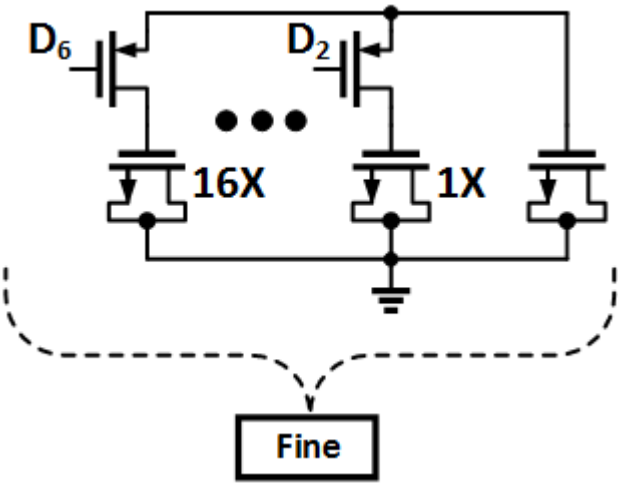

Figure 6.5: Fine Tuning

6.3.4 Divider

As shown in Figure 6.6, the dividers are realized using D flip-flop standard cells. They provide a dividing ratio of 8 so the output clock of 1.6GHz can be divided down to 200MHz, which complies with the reference frequency. This is similar to analog PLL's divider, but this uses standard cells.

6.4 Simulation and Results

Simulations were done in Verilog-AMS simulator and Spectre simulator.

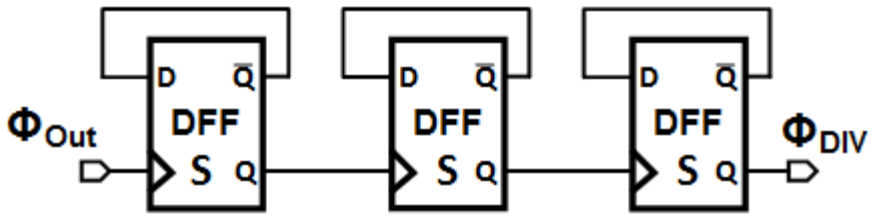

Figure 6.6: Standard Cell Divider

6.4.1 Verilog-AMS Simulations

The ADPLL Verilog-AMS model is simulated. Figure 6.7 shows the different frequencies. The first presents the output frequency of 1.6GHz. The second presents the divided frequency of 200MHz. The third row presents the reference frequency of 200MHz. Note that the divided frequency is not completely in phase with the reference frequency. This is because of the resolution of digital circuits. However, the ADPLL loop locks as long as frequency is locked.

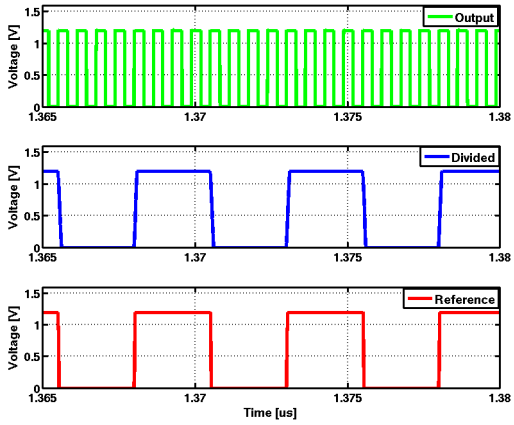

Figure 6.7: Locking in Modeling Simulation

Figure 6.8 shows the estimated frequency of the feedback divided clock signal. The signal goes above and below 200MHz; this is expected because the resolution cannot allow for exact lock of 200MHz, so the frequency will always go back and forth around the expected signal when it locks.

Figure 6.9 illustrates the control code of the ADPLL. Its behavior matches up with the behavior displayed above. The control code will oscillate back and forth between the nominal code as the loop locks.

