Techniques for High-Performance Digital Frequency Synthesis and Phase Control

by

Chun-Ming Hsu

Bachelor of Science in Engineering
National Taiwan University, June 1997

Master of Science
National Taiwan University, June 1999

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2008

Author ....
Department of Electrical Engineering and Computer Science
August 27, 2008

Certified by.....................................
Michael H. Perrott, Ph.D.
Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by.....................................
Terry P. Orlando, Ph.D.
Chairman, Committee on Graduate Students
Department of Electrical Engineering and Computer Science

Techniques for High-Performance Digital Frequency Synthesis and Phase Control

by

Chun-Ming Hsu

Submitted to the Department of Electrical Engineering and Computer Science
on August 27, 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This thesis presents a 3.6-GHz, 500-kHz bandwidth digital $\Delta\Sigma$ frequency synthesizer architecture that leverages a recently invented noise-shaping time-to-digital converter (TDC) and an all-digital quantization noise cancellation technique to achieve excellent in-band and out-of-band phase noise, respectively. In addition, a passive digital-to-analog converter (DAC) structure is proposed as an efficient interface between the digital loop filter and a conventional hybrid voltage-controlled oscillator (VCO) to create a digitally-controlled oscillator (DCO). An asynchronous divider structure is presented which lowers the required TDC range and avoids the divide-value-dependent delay variation. The prototype is implemented in a 0.13-μm CMOS process and its active area occupies 0.95 mm^2. Operating under 1.5 V, the core parts, excluding the VCO output buffer, dissipate 26 mA. Measured phase noise at 3.67 GHz achieves -108 dBc/Hz and -150 dBc/Hz at 400 kHz and 20 MHz, respectively. Integrated phase noise at this carrier frequency yields 204 fs of jitter (measured from 1 kHz to 40 MHz).

In addition, a 3.2-Gb/s delay-locked loop (DLL) in a 0.18-μm CMOS for chip-to-chip communications is presented. By leveraging the fractional-N synthesizer technique, this architecture provides a digitally-controlled delay adjustment with a fine resolution and infinite range. The provided delay resolution is less sensitive to the process, voltage, and temperature variations than conventional techniques. A new $\Delta\Sigma$ modulator enables a compact and low-power implementation of this architecture. A simple bang-bang detector is used for phase detection. The prototype operates at a 1.8-V supply voltage with a current consumption of 55 mA. The phase resolution and differential rms clock jitter are 1.4 degrees and 3.6 ps, respectively.

Thesis Supervisor: Michael H. Perrott, Ph.D.
Title: Associate Professor of Electrical Engineering

Acknowledgments

I would like to acknowledge many people for helping me during my doctoral work. First, I would especially like to thank my advisor, Professor Michael Perrott, for his continuous inspiration and encouragement on this work. It has been an exciting and valuable experience to work with him. I appreciate for what he has pushed me during the past few years, such that I could accomplish my dissertation and degree within a reasonable period of time.

I thank my committee, Professor Anantha Chandrakasan and Professor Joel Dawson, for their generous time and commitment. I also thank Professor Vladmir Stojanovic who supervised my RQE together with Professor Chandrakasan.

I need to thank my academic advisor, Professor Harry Lee, for his professional advise on my courses and schedule from my first day at MIT.

I am very grateful for having wonderful groupmates; I owe each of them a lot. Min Park and Kerwin Johnson did the wire bonding for me again and again. Matt Straayer generously allowed me to use his GRO design. He also contributed many useful suggestions on my work. Charlotte Lau provided me a lot of information about MIT and the group even before I came to Boston. She also shared her VCO design with me as well as drew the layout for part of my DLL chip. Belal Helal designed the shift-register for the group; without this cell, I might have missed the deadline of my synthesizer tape-out. Matt Park has been a good friend who has always heard my complaint. In addition, he and Min Park also introduced a lot of delicious Korean food to me. Scott Meneinger and Ethan Crain were also a great help to me in my first year.

I extend my thanks to many friends in MTL and EECS, especially Joyce Kwong, Vivienne Sze, and Yogesh Ramadass, with whom I went through the tough classes together in my first year.

I'd like to thank C2S2, the Focus Center for Circuit and System Solutions, one of five centers funded under FCRP, and SRC program, as well as MIT Center for Integrated Circuit and System for funding my research. In addition, Mr. Peter

Holloway helped me to access the process in National Semiconductor for my DLL chip.

I also need to thank for our group administrator, Valerie DiNardo, and all of the MTL staff.

The experts in MIT writing center, who helped me edit this dissertation, is also a wonderful resource to me.

I should also thank my former advisor, Professor Shen-Iuan Liu, in National Taiwan University for his continuous help. Professor Chorng-Kuang Wang in NTU also shared much of his life experience with me.

I thank my friends, no matter in Taiwan or in the States, who have been very supportive. I owe a special note of gratitude to Wei-Hung Chen for his valuable suggestions and help during my graduate-school application. In addition, I would like to especially thank Chi-Heng Wang, Ya-Ting Chou, Da-Yuan Tung, Hsin-Ning Keng, Hsin-Pei Shih, Yu-Chen Yeh, Chen-Hsiang Yu, and Julie Leh, who have brought me a lot of joy and pleasure in Boston. I will never forget those precious memories which have been built up with you.

Last but not least, I would like to share this pride with my family members. Dad and Mom - Thank you for what you have educated me. Also, your support all the time made it easier for me to pursue my dream. My wife Wan-Chen, to whom this dissertation is dedicated, - Thank you for always accompanying with me and cheering me up whenever I was down. Without you, life at MIT would be horrible.

There are still many others to thank as well. Thus, I would like to share this dissertation with those who have ever helped, influenced, and inspired me in my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As the capability of the digital calculation keeps on improving in modern sub-micron CMOS processes, there is increasing interest in developing digital approaches to assist or even replace the analog functions that encounter design difficulties due to the degrading analog device characteristics such as decreasing $g_m r_o$ and supply voltage, and increasing leakage current and variation. People have demonstrated some successful results in various digitally-assisted analog subsystems including data converters, RF transceivers, and phase-locked loops (PLL) [6][7][8][9].

Among various digitally-assisted analog techniques, the digital PLL has especially become a very hot research topic in the past few years after it was demonstrated to be able to meet the stringent wireless communication specifications [9][10][11][12][13]. However, existing low-noise digital fractional-N synthesizer techniques can only achieve about a 50-kHz loop bandwidth that may be not wide enough for some new applications [9][12]. Therefore, the goal of this research is to achieve a low-noise fractional-N PLL with a wider bandwidth and a mostly digital implementation [14]. Furthermore, we apply the fractional-N PLL technique to a new application: phase control of a high-speed clock [15].

The remainder of this chapter presents an overview of this thesis. We begin by narrowing the focuses of the thesis. Next, the proposed techniques are briefly discussed with implementation highlights. After that, the contributions of this thesis are summarized. Finally, an outline of the thesis is presented.

1.1 Area of focus

Digital PLLs have recently emerged as an attractive alternative to the more traditional analog PLL, with recent results demonstrating that digital frequency synthesizers with GSM-level noise performance can be achieved [9][12]. One of the key advantages of digital PLLs over their analog counterparts is that they remove the need for large capacitors within the loop filter by utilizing digital circuits to achieve the desired filtering function. The resulting area savings are critical for achieving a low-cost solution, and the overall PLL implementation is more readily scaled down in size as new fabrication processes are utilized. Also, by avoiding analog-intensive components such as charge pumps, a much more attractive "mostly" digital design flow is achieved.

While the benefits of a digital PLL approach are obvious to many, there remain basic questions regarding their attainable performance. In particular, can such structures achieve low jitter comparable to analog approaches? Can a high PLL bandwidth be achieved to more easily support wide-bandwidth modulation and fast settling? Can traditional voltage-controlled oscillators (VCO) be efficiently leveraged in such systems? We attempt to address the above questions in the first part of this thesis.

In the second part of this thesis, we target a digital approach to phase control of a high-frequency clock, which is essential to chip-to-chip communications. By leveraging the fractional-N PLL technique, we create a high-resolution infinite-range delay control scheme in a digital manner. In contrast, previous works rely on analog phase interpolators to achieve fine phase resolution, which is again undesired in modern CMOS processes [16][17][18][19].

1.2 Proposed Digital Frequency Synthesis Technique

A digital fractional-N frequency synthesizer is presented. This synthesizer leverages a noise-shaping time-to-digital converter (TDC) [2][3][20][21] and a simple quantization noise cancellation technique to achieve low phase noise with a wide PLL bandwidth of 500 kHz. In contrast to previous cancellation techniques [22][23][24][25][26], the proposed structure requires no analog components and is straightforward to implement with standard cell digital logic. With the cancellation technique enabled, the synthesizer achieves phase noise of -132 dBc/Hz at 3 MHz offset, and an integrated phase noise from 1 kHz to 40 MHz of 204 fs rms at 3.67 GHz.

1.2.1 Overview

Figure 1-1 shows a block diagram of the proposed synthesizer. The high-resolution digital phase detection is performed with a multipath gated ring oscillator (GRO) time-to-digital converter (TDC) presented in [3]. In contrast to previous digital PLL implementations [9], the digitally-controlled oscillator (DCO) is implemented as a conventional LC voltage-controlled oscillator (VCO) with coarse and fine varactors that are controlled by two 10-bit, 50-MHz digital-to-analog converter (DAC) structures. Both varactors are realized as accumulated-mode devices, and an additional four-bit MIM capacitor bank is included in the VCO to improve its tuning range. A unique aspect of the DAC implementations is that they are passive in nature and minimal in their analog complexity. Another interesting component of the architecture is an asynchronous frequency divider which avoids the divide-value delay variation at its output [22][24][26][27].

1.2.2 Quantization Noise Cancellation

Figure 1-2 displays the quantization noise cancellation circuit, which is completely digital in its implementation. The goal of this circuit is to remove the noise introduced by the dithering action of the divider, which is manifested in the GRO phase error signal $u[k]$ as a scaled version of the accumulated third-order $\Delta\Sigma$ quantization noise $x[k]$. Proper scaling of $x[k]$ must be performed before subtracting it from $u[k]$, and the scale factor is determined by a correlation circuit composed of a digital multiplier, an accumulator, and a first-order IIR filter. Due to the high resolution of the GRO TDC, the correlation feedback loop can be designed to have a reasonably fast settling

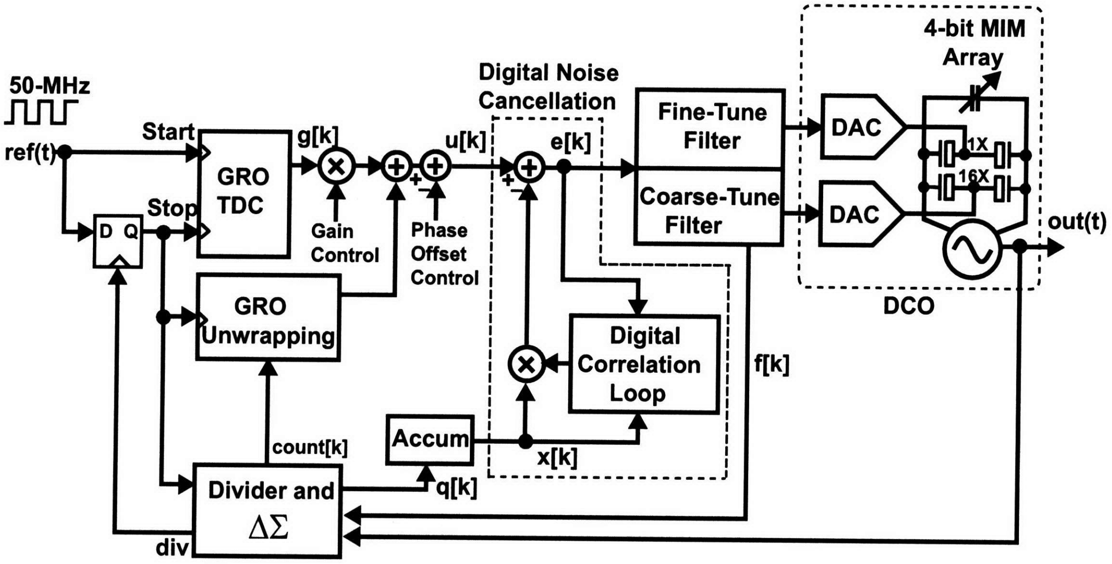


Figure 1-1: Detailed block diagram of the proposed digital $\Delta\Sigma$ synthesizer.

time without introducing a significant amount of additional noise into the synthesizer. Simulations indicate that this loop settles within 10 μs.

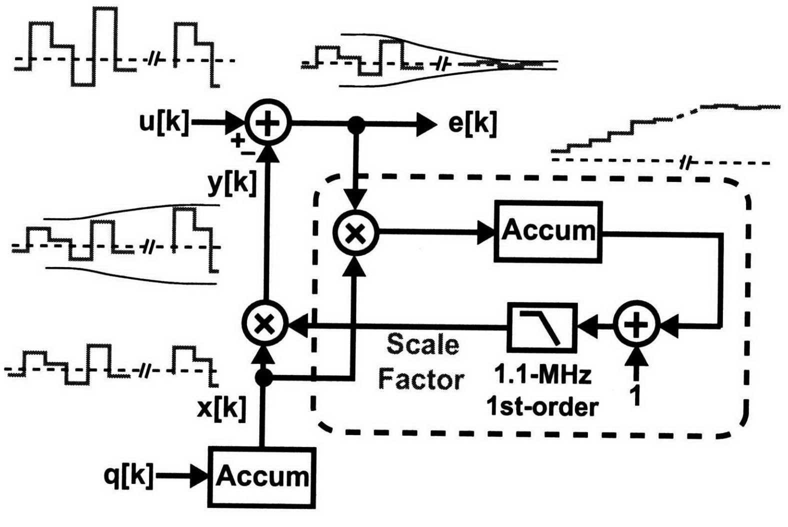


Figure 1-2: Simplified view of the all-digital quantization noise cancellation.

1.2.3 Digital-to-Analog Converter

Figure 1-3 displays a simplified circuit diagram of the 10-bit, 50-MHz DAC structure that is utilized in both the coarse and fine tuning paths. The key goals of the structure are to achieve a monotonic 10-bit DAC structure with minimal active circuitry and no transistor bias currents. The proposed topology essentially performs a two-step conversion, where the first step is performed by a five-bit resistor ladder, and the second step is performed by a five-bit zero-V_T NMOS capacitor array. In step one, the resistor ladder is used to form two voltages of value $V_L = \mathrm{M}/32 \cdot V_{DD}$ and $V_H = (\mathrm{M}+1)/32 \cdot V_{DD}$, where M ranges from 0 to 31, and V_{DD} corresponds to the 1.5-V supply voltage. Simultaneously, V_H is connected to N unit cell capacitors, and V_L to (32-N) unit cell capacitors, where N ranges from 0 to 31. In step two, the capacitors are first disconnected from the resistor ladder, and then connected to a common capacitor C_{load}. The combination of these steps at 50 MHz achieves 10-bit resolution as well as first-order filtering with cutoff frequency $f_o = 32C_u/(2\pi C_{load}) \cdot 50$ MHz. Therefore, the filtering bandwidth of each DAC is adjusted by proper selection of the C_{load} capacitor value. Note that the switches are implemented with low-V_T MOS devices.

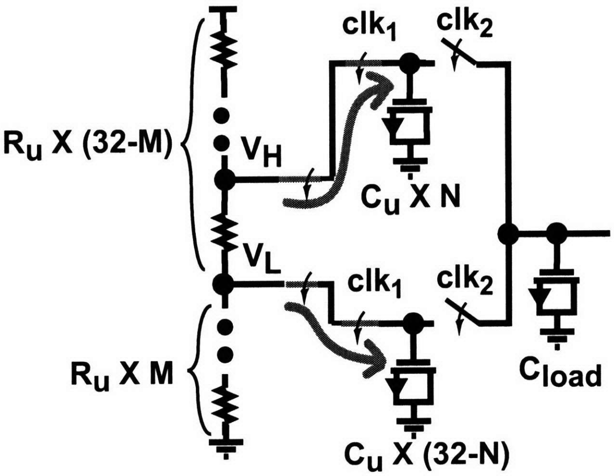


Figure 1-3: Simplified schematic of the proposed DAC.

1.2.4 Divider

Figure 1-4 displays the proposed divider structure, which leverages a common asynchronous divide-by-16-to-31 structure composed of cascaded divide-by-two/three stages [5], while achieving low noise without the use of re-timing at the divider output. As revealed by the figure, the divider structure realizes a given divide value as the addition of four values, three of which (i.e., N_0, N_1, and N_3) are always constant for a given frequency setting and one of which (i.e., N_2) is controlled by the third-order $\Delta\Sigma$ modulator to achieve fractional values. Due to the re-timing of the reference edge by the flip-flop shown in the figure, only the N_3 edge impacts the GRO phase detector, so that the divide-by-16-to-31 divider is set to a constant divide value before its output directly impacts the phase detector. The divider structure therefore avoids the divide-value-dependent jitter due to the $\Delta\Sigma$ dithering without the use of re-timing of the divider output [22][24][26].

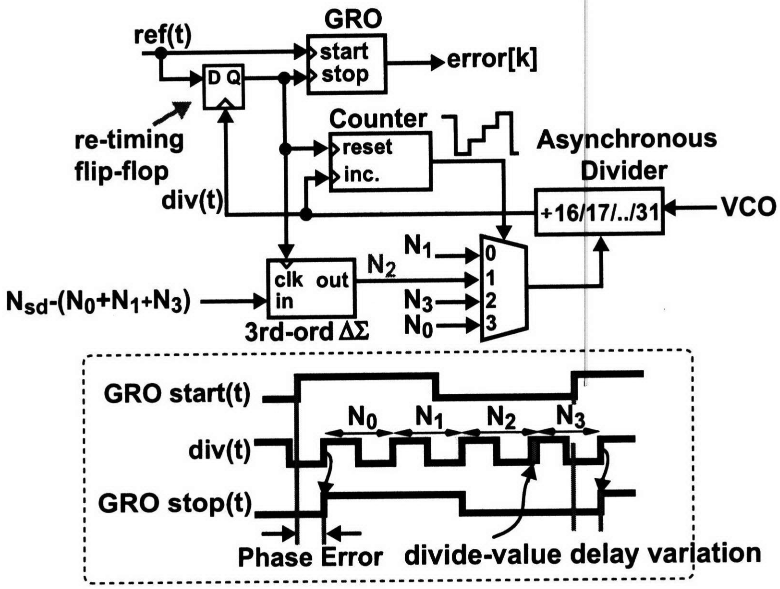


Figure 1-4: Proposed asynchronous divider structure achieving low power and jitter.

1.2.5 Loop Filter

To control both the coarse and fine varactors in the VCO, the loop filter consists of two paths, as shown in Figure 1-5. The coarse-tuning varactor, which has a K_v value that is 16 times higher than the fine-tuning varactor, is fed by a coarse-tuning DAC with eight times less bandwidth than the fine-tuning DAC to reduce the impact of its thermal noise. Further, the coarse-tuning DAC is allowed to vary only when the frequency value of the synthesizer is changed and is fixed in value during steady-state lock conditions, such that its quantization noise is eliminated from concern. During a frequency acquisition cycle, the fine-tuning DAC is held at its mid-point value during the coarse tuning, and is then allowed to vary according to the Type-II settling characteristics of the overall PLL once the coarse-tuning value is frozen. Note that a technique similar to that in [11] is used during the coarse tuning in order to allow the coarse-tuning DAC to quickly settle to its proper value while simultaneously achieving a desired phase error of zero at the overall loop filter input. The overall settling time of the synthesizer (i.e., the sum of the coarse- and fine-tuning times) is measured to be within 20 μs for 10-ppm accuracy.

1.3 Proposed Digital Phase Control Technique

Delay-locked loops (DLL) using analog phase interpolators as phase shifters have become popular because they provide reasonable phase resolution with an infinite phase range [16][17][18][19]. However, the interpolation circuits, which are implemented in the analog domain, must be accurately controlled to maintain the linearity of the phase shifter. Analog DLLs constructed with such phase interpolators usually provide good jitter performance, but the relatively high analog complexity of these blocks complicates the design of such DLLs. Therefore, we propose here a more digital DLL architecture, as shown in Figure 1-6 [15].

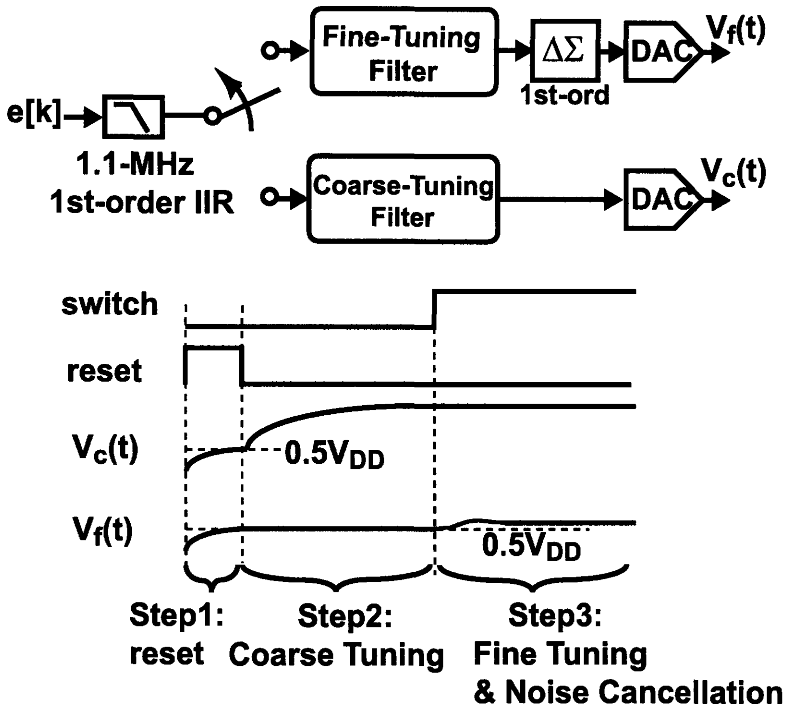


Figure 1-5: Coarse/fine tuning of the PLL output frequency.

1.3.1 Overview

The key idea behind the proposed DLL structure is to use a simple VCO instead of a phase interpolator to achieve the phase shifting functionality within a DLL. A VCO can be modeled as an integrator with the VCO phase being regarded as the output. Thus, if a positive or negative rectangular pulse is fed into the VCO, the VCO phase increases or decreases by a step at each time increment. By implementing the VCO as a standard ring oscillator, this approach offers a very simple, relatively digital implementation that has the ability to achieve very fine phase shifts and an infinite phase range.

Although it is difficult to use a stand-alone VCO as a phase shifter, as explained later, by applying feedback to the VCO in the form of a fractional-N synthesizer, as shown in Figure 1-6, the resulting fractional-N synthesizer functions as a digitally-controlled oscillator, with the $\Delta\Sigma$ modulator input being regarded as the control signal. In this way, the phase resolution can be digitally controlled and is less sensitive to the process, temperature, and voltage (PVT) variations than conventional structures based on phase interpolators.

A key element of the proposed structure is a digital $\Delta\Sigma$ modulator architecture that allows a high clock rate with a compact area and reasonable power dissipation. In addition, the output of the bang-bang phase detector (BBPD) is fed into a saturating integrator that allows the output of the detector to be averaged and converted from a three-level signal to a two-level signal. The integrator output is then sampled by a D-flip-flop (DFF) with a period of T_d; the sampled signal then controls the phase shifter.

As illustrated in Figure 1-6, only simple analog circuits are required in the proposed DLL architecture, without the need for good matching between any of their elements. The overall architecture is primarily digital and well suited for more advanced CMOS processes.

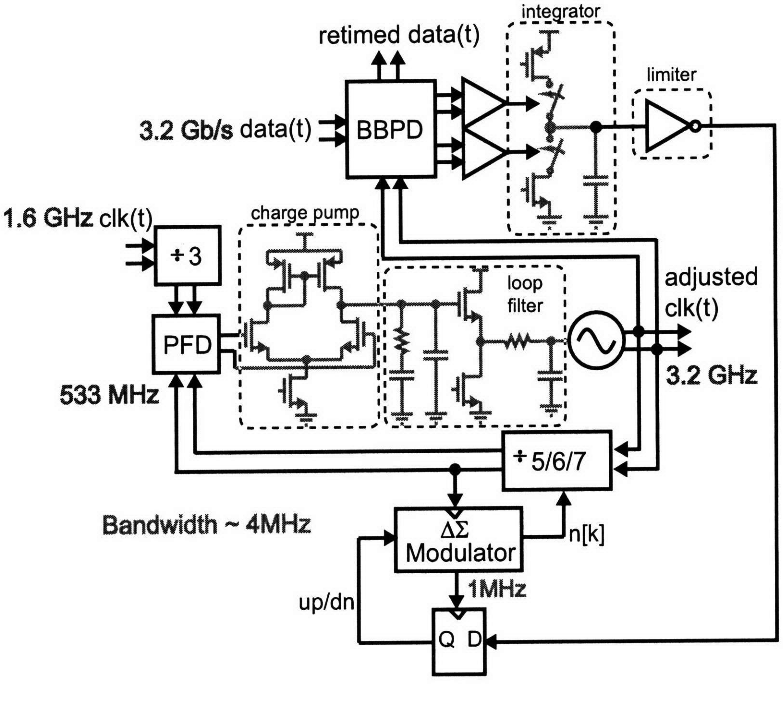


Figure 1-6: Proposed DLL with a synthesizer-based phase shifter.

1.3.2 Synthesizer-based Phase Shifter

Although in principle a ring oscillator can achieve the phase shifting function, it is quite difficult to accurately control the height and width of an analog pulse as well as to precisely set the nominal oscillation frequency of the VCO such that it is locked to the received clock of the DLL. However, by placing the VCO within a $\Delta\Sigma$ fractional-N frequency synthesizer [28][29][30][31], we can accurately control the VCO with digital precision by feeding digital pulses to the $\Delta\Sigma$ modulator, as illustrated in Figure 1-7. A phase resolution of $2\pi/2^n$ can be achieved by simply setting the number of fractional bits in the $\Delta\Sigma$ modulator to n. Thus, the resolution can be accurately and finely controlled and is independent of the PVT variations.

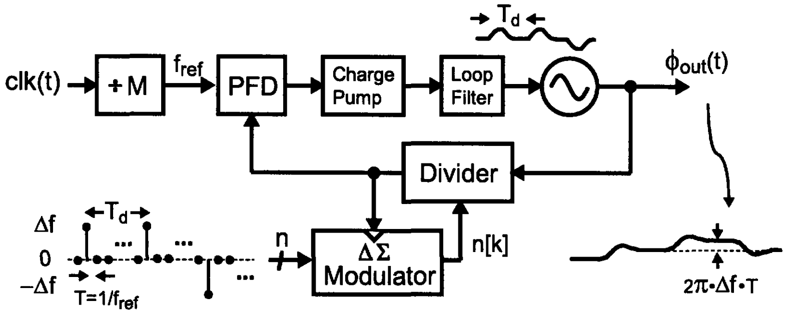


Figure 1-7: Proposed synthesizer-based phase shifter.

1.3.3 $\Delta\Sigma$ Modulator

Instead of using a standard second-order $\Delta\Sigma$ modulator, we propose a more compact and power-saving second-order modulator architecture, as shown in Figure 1-8. This architecture consists of an U/D counter, a multi-rate first-order $\Delta\Sigma$ modulator, and a differentiator. First, notice that the input to the $\Delta\Sigma$ is being updated at a rate of approximately $f_d = 1$ MHz, while the output is being updated at $f_{ref} = 533$ MHz. To connect these different sample rates, the first-order $\Delta\Sigma$ modulator must be progressively clocked from low to high frequencies. We achieve this goal by cascading three first-order $\Delta\Sigma$ modulators with different resolutions and clock rates. By using this approach, only a small portion of the overall $\Delta\Sigma$ modulator circuit operates at the highest frequency. Thus, the power consumption and design complexity are reduced at the expense of a slightly larger area.

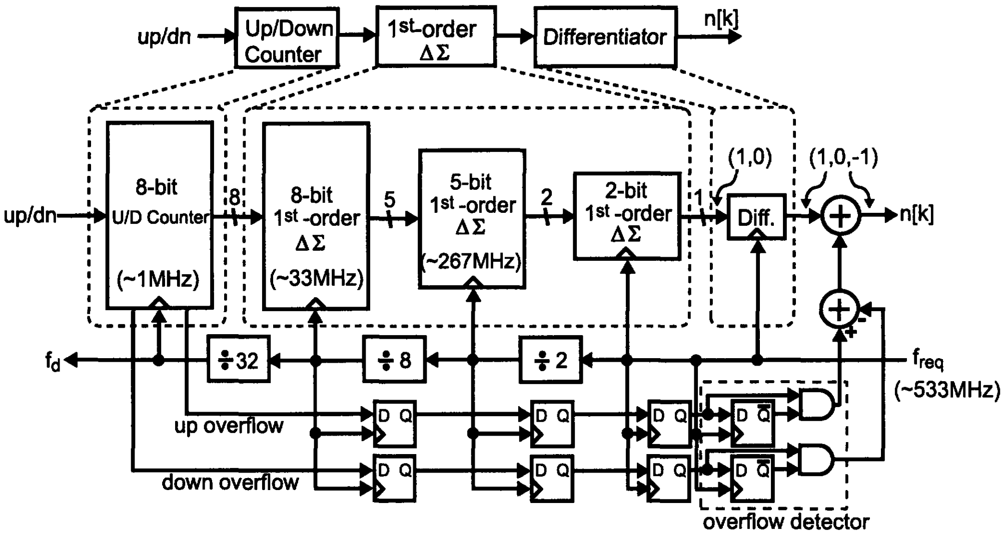


Figure 1-8: Multi-rate implementation of the proposed ΔΣ architecture.

1.4 Contributions

This thesis demonstrates techniques for high-performance digital frequency synthesis and phase control.

In the frequency synthesis part, we presents a 3.6-GHz low-noise, 500-kHz bandwidth digital ΔΣ frequency synthesizer architecture. The primary contributions of this part is as follows.

1. A synthesizer architecture that efficiently leverages a recently invented noise-shaping GRO TDC [3] to achieve the phase noise of -108 dBc/Hz at 400 kHz offset is presented. The needed peripherals for the GRO TDC are developed. (Note that the design of the GRO TDC core is not the contribution of this thesis.)

2. An all-digital quantization noise cancellation technique achieving the phase noise of -150 dBc/Hz at 20 MHz offset is presented. Proposed technique does

not need any analog components and can be implemented with digital standard cells.

3. A passive 10-bit 50-MHz digital-to-analog converter structure as an efficient interface between the digital loop filter and conventional LC oscillator is presented.

4. A 1.5-mW asynchronous divider structure that reduces the TDC range by a factor of four and avoids the divide-value-dependent delay variation without the need for re-timing the divider output is presented.

5. The measured jitter integrated from 1 kHz to 40 MHz achieves 204 fs at 3.67 GHz.

In the phase control part, we propose a digitally-controlled phase shifter based on the fractional-N synthesizer technique and demonstrate its application to a DLL for 3.2-Gb/s chip-to-chip communications. The primary contributions of this part is as follows.

1. A fractional-N-synthesizer-based phase shifter with an 1.4-degrees-resolution and infinite-range delay is presented. The delay provided by this phase shifter is less sensitive to the PVT variations than that of conventional techniques using a phase interpolator.

2. A digital $\Delta\Sigma$ modulator architecture that allows a 533-MHz clock rate with a compact area and reasonable power dissipation is presented.

3. A simple bang-bang detector supporting the proposed phase shifter is presented.

1.5 Overview of Thesis

The remaining chapters in this thesis provide further analysis and implementation details of the proposed techniques. An overview of the thesis is as follows.

In Chapter 2, we first provide sufficient background of fractional-N techniques and then focus on the key issues involved in achieving low jitter with a high PLL bandwidth in digital PLL structures. Here we see the need for a high-resolution TDC as well as a quantization noise cancellation scheme.

In Chapters 3 and 4, we provide details of the supporting blocks such as the DAC structure, which is used to control the VCO, and the low-jitter asynchronous divider.

Chapter 5 focuses on system-level issues associated with the coarse/fine-tuning approach used to control the PLL frequency. A systematic way to design a digital filter corresponding to the well known analog lead-lag filter is also described.

Chapter 6 presents the noise modeling of this digital synthesizer and compares the calculated noise performance with the time-domain behavior simulation results using *CppSim*. Trade-offs among the noise performance and several design parameters are also discussed.

In Chapter 7, measured results of the digital frequency synthesizer are demonstrated. The measured phase noise is also compared with the predicted value.

In Chapter 8, we apply the fractional-N technique to phase control of a high-speed clock. We introduce the proposed DLL structure utilizing a digitally-controlled phase shifter. Implementation details, including a $\Delta\Sigma$ modulator developed specifically for this application, are presented, followed by the measurement results.

Finally, Chapter 9 concludes this thesis and suggests some future research directions.

Chapter 2

Proposed Techniques for Achieving a Low-Noise and Wide-Bandwidth Digital PLL

In this chapter, we investigate the challenges in achieving a low-noise, wide-bandwidth digital fractional-N synthesizer. We show that the key challenges of attaining this goal lie in developing a high-resolution time-to-digital converter (TDC) and performing cancellation of the quantization noise caused by dithering of the divider. The proposed synthesizer architecture leverages a recently invented noise-shaping gated ring oscillator (GRO) TDC [2][3][14][20][21] to achieve the desired resolution and introduces an all-digital approach to quantization noise cancellation.

2.1 Background

One of the most important applications of a phase-locked loop (PLL) is the frequency synthesis. When the PLL is used as a frequency synthesizer, a digital counter divides the VCO frequency by N, and the output is compared with a clean reference frequency, as illustrated in Figure 2-1. After the loop is locked, the divider output is synchronized to the reference signal with the help of feedback. Therefore, VCO frequency becomes N times the reference frequency. With this structure, we can set the value of N to synthesize a desired frequency. This kind of PLL is called an integer-N PLL since the constraint here is that N must be an integer value. Because of this constraint, when a high output resolution is necessary, the reference frequency is usually limited to a low value because it needs to be equal to the targeted channel resolution. To maintain the stability, the PLL loop bandwidth is usually limited to less than one tenth of the reference frequency. For instance, when an output frequency resolution of 200 kHz is desired, the reference frequency must be equal to 200 kHz, resulting in a less than 20 kHz PLL bandwidth.

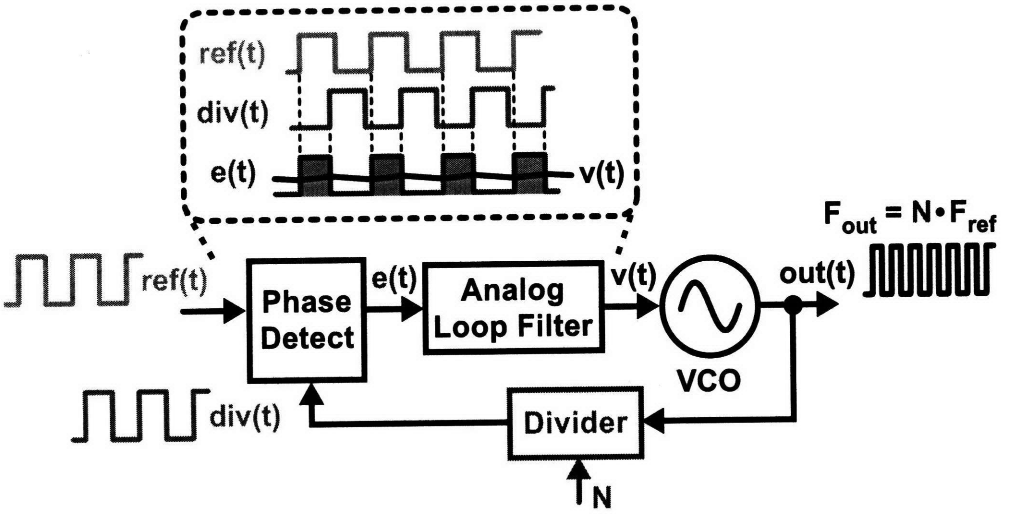


Figure 2-1: Integer-N frequency synthesizer.

The development of $\Delta\Sigma$ fractional-N frequency synthesizers has successfully broken the trade-off between the frequency resolution and bandwidth in an integer-N PLL [28][29][30][31]. In a $\Delta\Sigma$ fractional-N synthesizer, as illustrated in Figure 2-2, a fractional divide ratio is realized by dithering the divide ratio among several integer values with a $\Delta\Sigma$ modulator. The bit-length of the modulator can be extended to a high value easily to achieve a very high resolution of frequency. Since the reference frequency no longer determines the frequency resolution, a higher reference frequency can be used to obtain more freedom in setting the PLL bandwidth. Although the dithering action introduces a quantization noise, which is shaped to higher frequency offsets by the $\Delta\Sigma$ modulator, the lowpass action of the PLL dynamic can attenuate the shaped quantization noise.

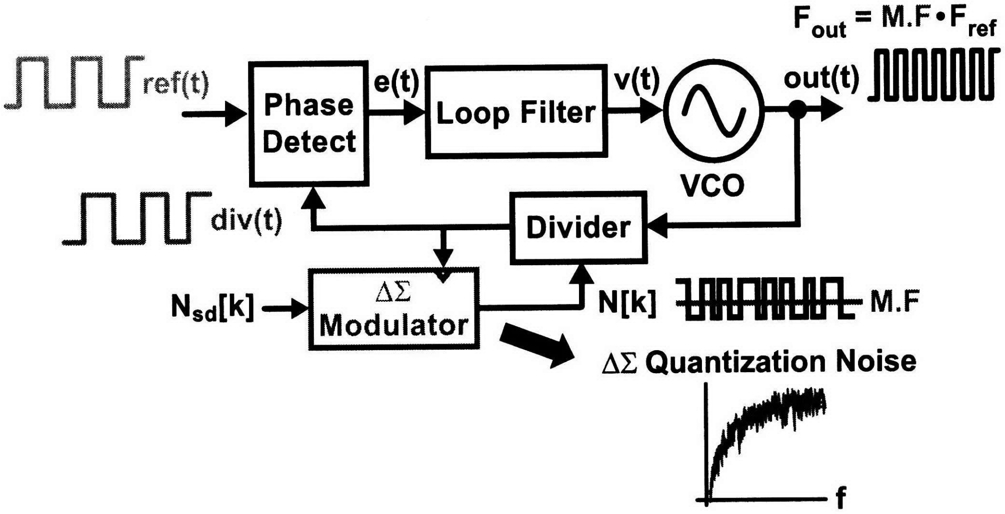


Figure 2-2: ΔΣ fractional-N synthesizer.

Recently, there have been two important trends in the research activities of fractional-N PLLs. The first trend is the development of technologies to achieve a wider loop bandwidth. The other trend is the digitalization of PLLs.

There are several advantages of using a wide loop bandwidth. First, it enables higher data-rate modulation without the pre-emphasis [32] or two-point modulation [9][33]. In addition, it enables greater VCO noise suppression and a shorter locking time. However, as illustrated in Figure 2-3, a wider PLL bandwidth leads to less quantization noise suppression. Therefore, the trade-off between the quantization noise and PLL bandwidth usually limits the possible increase in PLL bandwidth obtained by switching from an integer-N PLL to a fractional-N one.

The need for wider-bandwidth fractional-N synthesizers has motivated several researchers to develop phase noise cancellation techniques to avoid this trade-off. These techniques are reviewed in Section 2.4 [22][23][24][25][26]. State-of-the-art phase noise cancellation techniques have enabled wide PLL bandwidths of 700 kHz to 1 MHz without sacrificing the noise performance. However, all of these techniques heavily rely on analog-intensive circuits, complicating design and portability over future processes.

The continuing development of deep sub-micron CMOS processes has encouraged

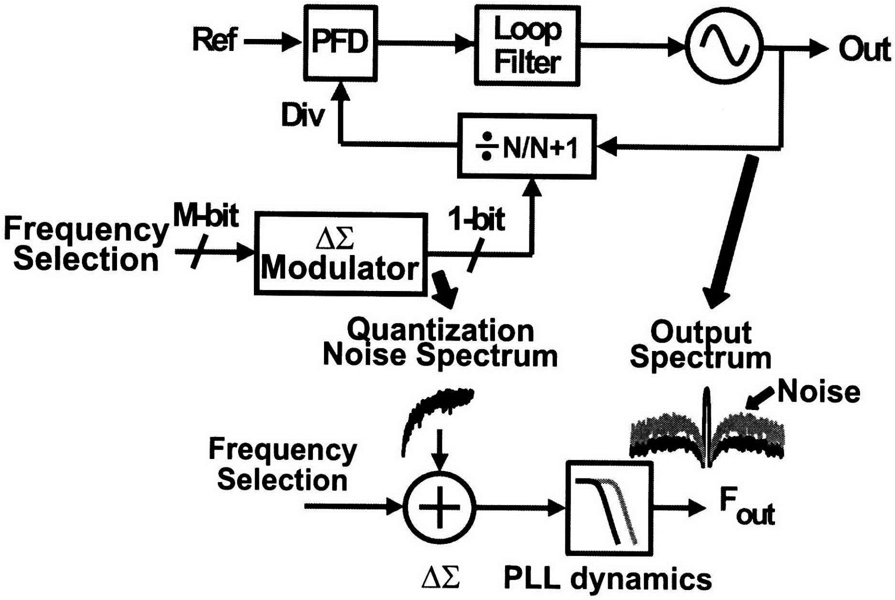


Figure 2-3: A wider PLL bandwidth results in less quantization noise suppression.

interest in an all-digital PLL. An all-digital PLL enables a compact and programmable on-chip loop filter with long time constants by leveraging the high-density digital capability available in a deep sub-micron process, as illustrated in Figure 2-4 [9]. Such a digital PLL would result in large area savings that are critical for achieving a low-cost solution and also avoid problems that conventional charge-pump PLLs would encounter in the future processes, including high variation and leakage current. Unlike an analog PLL, a digital PLL uses a TDC to perform phase detection because the TDC provides a digital phase error signal to the loop filter. Similarly, the oscillator needs to be controlled by the digital output of the filter. The work in [9] demonstrated that an all-digital synthesizer can meet the GSM specification, but its bandwidth of 40 kHz is an order lower than that achievable by analog phase noise cancellation techniques [22][23][24][25][26]. Although two recent digital synthesizers extended the loop bandwidths to 3 MHz and 142 kHz, respectively, they do not support high noise performance because the former work sacrifices its out-of-band noise performance while the latter one cannot achieve low in-band noise [10][11].

Therefore, the goal of this research is to achieve low noise, a wide bandwidth,

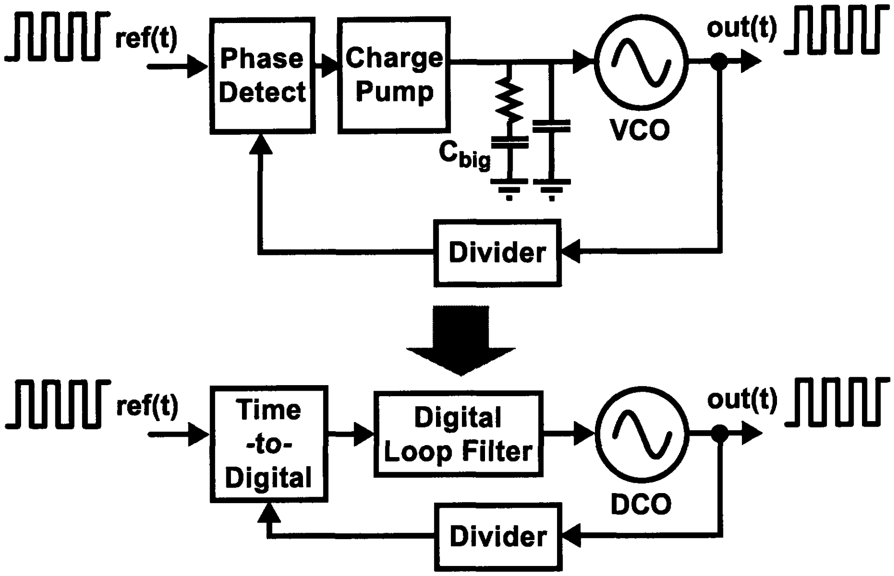


Figure 2-4: Progression from analog to digital PLL implementation.

and a digital implementation. We propose a digital fractional-N frequency synthesizer that leverages a noise-shaping TDC and a simple quantization noise cancellation technique to achieve low phase noise with a wide PLL bandwidth of 500 kHz [14]. Using this high-performance TDC, a 3.6-GHz synthesizer with <-100 dBc/Hz noise at low frequency offsets is demonstrated. In contrast to previous cancellation techniques [22][23][24][25][26], the proposed structure requires no analog components and is straightforward to implement with standard-cell digital logic. With the cancellation technique enabled, the synthesizer achieves the phase noise of -132 dBc/Hz at 3 MHz offset, and an integrated phase noise from 1 kHz to 40 MHz of 204 fs rms at 3.67 GHz. By utilizing quantization noise cancellation within a digital PLL, the proposed technique not only widens the bandwidths of digital frequency synthesizers without sacrificing their noise performance but also eliminates complicated analog circuits required in a conventional phase-noise-cancellation PLL.

More details of the fractional-N frequency synthesizer can be found in the literature [28][29][30][31]. In addition, a modeling approach for an analog fractional-N frequency synthesizer was introduced in [34]. With slight modification, the same approach was later applied to a digital PLL [35]. Note that the PLL model developed in Chapters 5 and 6 is based on this approach.

2.2 Challenge of a Low-noise Wide-bandwidth Digital PLL

The challenge of achieving a low-noise wide-bandwidth digital PLL is explored in this section. We begin by assuming that the quantization noise can be completely cancelled because it allows us to focus on the trade-off between the TDC and VCO noises as well as to understand the importance of the TDC resolution. Next, the impact of the quantization noise is discussed.

Figure 2-5 provides an intuitive view of the need for the improved TDC resolution when a high PLL bandwidth is desired. As shown in the figure, the output phase noise of a digital synthesizer is primarily influenced by the quantization noise of the TDC and the phase noise of the digitally-controlled oscillator (DCO), where the DCO is realized as the combination of a digital-to-analog converter (DAC) and hybrid VCO in our proposed system. As the figure shows, the TDC noise is lowpass-filtered by the PLL dynamics, whereas the DCO noise is highpass-filtered. Therefore, while raising the PLL bandwidth has the benefit of suppressing the DCO noise at low frequency offsets, it also carries the penalty of increasing the influence of the TDC noise. As such, the combination of a high bandwidth and low noise for the PLL demands a high-resolution TDC. Note that $G(f)$ in Figure 2-5 denotes the closed-loop PLL response, which is a lowpass filter [34].

As illustrated in Figure 2-6, the TDC in [1] uses a chain of delays to create multiple transitions and compares each transition with the VCO feedback signal to obtain a time error signal $e[k]$ in a discrete-value form. This action performs the continuous-to-discrete conversion in the time domain. Similar to analog-to-digital conversion in the voltage domain, this results in a quantization noise, whose level is determined by the unit delay value of the TDC. If we assume that the quantization noise of the TDC is white, then the in-band phase noise floor of the PLL (PN) for a given TDC

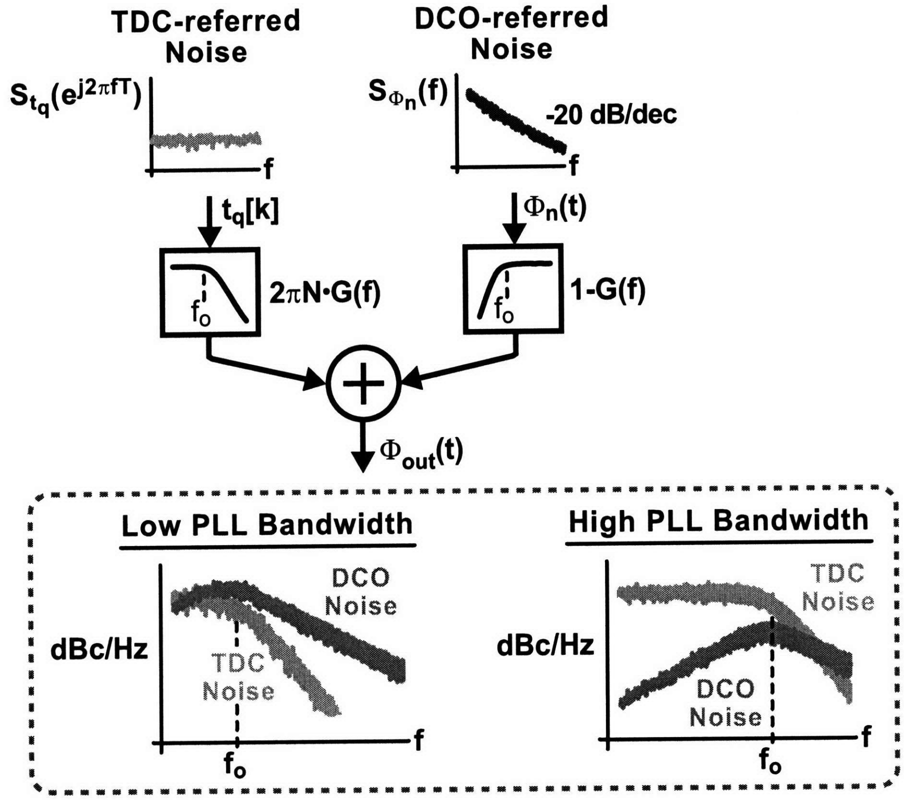


Figure 2-5: Phase noise of narrow-BW and wide-BW digital PLLs..

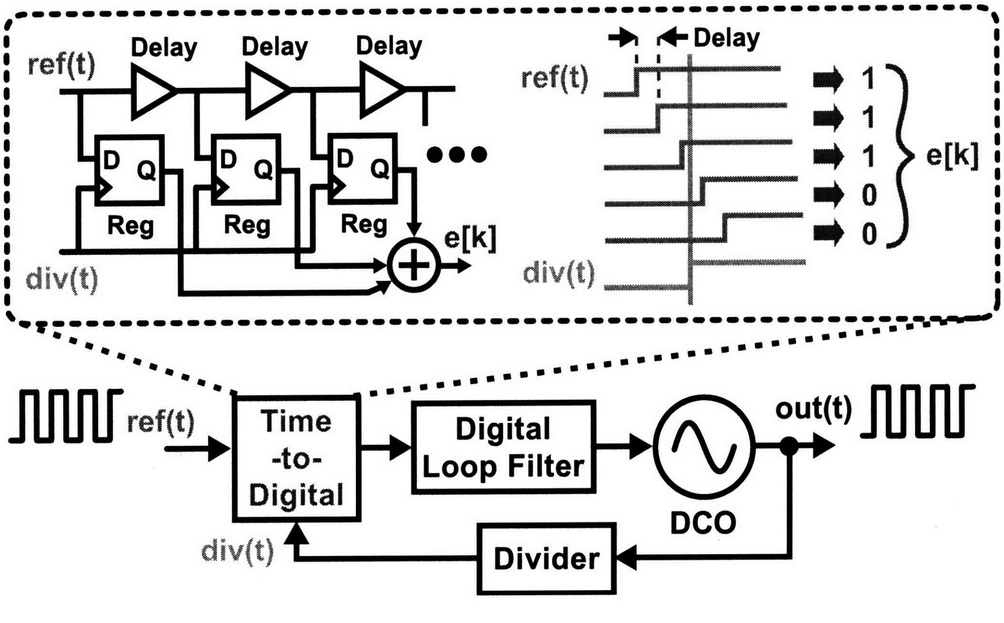


Figure 2-6: Classical time-to-digital converter in [1].

resolution (Δt_{del}) is calculated as:

$$PN = 10\log(1/T \cdot (2\pi N)^2 \cdot (1/12 \cdot \Delta t_{del}^2))(dBc/Hz) \tag{2.1}$$

where T is the reference period, and N is the nominal divide value.

To provide a sense of the TDC resolution requirements, let us first consider the example of GSM-level phase noise performance with a 20-ps TDC resolution, 3.6-GHz output frequency, and 50-MHz reference frequency (i.e., T=1/(50 MHz), N=(3.6 GHz)/(50 MHz)). According to equation 2.1, the TDC contributes an in-band noise of -95 dBc/Hz. In the case of a low PLL bandwidth, such as 50 kHz, as shown in the example in Figure 2-7(a), the overall in-band noise is usually dominated by the VCO, thus a 20-ps resolution is acceptable. However, to extend the PLL bandwidth while achieving the low noise required by GSM, this TDC architecture with a 20-ps resolution is not sufficient. As shown in Figure 2-7(b), GSM needs -100 dBc/Hz at 400 kHz offset referred to 3.6-GHz output frequency. When a PLL bandwidth larger than 400 kHz is desired, the in-band noise of -95 dBc/Hz contributed by the TDC is too high. Notice that, in this case, the VCO noise is suppressed more by the wider loop bandwidth, so the TDC noise now becomes the dominating noise source at low frequency offsets.

The noise analysis shown in Figure 2-5 ignores the fact that a quantization noise is produced by dithering of the divide value in a fractional-N synthesizer. As shown in Figure 2-3, this quantization noise is highpass-shaped due to the action of the $\Delta\Sigma$ modulator, and much of it is attenuated by the lowpass filtering action of the PLL dynamics. As shown in Figure 2-7(a), when the PLL bandwidth is narrow, the noise associated with the third-order $\Delta\Sigma$ modulator is so low that it causes no issue. However, the impact of seeking a higher PLL bandwidth is to let more of the quantization noise through such that the high-frequency noise performance of the PLL is adversely impacted. Following the same example given in the previous paragraph and assuming two poles at 1.1 MHz and 3 MHz, a 500-kHz PLL bandwidth with a third-order $\Delta\Sigma$ modulator results in -138 dBc/Hz output phase noise at 20 MHz

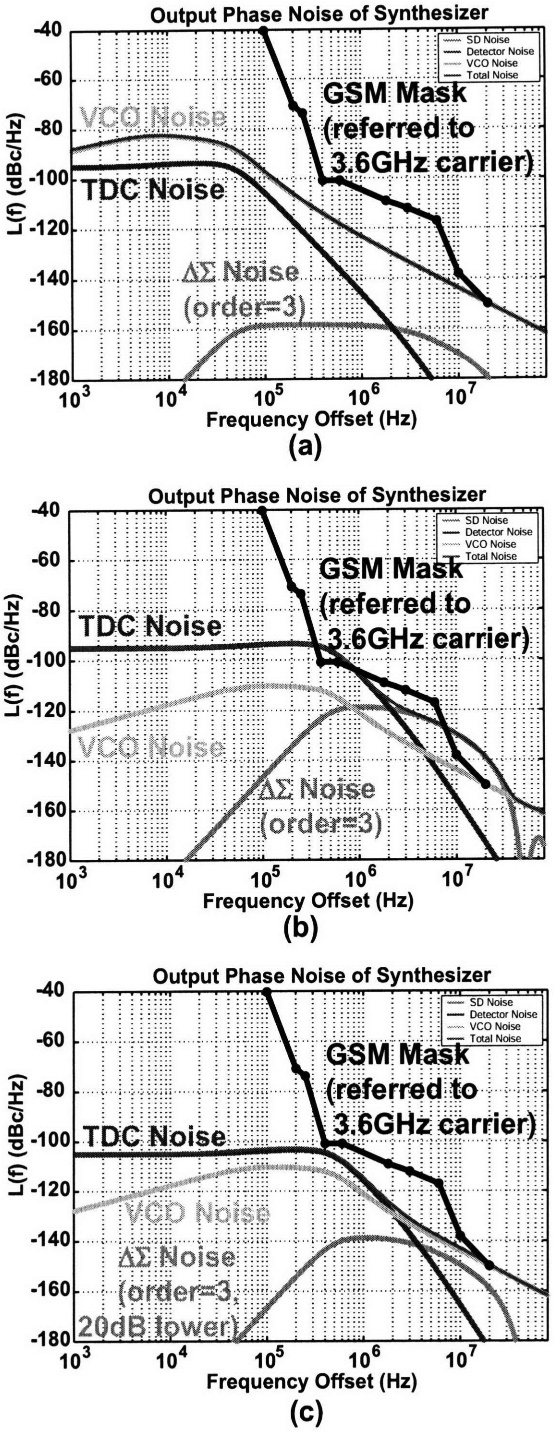


Figure 2-7: Phase noise performance: (a)50-kHz BW and 20-ps TDC resolution (b)500-kHz BW and 20-ps TDC resolution. (c)500-kHz BW, 6-ps TDC resolution, and 20-dB ΔΣ noise cancellation.

offset, as illustrated in Figure 2-7(b), which is 12 dB higher than the required -150 dBc/Hz to meet the GSM-level noise performance (referenced to a 3.6-GHz carrier frequency).

As illustrated in Figure 2-7(b), both the TDC and divider quantization noises cannot meet the GSM-level requirement. To meet the GSM specification, the TDC resolution needs to be reduced to 6 ps, as illustrated in Figure 2-7(c), which is not trivial even with today's processes. Furthermore, we also need to perform quantization noise cancellation to achieve at least 20 dB lower quantization noise to meet the mask. A GRO TDC and an all-digital quantization noise cancellation approach are introduced in Sections 2.3 and 2.5 in order to solve these two problems.

2.3 Review of the Gated Ring Oscillator TDC

For a classical TDC structure [1], the TDC resolution corresponds to an inverter delay. An inverter delay in a 0.13-μm process is about 35 ps, which is much larger than the goal of the 6-ps resolution. However, an alternative approach to obtain higher *effective* resolution is to pursue noise shaping of the TDC quantization noise and leverage the fact that the TDC output is lowpass-filtered by the PLL such that the high-frequency portion of that noise is removed.

Such noise shaping can be achieved by using a GRO topology for the TDC [2][3], as shown in Figure 2-8. As the figure reveals, a GRO TDC measures the phase error between two signals by enabling a ring oscillator during the measurement window and counting the resulting transitions that occur in the oscillator. Between measurements, the GRO is disabled such that its internal state is kept intact. When the GRO is enabled in the next measurement, it ideally picks up where it left off such that the quantization error from the end of the previous measurement is directly related to the quantization error at the beginning of the current measurement. The overall quantization noise becomes

$$e[k] = q[k] - q[k-1] \tag{2.2}$$

where $q[k]$ denotes the *raw* quantization error at the end of each measurement. The first-order difference operation indicated by the above equation reveals that first-order shaping of the quantization noise is achieved with the GRO structure. A more subtle advantage of the GRO structure is that it also scrambles the quantization noise of the TDC, which provides an important advantage in avoiding limit cycles in the PLL and improving spurs [21]. Also, another subtle point is that mismatch between delay stages is also first-order noise shaped due to the barrel-shifting action of the transitions through the ring oscillator structure, so that excellent linearity of the TDC can be achieved without the need for calibration [21].

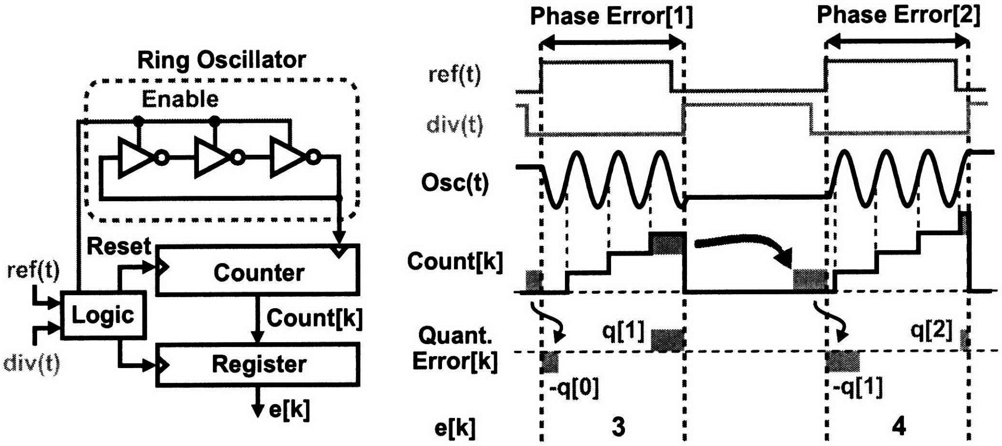


Figure 2-8: Concept of a gated ring oscillator TDC [2].

In practice, one can count transitions in all of the oscillator stages [2] rather than just transitions in a single stage as shown in Figure 2-8. By doing so, the *raw* resolution corresponds to an inverter delay, which is similar to the case for the commonly used TDC described in [1]. Again, the advantage of the GRO TDC over the conventional TDC is that the *effective* resolution is reduced well below an inverter delay by virtue of the noise shaping that it offers.

To further improve the GRO resolution, the multipath technique of reducing the delay per stage of a ring oscillator was applied by connecting the inputs of each delay stage to a combination of previous delay stages [36]. As shown in Figure 2-9, application of this technique to the GRO entails the use of multiple devices for each delay element and connection of their gates to an appropriate combination of delay stages [3]. The relative weight of each delay stage input is controlled through appropriate sizing of its given device. In the 0.13-μm CMOS prototype presented in this thesis, the multipath technique allows reduction of the delay per stage from 35 ps (i.e., one inverter delay) to 6 ps, hence yielding a factor of five improvement in TDC *raw* resolution. One should note that the *effective* resolution is further enhanced by the noise-shaping behavior of the GRO. Additional details of the final TDC used in this prototype synthesizer are described in [3][21]; the schematic of the multipath GRO is redrawn in Figure 2-10 for reference.

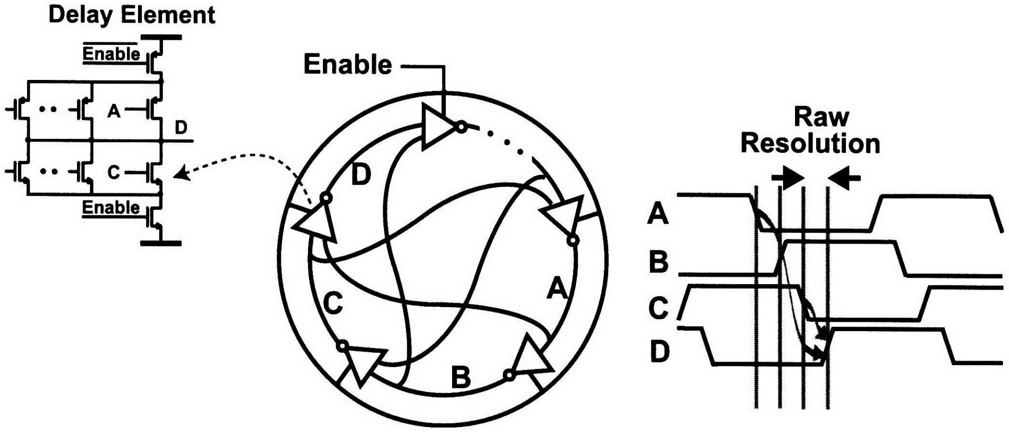


Figure 2-9: Concept of a multipath gated ring oscillator TDC.

The model of the GRO can be illustrated as Figure 2-11 [35]. The phase difference between $\Phi_{ref}[k]$ and $\Phi_{div}[k]$ is first calculated and then scaled to obtain the time difference with the gain of $T/2\pi$. After being summed together with the shaped quantization noise $t_q[k]$, the time difference is then scaled by the TDC gain $1/\triangle t_{del}$ to obtain $e[k]$.

In addition to the quantization noise, there are another two noise sources from the GRO [21]. As illustrated in Figure 2-12, the first noise is a white noise that is about 1 ps in time. The second noise has a -10 dB/dec roll-off and is caused by the flicker noise of the oscillator. In the end of this chapter, it is shown that this flicker noise dominates the PLL noise at low frequency offsets.

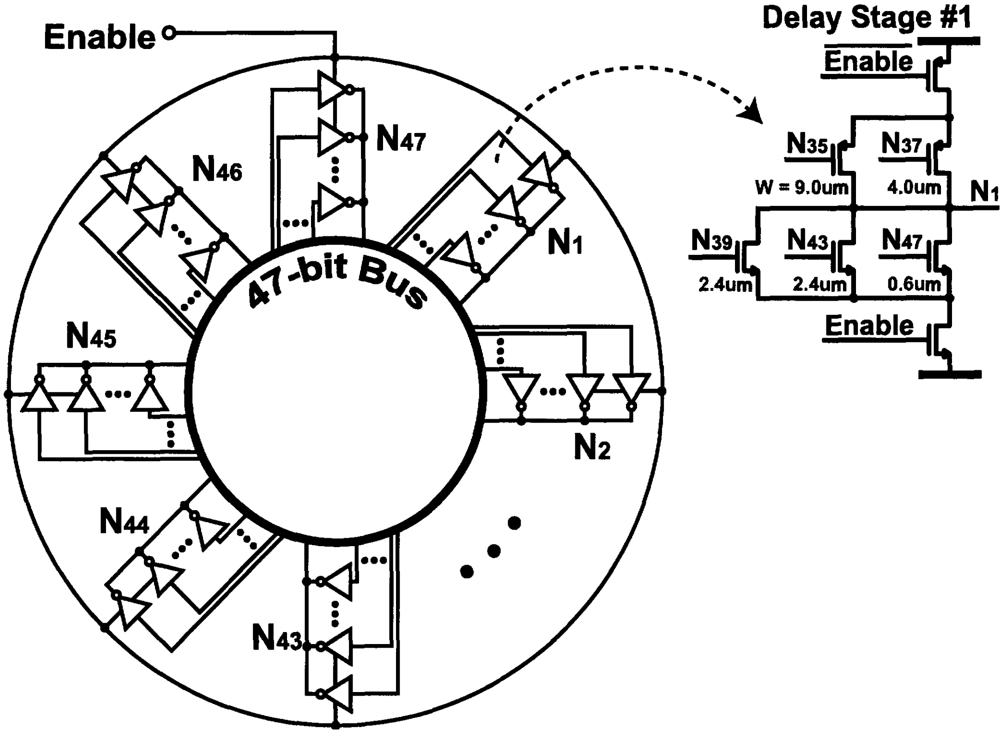


Figure 2-10: The prototype synthesizer in this thesis uses the multipath gated ring oscillator TDC in [3].

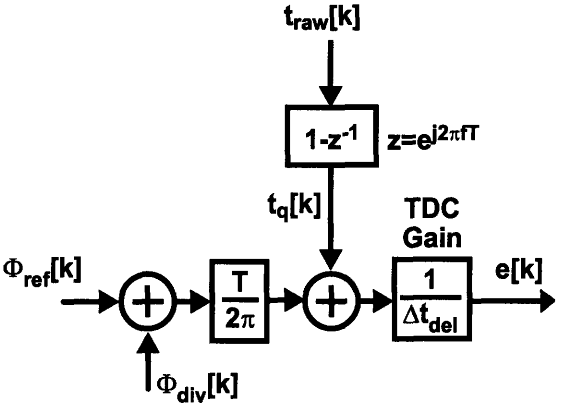


Figure 2-11: Model of the GRO TDC.

2.4 Review of the Previous Noise Cancellation Techniques in an Analog PLL

Rather than filtering the quantization noise with a narrow PLL bandwidth, recent research has demonstrated that the quantization noise in a $\Delta\Sigma$ fractional-N synthesizer

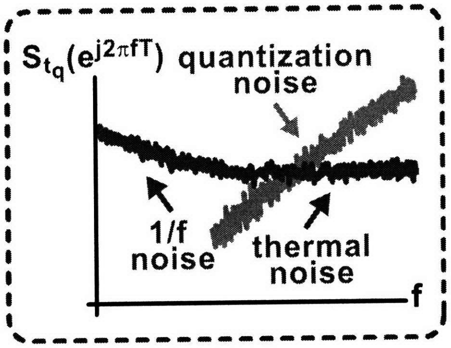


Figure 2-12: GRO causes another two noises other than the quantization noise.

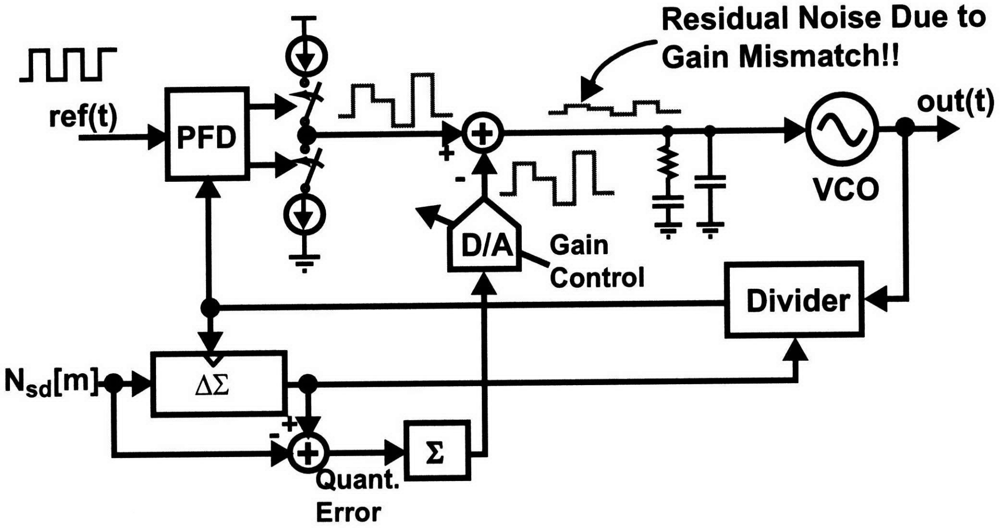


Figure 2-13: Classical phase noise cancellation PLL.

can be significantly reduced through cancellation [22][23][24][25][26]. In an analog fractional-N synthesizer, the quantization error due to the $\Delta\Sigma$ modulation results in a phase error at the phase detector output, as shown in Figure 2-13, but cancellation is achieved by first computing the quantization error using a simple digital subtraction circuit between the $\Delta\Sigma$ input and output, accumulating it (to convert from a frequency to phase signal), and then canceling it at the charge pump output through the use of a current DAC, which is necessary because the phase detector output is an analog signal while the quantization error is a digital signal. Unfortunately, high levels of cancellation require the gain of the DAC to be precisely matched to the effective gain of the charge pump. Any mismatch between these two paths leads to residual phase noise and tones. Therefore, matching in the analog domain limits the noise performance.

Earlier noise cancellation techniques did not try to calibrate DAC gain to match that of the PFD [22][23]. Instead, they reserved enough margins to tolerate the residual phase error due to the unmatched gains. As a result, the achievable noise performance is not good enough for some applications, such as GSM. The technique in [24] avoids the mismatch by embedding the DAC function within the PFD structure.

Recently, an adaptive calibration loop was proposed to dynamically set the DAC gain to minimize the residual error [25]. The VCO control voltage is multiplied by the sign of the accumulated quantization error associated with the $\Delta\Sigma$ modulator. This action calculates the absolute value of the PFD output phase error and uses it as an indicator of the mismatch amount. A feedback loop then accumulates the absolute value of the phase error and uses the output to control the DAC gain. The problem with this technique is that the DC value of the VCO control voltage is also multiplied by the sign, which introduces a large amount of tones and thus needs a low-bandwidth filter in the calibration loop to attenuate the tones [26]. The resulting one-second settling time of the calibration loop prevents this technique from being a practical solution for most applications.

Later, another similar technique multiplies only the AC component of the phase error with the sign of the accumulated quantization error [26]. This invention reduces the settling time to 35 μs, making the calibration technique more useful. However, the need for intensive analog circuits, including an operational amplifier, a differential loop filter, and a DAC, challenges the portability of this technique to future processes. In contrast, we propose an all-digital correlation loop implemented only with standard digital cells in this thesis.

2.5 Proposed Digital Noise Cancellation Technique

A digital fractional-N synthesizer can deal with the quantization noise directly in the digital domain, and thereby avoid the need for extra analog circuits in performing cancellation. Therefore, we propose an all-digital cancellation loop that can be implemented with standard logic cells, as shown in Figure 2-14. As in the analog approach, the quantization noise is fed into an accumulator (to convert from frequency to phase) and then subtracted from the TDC output after being properly scaled. Unlike the analog approach, our solution uses a digital multiplier to scale the quantization error, and the scale factor is easily computed by a simple digital correlator (i.e., a 16-bit digital multiplier) and accumulator circuit, as shown in Figure 2-15.

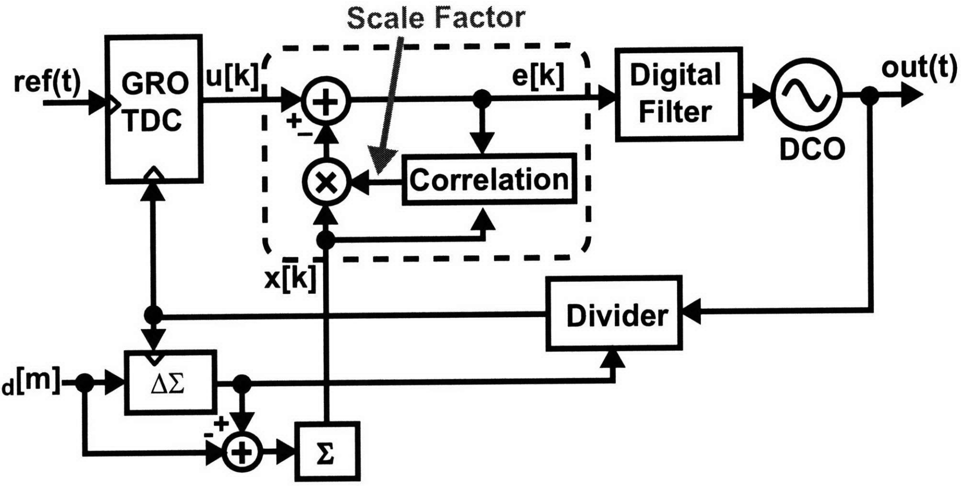


Figure 2-14: A digital PLL allows noise cancellation in the digital domain without the need for analog components.

Again, the goal of this circuit is to remove the noise introduced by the dithering action of the divider, which is manifested in the TDC phase error signal $u[k]$ as a scaled version of the accumulated ΔΣ quantization noise $x[k]$. Proper scaling of $x[k]$ must be performed before subtracting it from $u[k]$. The noise cancellation function and correlation feedback loop is enabled once the PLL has settled properly. In the beginning of the correlation process, the scale factor is set to one. Since the difference between $y[k]$ and $u[k]$ remains in $e[k]$, $e[k]$ is highly correlated to $x[k]$. Therefore, the

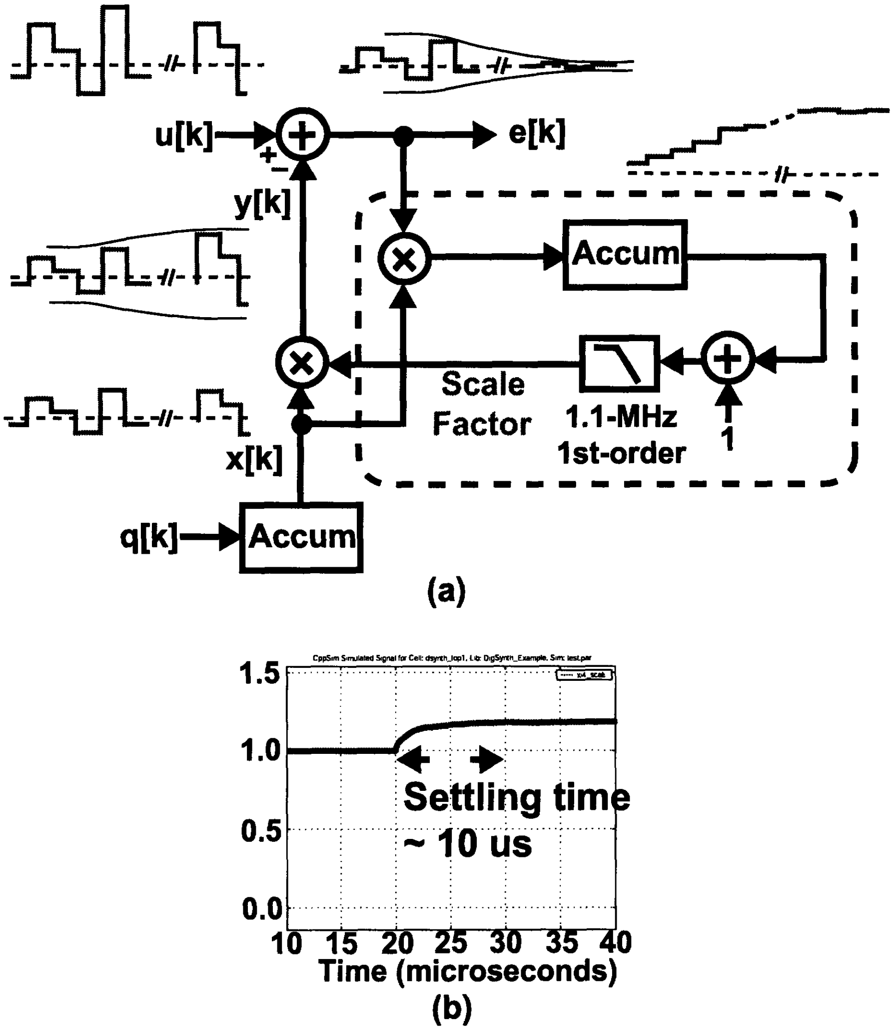


Figure 2-15: All-digital quantization noise cancellation: (a) simplified view of circuits, (b) settling behavior of the scale factor.

product of $e[k]$ and $x[k]$ is positive (negative) when the magnitude of $u[k]$ is larger (smaller) than $y[k]$. Since the scale factor is calculated by accumulating and filtering the correlation output, it ramps up or down, and the difference between $u[k]$ and $y[k]$ decreases gradually as a result. In the case where the quantization noise is completely cancelled, the correlation becomes zero in average because the residual noise in $e[k]$ is dominated by the TDC quantization noise, which is uncorrelated to the divider quantization noise. Thus, the accumulator can hold its value at the proper scale factor. Also, if there is some low-frequency variation in the TDC gain, this variation can be tracked by the correlation loop in order to keep the residual quantization noise small.