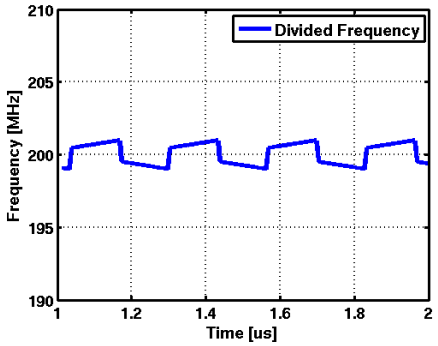

Figure 6.8: Frequency of Divided Signal in Modeling Simulation

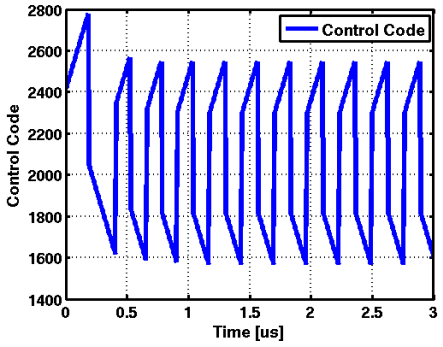

Figure 6.9: Control Code in Modeling Simulation

6.4.2 Transistors Design Simulation

After the simulation has confirmed the parameters of the loop, each component is replaced with the transistor design except the DLF. Then the ADPLL is simulated with the transistor design. The output's operating range is from 1.2GHz to 2.0GHz centering around 1.6GHz. Each LSB achieves resolution of 10MHz for fine tuning. The DLF parameters α and β were lowered to 25 and 5 to ensure the lock-in mechanism. However, this greatly reduces the lock-in time, which is shown below. Figure 6.10 shows the locking behavior with the transistor design. Note that because we are using transistors as resistor coarse control, the DCO's output voltage swing is limited and less than V_{dd} of 1.2V. However, even though full voltage swing cannot be achieved, the voltage is high enough for operation of the DCO. Figure 6.11 shows the locking behavior at a more zoomed out view. Since it is transistor level simulation, the lock-in time takes considerably longer when compared to the behavior modeling.

To examine the lock-in process, Figure 6.12 shows the frequency range of

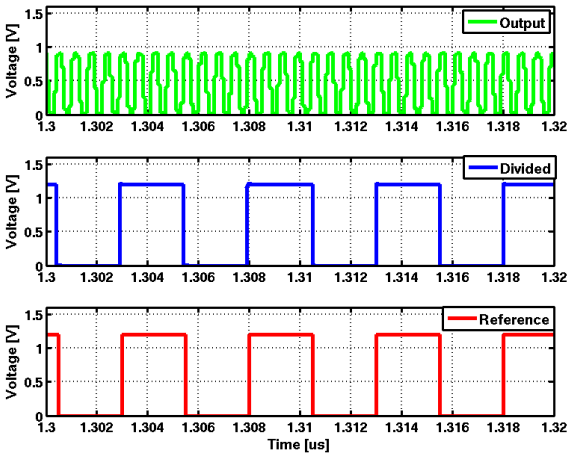

Figure 6.10: Locking in Transistor Simulation

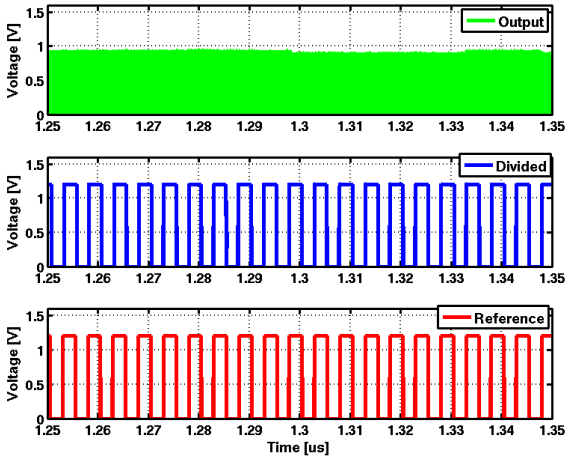

Figure 6.11: Zoomed-out View Locking in Transistor Simulation

the ADPLL. The center frequency of the divided signal is at about 200MHz. Note that the frequency is initially higher than 200MHz and slowly locks onto the center frequency. The lock-in time is about $1.3\mu s$.

The lock-in process can also be examined from the control code. Figure 6.13 shows control code versus time. The code word agrees with the frequency plot shown above, and the code words start to settle at around $1.3\mu s$.

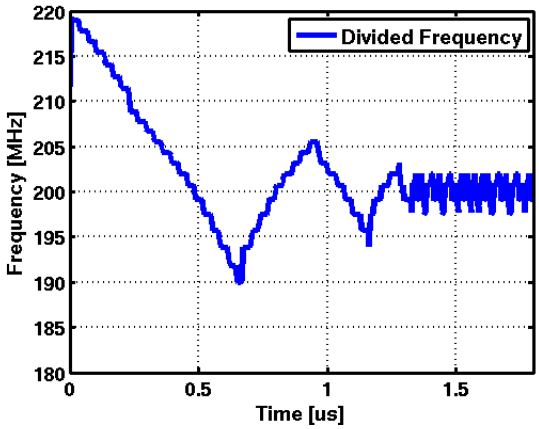

Figure 6.12: Frequency of Divided Signal in Transistor Simulation

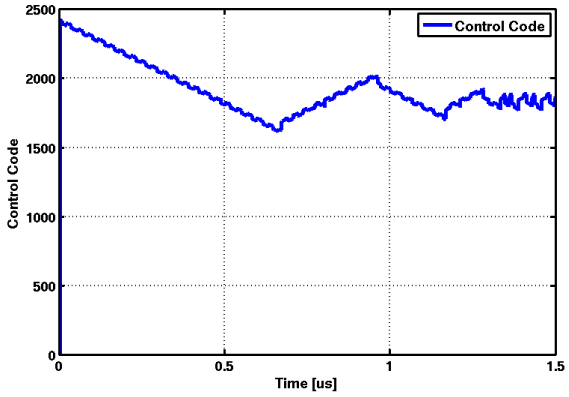

Figure 6.13: Control Code in Transistor Simulation

6.4.3 Potential Difficulties

Note that the above-and-below behavior is much more prominent in the transistor implementation. This is because the resolution of the digital control code is higher than that of the model case. This also causes the ADPLL to lose lock when the code word transition is too large. In addition, since the low resolution causes the code word to jump aggressively each time, the output frequency will go up and down, similar to that of the ripple of the control voltage in analog PLL. As a result, the jitter of the output clock will be very large (around 15% to 20% of output period) and will dominate the system.