An IIR lowpass filter with cutoff frequency of 1.1 MHz is used to further smooth the scale factor signal. Due to the high resolution of the TDC, the correlation feedback loop can be designed to have a reasonably fast settling time without introducing a significant amount of additional noise into the synthesizer. In the prototype system presented here, the loop is designed to settle in less than 10 μs without adverse effects to the phase noise of the synthesizer.

One side benefit of the quantization noise cancellation circuit is that it can be used to precisely track the TDC gain. In the prototype, this information is not used since a coarse open-loop gain calibration of the PLL by hand, which is implemented by a 12-bit digital multiplier following the GRO TDC, as shown in Figure 2-17, is sufficient for the academic context of this work. However, future applications may benefit from this information in the case where the TDC gain plays a critical role in the system performance.

Similar algorithms were implemented with analog-intensive circuits before this work [25][26]. With an all-digital implementation, analog non-idealities, such as DC offset [26], are completely eliminated. Furthermore, compared to the previous works, the proposed loop avoids the nonlinear sign function by multiplying the TDC phase error with the predicted phase error. The proposed correlation loop thereby generates fewer spurs than previous techniques and is easy to implement in the digital PLL.

Finally, another possible implementation of this noise cancellation technique is discussed in Section 9.2.

2.6 Proposed Digital $\Delta\Sigma$ Fractional-N Synthesizer

As described in Section 2.2, in a wide-bandwidth digital PLL, the noise at low to intermediate frequency offsets is limited by the resolution of the TDC, while the noise at high frequency offsets is limited by the $\Delta\Sigma$ quantization noise due to the divider dithering.

In order to achieve low noise with a wide bandwidth, we leverage the first-order noise-shaping multipath GRO TDC, described in Section 2.3, to achieve low in-band noise and propose an all-digital quantization noise cancellation technique, described in Section 2.5, to achieve low out-of-band noise. By combining these two techniques, we achieve a 500-kHz PLL bandwidth at 3.6-GHz carrier frequency with <-100 dBc/Hz in-band phase noise as well out-of-band phase noise of -150 dBc/Hz at 20 MHz offset.

Figures 2-16 and 2-17 show a simplified and detailed block diagram of the proposed synthesizer, respectively. In addition to the multipath GRO and the proposed all-digital quantization noise cancellation circuitry, other interesting components of the architecture include an asynchronous frequency divider that avoids the divide-value delay variation at its output. Furthermore, in contrast to previous digital PLL implementations [9], the DCO is implemented as a conventional LC VCO with coarse and fine varactors that are controlled by two passive 10-bit, 50-MHz DAC structures. To control the dual-port VCO, a dual-path digital filter (i.e., a coarse filter and a fine filter) is proposed. We discuss these blocks in more detail in the next three chapters.

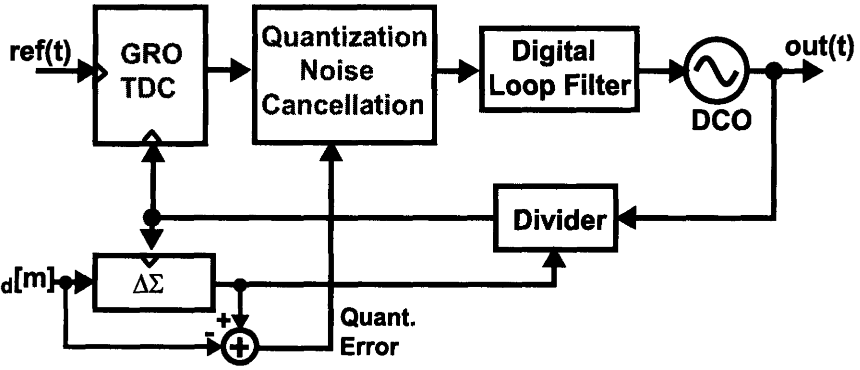


Figure 2-16: Proposed digital $\Delta\Sigma$ synthesizer utilizing the GRO TDC and the all-digital noise cancellation.

Note that a 50-MHz reference clock is used in this prototype. The reference clock is sampled by the divider output, as shown in Figure 2-17, and the resulting signal *stop* triggers the rest of the system such that all blocks are synchronous to the VCO edge. This point is discussed in more detail in Chapter 4.

Although the detailed analysis of the proposed synthesizer is deferred until Chap-

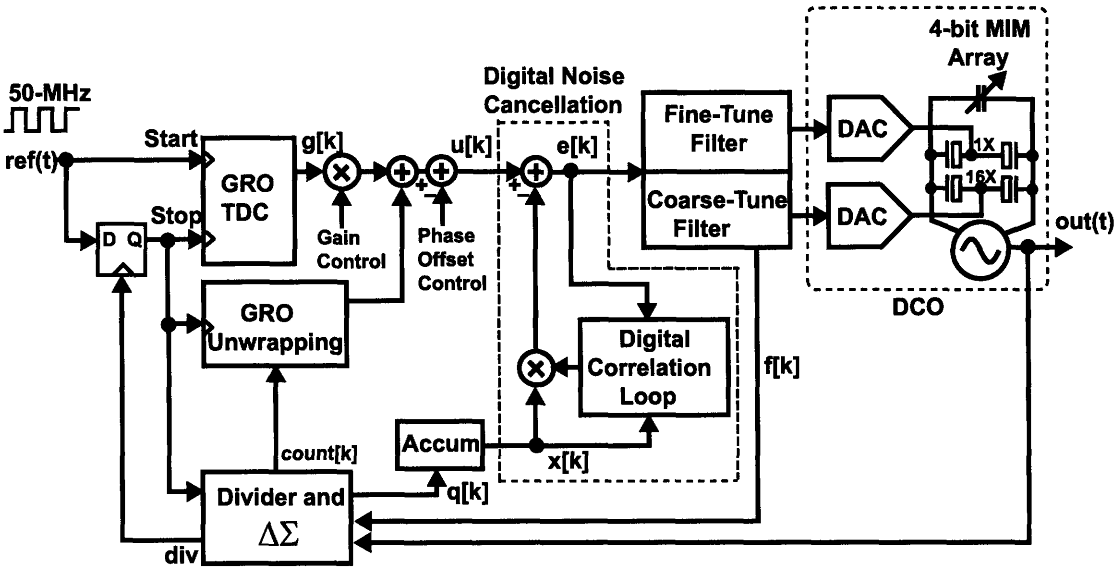


Figure 2-17: Detailed block diagram of the proposed digital ΔΣ synthesizer.

ter 6, the predicted noise performance of the synthesizer is shown here to demonstrate the advantages of the proposed PLL. First, Figure 2-18 illustrates the predicted phase noise when the GRO thermal and flicker noises are ignored. It is assumed that the GRO raw resolution is 6 ps, and the quantization noise is suppressed by 20 dB. One should see that although the shaped GRO quantization noise rises by 20 dB per decade, it is attenuated by the PLL loop filter at high frequency offsets. As a result, the GRO quantization noise is considerably below the VCO noise. At very low frequency offsets, noise of the crystal oscillator becomes the dominating source.

In addition, Figure 2-19 depicts the case where the GRO thermal and flicker noises are included. Even though the flicker noise of the GRO becomes the dominating noise source at low frequency offsets, the overall noise performance is still excellent.

To conclude, with the GRO and the noise cancellation technique, the bandwidth of the digital synthesizer can be extended to 500 kHz without violating the GSM mask, as shown in Figure 2-19. The resulting overall PLL noise is dominated by the VCO at high frequency offsets, while the low-frequency performance is limited by the flicker noise of the GRO. Details of the noise analysis can be found in Chapter 6, after the noise model of the proposed DAC and the coarse/fine-tuning scheme are

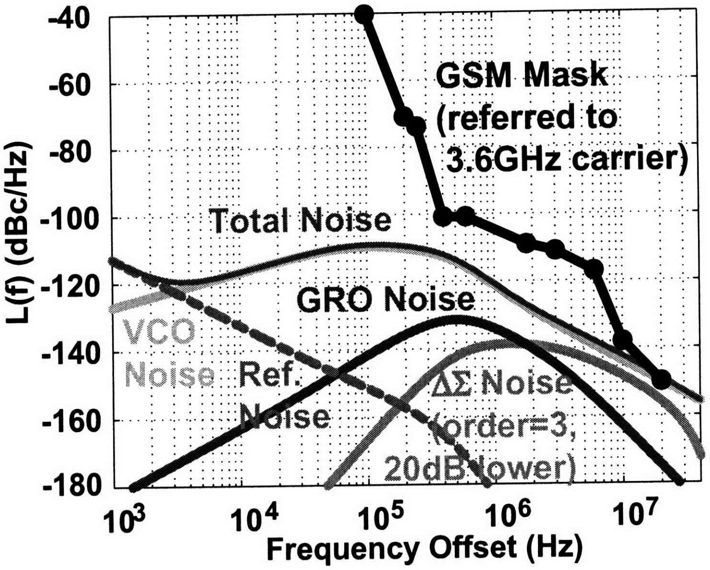


Figure 2-18: Predicted PLL noise performance using a multipath GRO TDC and an all-digital noise cancellation. (The thermal and flicker noises of the GRO are ignored.)

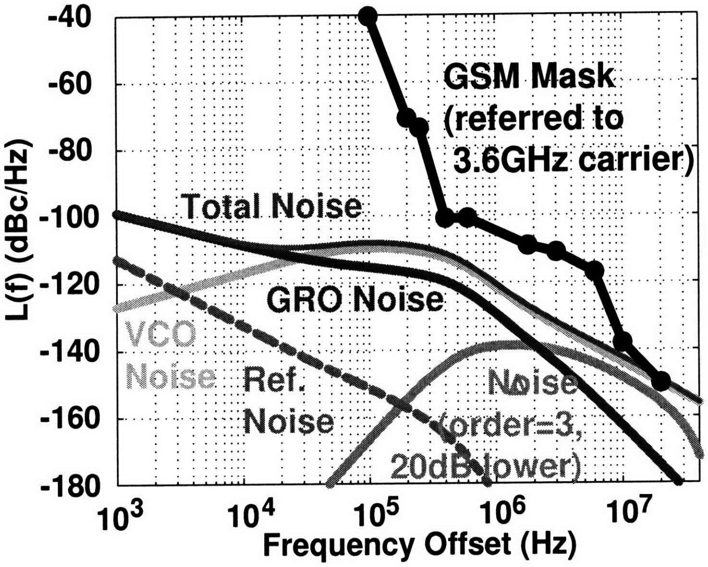


Figure 2-19: Predicted PLL noise performance including thermal and flicker noises of the GRO.

presented in Chapters 3 and 5, respectively.

2.7 Summary

There are two key challenges to extend the bandwidth of a digital fractional-N PLL. The first challenge is the need for a high-resolution TDC since its quantization noise becomes the dominating noise source at low frequency offsets in a wide-bandwidth PLL. The second challenge is the need for the divider quantization noise cancellation because a wide bandwidth also allows more quantization noise to go through.

To solve the first problem, we leverage a recently invented GRO TDC [3]. Implemented in a 0.13-μm process, this TDC achieves 6-ps *raw* resolution using a multipath ring oscillator. Furthermore, GRO TDC improves its *effective* resolution by first-order noise shaping. Therefore, we can move the TDC quantization noise to higher frequency offsets and leverage the PLL filtering to attenuate this undesired noise. By doing so, <-100 dBc/Hz noise is achievable within the PLL bandwidth in a 3.6-GHz digital PLL, where the low-frequency performance is limited by the flicker noise of the gated ring oscillator.

To solve the second problem, an all-digital quantization noise cancellation scheme is proposed. Unlike the analog PLL, in a digital PLL, scaling of the accumulated quantization noise can be performed purely with a digital multiplier. The scale factor is simply set by a digital correlation circuit that consists of a multiplier, an accumulator, and an IIR filter. With proper design, the correlation loop can settle within 10 μs without impacting the phase noise performance. The PLL noise at high frequency offsets is dominated by the VCO with this technique enabled.

Chapter 3

Digital-to-Analog Converter for VCO Control

While the time-to-digital converter (TDC) and digital noise cancellation circuits play the key roles in achieving low noise with a high bandwidth, the digitally-controlled oscillator (DCO) and frequency divider circuits present their own challenges in striving for an elegant implementation of the overall digital synthesizer.

In this chapter, we introduce the proposed DCO. As mentioned earlier, we consider the case of using a combination of a digital-to-analog converter (DAC) and hybrid voltage-controlled oscillator (VCO) to implement the DCO. Hybrid VCOs, which leverage a switched-capacitor array for frequency band selection and an analog varactor for fine tuning, have become a popular choice in many recent phase-locked loops (PLL) due to their ability to achieve a wide tuning range with excellent phase noise. While there is much literature on designing such VCOs [37][38], there has been very little research in determining appropriate DAC structures for this application space [11][39]. We propose an efficient passive DAC implementation that requires minimal analog content. We also say a few words about the hybrid VCO structure that is used as well as the modeling of the resulting DCO.

3.1 Passive Digital-to-Analog Converter

While the recent trend in digital PLLs is to create a sophisticated DCO using a switched-capacitor network [9], it is worthwhile to note that the design effort required to achieve good performance from such an approach may be prohibitive in many PLL applications. Also, some applications that could benefit from the small loop filter size of a digital PLL may be constrained to using an older technology that does not support the fine capacitor values required for a switched-capacitor DCO.

In addition, by putting a switched-capacitor array, that needs a high-speed operation clock (for example, 600 MHz in [40]) as well as complicated dynamic-element matching (DEM) algorithm [41], close to the VCO, it may be difficult to isolate the VCO core from the noises and tones generated on the digital side.

In such cases, it is worthwhile to consider the combination of a DAC and VCO for this function [11][39]. We therefore focus on the issue of achieving an efficient, "highly-digital" DAC implementation that avoids analog blocks, such as operational amplifiers and transistor bias networks. This also allows the use of an existing VCO design in a digital PLL.

3.1.1 DAC Operation

A five-bit resistor-ladder DAC is used in [11] to control a VCO, but the corner frequency of the RC low-pass filter following it suffers from the process variation. This variation in filter corner frequency may overwhelm the advantage of using a digital loop filter. Alternatively, a switched-capacitor DAC can provide a precise corner frequency that can be reconfigured by changing the capacitor ratio or clock frequency in a multi-standard application. The main idea of the proposed DAC structure is to utilize a five-bit switched-capacitor DAC to interpolate a finer voltage between two adjacent voltages provided from a five-bit resistor ladder.

Figure 3-1 displays a simplified circuit diagram of the proposed DAC structure, which provides 10-bit, 50-MHz operation with a full-supply output range using a passive circuit structure. The key idea of the proposed DAC structure is to perform a two-step conversion process using a five-bit resistor ladder in combination with a five-bit capacitor array. In step one, as illustrated in Figure 3-1(a), the resistor ladder is used to form two voltages of value V_L = M/32 · V_{DD} and V_H = (M+1)/32 · V_{DD} , where M ranges from 0 to 31, and V_{DD} corresponds to the 1.5-V supply voltage. Simultaneously, V_H is connected to N unit cell capacitors, and V_L to (32-N) unit cell capacitors, where N ranges from 0 to 31. The values of M and N are determined by the five MSBs and five LSBs of the 10-bit incoming data, respectively. In step two, as illustrated in Figure 3-1(b), the capacitors are first disconnected from the resistor ladder and then connected to a common capacitor C_{load} . The steady-state voltage of the DAC output can be derived to be:

$$\begin{aligned} V_o &= (N \cdot V_H + (32 - N) \cdot V_L) \cdot \frac{1}{32} \\ &= (N \cdot \frac{M+1}{32} \cdot V_{DD} + (32 - N) \cdot \frac{M}{32} \cdot V_{DD}) \cdot \frac{1}{32} \\ &= (M + \frac{N}{32}) \cdot \frac{V_{DD}}{32} \qquad (3.1) \end{aligned}$$

Therefore, the combination of these steps at 50 MHz achieves 10-bit resolution as well as first-order filtering with cutoff frequency [42]

$$f_o = 32C_u/(2\pi C_{load}) \cdot 50MHz \qquad (3.2)$$

Therefore, the filtering bandwidth of each DAC can be adjusted by proper selection of the C_{load} capacitor value.

3.1.2 Design Considerations and Implementation Details

Figure 3-2 illustrates the implementation details of the proposed DAC structure. Again, the 10-bit DAC consists of a five-bit resistor ladder and a five-bit capacitor array. There are 64 switches (S_1) between the resistor ladder and the capacitor array: half of them connect each node in the resistor ladder to V_H, and the other half connect these nodes to V_L. Each clock period, two adjacent switches are turned on to send voltages across one resistor, (M+1)/32·VDD and M/32·VDD, to V_H and V_L,

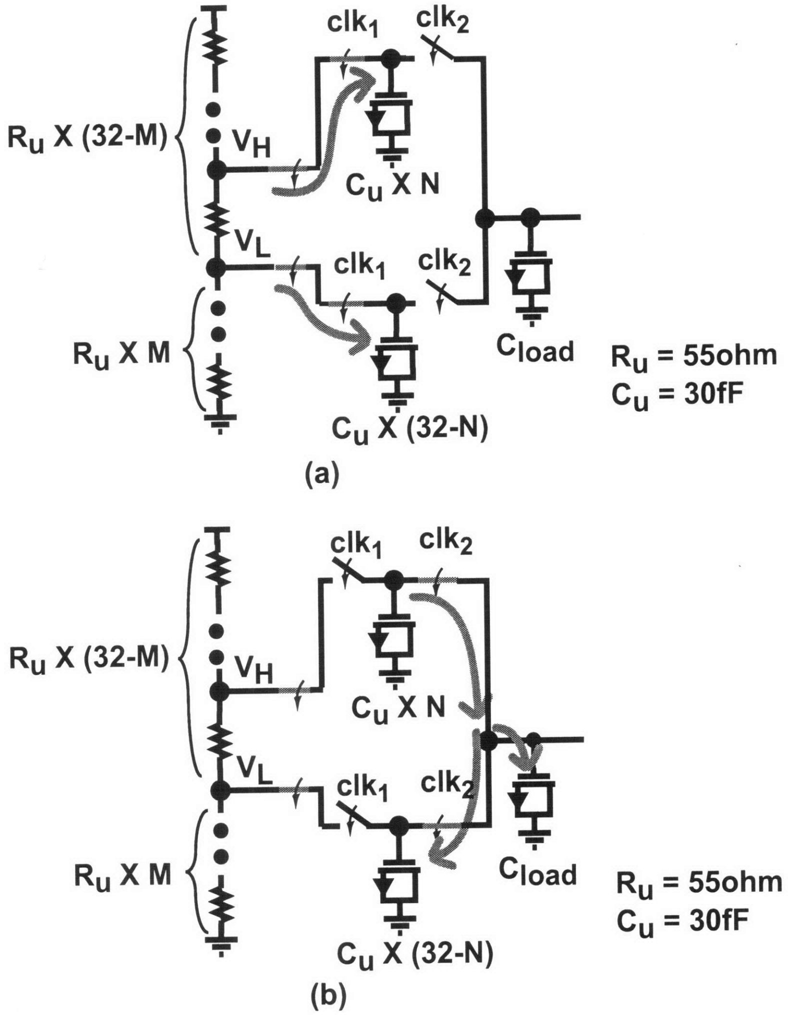


Figure 3-1: DAC operation: (a) step one: the unit capacitors are charged. (b) step two: the charges are redistributed and filtered.

respectively. The value of M is determined by the five MSBs of the 10-bit data, and a decoder is designed to control the switches, according to the value of M.

Each unit cell in the capacitor array consists of a zero-V_t NMOS device as a capacitor and four switches (two S_2 and two S_3). The first two S_2 switches pick up a voltage from V_H or V_L to charge the unit capacitor. According to the five LSBs of the 10-bit incoming data, a thermometer code is generated and used to decide the number of capacitors that are charged to V_H. The other two S_3 switches are controlled by a pair of non-overlapping clocks to achieve the switched-capacitor action. Compared

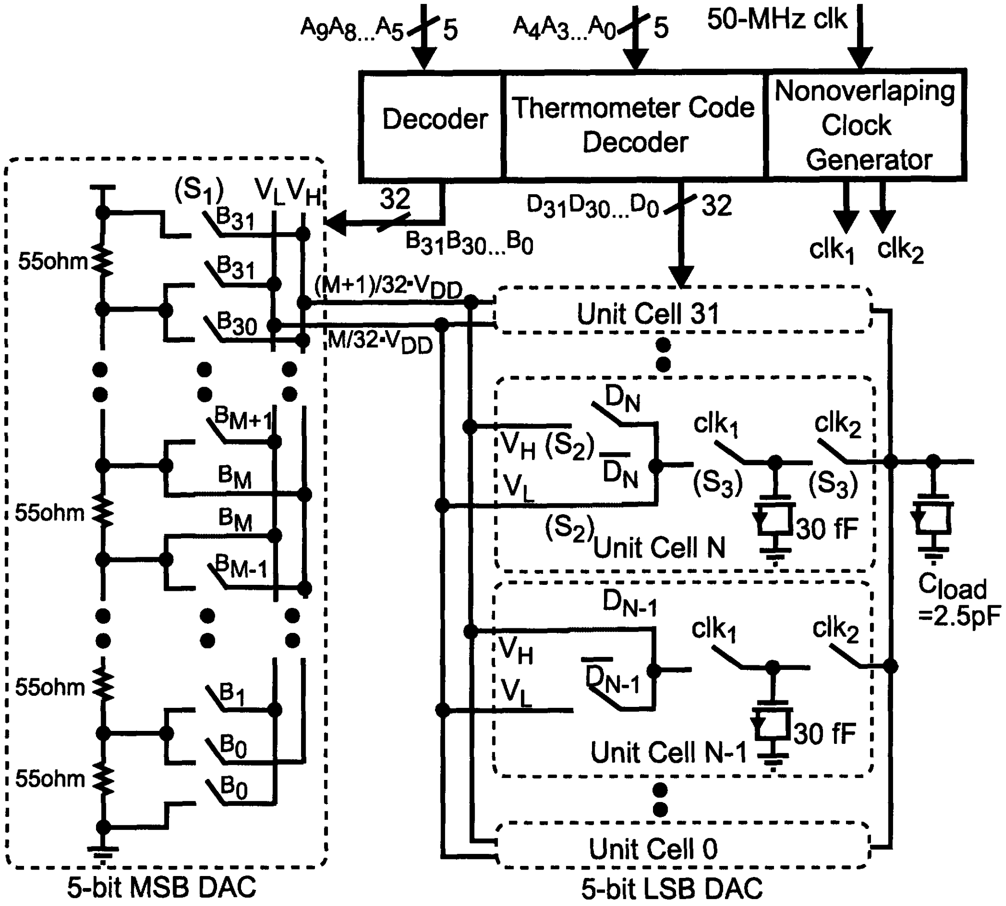
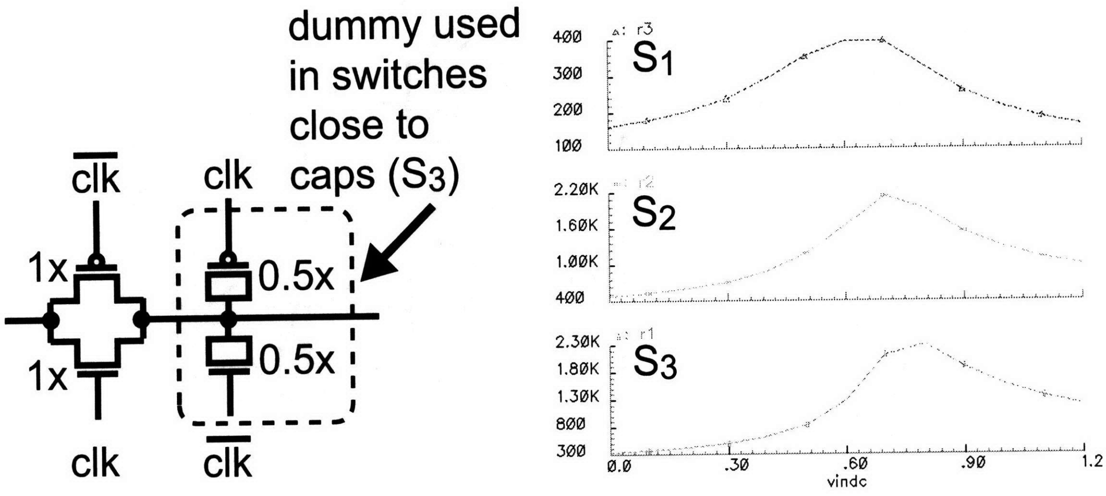


Figure 3-2: Implementation details of the proposed DAC.

to Figure 3-1, although the switches of S_2 in Figure 3-2 are additional, analysis in Section 3.1.3 shows that these extra switches do not adversely impact the settling time. The other way to implement it is to remove the S_3 switch on the left side of each unit capacitor and control the upper and lower S_2 switches with $D_N \cdot clk_1$ and $\bar{D_N} \cdot clk_1$, respectively. Although this way improves the settling time by eliminating one switch, operating on a high-speed clock signal is usually not a good idea due to the resulting complicated design.

The unit resistance R_u and on-resistance of the switches should be designed to be sufficiently low in value such that the top-plate voltages of the unit capacitors can completely settle to V_H and V_L during step one (see Section 3.1.3). Therefore, low-V_t MOS devices are used to implement the switches in order to minimize their on-resistance. All of the switches are composed of a low-V_t NMOS device and a low-V_t PMOS device to reduce the on-resistance over a wide voltage range. The schematic of the three different switches, their simulated on-resistance, and their device sizes are shown in Figure 3-3. It becomes more clear in the next section why the on-resistance of S_2 and S_3 can be relatively larger than that of S_1. Note that there is a trade-off between the value of R_u and the power dissipation of this DAC.



(a) (b)

switch type	W_N (um)	W_P (um)	Max. R_{ON} (ohm)
S_1	2.72	12.24	395
S_2	0.96	1.92	2130
S_3	1.36	1.58	2200

(c)

Figure 3-3: Switch: (a) schematic (b) simulated on-resistance (c) device size.

As for the capacitor array, the unit capacitor size must be chosen to be appropriately large to achieve acceptably low kT/C noise across the full range of the DAC (see Section 3.1.4). To achieve a low area for these capacitors, zero-V_t NMOS capacitors are used for their implementation (W/L = (6 μm)/(0.9 μm) for a 30 fF capacitor). Standard digital logic is used to perform the necessary decoding operations for control of the switch settings for a given input value to the DAC.

One crucial issue for the DAC is to appropriately clock it in a manner that does not introduce fractional spurs into the VCO. A standard clock generator is used to produce the non-overlapping clocks to drive the switches [42], but this generator must be driven by a clock that is synchronous to the VCO. For fractional-N synthesizers, this means that the divider output rather than the reference input must be used as the master clock source (i.e., the *stop* signal in Figure 2-17). The timing diagram of the non-overlapping clocks is shown in Figure 3-4. Note that a sufficiently long delay between *clk* and *clk*1 is created on purpose such that the unit capacitors can begin to be charged only after both decoder outputs are stable.

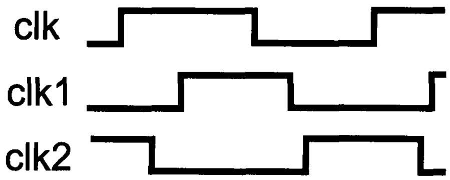


Figure 3-4: Timing diagram of the non-overlapping clocks.

If designed properly, the passive DAC structure supports monotonic operation without the need for any calibration. The key issues in design are to guarantee adequate settling of the resistor ladder to capacitor array voltage transfer as well as to minimize charge injection effects through proper design of the switches. For instance, dummy devices are added in the two S_3 switches to reduce the charge injection and clock feedthrough, as shown in Figure 3-3(a). These issues are commonly understood from the literature [43].

Unfortunately, while monotonic operation is fairly easy to achieve without calibration, the mismatch between the unit resistors and capacitors results in nonlinearity of the DAC transfer function. Since the DAC is driven by a first-order $\Delta\Sigma$ modulator to improve its effective resolution (see Figure 5-9), such nonlinearity may cause noise folding of the $\Delta\Sigma$ quantization noise. Fortunately, the 10-bit resolution offered by the passive DAC limits the magnitude of such noise folding, and the detailed behavioral simulation shows that mismatch with a standard deviation of 5% does not have a significant effect on the overall noise performance of the synthesizer given the coarse/fine-tuning method discussed later in this thesis [44].

3.1.3 Settling Time Calculation

We choose $R_u = 55\ \Omega$ and $C_u = 30$ fF in the implementation. We now check if these values can support a sufficiently short settling time, given the maximum on-resistances of the switches in Figure 3-3. First, a detailed schematic for settling time analysis in step one is redrawn in Figure 3-5. Notice that R_{sw2} counts for the two serial switches (i.e., S_2 and S_3) on the left side of each unit capacitor, so its maximum value is 2.1 kΩ + 2.2 kΩ = 4.3 kΩ , while R_{sw1} is the maximum on-resistance of S_1 (i.e., 395Ω). Before applying the open-circuit time-constant analysis [45], one should notice that the worst time constant should occur when M is around 16 since the output resistance looking back to the resistor ladder is maximized in this case. For simplicity, we can just approximate the output resistance of the resistor ladder to be $16R_u/2 = 8R_u$ to obtain the worst-case time constant. Since there are 32 capacitors in the circuit, open-circuit time-constant analysis suggests

$$\begin{aligned}\tau &= \sum \tau_i \\ &= 32(8R_u + R_{sw1} + R_{sw2})C_u \\ &= 32(8 \cdot 55 + 395 + 4300)30f = 4.93ns \qquad (3.3)\end{aligned}$$

Although it seems that the capacitor voltages cannot settle properly within a half period of 50 MHz (i.e., 10 ns), simulation results below show that this analysis overestimates the time constant.

Interestingly, by simplifying the previous circuit, we can obtain another much smaller time constant. One can argue that the top plates of the upper N unit capacitors can be connected together since they have the same voltage, as illustrated in Figure 3-6. By doing so, N of the resistors R_{sw2} can be considered to be in parallel, which results in a smaller time constant. After applying the same technique to the lower (32-N) capacitors, we can now use the open-circuit time-constant method again

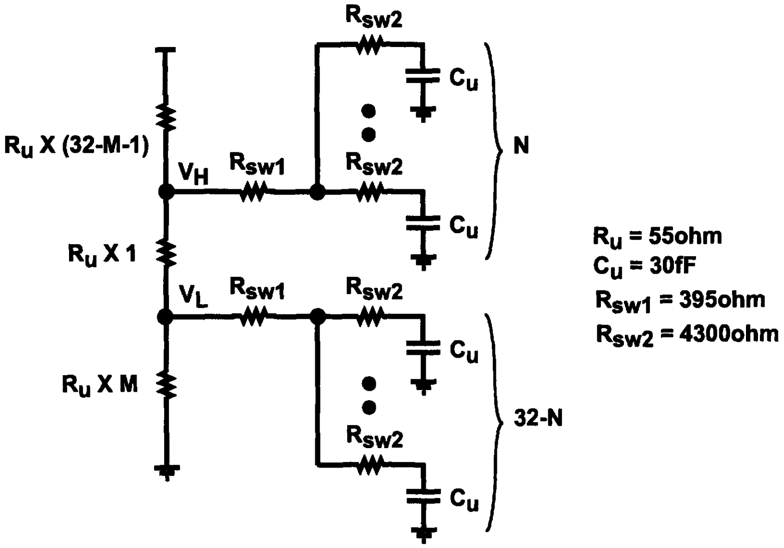


Figure 3-5: DAC schematic for time constant calculation.

to obtain

$$\tau_1 = (8R_u + R_{sw1} + R_{sw2}/N)NC_u = (8R_u + R_{sw1})NC_u + R_{sw2}C_u \quad (3.4)$$

$$\tau_2 = (8R_u + R_{sw1})(32 - N)C_u + R_{sw2}C_u \quad (3.5)$$

$$\begin{aligned} \tau &= \tau_1 + \tau_2 \\ &= (8R_u + R_{sw1})32C_u + 2R_{sw2}C_u \\ &= (8 \cdot 55 + 395)32 \cdot 30fF + 2 \cdot 4300 \cdot 30fF \\ &= 0.8ns + 0.26ns = 1.06ns \end{aligned} \quad (3.6)$$

which is much smaller than the value obtained by using equation 3.3. Thus, the signals have (10 ns)/(1.06 ns) = 9.4 time constants to settle. One should notice that although R_{sw2} is large, the time constant contributed by R_{sw2} is only 0.26/1.06=25% of the overall time constant because of the effect of the parallel resistors. Therefore, we do not need to design extremely small on-resistances for S_2 and S_3. Having small device sizes for S_2 and S_3 not only reduces the DAC area but also alleviates the negative impact of the charge injection from the switches.

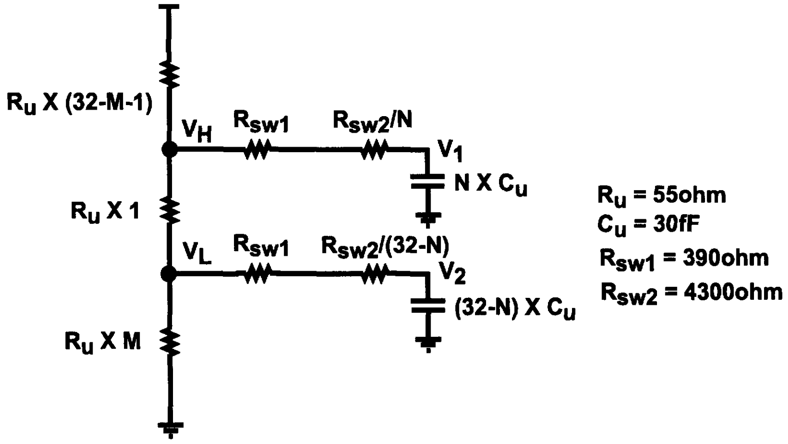


Figure 3-6: Simplified schematic for time constant calculation.

We now verify the above analysis with Spectre simulation. Both circuits in Figures 3-5 and 3-6 are stimulated by a square-wave V_{DD} to observe their transient responses, assuming M=N=16. The first two waveforms in Figure 3-7 correspond to capacitor voltages charged to V_H in Figures 3-5 and 3-6 , respectively. As seen here, the equivalence between both circuits is indeed true since these two waveforms are the same. (They actually overlap each other when being plotted together.) To further extract their time constants, the circuit in Figure 3-8 is also simulated by setting R_1=825Ω , R_2=935Ω , and C_L=1.65pF, such that its response best matches those of Figures 3-5 and 3-6. Therefore, we can conclude that the time constant is approximately $(825\Omega||935\Omega)1.65ps = 0.72ns$, which is even smaller than the value of 1.06 ns calculated with equation 3.6.

Another case where M=16 and N=31 is also verified, and its result is shown in Figure 3-9. Again, V_1 in Figure 3-6 overlaps V_O in Figure 3-8 when R_1=825Ω , R_2=935Ω , and C_L=2.1pF, indicating a time constant of $(825\Omega||935\Omega)2.1pF = 0.92ns$ at this node. As for V_2 in Figure 3-6, which has only one unit cell connected, one can observe that its sharper transition edge is somehow different from the response of a first-order system, but its time constant can still be claimed to be less than 0.86 ns. (Time constant of V_O in Figure 3-8 is chosen to be $(880\Omega||880\Omega)1.9pF = 0.86ns$ in this simulation.)

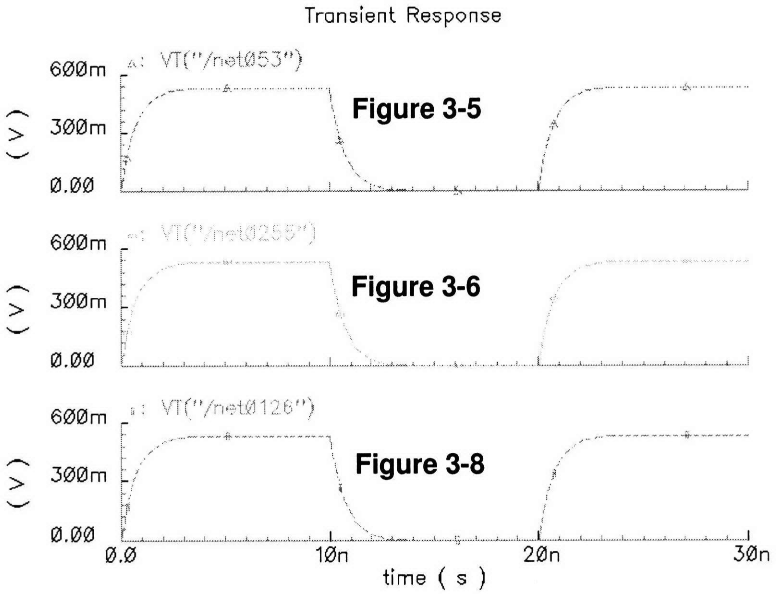


Figure 3-7: Simulated transient responses of Figures 3-5, 3-6, and 3-8 when M=N=16.

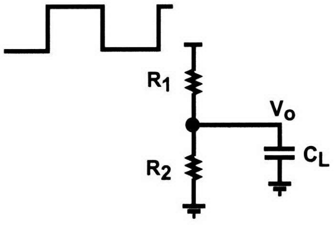


Figure 3-8: Equivalent circuit to extract time constants of Figures 3-5 and 3-6.

To conclude, the time constant of 1.06 ns obtained by applying open-circuit time-constant analysis to Figure 3-6 is sufficiently close to simulation results, and the DAC has 9.4 time constants to settle in step one. Note that parasitic capacitors contributed by the switches and wires are ignored in this analysis. In addition, settling time in step two is shorter than that in step one, according to simulation results.

3.1.4 Noise Calculation

We now calculate the noise spectral density of the DAC, and the results are used in the calculation of the overall PLL noise in Chapter 6. In order to simplify the analysis,

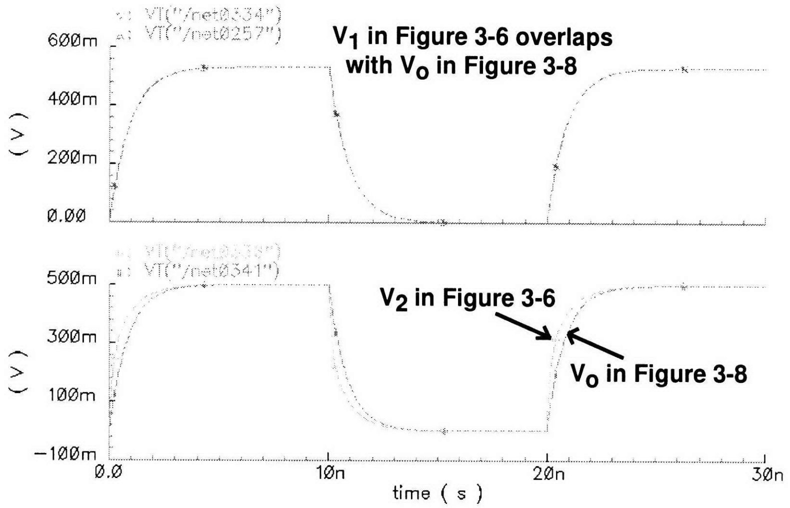


Figure 3-9: Simulated transient responses of Figures 3-6 and 3-8 when M=16 and N=31.

we first assume that all unit-capacitor cells can be merged together, as illustrated in Figure 3-10. R_{on1} and R_{on2} are equivalent on-resistances of the switches, whose values can be approximated by

$$R_{on1} = (R_{s2} + R_{s3})/32 + R_{s1}/2 + R_{DAC} \tag{3.7}$$

$$R_{on2} = (R_{s3})/32 \tag{3.8}$$

where R_{s1}, R_{s2}, and R_{s3} are listed in Figure 3-3, and R_{DAC} is the equivalent output resistance of the resistor ladder. The unit capacitors C_u are multiplied by 32 while the switch on-resistances are divided by 32, since 32 unit cells are in parallel. Note that this modification is verified with Spectre simulation later.

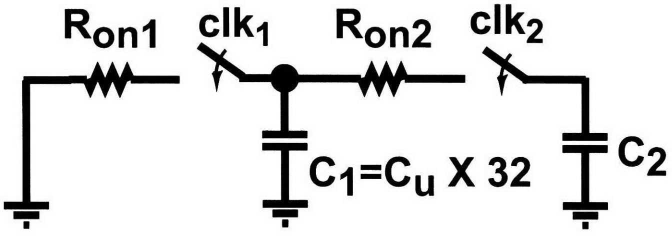


Figure 3-10: Equivalent circuit for noise analysis.

This simplification makes noise analysis of the proposed DAC structure the same as that of a passive switched-capacitor filter, which was derived in [46]. The complete expression of the output noise spectral density in [46] is

$$\begin{aligned} S_{vn}(f) &= A(f)\frac{2kT}{f_cC_1}sinc^2(\frac{\pi f}{f_c}) \\ &\quad +\frac{A(f)}{2}\frac{kT}{f_c\alpha C_1}(cos(\frac{\pi f}{2f_c})cos(\frac{\pi f}{4f_c})+\frac{1}{8}\frac{1}{1+\alpha})\cdot sinc^2(\frac{\pi f}{4f_c}) \\ &\quad +(\frac{1}{1+\alpha})^2\frac{kT}{2}R_{on2} \qquad (3.9) \\ A(f) &= \frac{1}{1+2\alpha(1+\alpha)(1-cos(\frac{2\pi f}{f_c}))} \qquad (3.10) \end{aligned}$$

where f_c is the clock rate of the filter, and α is the capacitor ratio C_2/C_1. Note that, first, these two equations are independent of R_{on1}. In addition, the low-frequency value of the first term is equal to $2kT/(f_cC_1)$, which is exactly the same as the noise density generated by the equivalent resistance $R_{eq} = 1/(f_cC_1)$ of the switched capacitor C_1. We later show that when α is reasonably large, the overall noise density is dominated by the first term, and the double-side noise density $2kTR_{eq}$ can be used as a good approximation of $S_{vn}(f)$ to simplify the analysis.

To demonstrate that the noise density at low frequency can be approximated as $2kT/(f_cC_1)$, the single-side noise density (i.e., $2S_{vn}(f)$) is plotted in Figure 3-11 using $\alpha = 2.5$, $f_c = 50$ MHz, $C_1 = 1$ pF, $R_{on2} = 2200\Omega/32$, all of which correspond to the fine-tuning DAC in the prototype chip. Although α is only 2.5 in this case, we can see that the total noise is indeed dominated by the first term in equation 3.9, and the second term only contributes 10% of the total noise at low frequency. The third term is practically negligible since its value is only 1.1×10^{-20}.

The next step is to develop a simple noise model for the behavior simulation in Chapter 6, since equation 3.9 is unnecessarily complicated to use directly. We propose to filter the equivalent resistor noise $2kTR_{eq}$ using a first-order filter with cutoff frequency $f_p = 1/(2\pi R_{eq}C_2)$ and use the filtered noise as the approximated

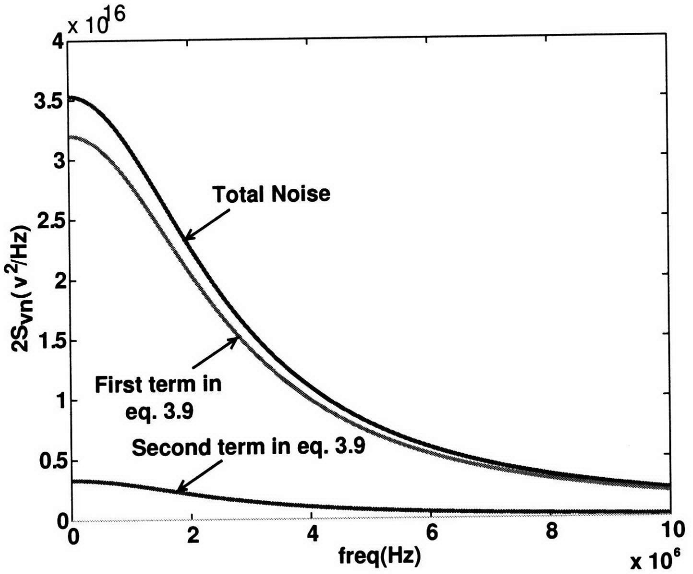


Figure 3-11: Calculated spectral noise density using $\alpha = 2.5$, $f_c = 50$ MHz, $C_1 = 1$ pF, $R_{on2} = 2200ohm/32$.

DAC noise, as described with the following equation:

$$S_{vn}(f) \quad = \quad 2kTR_{eq} \cdot \frac{1}{1 + (\frac{f}{f_p})^2} \tag{3.11}$$

Figure 3-12 compares the calculated value using equation 3.9, approximated value using equation 3.11, and simulated results using the PNoise function in Spectre RF. In this simulation, a simplified schematic of the DAC excluding the decoders in Figure 3-2 is used with M=N=16. In addition, on-resistors with their resistances listed in Figure 3-3 are connected in series with ideal switches; ideal capacitors are used. The result shows that the simulated value matches the calculated value very well. It also verifies that simplifying the proposed DAC structure to a passive switched-capacitor filter for noise analysis is reasonable.

We now investigate another case when α is much larger. Figure 3-13 illustrates the result when $\alpha = 20$. We can see that the approximated and calculated noises are very close in this case because the second term in equation 3.9 becomes very small when α is large. The simulation result is also reasonably close to the calculated value. In

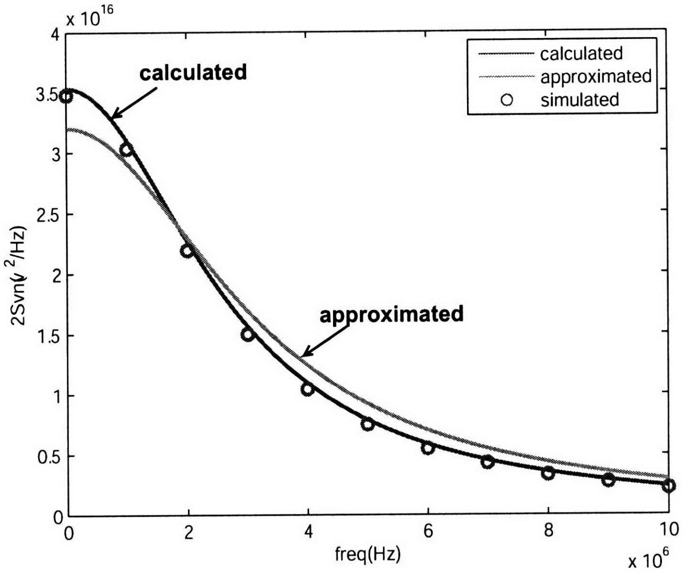


Figure 3-12: Calculated, approximated, and simulated noise densities using $\alpha = 2.5$, $f_c = 50$ MHz, $C_1 = 1$ pF, $R_{on2} = 2200ohm/32$.

addition, compared with Figure 3-12, the bandwidth of this case is reduced because a larger C_2 is used, as indicated by equation 3.2.

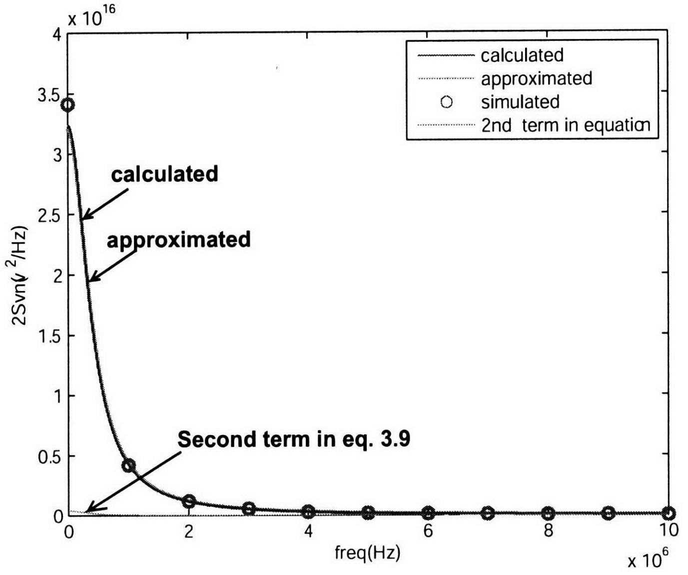


Figure 3-13: Calculated, approximated, and simulated noise densities using $\alpha = 20$, $f_c = 50$ MHz, $C_1 = 1$ pF, $R_{on2} = 2200ohm/32$.

To conclude, we can simplify the noise spectral density of the DAC from equation 3.9 to equation 3.11 without losing much accuracy, especially when α is large. Actually, the two cases in Figures 3-12 and 3-13 correspond to a fine-tuning DAC and a course-tuning DAC, respectively, in the prototype synthesizer. It is shown in Chapter 6 that the thermal noise contributed by the fine-tuning DAC is considerably below other noise sources, thus the 10% error in its noise density caused by using equation 3.11 does not have much negative impact on the accuracy of the predicted PLL noise.

3.2 Hybrid VCO

The hybrid VCO used in the prototype is a well understood structure [37] that consists of a four-bit switched-capacitor network for initial coarse frequency tuning, and two varactors for continuous tuning at coarse and fine levels. The coarse-tuning varactor is added to reduce the necessary resolution of the MIM array.

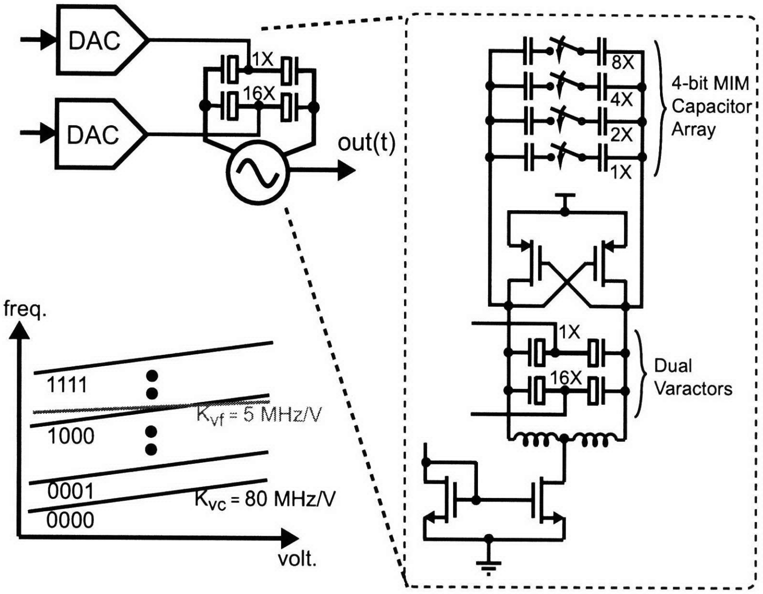


Figure 3-14: Schematic of the hybrid VCO.

A simplified view of the structure is shown in Figure 3-14. The four-bit switched-capacitor network is implemented with MIM capacitors and is tuned by hand in the prototype through a serial interface on the chip to achieve an overall VCO range of

3.15 to 4.25 GHz (see Figure 7-4). Note that the algorithm used to set the four-bit MIM capacitor bank is not included in this thesis.

The coarse- and fine-tuning varactors are accumulation-mode devices (i.e., a NMOS device in a N-WELL, as illustrated in the simplified diagram in Figure 3-15 [47]), with the coarse varactor sized to be 16 times larger than the fine varactor. Therefore, the K_v and tuning range of the coarse varactor is 16 times larger than the fine varactor. The consequence of this difference in K_v is discussed in the next chapter; the idea is to let the coarse varactor provide a moderate frequency range during the frequency acquisition cycle, while the fine varactor is used after the frequency settles to minimize the noise sensitivity.

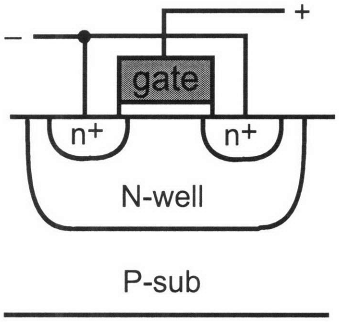


Figure 3-15: Simplified structure of the accumulation-mode varactor.

A center-tapped differential inductor consisting of the two top metal layers in parallel is used to achieve a peak Q factor at 3.6 GHz with a differential inductance value of 1.6 nH, according to the design kit.

Simulated coarse and fine VCO gains are 80 MHz/V and 5 MHz/V, respectively, when the control voltages are set around $V_{DD}/2$. Because of the characteristic of the varactors, VCO gains gradually decrease as control voltages increase (see Figures 7-5 and 7-6). Interestingly, the proposed synthesizer architecture, which utilizes both a coarse varactor and a fine varactor, can tolerate this nonlinear gain because of the high DAC resolution and the coarse/fine-tuning filter design. This point is explained in more detail in Section 5.4 after the filter architecture is presented.

Instead of using the complementary topology [48][49], which consists of both a

NMOS and PMOS cross-coupled pair, the chosen VCO structure utilizes only a PMOS cross-coupled pair to obtain a larger signal swing under a low supply voltage (1.5 V). A PMOS cross-coupled pair rather than a NMOS one is used because of its lower flicker noise in this process. In addition, the topology using a NMOS current source on the bottom and the PMOS cross-coupled pair on the top is chosen, as shown in Figure 3-14. To explain that, in the topologies that place the common-mode voltage of the VCO signals at V_{DD} or ground [49], the signal swing is limited by the maximum $|v_{GB}|$ a device can tolerate. For instance, if the topology utilizing a NMOS tail current and a NMOS cross-coupled pair is used [49] with a supply voltage of 1.5 V and maximum $|v_{GB}|$ of 1.8 V, the amplitude of the single-ended oscillation signal is limited to only 0.3 V, which degrades the phase noise. Instead, the chosen topology allows us to place the common-mode voltage around the middle supply to achieve a higher swing than other topologies without breaking the devices [50][51].

The switch proposed in [4] is used in the MIM capacitor array. The schematic of ths switch (Figure 3-16) and the design considerations, described in [4], are repeated here for convenience: "When VDIG is high, Ma0 is on and the whole cell is on. Transistors Ma3 and Ma4 provide DC bias to ground for the drain and source of Ma0 to ensure minimum on-resistance for Ma0 and thus to maximize the Q when the cell is on. Since these two transistors are used to provide DC bias to the drain and source of Ma0 when the switch is on, the minimum size is used so as not to degrade the tuning ratio. When VDIG is low, Ma0, Ma3, and Ma4 are off and the cell is off. Since the drain and source of Ma0 are floating and due to large signal swing at the VCO outputs, the drain and source of Ma0 can swing below ground and slightly turn on Ma0, which leads to poor Q when the cell is off. Two PMOS transistors Ma1 and Ma2 are added to bias the drain and source of Ma0 to VP to ensure Ma0 is off." In this chip, VP is connected to V_{DD}. More details of the switch design can be found in [4].

Inverter-based buffers are used both between the VCO and the divider and between the VCO and the output pad, as illustrated in Figure 3-17. First, the differential VCO signals are AC coupled to two inverters with resistor feedback. By doing so,

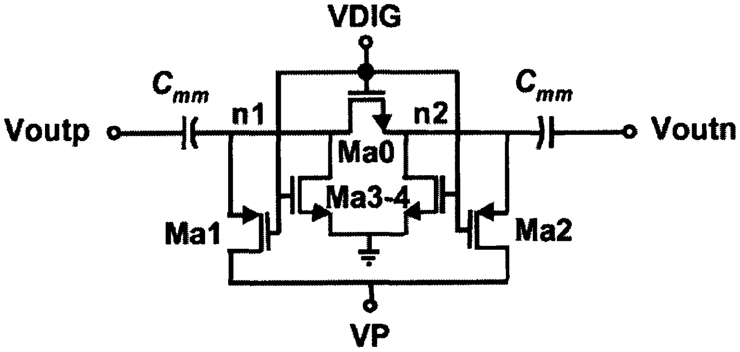


Figure 3-16: The switch structure in [4] is used in the four-bit MIM array.

the duty cycle of the inverted signal and its harmonic contents do not strongly depend on the common-mode voltage of the VCO so that we can have more freedom in choosing the V_{GS} of the cross-coupled pair devices. Another inverter follows each resistor-feedback inverter to complete the first-stage buffer. One of the differential buffered signals is then used to drive the divider, and the other one drives a second-stage buffer with device sizes sufficiently large to drive the 50-Ω load. The V_{DD} of the first-stage buffers are connected to the VCO supply, but a separate V_{DD} pad is reserved for the second-stage buffer to prevent this buffer from disturbing the main VCO supply because of its high current (7 mA).

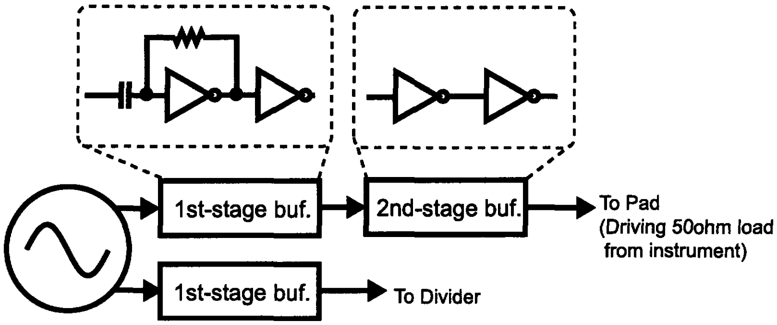


Figure 3-17: Buffers after VCO to drive the divider and the output pad.

3.3 DCO Model

Figure 3-18 illustrates the model of the proposed DCO. The input digital code is first scaled by a factor of $V/2^B$, where V and B denote the power supply voltage and number of bits of the DAC, respectively, to convert the input digital code to a voltage. Another scale factor of T accounts for the DT-CT conversion due to the DAC [35]. In addition, the first-order lowpass pole created by the switched-capacitor structure is described by its equivalent analog response, $H_{LP,f}(s)$. Coarse and fine varactor gains are described by the two integration functions. Approximated noises of the coarse and fine DAC, which are developed in Section 3.1.4, as well as the VCO noise are also included in the model for further noise analysis in Chapter 6.

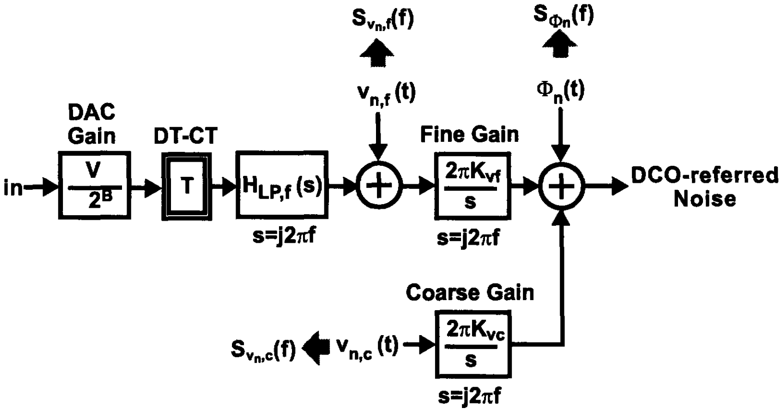


Figure 3-18: Model of the proposed DCO.

According to Section 3.1.4, thermal noises from the fine-tuning and coarse-tuning DACs can be approximated as:

$$S_{v_n,f}(f) = 2 \cdot 2kTR_{eq,f} \cdot \frac{1}{1+(\frac{f}{f_{pf}})^2} \tag{3.12}$$

$$S_{v_n,c}(f) = 2 \cdot 2kTR_{eq,c} \cdot \frac{1}{1+(\frac{f}{f_{pc}})^2} \tag{3.13}$$

where $R_{eq,f}$ and f_{pf} are the equivalent resistance and corner frequency of the fine-tuning DAC, and $R_{eq,c}$ and f_{pc} are those of the coarse-tuning DAC. Note that a scale factor of two is added in each equation to modify it from a double-side density into a single-side one.

3.4 Summary

The case of using a combination of a DAC and hybrid VCO to implement the DCO is considered. We propose an efficient 50-MHz 10-bit passive DAC that requires minimal analog content. A first-order switched-capacitor filter is also embedded in the DAC structure, providing a lowpass pole for the overall PLL. In addition, the hybrid VCO used in this prototype is discussed. Finally, a noise model of the overall DCO is provided.

Chapter 4

Asynchronous Divider

Current digital phase-locked loop (PLL) structures commonly use a synchronous divider with the argument that it has excellent jitter characteristics. Unfortunately, such structures also have relatively high power consumption due to the fact that many elements must be clocked at the highest frequency in the system (i.e., frequency of the voltage-controlled oscillator (VCO)) [9].

In this chapter, we propose an asynchronous divider structure that has low power consumption while still maintaining excellent noise performance. The time-to-digital converter (TDC) unwrapping function and offset control are also discussed in the end of this chapter.

4.1 Asynchronous, Low-Jitter Divider

For classical analog fractional-N synthesizers, it is common to use an asynchronous divider structure [5] due to its low power and compact layout. The low power is achieved by operating only a small portion of the structure at the highest frequency. As shown in Figure 4-1, application of this structure to a digital fractional-N synthesizer is straightforward in principle. However, the key issue arising is that the gated ring oscillator TDC must support a very large time range during locking since the phase error can span the entire reference period. Because the nominal phase range required after the PLL is locked is much smaller than the reference period (see Section

4.3), this constraint can lead to wasted power and area in the GRO to support such a wide range that is only briefly utilized during locking. In addition, note that the phase error seen by a detector can even be larger than 2π or become negative during frequency acquisition, as illustrated in Figure 4-2. In order to have a well-defined TDC measurement each reference cycle, only one reference edge and one divider edge are desired in each period. When a phase error is larger than 2π (Figure 4-2(a)), the divider edge is missing between two reference edges such that the TDC output is not defined in this period. When a negative phase error follows a positive phase error (Figure 4-2(b)), two divider edges occur in one period such that the phase error between the second divider edge and the next reference edge cannot be measured. The reason for the latter problem is that a TDC cannot handle a negative phase error directly. These issues make the application of a TDC to the classical fractional-N architecture complicated in practice. Some previous works thus combined a classical phase detector and a TDC [12], but this solution ruined the beauty of a digital PLL structure.

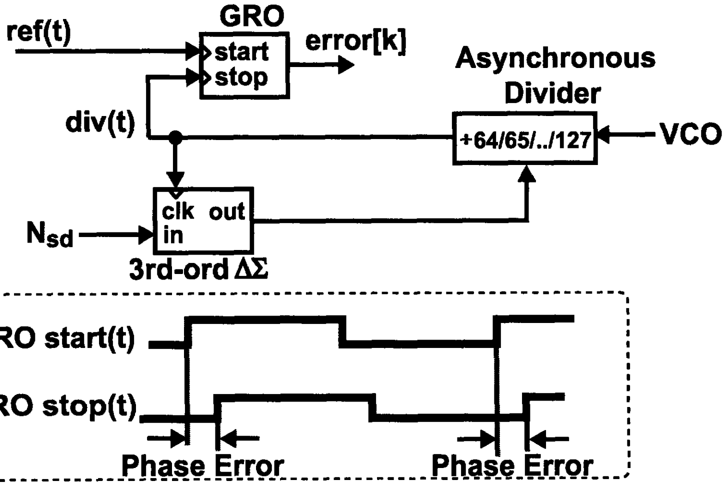


Figure 4-1: Classical approach to using an asynchronous divider in a digital fractional-N PLL.

Also, a subtle issue with asynchronous divider structures is that the delay from input to output can shift slightly as a function of the divide value because of the varying internal-node capacitors. This leads to additional jitter when dynamically varying

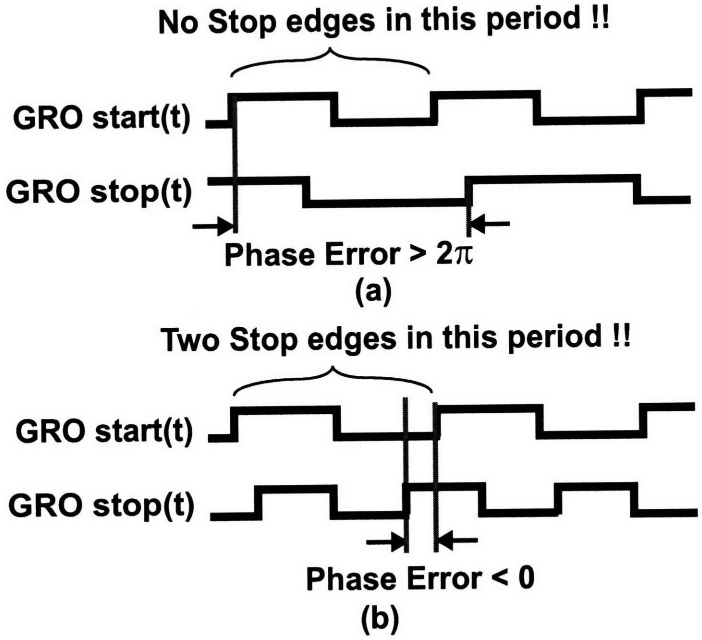


Figure 4-2: The GRO output is not well-defined when one reference cycle includes (a) no *stop* edge (b) two *stop* edges.

the divide value according to a $\Delta\Sigma$ modulator in a fractional-N PLL [22][24][26][27]. The common approach to dealing with such delay variation is to re-clock the divider output with a register that is timed by the VCO output, but this approach is costly in power and also opens the door to metastability problems [24]. To explain that, the possible divider delay may vary over a wide range due to the process, voltage, and temperature variations. When the divider output edge is too close to the VCO sampling edge, the metastability issue arises, making the design of the re-timing circuit complicated. Therefore, the goal of the proposed divider structure is to eliminate the divide-value delay variation without a complicated re-timing circuit.

4.1.1 Divider Operation

We propose a very simple divider modification that alleviates both of the issues described above. As shown in Figure 4-3, the proposed structure reduces the divide value range of the core asynchronous divider such that the nominal frequency of the core output $div(t)$ becomes four times the reference frequency. Since four divider edges exist within each reference period in this case, the signal *div*(t) cannot be used for phase comparison directly. By using the core divider output to re-time the reference (i.e., shown as the re-timing flip-flop in the figure), the effective divider output impacting the GRO TDC has the same frequency as the reference. In addition, it can be guaranteed that only one *stop*(t) edge is allowed to occur every reference cycle such that the TDC output is always well-defined. This re-timing technique has similarities to a technique proposed in [9], but has the advantage of much fewer components operating at high frequencies. By multiplexing a series of four divide values into the divider each reference period, the effective divide value N becomes the sum of those values (i.e., $N=N_0+N_1+N_2+N_3$). Note that only one of the divide values needs to be dithered by the $\Delta\Sigma$ modulator (i.e., N_2 in this example), whereas the rest can be kept at static values that are chosen according to the desired output frequency of the synthesizer. Whenever the carrier frequency changes, the values of N_0, N_1, and N_3 are set through a shift register together with N_{sd}, which contains both an integer value and the fractional value for the $\Delta\Sigma$ modulator.

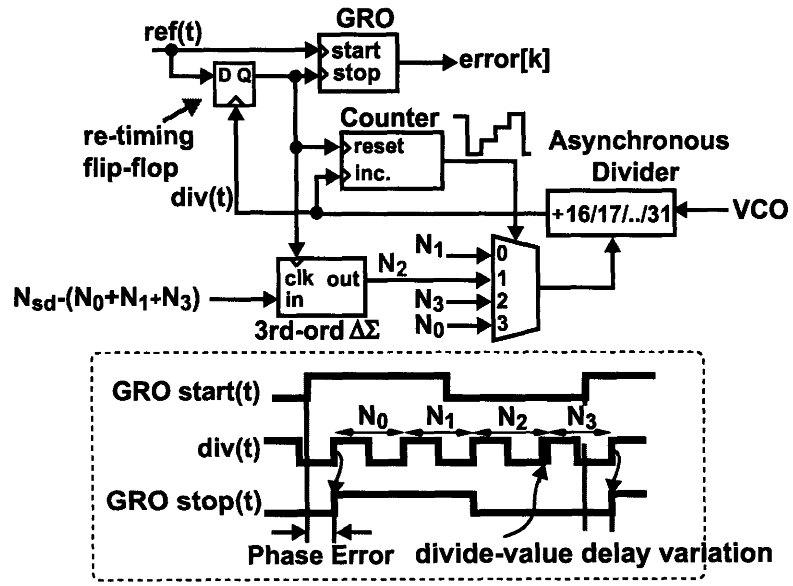


Figure 4-3: Proposed asynchronous divider structure achieving low power and jitter.

To explain the advantage offered by the proposed divider structure with respect to the TDC range, consider the fact that since the re-timing flip-flop (shown in Figure

4-3) is clocked at four times the reference frequency, the maximum time span seen by the TDC is 1/4 the reference period rather than the full reference period. The factor-of-four reduction in the phase range eases the dynamic range requirements of the TDC phase detector. In the case where the actual phase error exceeds the TDC range, it is a simple manner to keep track of the resulting cycle slips such that a net "unwrapped" phase is computed, as described in Section 4.2. Once the PLL is locked, such cycle slipping disappears and a TDC range of one fourth the reference period is more than adequate to track the PLL phase error (see Section 4.3). In the prototype presented here, the required TDC range becomes $1/(50\text{ MHz})/4 = 5$ ns, leading to an 11-bit GRO implementation given that the raw resolution of the GRO is 6 ps.

As for the advantage offered with respect to the divider delay variation, note that only one of the four edges of the core asynchronous divider output has an impact on the TDC each reference period. By choosing the divide value associated with that key TDC edge to be constant, the corresponding core divider delay from input to output is also constant for that key edge (ignoring the thermal noise effects). Therefore, if we simply choose the $\Delta\Sigma$ modulated divide value to control any of the other three core divider edges not corresponding to the key edge that impacts the TDC, we can avoid variation in the timing of that key edge due to the $\Delta\Sigma$ divide value variation. As shown in the figure, we choose N_2 to be the divide value controlled by the $\Delta\Sigma$ modulator. The divider structure therefore avoids the divide-value-dependent jitter due to the $\Delta\Sigma$ dithering without re-timing the divider output using the VCO edge. Simulation results are presented in the next section to support this idea.

4.1.2 Implementation Details

The divide-by-16-to-31 divider core shown in Figure 4-4 is based on [5]. The beauty of this topology is the simple design resulting from the modular structure. Instead of using six divide-by-two/three stages to achieve the necessary divide-by-64-to-127 range as usual, the proposed divider uses only four stages to perform divide-by-16-to-31 each time but needs four division cycles to complete the overall divide-by-64-to-127 action, as explained in Section 4.1.1.

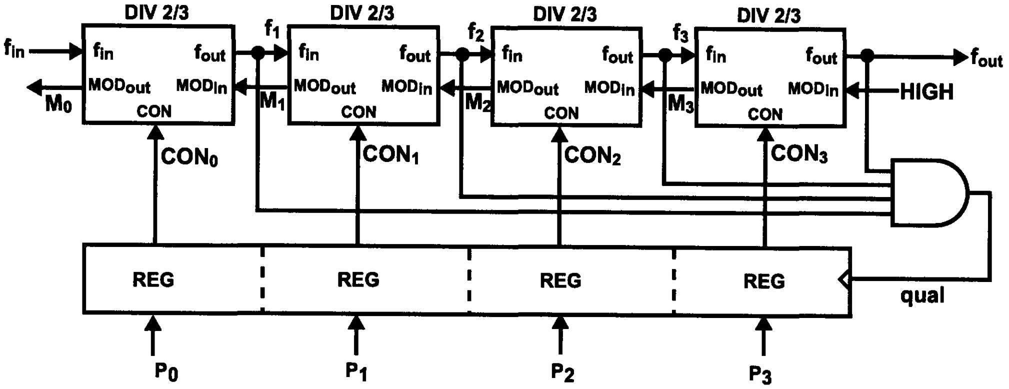


Figure 4-4: Schematic of the modular divider structure.

Different from the original implementation based on the current-mode logic in [5], the implementation of the asynchronous divider in this prototype is achieved with the full-swing TSPC logic in order to save the chip area and power dissipation [52]. Figures 4-5 and 4-6 illustrate the original divide-by-two/three cell in [5] and the modified TSPC version used in this chip, respectively. The upper two latches in Figure 4-5 are combined as a TSPC DFF, and the lower two are implemented as P-type and N-type latches, respectively. Notice that the necessary logic functions (i.e., AND) are merged with the DFF or latch with the shaded devices in Figure 4-6 to reduce the propagation delay. The resulting savings in area and power are huge since no resistors and current sources are needed in a TSPC implementation. The average current dissipation of the overall divider is only 1 mA in 0.13-μm CMOS when the VCO frequency is 3.6 GHz. Simulation indicates that the divider can operate up to 5 GHz. Note that this design is negative-edge triggered.

A detailed timing diagram of the core divider in Figure 4-4 is illustrated in Figure 4-7 to better understand this divider structure. First, the divide ratio is

$$N = 16 + CON_0 \cdot 2^0 + CON_1 \cdot 2^1 + CON_2 \cdot 2^2 + CON_3 \cdot 2^3 \tag{4.1}$$

When N is 16 ($CON_3CON_2CON_1CON_0$=0000), each cell divides the incoming frequency by two, and the action of divide-by-16 is completed at the 17th falling edge of f_{in}. However, when we increase the divide value by setting $CON_3CON_2CON_1CON_0$

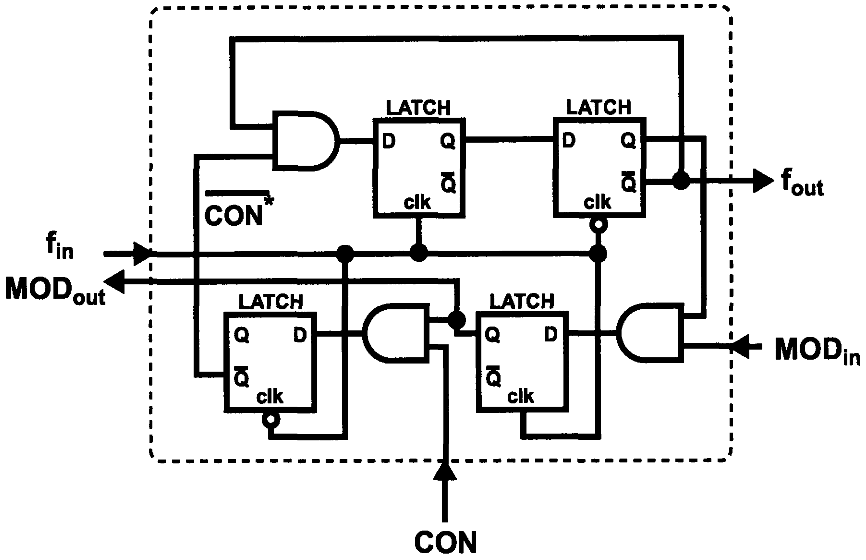


Figure 4-5: Schematic of the divide-by-two/three stage in [5].

to a value other than 0000, each cell is designed to perform divide-by-three once within the divide-by-N cycle if its CON signal is set to one. As a result, the cells can extend the divider output period by 2^0, 2^1, 2^2, and 2^3 VCO periods, respectively, to achieve the goal of divide-by-N, as shown in Figure 4-7(a). Notice that Figure 4-7(a) illustrates the case of divide-by-31. To illustrate the timing diagram for an N value other than 31, one can simply remove $region_i$ from Figure 4-7(a) if CON_i is zero.

Each divide-by-two/three stage can perform divide-by-three only once in each divide-by-N action because a second control signal MOD_{in} is added in each stage, in addition to CON. To explain that, this design allows CON_i to be loaded to $\overline{CON_i^*}$ only when MOD_i is HIGH, as illustrated in Figure 4-7(b) and (c). One can see that the MOD signal (i.e., M_i) propagates from the last stage to the first stage gradually, so there is only one chance for every stage to perform divide-by-three when $\overline{CON_i^*}$ is LOW. If CON_i is set to zero, the corresponding $\overline{CON_i^*}$ keeps at HIGH, so no divide-by-three action can occur in that stage, and $region_i$ in Figure 4-7(a) should be removed.