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this work, introduction and basics of PLL are first introduced in Chapter 1 and Chapter 2. Chapter 3 examines the ADPLL from block level and zooms for analysis. Chapter 4 discusses the noise sources and jitter theory of ADPLL. In Chapter 5, ADPLL is designed at transistor level operating nominally at 1.6GHz, and operates from 1.2GHz to 2.0GHz. Each fine control bit achieves 10MHz resolution.

As future work, the DLF can be also designed at transistor level for a complete transistor design. In addition, jitter reduction techniques can be examined. The quantization noise from the PFD and the DCO can be greatly reduced in order to achieve better performance. For example, a higher resolution TDC will allow for more accurate proportional path, which greatly reduces jitter residing in a BBPFD (only UP and DN). In addition, dual-loop techniques can be added to decrease the lock-in time and the tuning range of the ADPLL. This typically requires a frequency-locked loop (FLL) in addition the phase-locked loop. Another trend in recent years is the multiplying delay-locked loop (MDLL), which replaces the output clock edges with the injection of reference edge. This allows DCO/VCO jitter to reset and do not accumulate. Lastly, power reduction techniques also represent an intriguing field. Even though ADPLL saves power compared to an analog PLL, different and novel architectures and techniques allow even more power reduction in the ADPLL.

REFERENCES

[1] K. Iniewski, *CMOS Nanoelectronics: Analog and RF VLSI Circuits*. McGraw-Hill, 2011.

[2] D. Wei, "Clock synthesizer design with analog and digital phase locked loop," M.S. thesis, Univ. of Illinois at Urbana-Champaign, 2014.

[3] D. Pfaff, S. Kanesapillai, V. Yavorskyy, C. Carvalho, R. Yousefi, M. Khan, T. Monson, M. Ayoub, and C. Reitlingshoefer, "A 1.8W 115Gb/s serial link for fully buffered DIMM with 2.1ns pass-through latency in 90nm CMOS," in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, Feb 2008, pp. 462–628.

[4] J.-S. Lee, M.-S. Keel, S.-I. Lim, and S. Kim, "Charge pump with perfect current matching characteristics in phase-locked loops," *Electronic Letters*, vol. 36, no. 23, pp. 1907–1908, Nov 2000.

[5] P. K. Hanumolu, M. Brownlee, K. Mayaram, and U.-K. Moon, "Analysis of charge-pump phase locked loops," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 9, pp. 1665–1674, Sept 2004.

[6] M. Mansuri, "Low-power low-jitter on-chip clock generation," Ph.D. dissertation, UCLA, 2003.

[7] S. R. Al-Araji, Z. M. Hussain, and M. A. Al-Qutayri, *Digital Phase Lock Loop, Architectures and Applications*. Springer, 2006.

[8] J. Zhao, "A low power CMOS design of an all digital phase locked loop," Ph.D. dissertation, Northeastern University, 2011.

[9] W. Yin, "Design techniques for high-performance digital PLLs and CDRs," Ph.D. dissertation, Oregon State University, 2010.

[10] S. Bashiri, S. Aouini, N. Ben-Hamida, and C. Plett, "Analysis and modeling of the phase detector hysteresis in bang-bang PLLs," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, vol. 62, no. 2, pp. 347–355, Dec 2015.

[11] M. E. Elrabaa, "A portable high-frequency digitally controlled oscillator (DCO)," *The VLSI Journal of Integration*, vol. 47, no. 3, pp. 339–346, June 2013.

[12] S. Henzler, *Time-to-Digital Converters*, ser. Springer Series in Advanced Microelectronics. Springer, 2010, vol. 29.

[13] V. Kratyuk, P. K. Hanumolu, U.-K. Moon, and K. Mayaram, "A design procedure for all-digital phase-locked loop based on a charge-pump phase-locked loop analogy," *IEEE Transactions on Circutis and Systems - II: Express Briefs*, vol. 54, no. 3, pp. 247–251, March 2007.

[14] M. Anderson, J. Wernehag, A. Axholt, and H. Sjoland, "Teaching top down design of analog/mixed signal ics through design projects," in *Frontiers In Education Conference*, Oct 2007, pp. T1C–1 – T1C–4.