When designing a divider for a fractional-N PLL, one critical thing is how to update the divide value every reference cycle without causing any operation error

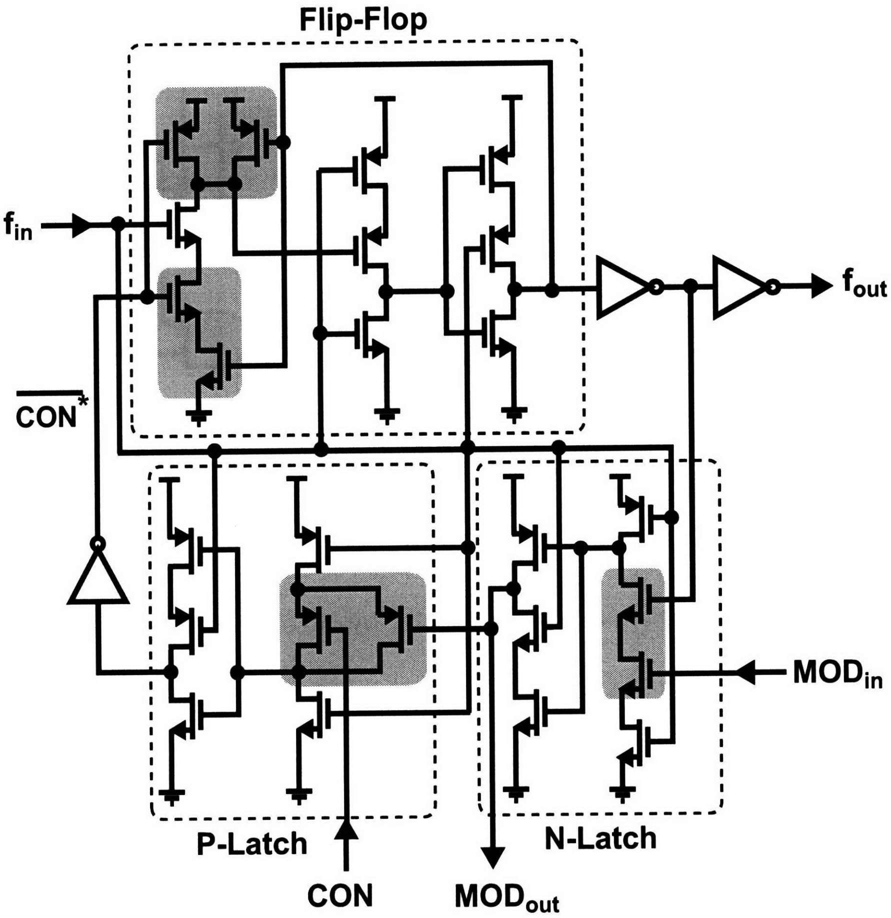


Figure 4-6: TSPC implementation of the divide-by-two/three stage.

[53]. With the help of Figure 4-7, it is clear that we can safely update CON_i once $\overline{CON}_i^*$ becomes LOW since the divide-by-three action is already launched. In this prototype, a simple control qualifier design, as shown in Figure 4-4, is achieved by gating the incoming divide value $P_n[k]$, which is triggered by the negative edge of f_{out}, with a register that is triggered by the result of $f_1 \cdot f_2 \cdot f_3 \cdot f_{out}$. As illustrated in Figure 4-7(d), the first positive *qual* edge does not occur until all $\overline{CON}_i^*$ become LOW. Thus, the divider core cannot see the updated divide value until the operation of divide-by-N is completely launched. Notice that although there are another three positive *qual* edges in Figure 4-7(d) due to this simple control qualifier design, the operation is not ruined since $P_3P_2P_1P_0$ cannot change during this period.

Figure 4-8 illustrates a more detailed schematic of the overall divider architecture.

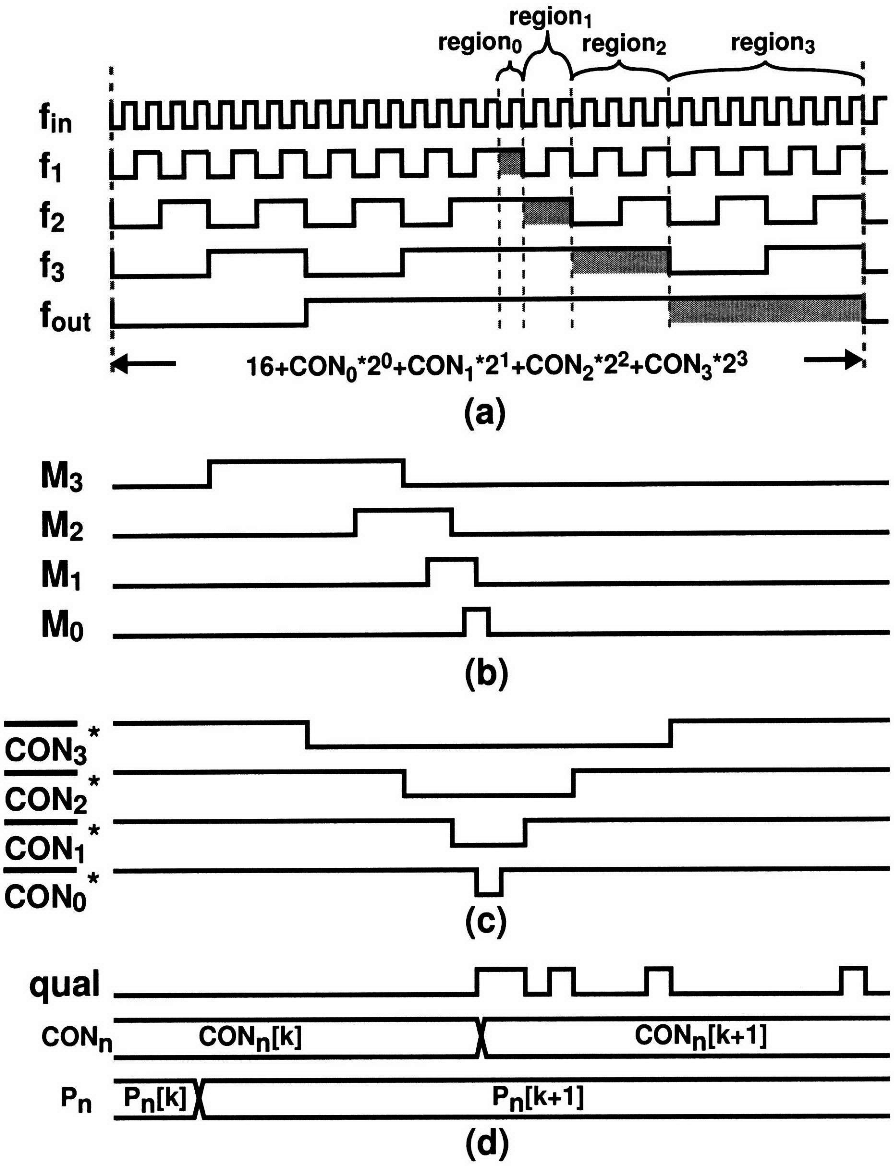


Figure 4-7: Timing diagram of the divide-by-16-to-31 divider (a)main signals (b)MOD signals (c)$\overline{CON}_i^*$ (d)control qualifer.

The difference between Figures 4-8 and 4-3 explains the timing details and the way to reset the counter that is used to control the multiplexer as well as to unwrap the GRO output. First, since the divider is negative-edge triggered, we use the negative node of the VCO differential signals to drive the divider (the positive node drives the pad), and add an inverter at the divider output to let each $div(t)$ cycle begin with a positive edge. Again, $div(t)$ samples the $ref(t)$ using a re-timing flip-flop with the

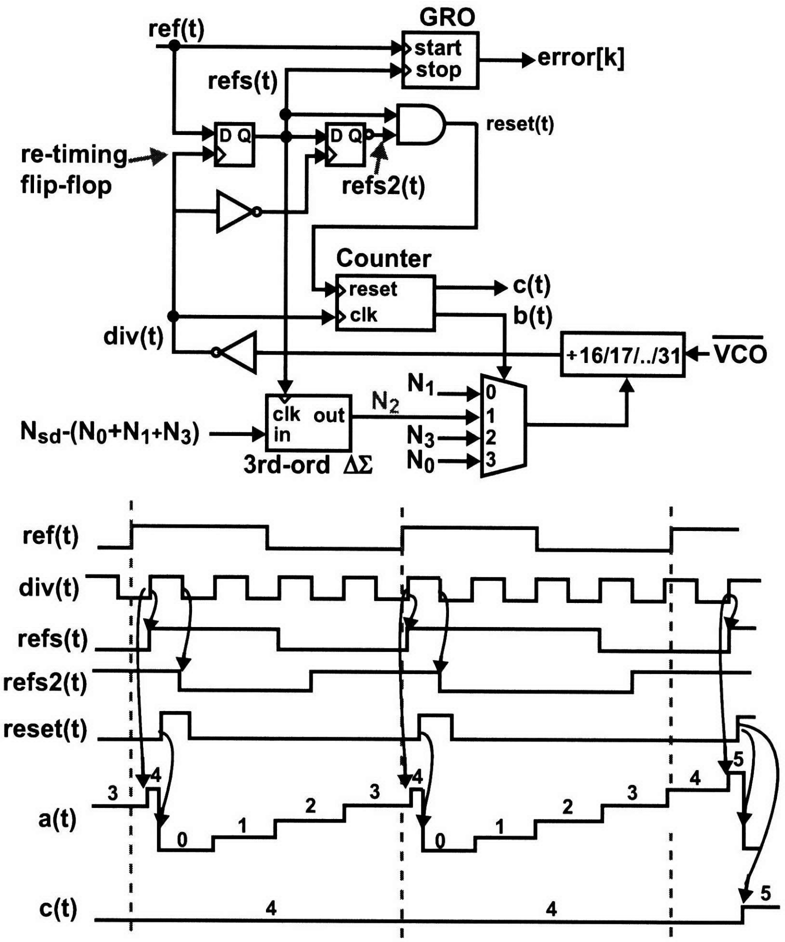


Figure 4-8: Detailed schematic of the proposed divider structure.

output labeled as $refs(t)$. The realigned reference signal $refs(t)$ is not only used as a disable signal of the GRO, but it also triggers the third-order $\Delta\Sigma$ modulator as well as the rest of the system (i.e., noise cancellation, loop filter, DAC). By doing so, the whole system is synchronous to the VCO edge.

The signal $div(t)$ also triggers a three-bit counter whose schematic is shown in Figure 4-9. There are two outputs in this counter. One output $b(t)$, which is the two LSBs of $a(t)$ and is triggered by the $div(t)$ edge, is used to control the 4-to-1 multiplexer such that we can sequentially choose a divide value from N_0 to N_3 every reference cycle. The other output of the counter $c(t)$ is triggered by a $reset(t)$ signal that occurs only once every reference cycle. The $reset(t)$ pulse is the result of $refs(t) \cdot refs2(t)$, where $refs2(t)$ is an inverted, sampled version of $refs(t)$. Whenever the *reset*(t) edge occurs, the instantaneous value at $a(t)$ is loaded to $c(t)$. This value of $c(t)$ indicates the number of positive $div(t)$ edges within every reference cycle (i.e., the number of $div(t)$ cycles in each $refs(t)$ cycle). In the steady state, the value of $c(t)$ should always be four. However, during the initial frequency acquisition cycle, $c(t)$ can be five or three when the VCO frequency is too high or too low, respectively. The timing diagram in Figure 4-8 illustrates the case when the VCO frequency is slightly higher such that $c(t)$ is four in the first cycle and five in the second cycle. Notice that $a(t)$ also increases by one at every $refs(t)$ edge but is reset right after the $reset(t)$ edge occurs. The time delay between $div(t)$ and $reset(t)$ allows $c(t)$ to sample the increasing $a(t)$ value. Resetting the value of $a(t)$ from four or five to zero during the divide cycle in Figure 4-8 does not impact the divider core because of the gating function created in Figure 4-4. The value of $c(t)$ is then used to unwrap the GRO output phase such that an almost linear TDC transfer function can be achieved to avoid the cycle slipping, shortening the PLL locking time. The circuit that unwraps the GRO output is described in Section 4.2.

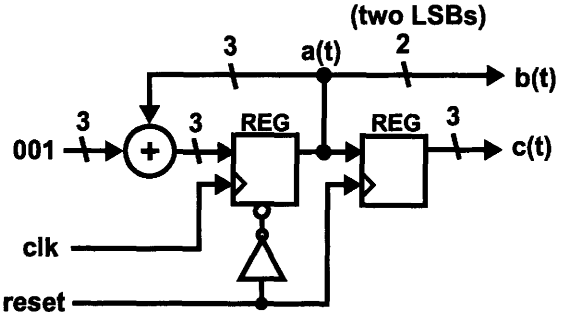


Figure 4-9: A three-bit counter used to control the multiplexer and to record the number of divider edges.

We now check the issue of the divide-value-dependent delay in an asynchronous divider with Spectre simulation and then demonstrate how the proposed structure improves it. First, we examine how the input-output delay changes in the core divider (Figure 4-4) by sweeping the divide value from 16 to 31. The variable delay can be observed in the eye-diagram of the divider output, as illustrated in Figure 4-10, and the resulting pk-pk jitter is about 2 ps at the negative edge. This experiment verifies that the divide-value-dependent delay indeed exists. Then, we simulate the proposed divider (Figure 4-8) with four different setups. In each case, we select one of the four divide values to toggle between 19 and 21 and set the other three divide values to 20. By doing so, the effective divide value (i.e., $N_0+N_1+N_2+N_3$) of these four simulations are identical, but we can see the negative impact of the divide-value-dependent delay when N_3 is dithered. As shown in Figure 4-11(c), when N_3 toggles between 19 and 21, we can clearly see two different rising edges caused by two different delay values (separate by 0.7 ps) at the re-timing flip-flop output (i.e., *refs*(t) in Figure 4-8). As for the other three cases, we can only see one rising edge caused by the divide value of 20. This result is provided here as evidence that the proposed divider structure can avoid the divide-value-dependent delay (jitter) without requiring re-timing.

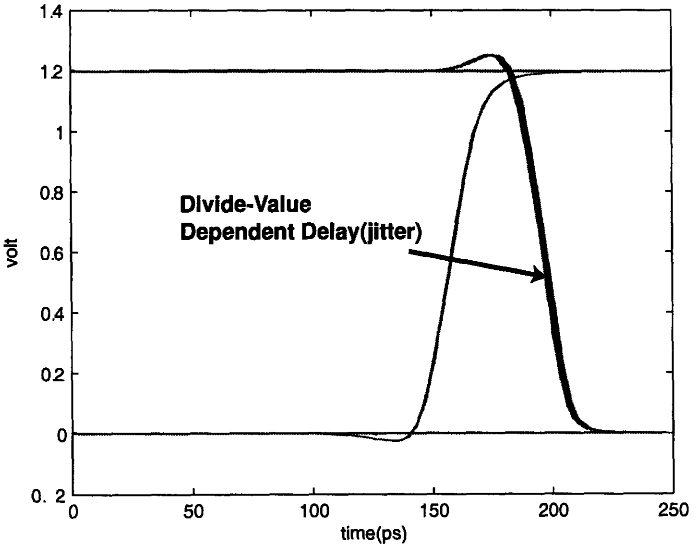


Figure 4-10: Simulated jitter of the divide-by-16-to-32 divider.

4.2 The TDC Unwrapping Function

We now continue the discussion of "unwrapping" the TDC phase. First, Figure 4-12 explains the meaning of "phase wrapping" and "phase unwrapping". In this figure, we assume that the frequency of *stop* is smaller than that of *start* in the beginning,

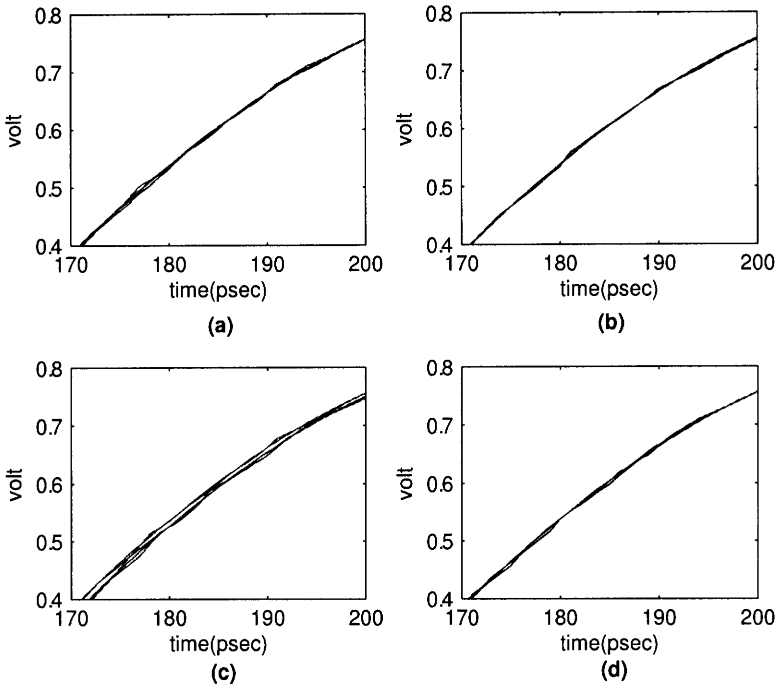


Figure 4-11: Simulated jitter of the resampled reference clock: (a)N_1 toggles between 19 and 21, $N_2 = N_3 = N_0 = 20$ (b)N_2 toggles between 19 and 21, $N_1 = N_3 = N_0 = 20$(c)N_3 toggles between 19 and 21, $N_1 = N_2 = N_0 = 20$ (d)N_0 toggles between 19 and 21, $N_1 = N_2 = N_3 = 20$.

so the phase error detected by the TDC (i.e., $g[k]$ or $m[k]$) keeps increasing until only three *div*(t) edges occur in one *ref*(t) cycle, as shown in Figure 4-13(a). What happens at this point is that the TDC underestimates the phase error because it cannot see the phase error corresponding to the fourth *div*(t) cycle that occurs after the *refs*(t) edge. The result is that $g[k]$ and $m[k]$ "wrap," as illustrated in Figure 4-12. This phenomena is similar to the cycle-slipping effect in a tri-state PFD, which leads to a much longer locking time than a linear system can achieve [53]. Therefore, the idea is to switch in an offset *unwrap*$[k]$ to eliminate the negative effect of phase wrapping, as illustrated in Figure 4-12. With the phase unwrapping function, the net phase error $u[k]$ seen by the loop filter can keep increasing even if it is larger than $\pi/2$ (i.e., one-fourth of the reference period). In the other case where $f_{stop} > f_{start}$, phase error keeps decreasing. At some point, there can be five *div*(t) edges occurring in one *ref*(t) cycle, as shown in Figure 4-13(b). The problem now is that instead of seeing the real phase error that is negative, TDC sees the time difference between the *ref*(t) edge and the *next refs*(t) edge. Again, the TDC output wraps, so we need to switch in a negative step such that the net phase error can decrease seamlessly.

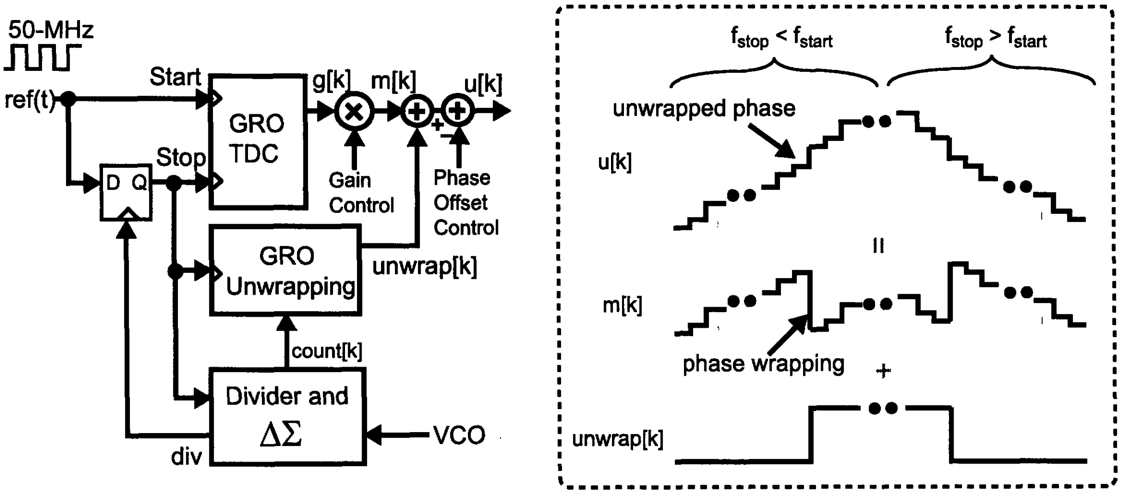


Figure 4-12: A phase unwrapping function eliminates the phase wrapping at the TDC output.

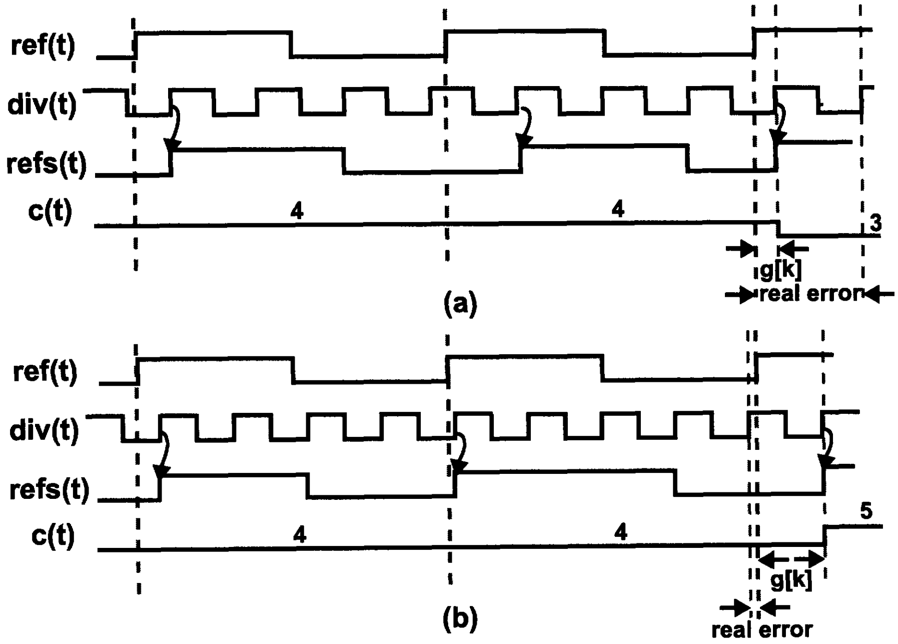


Figure 4-13: Timing diagram when phase wrapping occurs (a)$f_{stop} < f_{start}$ (b)$f_{stop} > f_{start}$.

The unwrapping function in this prototype is implemented as in Figure 4-14. The counter output $c[k]$ in Figure 4-9 is leveraged to indicate whether phase wrapping occurs. As shown in Figure 4-14, this number compares with four, and their difference is scaled and accumulated to generate $unwrap[k]$. Therefore, when no phase wrapping occurs, $c[k]$ is four and nothing is accumulated. However, when $c[k]$ is different from four because of phase wrapping, the deviation from four accumulates to provide an offset value to the TDC output. A six-bit multiplier scales the deviation value before it is accumulated such that the step $unwrap[k]$ provides can roughly match the step in the wrapped phase signal (i.e., $m[k]$). Notice that since phase wrapping can only occur during frequency acquisition, accurate cancellation of this step is not necessary. The scale factor is programmable through the shift register in the prototype but is set to a fixed value during the measurement.

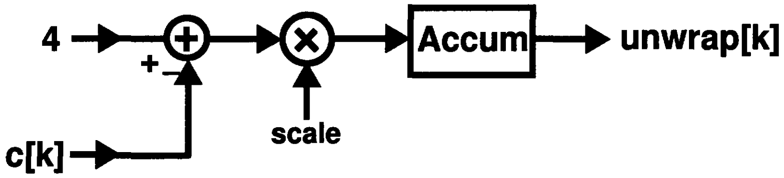


Figure 4-14: Implementation of the TDC unwrapping function.

4.3 TDC Offset

Since a TDC cannot handle a negative time difference directly, the average time difference between the GRO *start* and *stop* edges needs to be biased to a certain offset after the PLL locks, as illustrated in Figure 4-15.

On the one hand, the offset value must be large enough, such that the *stop* edge never leads the *start* edge to guarantee a positive time difference in the steady state. On the other hand, a small offset value reduces the duty cycle of the GRO, resulting in lower power and noise. As shown in Figure 2-17, the offset value, which is programmable through a shift register in the prototype, is controlled by a subtractor. In the measurement results in Chapter 7, this offset value is set to around 1.2 ns.

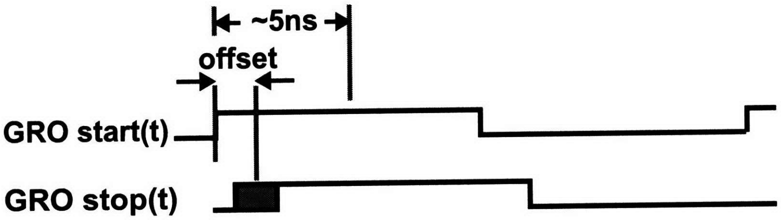


Figure 4-15: The time difference between the GRO *start* and *stop* edges is biased to an offset value in the steady state.

Figure 4-15 also explains why the divider output frequency is chosen to be about four times the reference frequency. By doing so, the *stop* edge can move safely without causing cycle slipping in the steady state. There might be insufficient margin if the divider output frequency is higher than four times the reference frequency. On the other hand, setting divider output frequency to be only two times the reference frequency increases the necessary TDC range during frequency acquisition.

Notice that if direct modulation is built into the synthesizer, the deviation in the location of the *stop* edge may become larger. Careful investigation with detailed behavior simulation is necessary in that case to guarantee correct operation.

4.4 Summary

We propose an asynchronous divider structure that has low power consumption while still maintaining excellent noise performance by avoiding the divide-value-dependent delay in an asynchronous divider. In addition, this divider also lowers the required TDC range by a factor of four. The TDC unwrapping function and offset control are also discussed.

Chapter 5

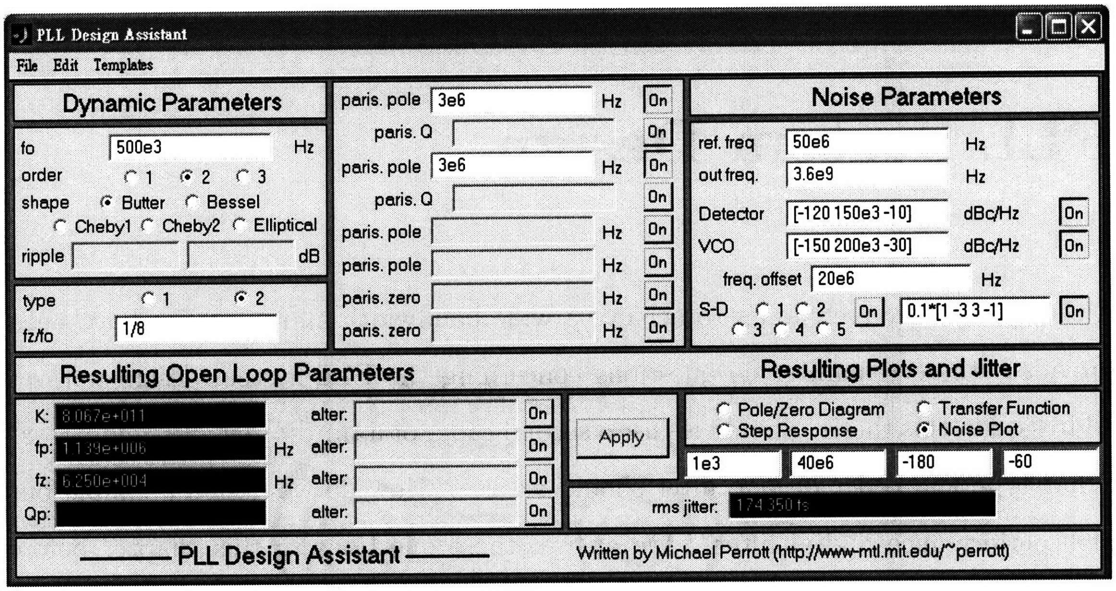
PLL System Design

Although the key techniques to achieve a wide bandwidth and low noise have been provided, there remain some questions concerning the PLL system design not yet addressed in this thesis. Should we use a second-order or a third-order $\Delta\Sigma$ modulator? What type and order of loop filter should we use? How do we design a simple but high-performance digital filter? We answer these questions in this chapter before moving on to the noise analysis and measurement results in the next two chapters.

5.1 System Design Using PLL Design Assistant

In this section, we use a tool, PLL Design Assistant [54][55], to investigate the noise performance of the proposed techniques quickly. Based on these results, details about how to choose the $\Delta\Sigma$ modulator order, PLL type, and PLL order are discussed in the next few sections.

Figure 5-1 illustrates the parameters assumed in this analysis. Key parameters are: f_{ref} = 50 MHz, f_{vco} = 3.6 GHz, f_o = 500 kHz, $order$ = 2, f_z/f_o = 1/8. In addition, the VCO phase noise is assumed to be -150 dBc/Hz at 20 MHz offset with flicker corner frequency of 200 kHz. The GRO TDC noise is assumed to have a -120 dBc/Hz floor with a -10 dB/dec roll-off at low frequency offsets. Although the TDC quantization noise is ignored in the analysis in this section, it is included in further analysis in Chapter 6. Furthermore, a third-order $\Delta\Sigma$ modulator is chosen, and we assume 10% of its quantization noise is left after the noise cancellation is performed. Note that although a high reference frequency of 50 MHz is chosen to lower the quantization noises of the TDC, DAC, and divider in this system, the analysis in Section 6.2.2 and the measurement results in Section 7.3 show that the proposed synthesizer can also utilize a lower reference frequency.



fo 500e3 Hz
order ○ 1 ● 2 ○ 3
shape ● Butter ○ Bessel ○ Cheby1 ○ Cheby2 ○ Elliptical
ripple dB
type ○ 1 ● 2
fz/fo 1/8
paris. pole 3e6 Hz
paris. Q
paris. pole 3e6 Hz
paris. Q
paris. pole Hz
paris. pole Hz
paris. zero Hz
paris. zero Hz
ref. freq 50e6 Hz
out freq. 3.6e9 Hz
Detector [-120 150e3 -10] dBc/Hz
VCO [-150 200e3 -30] dBc/Hz
freq. offset 20e6 Hz
S-D ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 0.1*[1 -3 3 -1]
K: 8.067e+011
fp: 1.139e+006 Hz
fz: 6.250e+004 Hz
Qp:
alter:
alter:
alter:
alter:
○ Pole/Zero Diagram ○ Step Response ○ Transfer Function ● Noise Plot
1e3 40e6 -180 -60
rms jitter: 174.350 fs

Figure 5-1: Parameters assumed in this PLL analysis.

In order to suppress the third-order shaped quantization noise, which has a 40 dB/dec roll-off at PLL output, parasitic poles need to be included in the second-order PLL because a second-order PLL can only provide a -40 dB/dec roll-off in its closed-loop transfer function. Note that two poles at 3 MHz are included in this analysis. The first one is to model the switched-capacitor filter inside the fine-tuning DAC. The second pole was added because we intended to add another RC lowpass filter after each DAC to attenuate the clock feedthrough. However, we decided to remove this extra filter to simplify the circuit complexity but did not modify the parameters used in this analysis and the following design.

Figure 5-2 illustrates the corresponding phase noise. As revealed in this figure, excellent in-band and out-of-band phase noises are achieved. Also, the integrated jitter is 174 fs, according to the calculation of PLL Design Assistant.

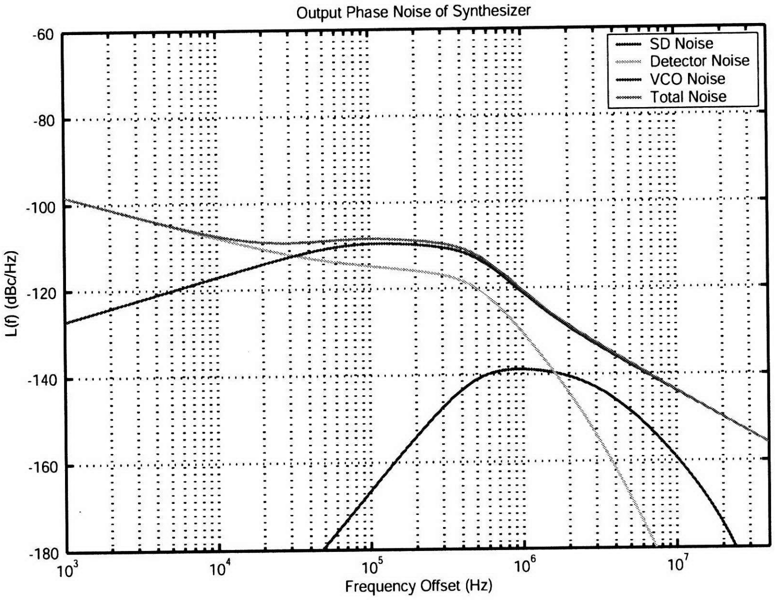


Figure 5-2: Noise analysis using PLL Design Assistant.

5.2 $\Delta\Sigma$ Modulator Design

When designing the $\Delta\Sigma$ Modulator, one should avoid the first-order modulator since the resulting tones are usually too high to be cancelled completely. Second- or third-order modulators are most popular nowadays since they sufficiently scramble the quantization noise. One subtle difference between a second-order and a third-order modulator is the needed accuracy of the noise cancellation in order to achieve the targeted phase noise. Let us revisit the analysis in Section 5.1 but disable the noise cancellation function by setting the S-D transfer function from 0.1*[1 -3 3 -1] to 1*[1 -3 3 -1]. Note that this scale factor (i.e., 0.1 or 1) is denoted as ϵ in equation 6.17. The results in Figure 5-3 reveal that the peak phase noise contributed by the quantization noise is about -120 dBc/Hz at 1-2 MHz offset before the noise cancellation is performed. To make the quantization noise lower than the VCO intrinsic noise, we need to suppress it by a certain amount. As illustrated in Figure 5-2, assuming a 20 dB suppression, the residual quantization noise is considerably below the VCO noise. In other words, we can tolerate 10% of the noise left after the cancellation.

As for the case where a second-order modulator is used, the peak phase noise is -96

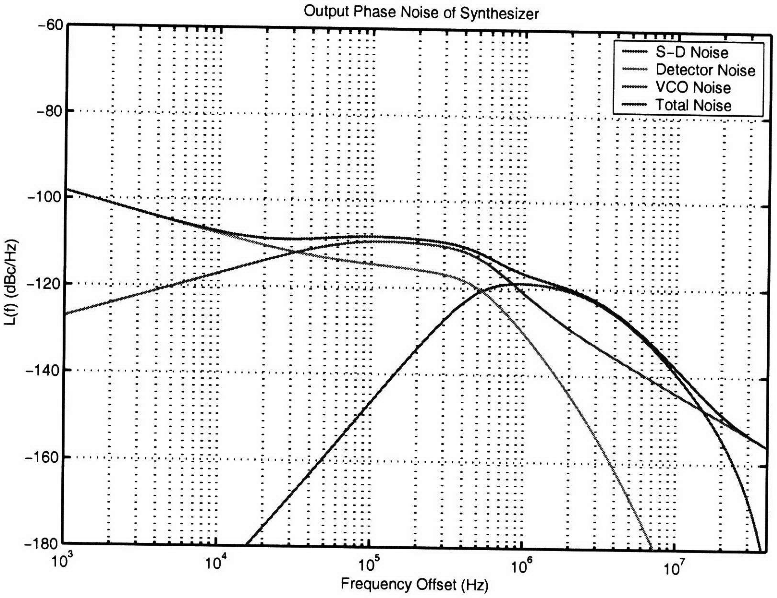


Figure 5-3: Phase noise of a PLL using a third-order $\Delta\Sigma$ modulator without noise cancellation.

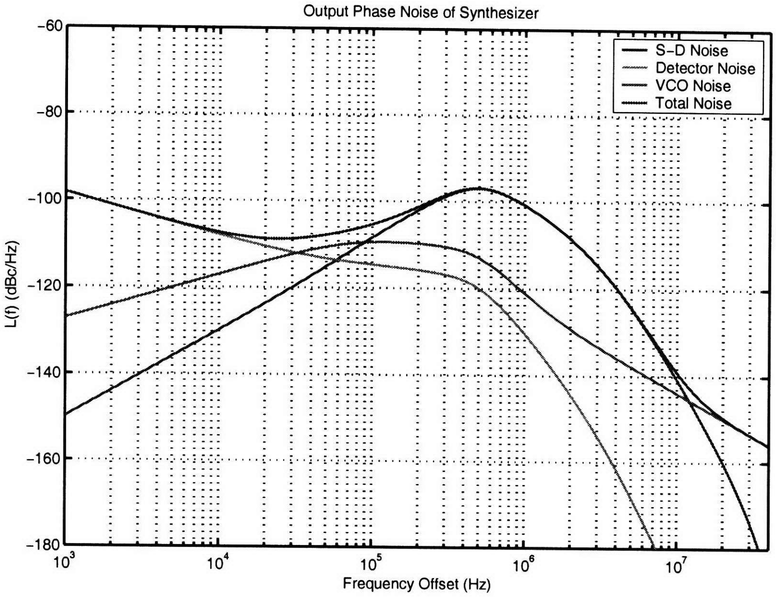


Figure 5-4: Phase noise of a PLL using a second-order $\Delta\Sigma$ modulator without noise cancellation.

dBc/Hz at 500 kHz offset without the noise cancellation, as shown in Figure 5-4. With only 20 dB suppression of this noise, Figure 5-5 reveals that the quantization noise is still too high compared to the intrinsic VCO noise. More than 26 dB quantization noise suppression is necessary such that the residual quantization noise can be lower than the VCO noise, as depicted in Figure 5-6. In other words, to use a second-order modulator to reach similar noise performance as in the case where a third-order one is used, the scale factor of the quantization noise for the noise cancellation must be more accurate. We thus choose a third-order modulator to decrease the necessary accuracy in the noise cancellation.

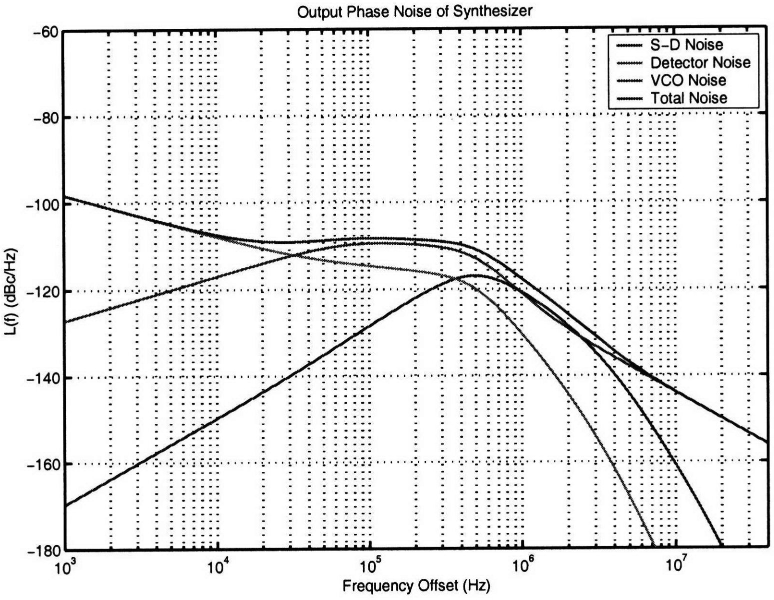


Figure 5-5: Phase noise of a PLL using a second-order $\Delta\Sigma$ modulator with 20-dB noise cancellation.

As a side point, the third-order $\Delta\Sigma$ modulator is implemented with the MASH structure [53]. One critical point in designing the modulator circuit is that the number of clock delays from the quantization error to the multiplier output (delay2 in Figure 5-7) should match that on the TDC side (delay1 in Figure 5-7) such that the corresponding errors on both paths can line up to each other before being cancelled. Since there is some latency in the GRO TDC, extra registers, which are not shown in Figure 5-7, are added between the accumulator and multiplier to match the delays.

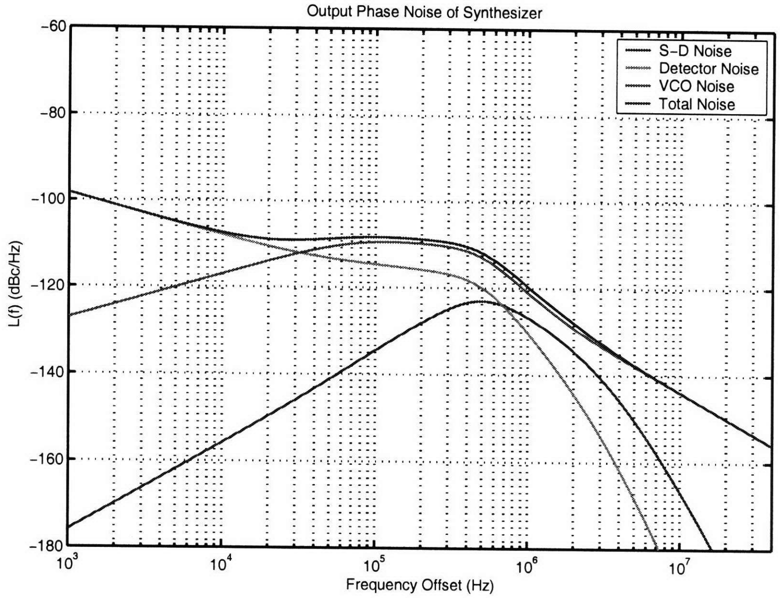


Figure 5-6: Phase noise of a PLL using a second-order $\Delta\Sigma$ modulator with 26-dB noise cancellation.

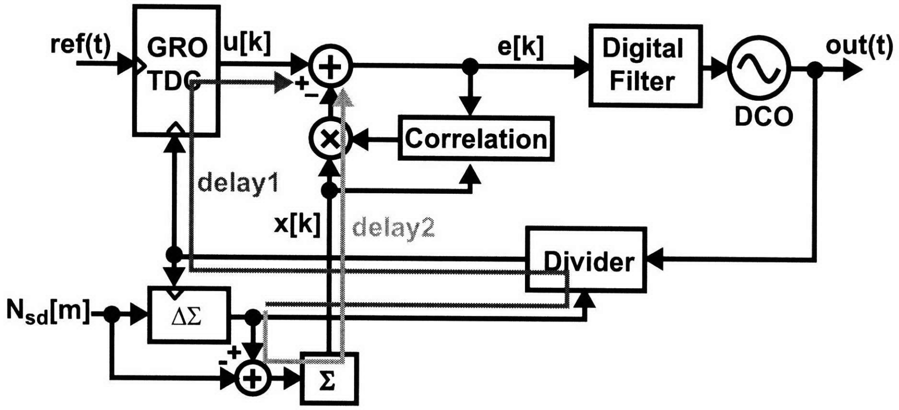


Figure 5-7: The delays on both paths need to be equal to each other such that the quantization noise can be cancelled correctly.

5.3 PLL Type and Order

A Type-II PLL has several advantages over its Type-I counterpart. The integrator in the loop filter not only further suppresses the in-band noise of the VCO, but also avoids a nonzero phase error between the phase detector inputs in the steady state, minimizing the necessary TDC range. We therefore design a Type-II loop in the steady state (i.e., the fine-tuning loop in Figure 5-9). As explained in Section 5.4, a Type-I coarse-tuning loop is designed to improve the overall noise performance by reducing the bandwidth of the coarse-tuning DAC.

Since a third-order $\Delta\Sigma$ modulator, which introduces a 40 dB/dec slope quantization noise, has been chosen, we need a -60 dB/dec roll-off at high frequency offsets in the PLL closed-loop transfer function to suppress this $\Delta\Sigma$ noise. Note that the slope of the third-order $\Delta\Sigma$ noise at the PLL output is 40 dB/dec instead of 60 dB/dec because of the integration function of the divider [34]. Instead of using a third-order PLL to achieve the goal of -60 dB/dec roll-off, a second-order PLL with an additional pole, which is provided by the proposed DAC, is utilized to simplify the filter design.

In addition, although the -60 dB/dec roll-off PLL closed-loop transfer function is intended to filter the $\Delta\Sigma$ noise, it also attenuates the shaped GRO quantization noise without the need for any extra cost in the hardware.

5.4 Proposed Loop Filter

Figure 5-8 provides a conceptual picture of the coarse/fine-tuning method used to acquire phase-lock for the PLL, where we assume that the four-bit control of the MIM capacitor array in the VCO has already been set to achieve the proper frequency band of operation. We see that the TDC output is first filtered by a 1.1-MHz IIR filter in order to reduce the high-frequency quantization noise of the TDC as well as any residual quantization noise produced by the dithered divide value that is not eliminated by the all-digital quantization noise cancellation circuit. During frequency acquisition, the filtered TDC output is first fed into a coarse-tuning path while the fine-tuning path is locked to its mid-range value. After the coarse-tuning path is given a specified amount of time to settle, its value is locked in place. The filtered TDC output is then fed into the fine-tuning path, and the digital quantization noise cancellation is enabled. The state of the filters (i.e., reset, coarse tuning, and fine tuning) and the amount of time assigned to each state are controlled through a shift-register in the prototype. We discuss each of these tuning paths in more detail in the rest of this section.

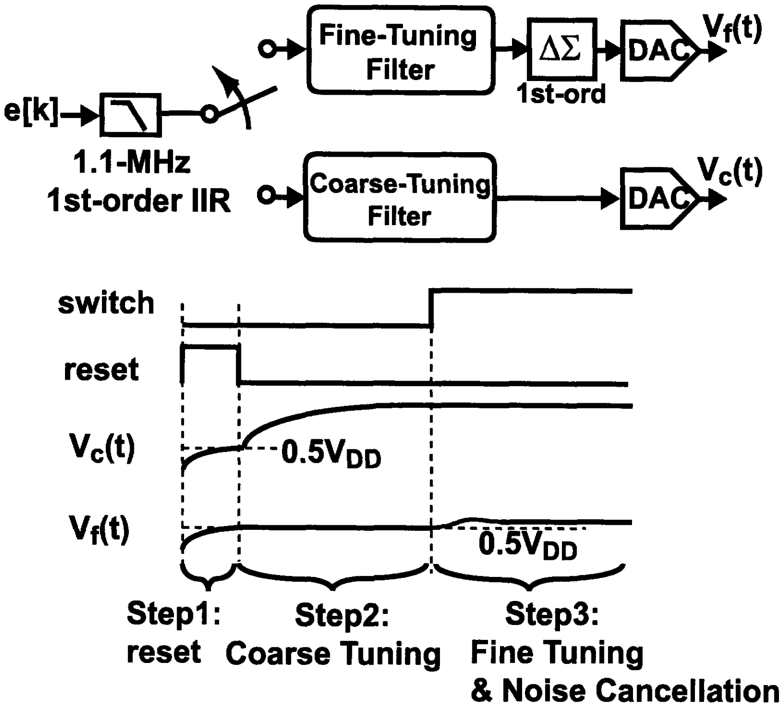


Figure 5-8: Coarse/fine-tuning of the PLL output frequency.

We begin by providing further details of the simpler fine-tuning path. As shown in Figure 5-9, this path is designed to correspond to the analog lead-lag filter topology. A digital accumulator and feedforward gain of K_1 realize a zero of 62.5 kHz, while the initial IIR filter and switched-capacitor network in the fine-tuning DAC realize poles of 1.1 and 3.1 MHz, respectively. Note that the DAC bandwidth is set according to its load capacitor, which has a value of 2.5 pF. Also, note that a first-order $\Delta\Sigma$ modulator is placed between the fine-tuning loop filter and DAC in order to increase the effective resolution of the DAC.

In contrast, the more complicated coarse-tuning path is shown in Figure 5-10. The key challenge in designing this path is to achieve fast settling despite the fact that the coarse-tuning DAC bandwidth must be set to a value eight times lower than the fine-tuning DAC bandwidth. The decrease in bandwidth is achieved by increasing the load capacitor of the coarse-tuning DAC to 20 pF (as compared to the 2.5 pF capacitance of the fine-tuning DAC). The reason for the lower bandwidth is that the coarse-tuning

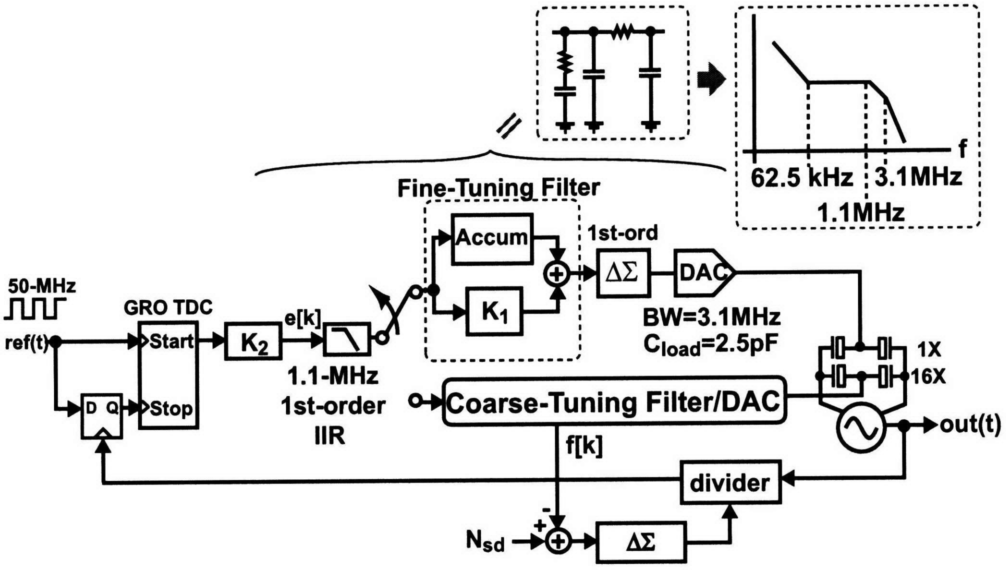


Figure 5-9: Fine-tuning digital loop filter.

varactor K_v is 16 times higher than the fine-tuning varactor (due to its 16 times larger tuning range), so that the thermal noise of the coarse-tuning DAC needs to be more aggressively filtered to avoid degradation of the synthesizer noise performance. Notice that although we freeze the coarse-tuning filter in the steady state, the coarse DAC is still operating due to its switched-capacitor structure. Therefore, the kT/C noise of the coarse-tuning DAC still exists in the steady state.

To improve the coarse-tuning settling time, we alter the loop filter topology of the fine-tuning path such that only the accumulator path feeds into the coarse-tuning DAC, as shown in Figure 5-10. Since the accumulator path requires much less bandwidth to operate than the feedforward path, a much lower DAC bandwidth can be tolerated while still achieving a reasonable settling time. Of course, the feedforward path is required to stabilize the PLL feedback loop, but this path can be implemented by bypassing the coarse-tuning DAC and instead making use of the $\Delta\Sigma$ modulator and divider circuits as shown. This technique is similar to that proposed in [11] and has the interesting property of effectively turning the PLL feedback dynamics into a Type-I system despite the fact that two integrators are in the open-loop system (i.e., the accumulator and VCO). A Type-I system has the advantage of a faster settling

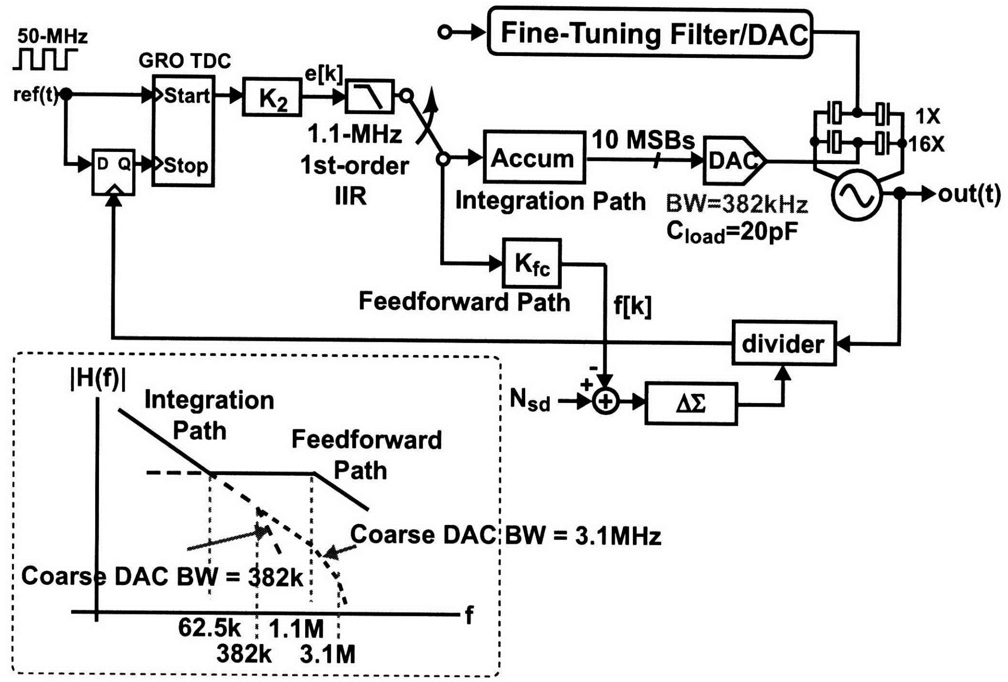


Figure 5-10: Coarse-tuning digital loop filter.

time than its Type-II counterpart, but has the disadvantage of providing less attenuation of the VCO noise at low frequency offsets. However, since the coarse-tuning path is used only during initial frequency acquisition, the reduced suppression of the VCO noise is not of concern.

Note that although a lower coarse-tuning DAC bandwidth improves the noise performance, the coarse-tuning DAC bandwidth must still be sufficiently higher than the targeted zero position (i.e., 62.5 kHz in this example). As illustrated in Figure 5-11, if the coarse-tuning DAC bandwidth becomes lower than the zero position, the overall transfer function becomes unacceptable since an additional pole and zero are introduced.

One additional benefit of reducing the coarse-tuning DAC bandwidth is that it reduces the magnitude of the reference spur caused by clock feedthrough within the DAC. While the fine-tuning DAC also has such clock feedthrough, its impact on the PLL output is 16 times lower due to the lower K_v of the fine-tuning varactor.

Also, note that a $\Delta\Sigma$ modulator is not required in the coarse-tuning path since

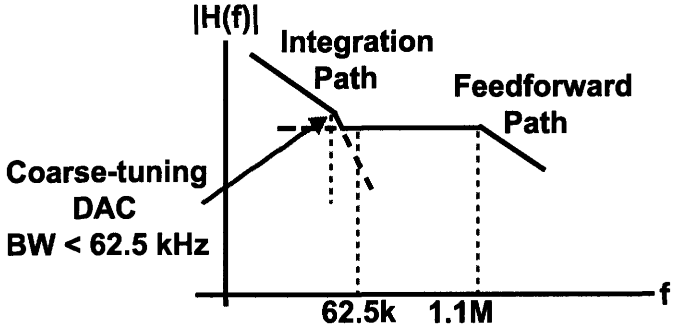


Figure 5-11: The bandwidth of the coarse-tuning DAC needs to be sufficiently higher than the targeted zero frequency.

the 10-bit resolution is more than adequate for achieving a small enough frequency error for the fine-tuning path to stay within range. To explain that, the coarse DAC output voltage may toggle between two adjacent levels after frequency acquisition is completed. This voltage step results in an 117 kHz frequency step in the end of coarse tuning ($1.5V/2^{10} \cdot 80$ MHz/V = 117 kHz). To achieve more accurate frequency locking in the steady state, the maximum offset voltage the fine-tuning DAC output needs to provide is only 23 mV (117kHz/5MHz=23.4mV). Since the fine-tuning voltage stays around $V_{DD}/2$ independently of the VCO frequency, a nonlinear fine-tuning VCO gain (Figure 7-6) does not seriously impact the PLL response over different VCO frequencies. This is one advantage of the proposed PLL, which uses two varactors and filters in its implementation. Although the variation of the coarse-tuning VCO gain may change the PLL response and thus the locking time during the coarse tuning, the locking time is usually not as much of a concern as the steady-state PLL transfer function that determines the overall noise performance.

5.5 Calculation of the Loop Filter Parameters

This section introduces the method to determine the various parameters in the fine and coarse filters. First, we begin from the fine-tuning filter by redrawing its equivalent model as Figure 5-12. The GRO TDC and DAC models in Figures 2-11 and 3-18, respectively, are plugged into the $\Delta\Sigma$ synthesizer model in [34] with all of the noise sources ignored at this point. Also, the pole caused by the DAC, which is already set to 3 MHz, can be ignored here for simplicity. The parameters which need to be determined here are K_2, α, and K_1. K_2 and K_1 are two gain factors used to determine the overall filter gain and zero position, respectively. Note that K_2 is implemented with a multiplier following the GRO, as shown in Figure 2-17. The value of α determines the cutoff frequency of the first-order IIR [9].

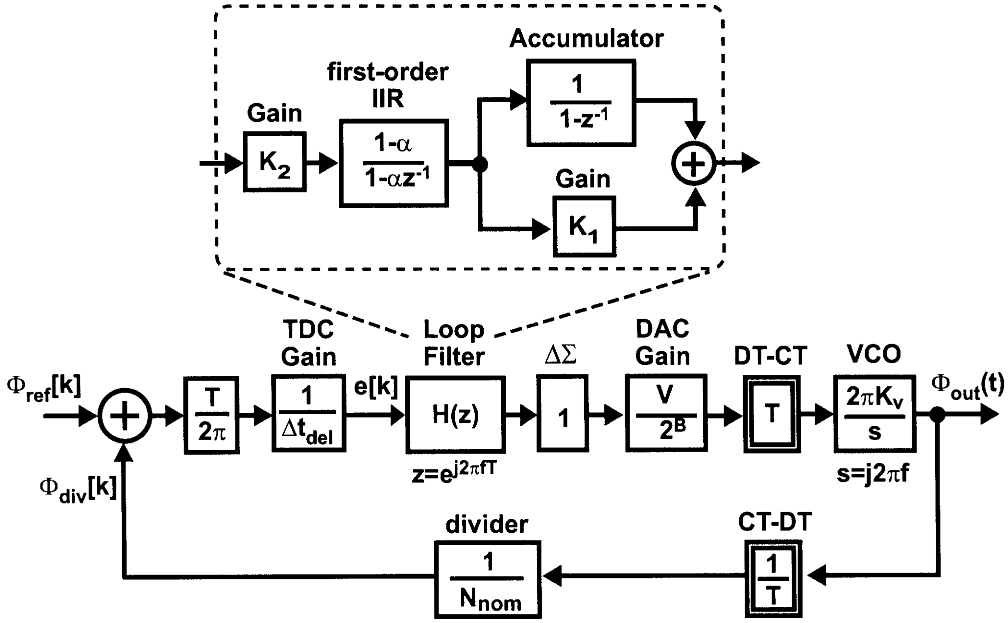


Figure 5-12: Modeling of the PLL in the fine-tuning mode for the PLL response calculation.

The task now is to derive the approximated S-domain open-loop transfer function of our PLL and compare it to that provided by PLL Design Assistant:

$$A_{calc}(s) \quad = \quad \frac{K}{S^{type}} \cdot \frac{1+\frac{s}{\omega_z}}{1+\frac{s}{\omega_p}} \tag{5.1}$$

where *type* is one or two for a Type-I and Type-II PLL, respectively, and ω_z and ω_p

are the zero and pole frequencies. The open-loop transfer function of our PLL is:

$$\begin{aligned} A(s) &= \frac{T}{2\pi} \cdot \frac{1}{\Delta t_{del}} \cdot H(z)|_{z=e^{sT}} \cdot \frac{V}{2^B} \cdot \frac{2\pi K_v}{S} \cdot \frac{1}{N_{nom}} \\ &= \frac{T}{\Delta t_{del}} \cdot \frac{V}{2^B} \cdot \frac{K_v}{N_{nom}} \cdot \frac{1}{s} \cdot H(z)|_{z=e^{sT}} \end{aligned} \tag{5.2}$$

$$\begin{aligned} H(z) &= K_2 \cdot \frac{1-\alpha}{1-\alpha z^{-1}} \cdot (K_1 + \frac{1}{1-z^{-1}}) \\ &= K_2 \cdot \frac{1-\alpha}{1-\alpha z^{-1}} \cdot \frac{K_1 - K_1 z^{-1} + 1}{1-z^{-1}} \end{aligned} \tag{5.3}$$

By using the approximation of $z^{-1} = e^{-sT} \sim 1 - sT$, the S-domain filter transfer function can be approximated as

$$\begin{aligned} H(s) &= H(z)|_{z=1-sT} \\ &= K_2 \cdot \frac{1-\alpha}{1-\alpha(1-sT)} \cdot \frac{K_1 - K_1(1-sT)+1}{sT} \\ &= \frac{K_2}{sT} \cdot \frac{1}{1+\frac{s\alpha T}{1-\alpha}} \cdot (1+sK_1T) \end{aligned} \tag{5.4}$$

Now, we can plug equation 5.4 into equation 5.2 to obtain the final S-domain open-loop transfer function of our PLL:

$$A(s) = (\frac{K_2}{\Delta t_{del}} \cdot \frac{V}{2^B} \frac{K_v}{N_{nom}}) \cdot \frac{1}{s^2} \cdot \frac{1+sK_1T}{1+\frac{s(\alpha T)}{1-\alpha}} \tag{5.5}$$

The last step is to compare the opne-loop transfer function of our PLL (equation 5.5) with the desired open-loop transfer function (equation 5.1) to obtain the following relationships:

$$K_1 = \frac{1}{\omega_z T} = \frac{1}{2\pi f_z T} \tag{5.6}$$

$$K_2 = K \cdot \Delta t_{del} \cdot \frac{2^B}{V} \cdot \frac{N_{nom}}{K_v} \tag{5.7}$$

$$\alpha = \frac{1}{1+\omega_p T} = \frac{1}{1+2\pi f_p T} \tag{5.8}$$

In addition to the five parameters that have been determined with PLL Design Assistant in Section 5.1($K = 8.067 \cdot 10^{11}$, $f_z = 62.5$ kHz, $f_p = 1.139$ MHz, $N_{nom} \sim 72$, T=1/(50 MHz)=20 ns), this prototype uses the following parameters according to the circuit design results:

$$\Delta t_{del} = 6ps, V = 1.5volt, B = 10bit, K_v = 5MHz/V \tag{5.9}$$

By plugging these numbers into equations 5.6, 5.7, and 5.8, we obtain the following fine-tuning filter parameters:

$$K_1 = 127.3, K_2 = 0.0476, \alpha = 0.8748 \tag{5.10}$$

Notice that the final values used for K_1 and α in the prototype are 128 and 0.875, respectively, to leverage the fact that a gain factor of a power of two can be easily implemented as bit-shifting in the hardware. One should adjust the original chosen parameters in PLL Design Assistant and equation 5.9 if necessary to make the resulting K_1 and α sufficiently close to a power of two. Again, K_2 is implemented with a 12-bit multiplier between the TDC and digital filter, as illustrated in Figure 2-17. This multiplier also provides a knob to adjust the open-loop gain in order to compensate for the gain variations in the VCO and TDC.

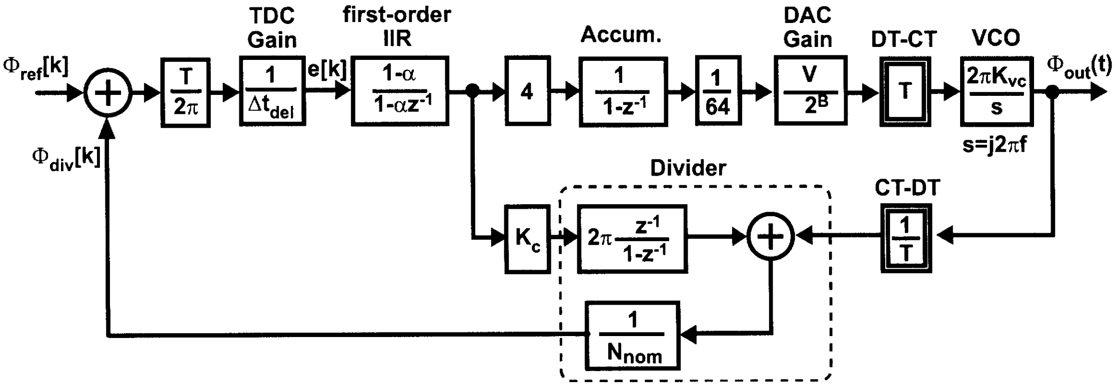


Figure 5-13: Modeling of the PLL in the coarse-tuning mode for the PLL response calculation.

We now analyze the modeling of the PLL in the coarse-tuning mode in order to determine the value of the feedforward gain K_c. The main difference between the coarse-tuning and fine-tuning filters is that the feedforward signal is fed to the divider through the $\Delta\Sigma$ modulator, instead of the VCO, in the coarse-tuning one. We can obtain a similar open-loop response because the divider also behaves as an integrator like the VCO, as illustrated in Figure 5-13 [34]. Notice that the coarse VCO gain K_{vc} in this figure is chosen to be 16 times larger than the fine VCO gain, so the accumulator gain needs to be reduced by 16 times in order to obtain the same open-loop gain as the fine-tuning filter. Again, the gain factor of 1/16 is easily achieved by bit-shifting (i.e., 4 and 1/64 before and after the accumulator, respectively).

As for the feedforward path gain, its S-domain transfer function can be approximated as:

$$\begin{aligned} K_c \cdot 2\pi \frac{z^{-1}}{1-z^{-1}} &\sim K_c \cdot 2\pi \frac{e^{-sT}}{1-e^{-sT}} \\ = K_c \cdot 2\pi \frac{e^{-sT}}{1-(1-sT)} &= \frac{2\pi K_c}{sT} e^{-sT} \end{aligned} \tag{5.11}$$

where the approximation of $z^{-1} = e^{-sT} = 1 - sT$ is used again. In the fine-tuning loop, the feedforward gain from the IIR output to the divider input is:

$$\begin{aligned} K_1 \cdot \frac{V}{2^B} \cdot T \cdot \frac{2\pi K_v}{s} \cdot \frac{1}{T} \\ = K_1 \cdot \frac{V}{2^B} \cdot \frac{2\pi K_v}{s} \end{aligned} \tag{5.12}$$

By making equation 5.12 equal to equation 5.11, we can obtain the following equation:

$$K_c = K_1 \cdot \frac{V}{2^B} \cdot K_v \cdot T \tag{5.13}$$

By plugging the parameters of $K_1 = 128$, $V = 1.5$, $B = 10$, $K_v = 5$ MHz, $T = 20$ ns into the above equation, the resulting K_c is 0.0187. The closest power of two number, 0.0156 (i.e., 1/64), is chosen and implemented as bit-shifting again.

5.6 Summary

A third-order MASH $\Delta\Sigma$ modulator is used to decrease the necessary accuracy of the quantization noise cancellation, so a second-order PLL with a parasitic pole is needed to attenuate this $\Delta\Sigma$ quantization noise at high frequency offsets. A Type-II PLL is chosen to greatly suppress the VCO in-band noise as well as to force a zero phase error at the filter input. The fine-tuning filter is equivalent to an analog lead-lag filter. The two necessary poles are created by a digital IIR filter (1.1 MHz) and the first-order filtering function (3.1 MHz) embedded in the fine-tuning DAC. The zero is set at 62.5 kHz. In addition, the feedforward signal of the coarse-tuning filter is fed to the $\Delta\Sigma$ modulator, instead of the VCO as usual, so that the coarse-tuning DAC bandwidth can be narrowed dramatically to reduce its negative impact on the overall PLL noise. Finally, a systematic way to determine the parameters in the loop filter is derived.

Chapter 6

Noise Analysis and Behavior Simulation

In this chapter, we first analyze the noise performance of the proposed digital synthesizer. After the noise model is built, a short discussion on the trade-offs among the PLL noise and several design parameters is given. Finally, we use a C++ based simulator, *CppSim*, to verify the system [56].

6.1 Noise Analysis of the Proposed Digital Synthesizer

In this section, we first build a complete noise model of the proposed synthesizer and then calculate the overall PLL noise using this model.

6.1.1 PLL Noise Modeling

Figure 6-1 illustrates the modeling of the proposed digital frequency synthesizer with the TDC and DCO models, including their various noise sources [35]. As shown in this figure, the main noise sources in the overall system are

1. TDC quantization noise ($t_q[k]$)
2. Reference noise ($\phi_{ref}[k]$)

3. Divider $\Delta\Sigma$ quantization noise ($n[k]$)
4. Fine-tuning DAC quantization noise ($q[k]$)
5. VCO phase noise ($\phi_n(t)$)
6. Fine-tuning and coarse-tuning DAC thermal noises ($v_{n,f}(t)$ and $v_{n,c}(t)$)

Notice that the effect of the noise cancellation is not included in this figure but is considered in equation 6.17.

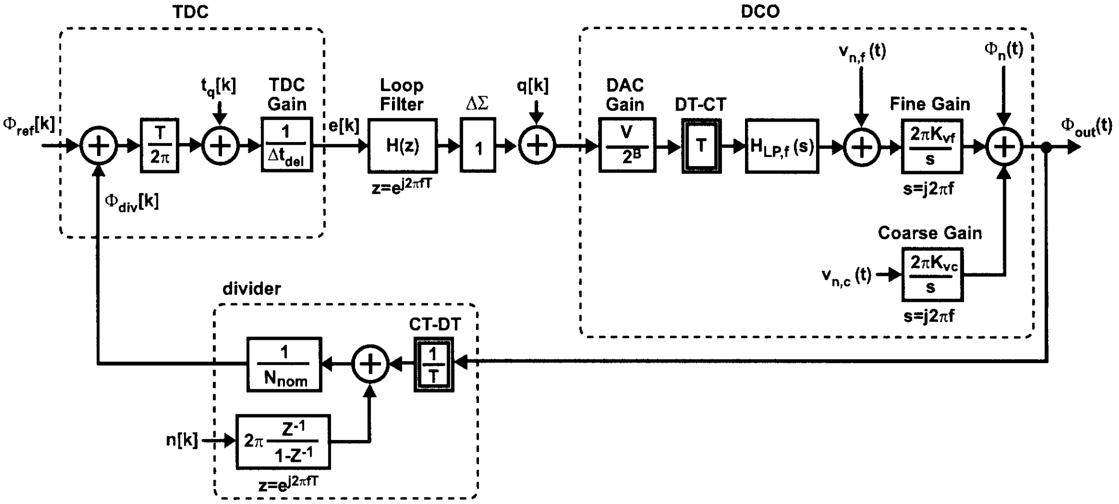


Figure 6-1: Modeling of the proposed digital synthesizer with various noise sources.

The goal here is to first derive the individual noise transfer functions, then characterize the spectral noise density of each noise source, and finally calculate the PLL output phase noise contributed from each of them. Before being able to do it in a systematic way, we need to define several transfer functions and terms in advance. First, the open-loop and closed-loop transfer functions are defined as [34][35]:

$$A(f) = \frac{T}{2\pi}\cdot\frac{1}{\Delta t_{del}}\cdot H(z)|_{z=e^{j2\pi fT}}\cdot\frac{V}{2^B}\cdot H_{LP,f}(f)\cdot\frac{K_{vf}}{jf}\cdot\frac{1}{N_{nom}} \tag{6.1}$$

$$G(f) = \frac{A(f)}{1+A(f)} \tag{6.2}$$

Since $A(f)$ is lowpass in nature with an infinite gain at DC, $G(f)$ has the following properties:

$$G(f) \longrightarrow 1 \;\; as \;\; f \longrightarrow 0$$
$$G(f) \longrightarrow 0 \;\; as \;\; f \longrightarrow \infty \tag{6.3}$$

, implying that $G(f)$ is a lowpass filter with a low-frequency gain of one.

Next, we refer all noise sources before the loop filter (i.e., $\phi_{ref}[k]$, $t_q[k]$, and $n[k]$) to the reference input and call it "reference-referred noise." In addition, we refer all noise sources after the loop filter (i.e., $\phi_n(t)$, $q[k]$, $v_{n,f}(t)$, and $v_{n,c}(t)$) to the DCO output and call it "DCO-referred noise." Dividing the seven noise sources into two groups, as illustrated in Figure 6-2, allows us to calculate and understand the overall output phase noise easily, since these two equivalent noises have very different characteristics after being filtered by the PLL. To understand the difference, we define the noise transfer functions from the reference-referred and DCO-referred noises to the PLL output as following:

$$\frac{\phi_{out}}{\phi_{ref}} = T \cdot N_{nom} \cdot |G(f)| \tag{6.4}$$
$$\frac{\phi_{out}}{\phi_n} = |1 - G(f)| \tag{6.5}$$

One should see the key difference of these two equations is that although the reference-referred noise is filtered by the lowpass filtering action of the PLL (i.e., $G(f)$), the DCO-referred noise is highpass filtered by $1-G(f)$. In addition, the reference-referred noise is amplified by a factor of $T \cdot N_{nom}$, but the DCO referred-noise is not. As for the transfer functions from each noise source to either the reference input or DCO output, we can derive them with the help of Figures 6-3 and 6-4.

One subtle point is that the calculating of the spectral noise densities with Figures 6-2, 6-3, and 6-4 involves both discrete-time (DT) and continuous-time (CT) signals. Therefore, the following two equations need to be applied properly [34][57]:

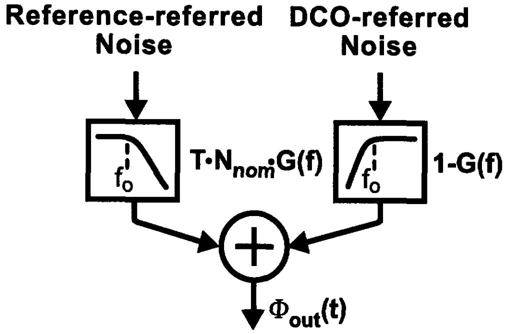


Figure 6-2: Dividing the noise sources into two groups: reference-referred noise and DCO-referred noise.

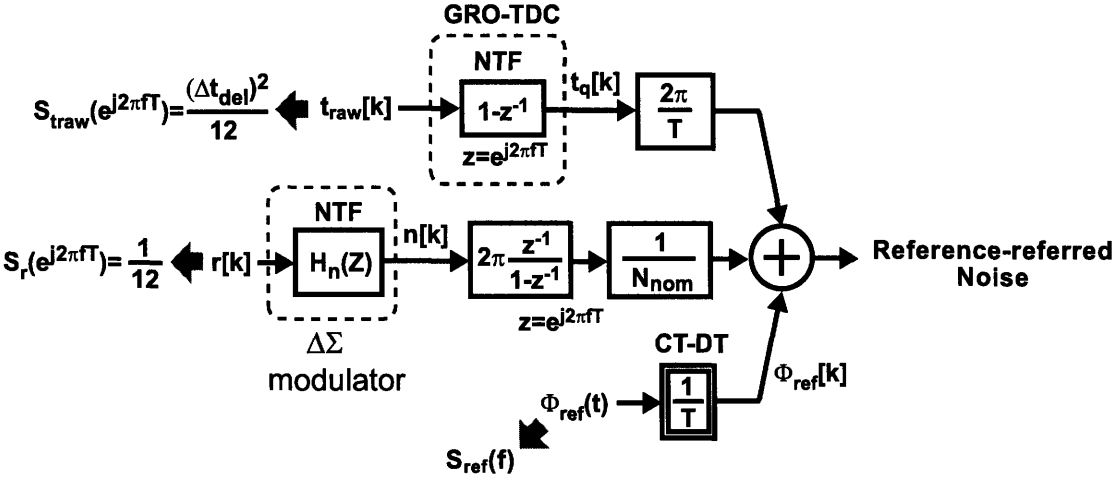


Figure 6-3: Calculation of the reference-referred noise.

case 1) CT input $x(t)$ fed to CT filter $H(f)$ to produce a CT output $y(t)$:

$$S_y(f) = |H(f)|^2 S_x(f) \tag{6.6}$$

case 2) DT input $x[k]$ fed to CT filter $H(f)$ to produce a CT output $y(t)$:

$$S_y(f) = \frac{1}{T}|H(f)|^2 S_x(e^{j2\pi fT}) \tag{6.7}$$

Now with the foregoing three figures and two equations, the PLL output noise densities contributed by various noise sources can be derived one by one in the next

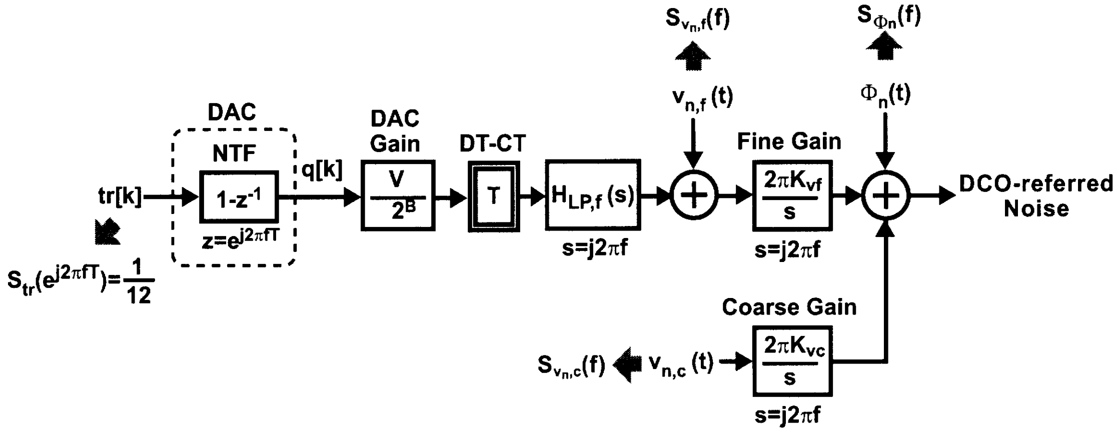


Figure 6-4: Calculation of the DCO-referred noise.

section.

6.1.2 Overall Phase Noise Calculation

The goal of this section is first to calculate the amount of each noise at the PLL output and then to plot the overall PLL noise using MATLAB. Note that the first three noises below are DT noises such that equation 6.7 should be used; the rest of the noises are CT noises, so equation 6.6 should be used.

A. TDC Noise

The noise due to the TDC quantization can be calculated as:

$$S_{out,traw}(f) \;=\; \frac{1}{T}|T \cdot N_{nom} \cdot G(f)|^2 \cdot (\frac{2\pi}{T})^2 \cdot |1-e^{-j2\pi fT}|^2 \cdot S_{traw}(e^{j2\pi fT}) \quad (6.8)$$

which can also be expressed as

$$S_{out,traw}(f) \;=\; \frac{1}{T}|2\pi N_{nom} G(f)|^2 (2sin(\pi fT))^2 \cdot S_{traw}(e^{j2\pi fT}) \quad (6.9)$$

By assuming that the raw quantization noise of the TDC is white, its noise density can be expressed as:

$$S_{traw}(e^{j2\pi fT}) = \frac{(\Delta t_{del})^2}{12} \tag{6.10}$$

Therefore, we obtain the output noise contributed by the TDC quantization noise as:

$$S_{out,traw}(f) = \frac{1}{T}|2\pi N_{nom}G(f)|^2(2sin(\pi fT))^2 \cdot \frac{(\Delta t_{del})^2}{12} \tag{6.11}$$

Recall that there are another two noise sources in the GRO in addition to the quantization noise, as illustrated in Figure 2-12. Therefore, we need to add these two noise components to equation 6.11. The flat noise floor in Figure 2-12 can be described with a new parameter Δt_{floor}, which is 1 ps, in equation 6.12. The flicker noise is described with the last term in this equation. It can be shown that the PLL noise performance is limited by this flicker noise at low frequency offsets.

$$\begin{aligned} S_{out,traw}(f) &= \frac{1}{T}|2\pi N_{nom}G(f)|^2((2sin(\pi fT))^2 \cdot \frac{(\Delta t_{del})^2}{12} + \frac{(\Delta t_{floor})^2}{12} \\ &\quad + \frac{K_{flicker}}{f}) \end{aligned} \tag{6.12}$$

B. Reference Noise

The reference noise is caused by not only the off-chip 50-MHz reference source but also the buffer between the reference source and TDC. Similar to the sampling action on the VCO side [34], a scale factor of $1/T$ is necessary in Figure 6-3 to convert the CT noise density $S_{ref}(f)$, which is usually reported in the datasheet or measurement results, to its DT version. The output noise because of the reference noise can be calculated as:

$$\begin{aligned} S_{out,ref}(f) &= |T \cdot N_{nom} \cdot G(f)|^2 \cdot (\frac{1}{T})^2 \cdot S_{ref}(f) \\ &= |N_{nom} \cdot G(f)|^2 \cdot S_{ref}(f) \end{aligned} \tag{6.13}$$

The value of $S_{ref}(f)$ should be estimated using the measurement result of the off-chip reference source and the simulation result of the reference buffer. One should notice that $S_{ref}(f)$ is amplified and lowpass filtered by the PLL. It indicates that an excellent reference source and buffer design is critical to achieve a low-noise wide-bandwidth PLL since a wide bandwidth leads to less reference noise suppression.

C. Divider $\Delta\Sigma$ Quantization Noise

Assuming a m-th order $\Delta\Sigma$ modulator is used next to the divider, noise due to this modulator can be calculated as:

$$\begin{aligned} S_{out,r}(f) &= \frac{1}{T}|T \cdot N_{nom} \cdot G(f)|^2 \cdot (\frac{1}{N_{nom}})^2 \cdot |2\pi \frac{e^{-j2\pi fT}}{1-e^{-j2\pi fT}}|^2 \cdot |1-e^{-j2\pi fT}|^{2m} \\ &\cdot S_r(e^{j2\pi fT}) \end{aligned} \tag{6.14}$$

which can be simplified as

$$S_{out,r}(f) = T|2\pi G(f)|^2 (2sin(\pi fT))^{2(m-1)} \cdot S_r(e^{j2\pi fT}) \tag{6.15}$$

By assuming that the raw quantization noise of the $\Delta\Sigma$ modulator is white, its noise density can be expressed as:

$$S_r(e^{j2\pi fT}) = \frac{1}{12} \tag{6.16}$$

Therefore, we obtain the output noise contributed by the $\Delta\Sigma$ quantization noise as:

$$S_{out,r}(f) = \epsilon^2 \cdot T|2\pi G(f)|^2 (2sin(\pi fT))^{2(m-1)} \cdot \frac{1}{12} \tag{6.17}$$

with the parameter ϵ used to model the effect of the noise cancellation. This parameter corresponds to the scale factor in the S-D transfer function used in the PLL Design Assistant, as illustrated in Figure 5-1. The value of ϵ, ranging from zero to one, represents the amount of noise left after the noise cancellation is performed. In the ideal case where the quantization noise is completely removed, ϵ is zero, making $S_{out,r}(f)$ zero. As for the case in Figure 5-1, where 10% of the quantization noise is left, ϵ is 0.1.

D. Fine-tuning DAC Quantization Noise

Although the 10-bit DAC alone does not contribute a quantization noise, truncating the digital filter output, which is longer than 10 bits, to 10 bits results in another quantization noise in the system. In order to reduce this noise at low frequency offsets as well as to avoid spurs due to this truncation, a first-order $\Delta\Sigma$ modulator is put between the digital loop filter and DAC, as illustrated in Figure 5-9. The noise due to this truncation can be calculated as:

$$\begin{aligned} S_{out,tr}(f) &= \frac{1}{T}|1-G(f)|^2 \cdot (\frac{V}{2^B} \cdot T)^2 \cdot |H_{LP,f}(j2\pi f)|^2 \cdot |\frac{2\pi K_{vf}}{j2\pi f}|^2 \\ &\cdot |1-e^{-j2\pi fT}|^2 S_{tr}(e^{j2\pi fT}) \end{aligned} \tag{6.18}$$

which can also be simplified as

$$\begin{aligned} S_{out,tr}(f) &= T \cdot |1-G(f)|^2 \cdot (\frac{V}{2^B})^2 \cdot \frac{1}{1+(\frac{f}{f_{pf}})^2} \cdot (\frac{K_{vf}}{f})^2 \cdot (2sin(\pi fT))^2 \\ &\cdot S_{tr}(e^{j2\pi fT}) \end{aligned} \tag{6.19}$$

By assuming that the truncation noise is white, its noise density can be expressed as:

$$S_{tr}(e^{j2\pi fT}) = \frac{1}{12} \tag{6.20}$$

Therefore, we obtain the PLL output noise contributed by this $\Delta\Sigma$ modulated truncation noise as:

$$\begin{aligned} S_{out,tr}(f) &= T \cdot |1-G(f)|^2 \cdot (\frac{V}{2^B})^2 \cdot \frac{1}{1+(\frac{f}{f_{pf}})^2} \cdot (\frac{K_{vf}}{f})^2 \cdot (2sin(\pi fT))^2 \\ &\cdot \frac{1}{12} \end{aligned} \tag{6.21}$$

E. VCO Noise

The PLL output noise contributed from the VCO noise can be simply calculated as:

$$S_{out,\phi_n}(f) = |1 - G(f)|^2 \cdot S_{\phi_n}(f) \tag{6.22}$$

F. Fine-tuning and Coarse-tuning DAC Thermal Noises

The noise due to the fine-tuning and coarse-tuning thermal noises can be calculated as:

$$S_{out,v_n,f}(f) = |1 - G(f)|^2 \cdot (\frac{K_{vf}}{f})^2 \cdot S_{v_n,f}(f) \tag{6.23}$$

$$S_{out,v_n,c}(f) = |1 - G(f)|^2 \cdot (\frac{K_{vc}}{f})^2 \cdot S_{v_n,c}(f) \tag{6.24}$$

where the noise spectral densities $S_{v_n,f}(f)$ and $S_{v_n,c}(f)$ have been derived in Chapter 3. These equations are listed below again for convenience:

$$S_{v_n,f}(f) = 4kTR_{eq,f} \cdot \frac{1}{1 + (\frac{f}{f_{pf}})^2} \tag{6.25}$$

$$S_{v_n,c}(f) = 4kTR_{eq,c} \cdot \frac{1}{1 + (\frac{f}{f_{pc}})^2} \tag{6.26}$$

where $R_{eq,f}$ and f_{pf} are the equivalent resistance and corner frequency of the fine-tuning DAC , and $R_{eq,c}$ and f_{pc} are those of the coarse-tuning DAC.

G. Overall Noise

Because the above noises are uncorrelated to each other, the overall noise spectral density at the PLL output can be obtained by summing the above results:

$$\begin{aligned} S_{out}(f) = {} & S_{out,traw}(f) + S_{out,ref}(f) + S_{out,r}(f) + S_{out,\phi_n}(f) + S_{out,tr}(f) + S_{out,v_n,f}(f) \\ & + S_{out,v_n,c}(f) \end{aligned} \tag{6.27}$$

To observe the relative contribution of each noise component, a MATLAB script, which can be found online [44], is developed to plot the overall PLL noise with the result illustrated in Figure 6-5. The parameters used in this calculation are listed in Table 6.1. One should see that most noises are lower than the VCO phase noise except the thermal noise from the coarse-tuning DAC and flicker noise of the GRO. Although the overall noise is limited by the thermal noise of the coarse-tuning DAC from 40 kHz to 600 kHz, we can still achieve excellent noise performance of -108 dBc/Hz at 400 kHz offset, which is 8 dB lower than the GSM requirement (after being referenced to 3.6 GHz). The noise at 20 MHz offset is -150 dBc/Hz, limited by the intrinsic phase noise of the VCO. In addition, notice that the quantization noise from the fine-tuning DAC is so low that its folded amount because of the DAC nonlinearity is not sufficiently high to impact the overall performance. This effect is investigated with *CppSim* simulation in Section 6.3.

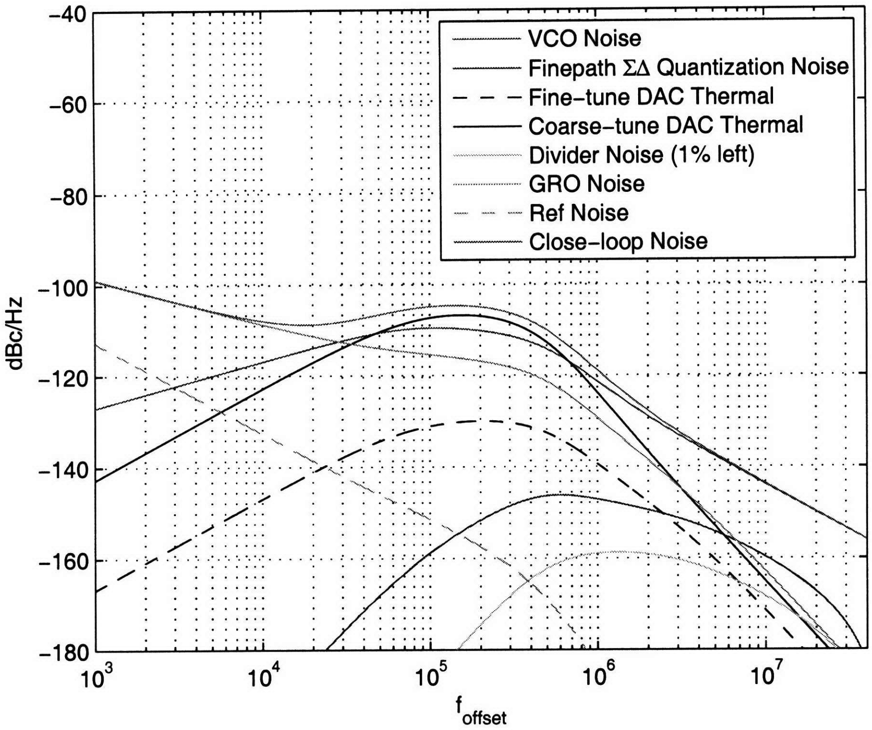


Figure 6-5: Overall calculated noise using the parameters in Table 6.1

6.2 Design Considerations

Since the noise model is built, we can now discuss the trade-offs among the phase noise and some design parameters in this section.

6.2.1 PLL Bandwidth

The main limiting factor of the PLL bandwidth is the quantization noise of the GRO. Because of its 20 dB/dec slope, even though a PLL can attenuate it, this noise still has a negative impact on the overall PLL noise because extending the PLL bandwidth allows more of it to go through. To understand this better, another noise plot is

Table 6.1: Parameters used for calculation in Figure 6-5

Parameter	Value
f_{clk}	50 MHz
N_{nom}	73
Δt_{del}	6 ps
Δt_{floor}	1 ps
$K_{flicker}$	1.2×10^{-20}
Reference noise	-150 dBc/Hz at 1 kHz offset
V	1.5 V
B	10 bit
C_u	30 fF
$C_{2,f}$	2.5 pF
$C_{2,c}$	20 pF
K_{vf}	5 MHz/V
K_{vc}	80 MHz/V
VCO noise	-150 dBc/Hz at 20 MHz offset, 200-kHz flicker noise corner
K_1	128
K_2	0.0476
α	0.875
m	3
ϵ	0.01

illustrated in Figure 6-6, assuming a PLL bandwidth of 1 MHz. In this case, the GRO quantization noise becomes comparable with the VCO intrinsic noise at 1-6 MHz offset, degrading the overall noise performance at high frequency offsets. We therefore choose 500 kHz as the bandwidth in this prototype such that the overall noise can be dominated by the intrinsic VCO noise. For applications with less strict noise performance, a bandwidth wider than 500 kHz is of course achievable, as already illustrated in Figure 6-6. Also, one can add another filter in the loop to attenuate the GRO quantization noise if necessary.

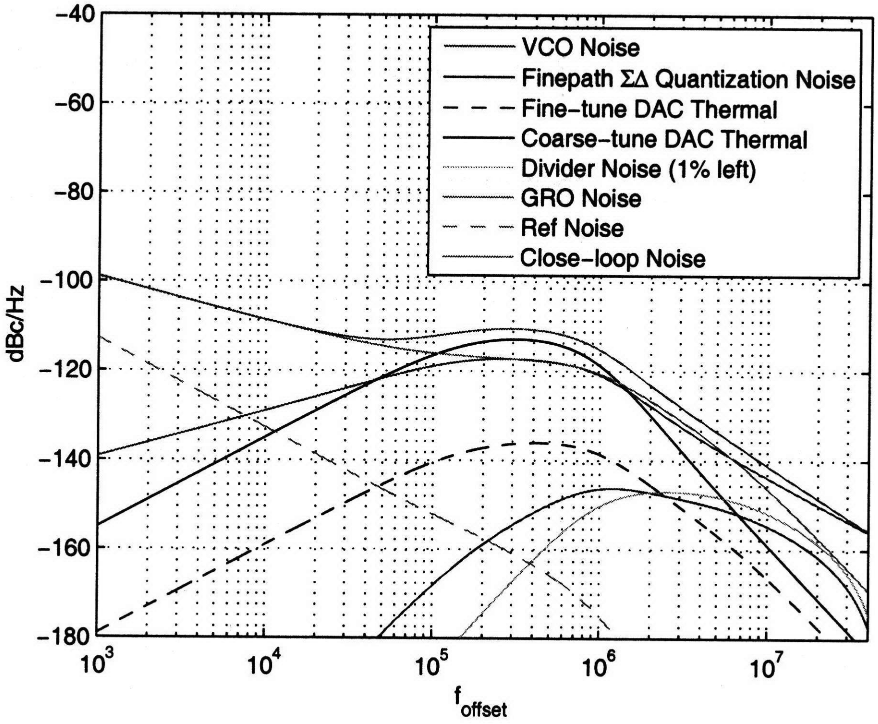


Figure 6-6: Overall calculated noise assuming a 1-MHz bandwidth.

One should notice that the divider quantization noise also becomes higher when the bandwidth is extended to 1 MHz, although it is still lower than the VCO noise. Actually, a 90% cancellation of this noise as assumed in Section 5.1 is not good enough for a 1-MHz bandwidth PLL. The accuracy of this cancellation has to be around 99%, as shown in Figure 6-6.

Another minor factor limiting the PLL bandwidth is the latency caused by the digital circuits between the TDC and DAC. Pipelining [58] is necessary here because a 50-MHz clock rate is utilized. The resulting latency degrades the phase margin of a high-bandwidth PLL, so one should include it in the phase margin calculation. To explain this, assume that there are n clock delays in the digital path; then a gain factor of $z^{-n} \sim e^{-nsT}$ needs to be added to the open-loop transfer function of the PLL (i.e., equation 6.1).

6.2.2 Reference Frequency

The implemented prototype uses a 50-MHz reference clock. However, some applications require a lower reference frequency since crystal oscillators at lower frequencies are usually cheaper. For example, GSM usually uses a reference frequency of 13 or 26 MHz.

To understand the performance of the proposed PLL with a lower reference frequency, the calculated noise with a 26-MHz clock is plotted in Figure 6-7. The parameters used in this plot are K_1=66.2, K_2=0.092, α=0.78, and N_{nom}=138.46. All of the other parameters are kept at the same values as those in Table 6.1.

The quantization noises of the GRO, divider, and fine-tuning DAC as well as the thermal noises of both DACs become higher due to a lower clock rate in this case, but only the coarse-tuning thermal noise has an impact on the overall performance, according to Figure 6-7. The phase noise at 400 kHz offset becomes -107 dBc/Hz, which is only 1 dB worse than the case where a 50-MHz reference clock is used. The highest phase noise at intermediate frequency offsets is -103.2 dBc/Hz at 125 kHz offset, compared to -104.5 dBc/Hz in Figure 6-5.

To conclude, although the proposed PLL is demonstrated with a 50-MHz clock in this thesis, it is possible to use a lower reference frequency in the future implementation to achieve the compatible performance. Note that in Chapter 7, a measurement result with a 30.5-MHz clock is also provided.

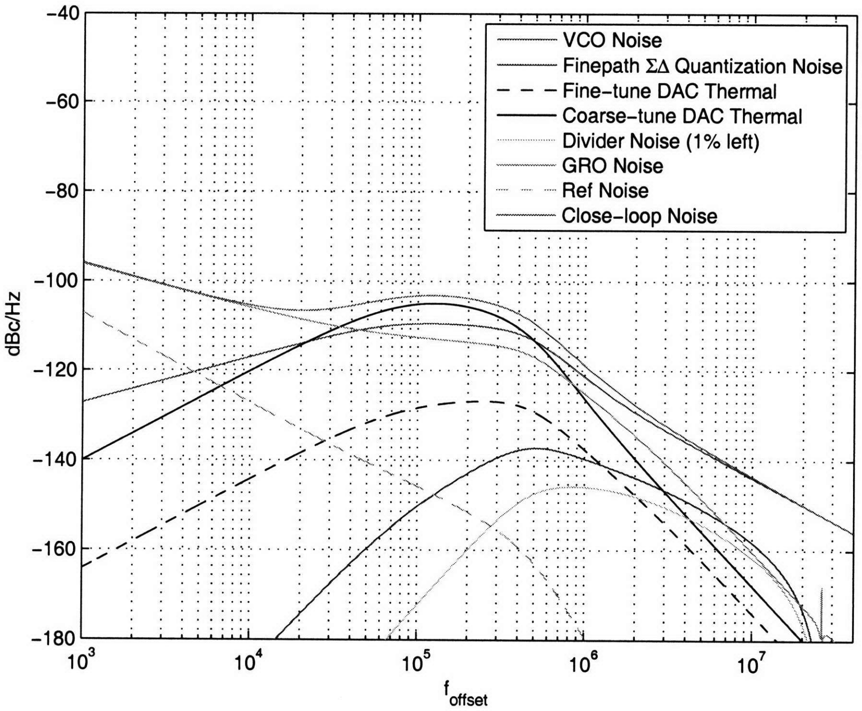


Figure 6-7: Overall calculated noise with a 26-MHz reference clock.

6.2.3 Bandwidth of the Coarse-tuning DAC

As discussed in Section 5.4, a coarse-tuning loop filter, which is different from the fine-tuning filter, is utilized, such that the bandwidth of the coarse-tuning DAC can be lowered to around 400 kHz to reduce the impact of its thermal noise on the PLL phase noise.

We now check the case where the bandwidth of the coarse-tuning DAC is set to 3 MHz like the fine-tuning DAC, and the result is plotted in Figure 6-8. As revealed in this plot, the thermal noise of the coarse-tuning DAC becomes higher than the VCO noise over a wide range. By comparing Figures 6-8 and 6-5, one can see how a lower coarse-tuning DAC bandwidth improves the overall noise performance.

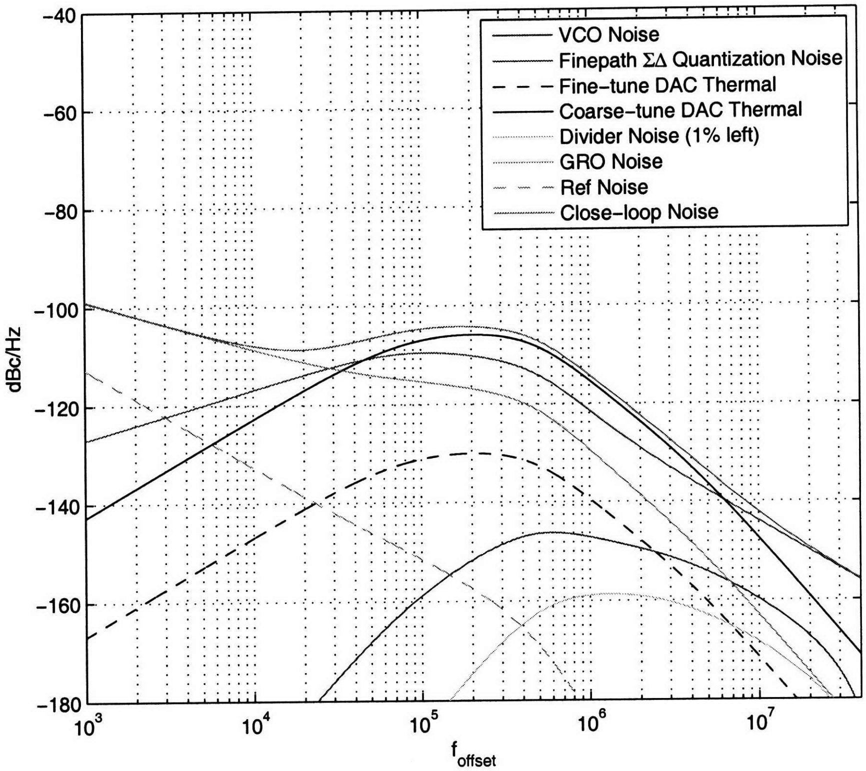


Figure 6-8: Calculated noise when the bandwidth of the coarse-tuning DAC is set to 3 MHz.

6.2.4 Coarse-tuning VCO Gain

As shown in Figure 6-5, the thermal noise of the coarse-tuning DAC is slightly higher than the VCO noise. To further improve the noise at 400 kHz offset, we can decrease the coarse-tuning VCO gain, but it carries the penalty of reducing the VCO coarse-tuning range and thereby needs a finer MIM array resolution. Figure 6-9 illustrates the calculated noise, assuming the coarse-tuning VCO gain is 20 MHz/V. Notice that the overall noise is now dominated by the VCO at intermediate frequencies instead of the coarse-tuning DAC, with a factor of four reduction in the coarse-tuning range.

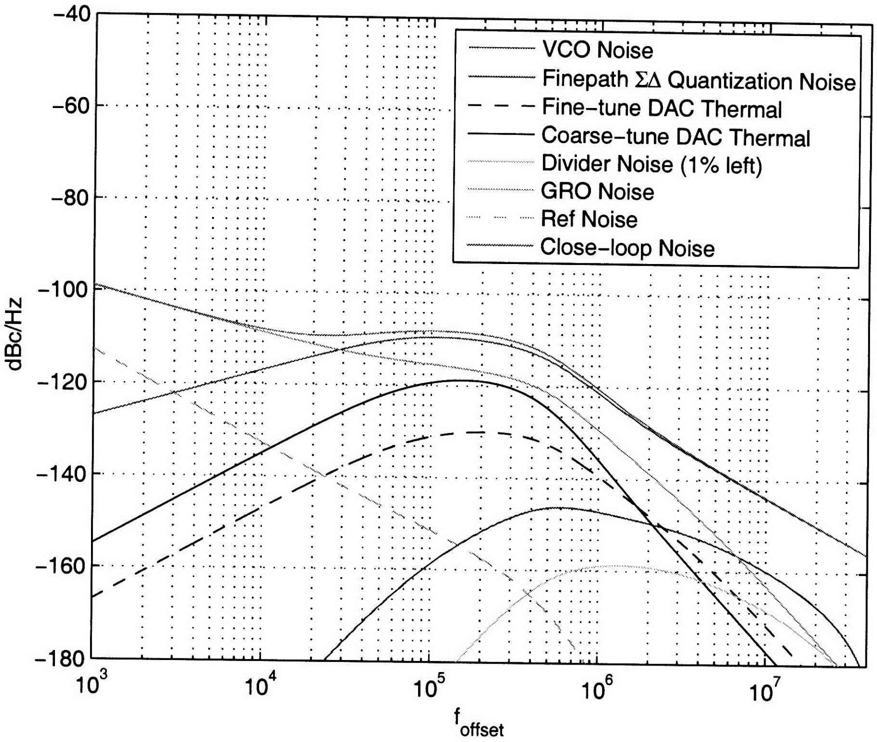
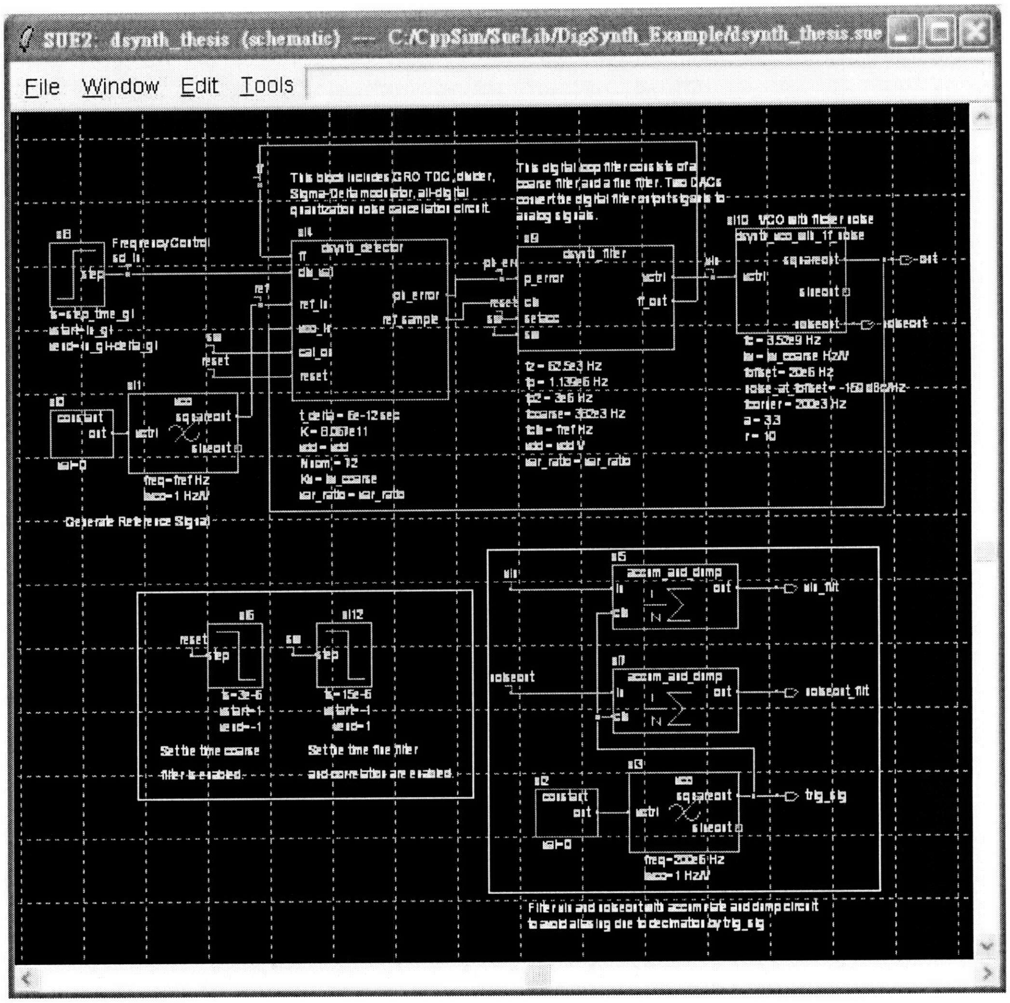


Figure 6-9: Calculated noise when the coarse-tuning VCO gain is reduced to 20 MHz/V.

6.3 Behavior Simulation with CppSim

To verify the proposed synthesizer architecture in the time domain, a C++ based tool, *CppSim*, is used to build the behavior model, as illustrated in Figure 6-10 [44][56][59]. A detailed introduction to this model is available in [44], so we only demonstrate the important simulation results here.

First, the locking behavior of the synthesizer is checked. Figure 6-11 illustrates the coarse-tuning and fine-tuning voltages. As described in Section 5.4, both voltages are brought to $V_{DD}/2$ first. The longer time the coarse-tuning voltage needs to settle in this stage reflects the eight times lower bandwidth of the coarse-tuning DAC. From t = 3 μs, the fine-tuning path is frozen, but the coarse-tuning path becomes enabled to achieve frequency acquisition. A zoomed-in snapshot in Figure 6-12 reveals that the coarse-tuning voltage settles around t = 15 μs, so we can enable the fine-tuning


SUE2: dsynth_thesis (schematic) — C:/CppSim/SueLib/DigSynth_Example/dsynth_thesis.sue

This block includes GRO TDC, divider, Sigma-Delta modulator, all-digital quantization noise cancellation circuit.

This digital loop filter consists of a coarse filter, and a fine filter. Two DACs convert the digital filter outputs signals to analog signals.

Set the time coarse filter is enabled.

Set the time fine filter and correlation are enabled.

Filter vin and noiseout with accumulate and dump circuit to avoid aliasing due to decimation by trig_sig.

Figure 6-10: *CppSim* behavior model of the proposed digital synthesizer.

path and freeze the coarse-tuning code at this point to achieve phase locking. Figure 6-12 also indicates that the entire locking time (i.e., reset, coarse tuning, and fine tuning) is around 20 μs.

The effect of the noise cancelling can also be observed through the simulation. Figure 6-13 depicts the simulated scale factor and phase error signal (i.e., *scale factor* and $e[k]$ in Figure 2-14) with the noise cancellation enabled at t = 15 μs. After t = 15 μs, the magnitude of $e[k]$ drops immediately, and *scale factor* gradually settles to 1.1 V. The settling time of the calibration loop is around 10 μs.

Figure 6-14 illustrates the simulated phase noise overlapped with the calculated noise in Figure 6-5. One can see the good agreement between the analysis and simu-

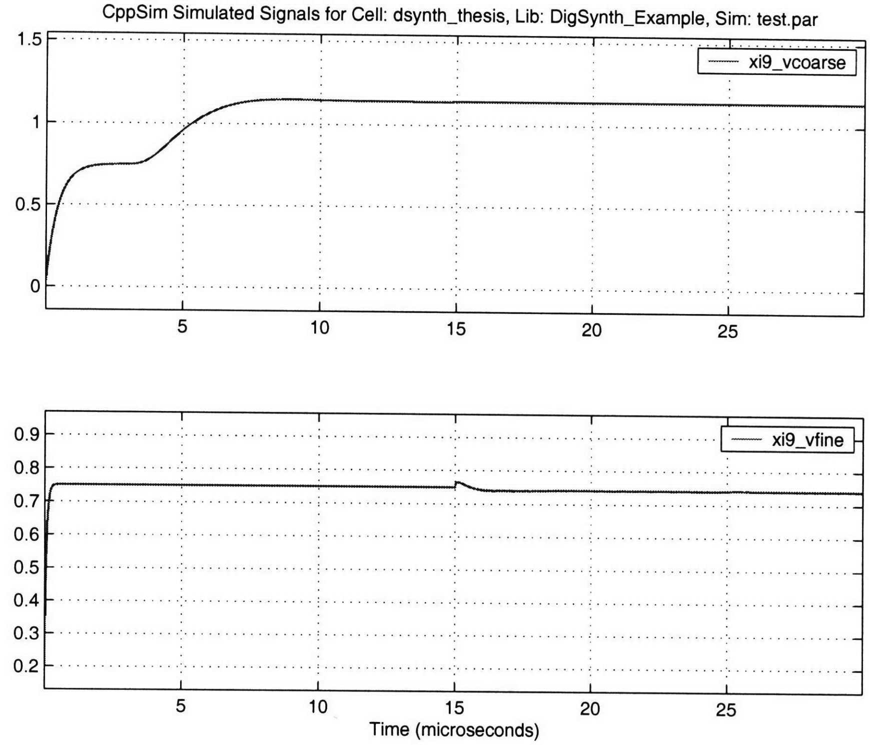


Figure 6-11: Simulated coarse-tuning and fine-tuning voltages.

lation.

Also, the impact on the phase noise due to the variations of the unit resistors and capacitors in the DAC is also investigated. Figure 6-15 plots ten simulation results with a 5% standard deviation assigned to the unit resistors and capacitors in Figure 3-2. The results show that the mismatch with a standard deviation of 5% does not seriously affect the overall noise performance. Note that this plot looks noisier because the number of the simulation steps is reduced by a factor of four to save the simulation time.

6.4 Summary

We present the noise modeling of this synthesizer as well as a behavior model based on *CppSim*. Based on the noise model, we discuss the trade-offs among the PLL noise and several design parameters. In addition, the time-domain simulation result agrees

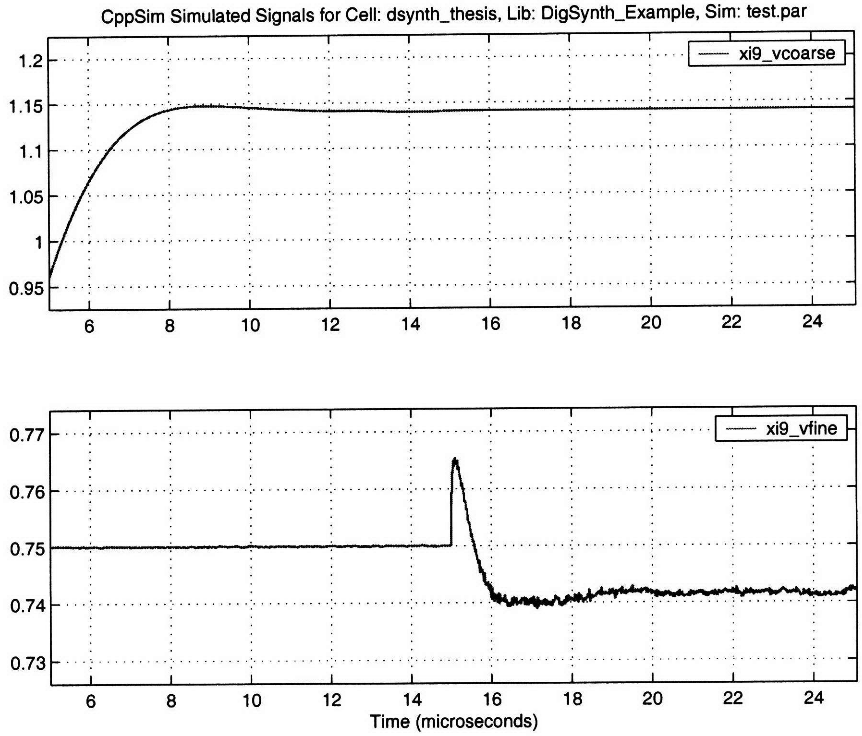


Figure 6-12: Zoomed-in coarse-tuning and fine-tuning voltages.

with the frequency-domain noise analysis quite well. The development of these two models allows us to verify the performance of the system before the chip is actually implemented.

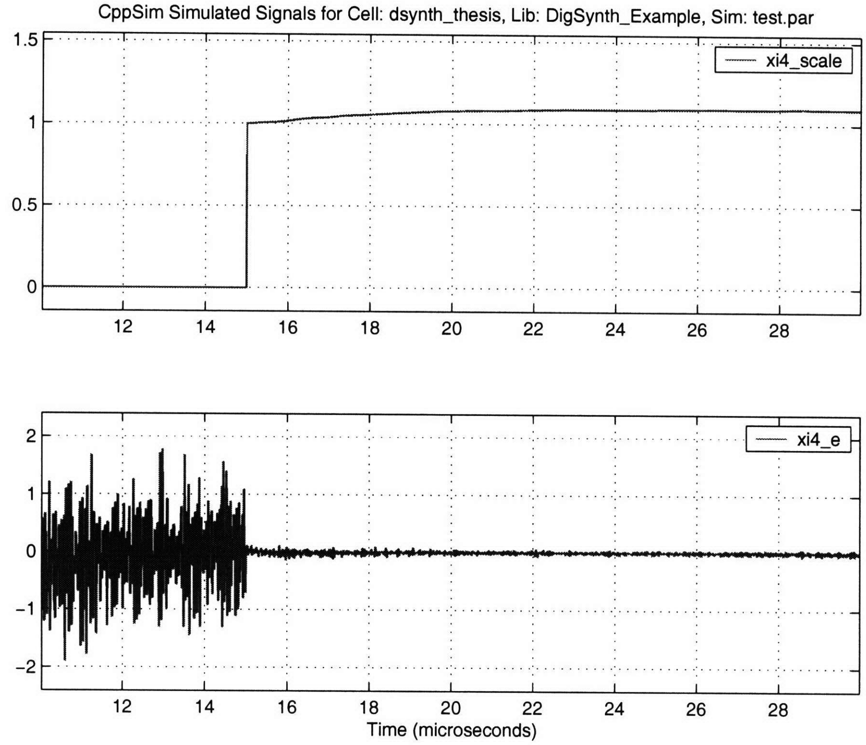


Figure 6-13: Simulated *scale factor* and phase error signal $e[k]$ with the noise cancellation enabled at t=15μs.

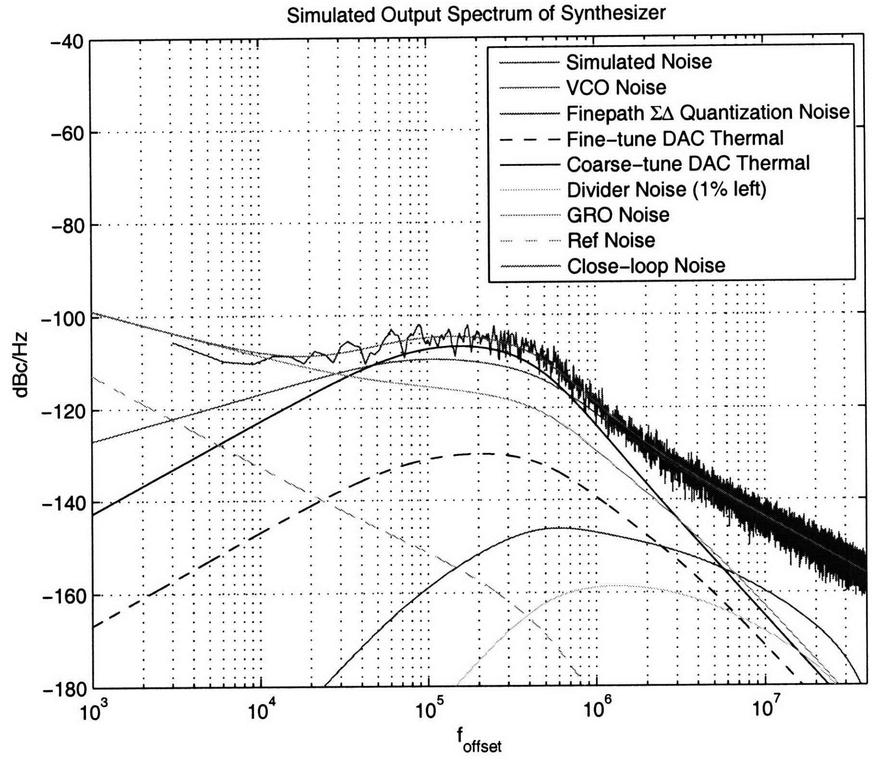


Figure 6-14: Comparison between the calculated noise with MATLAB and simulated noise with *CppSim*.

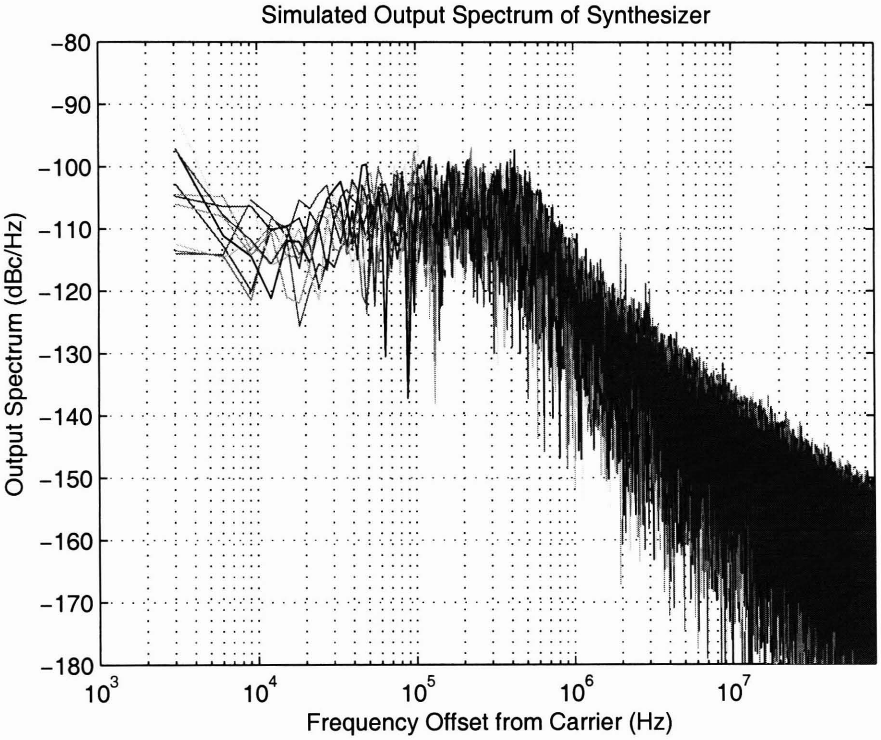


Figure 6-15: Ten phase noise simulation results with a 5% device standard deviation.

Chapter 7

Digital Synthesizer Measurement

This chapter demonstrates the performance of the proposed digital synthesizer, including the area and power, noise and spur performance, and locking time. We also compare the measured phase noise with that obtained with the frequency-domain analysis. Comparison among this chip and other published digital synthesizers as well as analog synthesizers utilizing noise cancellation are also given.

7.1 Area and Power Dissipation

To verify the techniques presented in this thesis, a prototype chip with its die photo shown in Figure 7-1 is implemented in a 0.13-μm CMOS process. The chip has a total area of 1.4×1.4 mm^2 and an active area of 0.95 mm^2, of which the GRO TDC occupies 157×252 μm^2. Each DAC occupies 240×180 μm^2, and the loop filter occupies 160×190 μm^2. Although the proposed synthesizer uses two DACs, the area of the digital loop filter plus these two DACs is still less than one fourth of that of an analog loop filter in a 730-kHz bandwidth PLL [26]. This demonstrates the main advantage of using a digital PLL structure.

The die is bonded on the printed circuit board directly for testing (i.e., no package is used). A photo of the evaluation board is shown in Figure 7-2. As shown in this figure, a FPGA board, which is used to generate the control signals for the synthesizer, is connected to the main evaluation board.

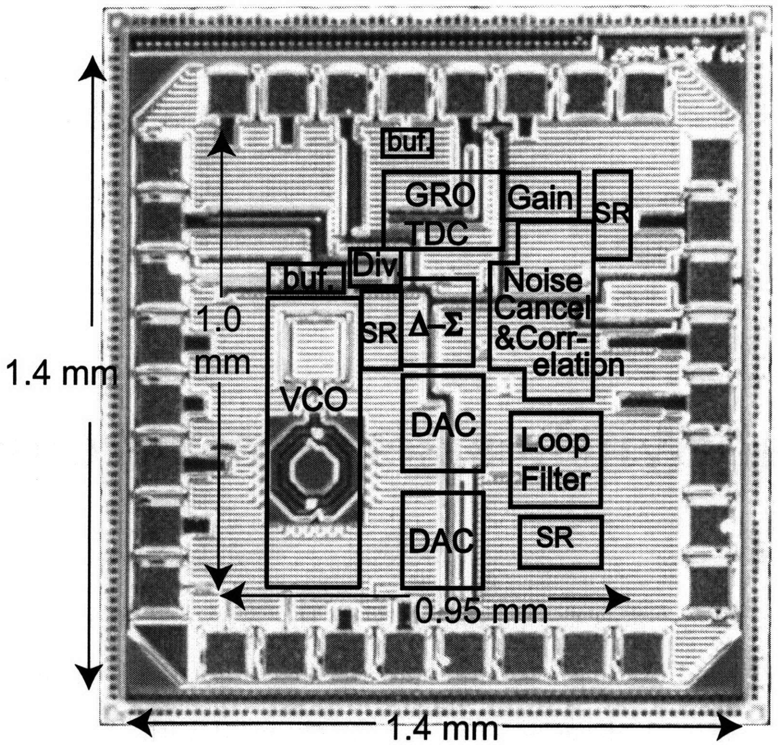


Figure 7-1: The active area of the implemented 0.13-μm digital frequency synthesizer is 0.95 mm^2.

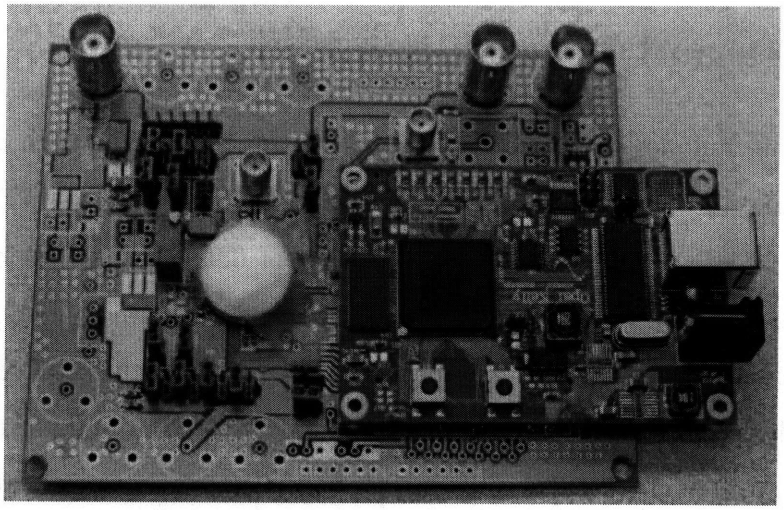


Figure 7-2: Photo of the evaluation board.

The chip has 32 pads in total: eight of them are used for separate V_{DD}; another eight of them are used for grounds, but these grounds are connected together within the chip; the rest of the pads are used for signals. Suitable ESD circuits are allocated to different types of pads. Table 7.1 summarizes the measured current dissipation of the core circuits operating at the supply voltage of 1.5 V. The overall current consumption of the core circuits is 26 mA, excluding the VCO pad buffer which consumes 7 mA from a 1.1-V supply. Assuming a steady-state time offset of about 1.2 ns between the *start* and *stop* edges of the GRO (i.e., 4 to 5 VCO cycles), the GRO dissipates 2.3 mA. This offset value is programmable in the prototype and is set to a small value to both lower the average GRO power dissipation as well as its in-band noise. A pie chart in Figure 7-3 illustrates the distribution of the total power consumption in this chip. Note that the V_{DD} pad used for the digital I/O buffer consumes no power in the steady state, so its power consumption is not included in the table or chart.

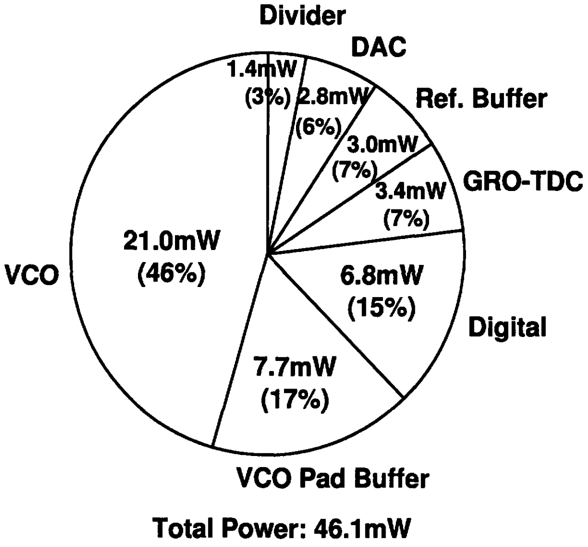


Figure 7-3: Power distribution of the chip.

7.2 VCO Gain

We first measure the DCO frequency range across different MIM capacitor values with the results shown in Figure 7-4. The fine-tuning code of the DCO is set to 512, while three coarse-tuning codes (0, 512 and 1023) are swept to get a rough sense of the frequency range that each band provides. Notice that the bank is named to represent the number of unit MIM capacitors switched into the tank. The frequency increases as the number of unit MIM capacitors decreases. The results show that the DCO covers a wide frequency range from 3.15 GHz to 4.23 GHz, and the curves are overlapped properly.

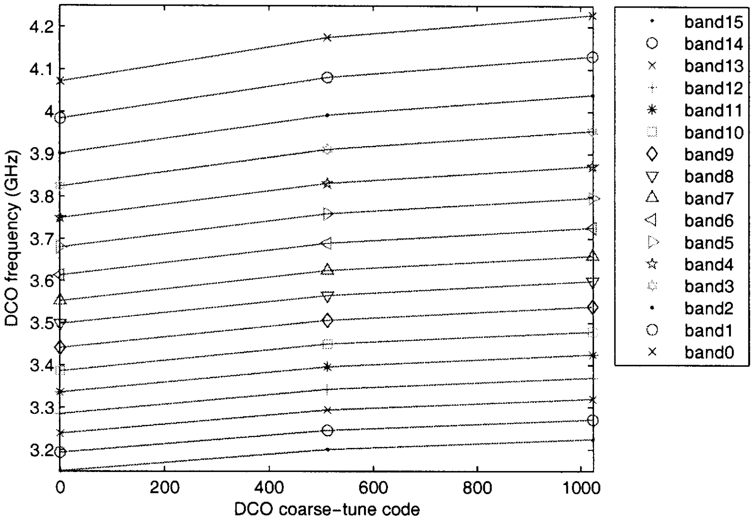


Figure 7-4: Measured frequency range of the DCO (fine-tuning code is set to 512).

In the rest of this chapter, we focus on band7 to characterize the PLL performance. Figure 7-5 illustrates the measured coarse-tuning DCO frequencies and the extracted

Table 7.1: Measured Current Dissipation

Block	Current(mA)
VCO	14.027
Digital	4.507
GRO-TDC	2.263
Reference Buffer	2.001
DAC	1.875
Divider	0.959
Total	25.63

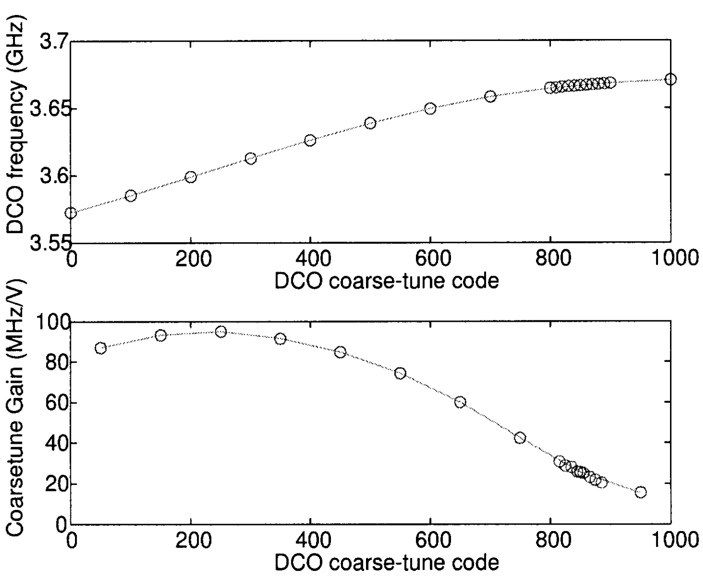


Figure 7-5: Measured DCO frequency at band7 and the extracted coarse-tuning analog VCO gain (The fine-tuning code is set to 512).

analog coarse-tuning VCO gain (i.e., DCO gain divided by DAC gain ($1.5V/2^{10}$). It can be seen that the VCO gain is about 80 MHz/V in the middle of the tuning range but decreases as the tuning code and analog control voltage increase. Calculated VCO gain is about 29 MHz/V when the coarse-tuning code is around 825. The best measured phase noise in the next section is biased around this point.

Figure 7-6 shows the measured fine-tuning DCO frequencies and the extracted *analog* fine-tuning VCO gain, when the coarse-tuning code is set to 825. The fine-tuning VCO gain is about 5 MHz/V in the middle supply, which is 16 times lower than the coarse-tuning gain as expected.

7.3 Phase Noise and Spurs

The synthesizer is first tested with a 50-MHz reference clock. Figure 7-7 illustrates the measured open-loop phase noise of the DCO (i.e., VCO and DAC) at 3.6657 GHz from an Agilent Signal Source Analyzer E5052A. This measurement reveals that the DCO achieves -115 and -151 dBc/Hz phase noise at 400 kHz and 20 MHz, respectively,

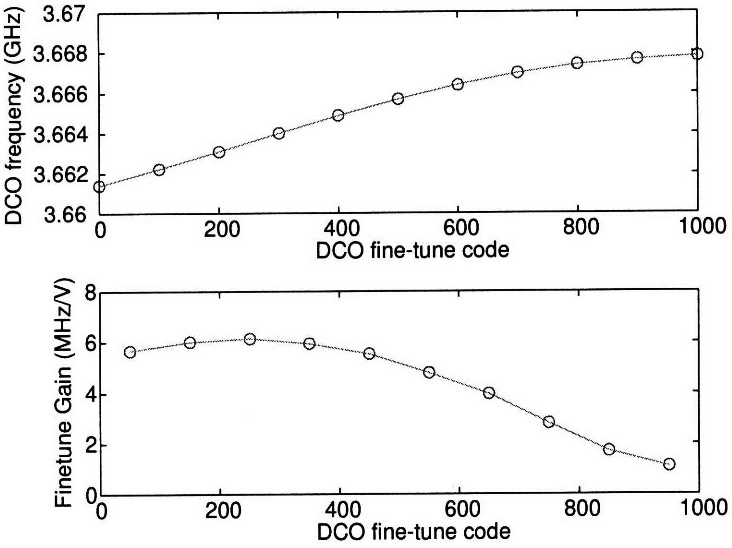
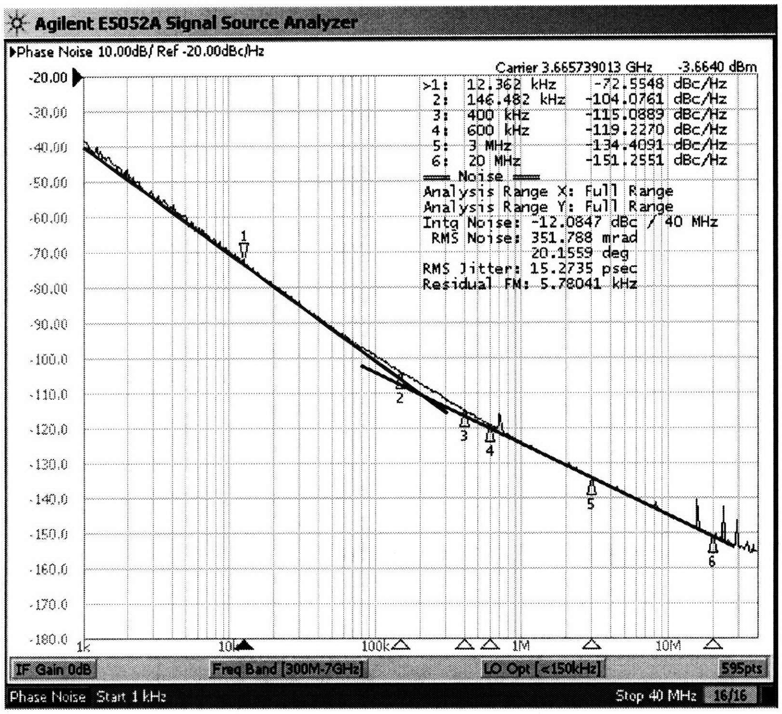


Figure 7-6: Measured DCO frequency at band7 and the extracted fine-tuning analog VCO gain (The coarse-tuning code is set to 825).

with flicker corner around 200 kHz.

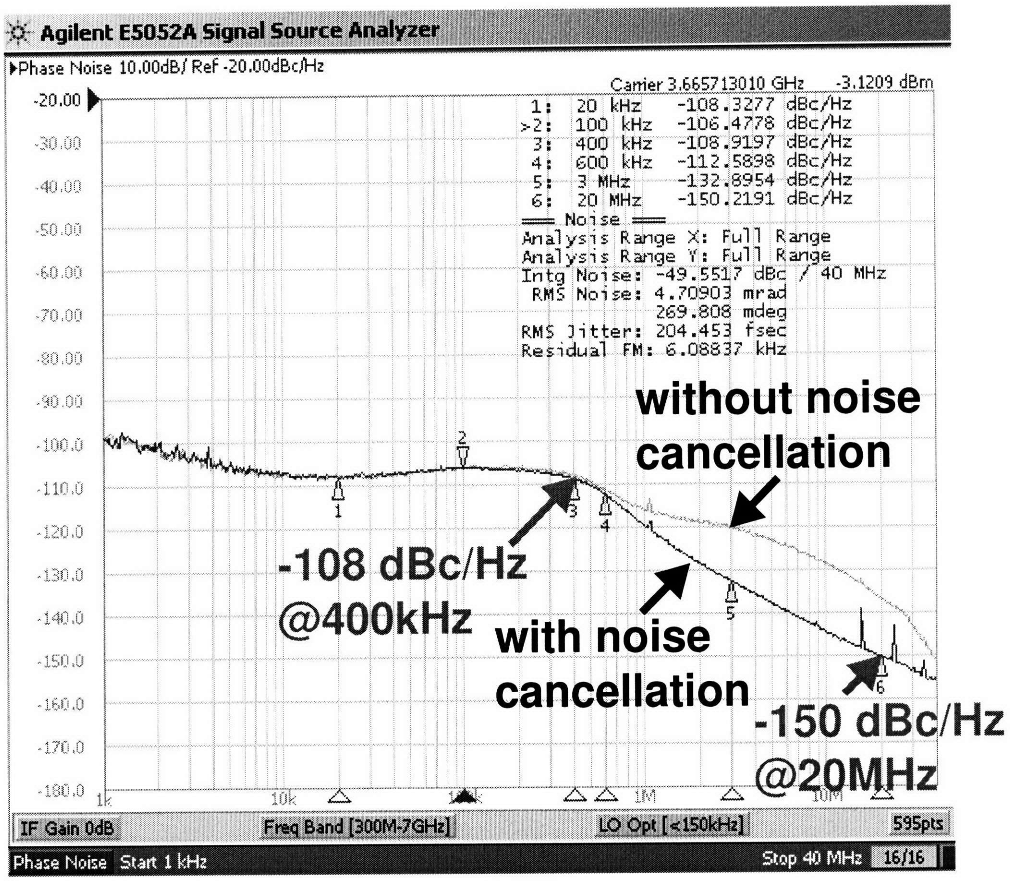


```
                              Carrier 3.665739013 GHz   -3.6640 dBm
>1:  12.362 kHz       -72.5548 dBc/Hz
 2:  146.482 kHz     -104.0761 dBc/Hz
 3:  400 kHz         -115.0889 dBc/Hz
 4:  600 kHz         -119.2270 dBc/Hz
 5:   3 MHz          -134.4091 dBc/Hz
 6:  20 MHz          -151.2551 dBc/Hz
═══ Noise ═══
Analysis Range X: Full Range
Analysis Range Y: Full Range
Intg Noise: -12.0847 dBc / 40 MHz
 RMS Noise: 351.788 mrad
            20.1559 deg
RMS Jitter: 15.2735 psec
Residual FM: 5.78041 kHz
```

```
Start 1 kHz                                   Stop 40 MHz
```

Figure 7-7: Measured DCO phase noise at 3.67 GHz

Figure 7-8 shows the best measured closed-loop phase noise at 3.6657 GHz, where the results are shown with and without cancellation of the quantization noise. As the figure reveals, greater than 15 dB noise cancellation is achieved such that the out-of-band noise is dominated by the VCO. With the noise cancellation enabled, the in-band noise is -108 dBc/Hz at 400 kHz offset, and the out-of-band noise is -132 and -150 dBc/Hz at 3 and 20 MHz offsets, respectively. The integrated noise from 1 kHz to 40 MHz is 204 fs at this frequency. Since the jitter number is not usually reported for a frequency synthesizer, we estimate the corresponding jitter of the phase noise plot in [9], with PLL Design Assistant, and obtain a value of 1.5 ps for comparison. Therefore, our jitter (204 fs) is more than five times better than that of [9].



```
Carrier 3.665713010 GHz     -3.1209 dBm
  1:  20 kHz    -108.3277 dBc/Hz
 >2:  100 kHz   -106.4778 dBc/Hz
  3:  400 kHz   -108.9197 dBc/Hz
  4:  600 kHz   -112.5898 dBc/Hz
  5:  3 MHz     -132.8954 dBc/Hz
  6:  20 MHz    -150.2191 dBc/Hz
=== Noise ===
Analysis Range X: Full Range
Analysis Range Y: Full Range
Intg Noise: -49.5517 dBc / 40 MHz
 RMS Noise: 4.70903 mrad
            269.808 mdeg
RMS Jitter: 204.453 fsec
Residual FM: 6.08837 kHz
```

Figure 7-8: Measured PLL phase noise at 3.67 GHz.

We now compare the measured results to the calculated performance with the analysis model presented in Chapter 6. Figures 7-9 and 7-10 illustrate the cases where the noise cancellation function is enabled and disabled, respectively. Note that Δt_{floor} and $K_{flicker}$ in equation 6.12 are picked to match the measured results directly, since these two numbers are difficult to estimate in the design phase. In addition, in this measurement, the VCO coarse-tuning gain decreases to 29 MHz/V from the nominal value of 80 MHz/V because a higher carrier frequency is set. Measured results match the analysis model very well except at the intermediate frequencies. It is possible that this extra noise is caused by digital ground and substrate noises that couple to the VCO output through the coarse-tuning DAC.

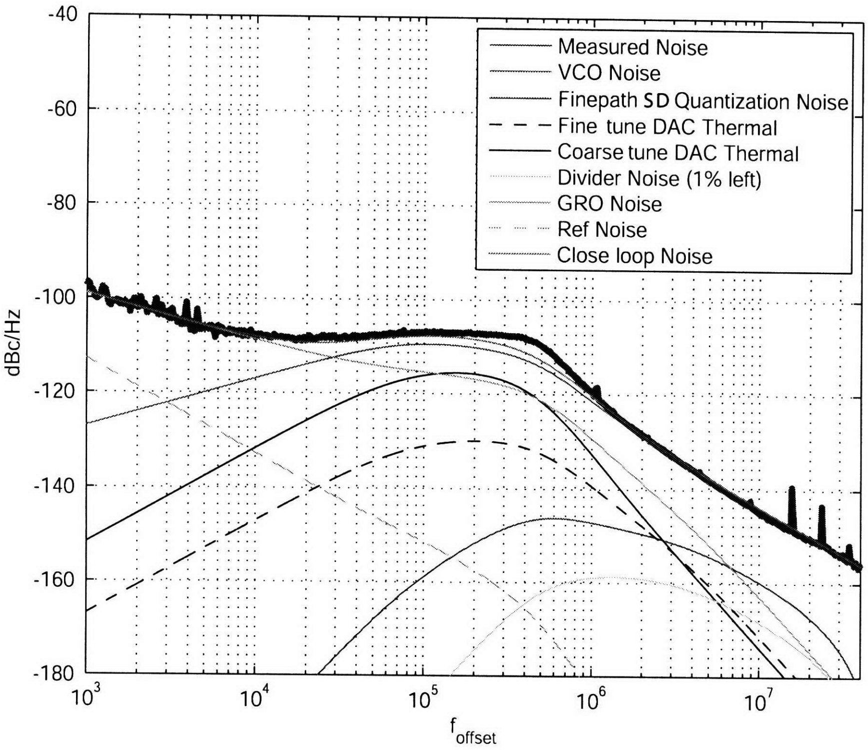


Figure 7-9: Comparison between the measured and calculated noises with the noise cancellation.

We now demonstrate another case when the VCO frequency is set to 3.638 GHz. As illustrated in Figure 7-5, the coarse-tuning VCO gain corresponding to this frequency is about 80 MHz/V, as assumed in Table 6.1. The measured phase noise at 3.638 GHz is depicted in Figure 7-11 overlapped with that at 3.6657 GHz. One should

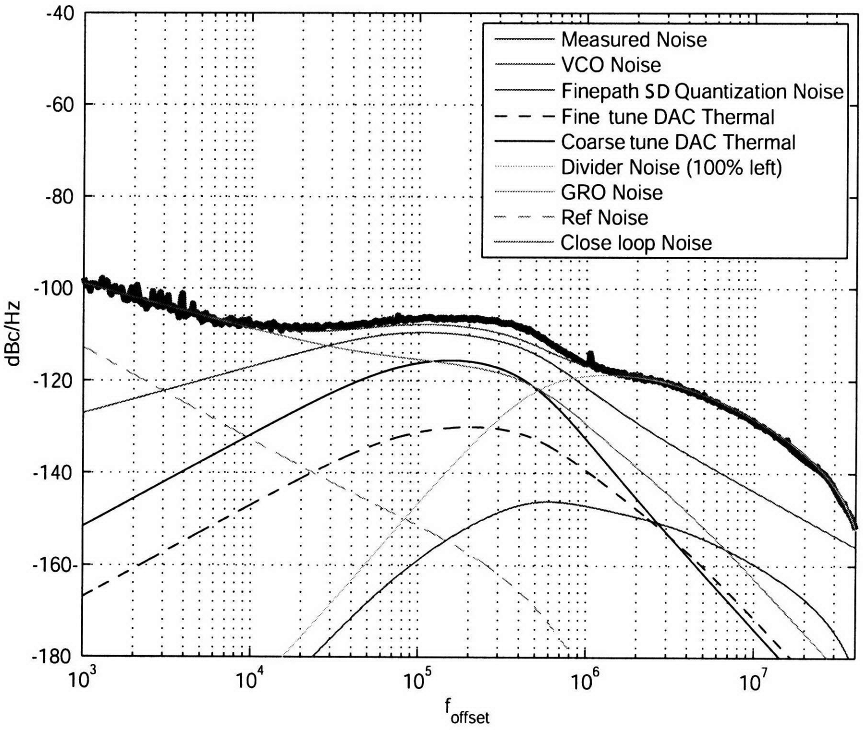
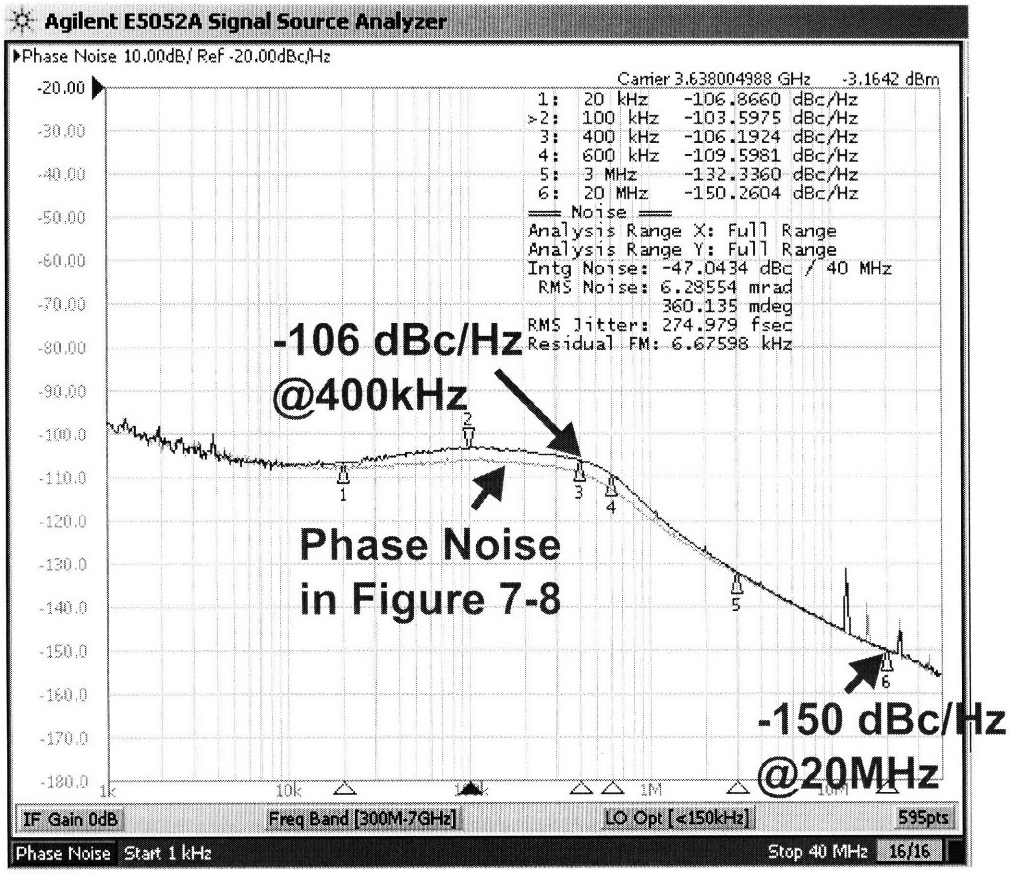


Figure 7-10: Comparison between the measured and calculated noises without the noise cancellation.

see that the phase noise at intermediate frequency offsets becomes higher because of the higher coarse-tuning VCO gain and noise. This measurement is also compared with the calculated noise in Figure 6-5, as shown in Figure 7-12.

The phase noise is also tested from 3.620 to 3.670 GHz with intervals of 1 MHz. As illustrated in Figure 7-13, the phase noise at 400 kHz offset as well as the integrated noise (i.e., jitter) degrade as the carrier frequency is lowered, but the overall jitter still remains less than 300 fs for most of that frequency range. The degradation of the phase noise comes from two possible reasons. First, the thermal noise of the coarse-tuning DAC is amplified more as the VCO gain increases (Figure 7-5). Second, we suspect that the switching noise of the digital circuits, which couples to the VCO output through the common ground, is also amplified as the VCO control voltage decreases. To explain that, in the DAC structure in Figure 3-2, the ground is connected together with that of the digital filter. Therefore, the digital noise may



```
Carrier 3.638004988 GHz     -3.1642 dBm
 1:  20 kHz    -106.8660 dBc/Hz
>2:  100 kHz   -103.5975 dBc/Hz
 3:  400 kHz   -106.1924 dBc/Hz
 4:  600 kHz   -109.5981 dBc/Hz
 5:  3 MHz     -132.3360 dBc/Hz
 6:  20 MHz    -150.2604 dBc/Hz
=== Noise ===
Analysis Range X: Full Range
Analysis Range Y: Full Range
Intg Noise: -47.0434 dBc / 40 MHz
 RMS Noise: 6.28554 mrad
            360.135 mdeg
RMS Jitter: 274.979 fsec
Residual FM: 6.67598 kHz
```

Figure 7-11: Measured PLL phase noise at 3.638 GHz.

couple to the resistor ladder output with a scale factor of $(32-M)/32$. Since a smaller value of M is necessary to support a lower carrier frequency, the digital switching noise begins to have a more serious impact on the PLL noise when the carrier frequency is lower. Furthermore, the measured worst-case phase noise at 3 and 20 MHz offsets are -131.5 and -148.5 dBc/Hz, respectively, in this frequency range.

The reference spur is measured with an Agilent Spectrum Analyzer 8595E to be -65 dBc at 3.67 GHz, as illustrated in Figure 7-14.

Fractional spurs are also tested from 3.620 to 3.670 GHz with intervals of 1 MHz, as illustrated in Figure 7-15. The worst-case spurs occur close to the integer boundary and are measured to be -53 dBc at carrier frequencies of 3.649 and 3.651 GHz, -64 dBc at carrier frequencies of 3.648 and 3.652 GHz, and are less than -65 dBc at all the other carrier frequencies. Note that the frequency offset at which the worst spur

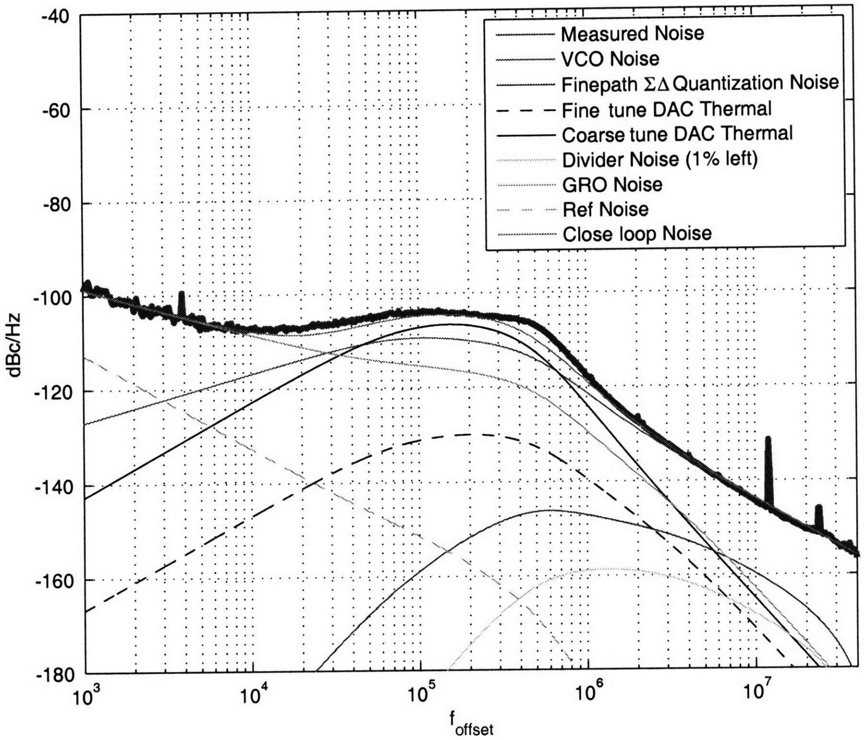


Figure 7-12: Comparison between the measured and calculated noises at 3.638 GHz.

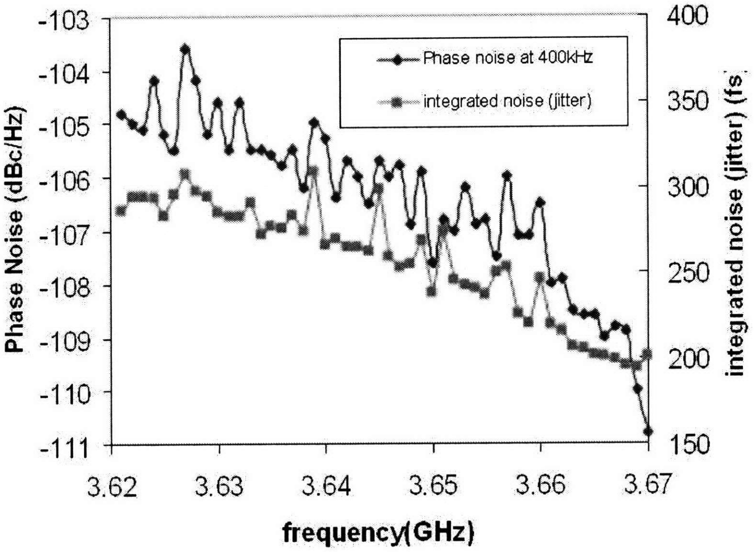


Figure 7-13: Measured jitter and phase noise at 400 kHz offset over a 50-MHz range with 1-MHz increments.

is seen is also recorded. When the carrier frequency is more than 3 MHz away from 3.65 GHz, the worst spur usually occurs around 1 MHz. It is suspected that this spur around 1 MHz is from the FPGA and coupled to the VCO output through the common ground on the board.

At carrier frequencies less than 1 MHz away from the integer boundary (3.65 GHz), worst-case fractional spurs are measured to be -42 dBc at a 400 kHz offset frequency (i.e., a carrier frequency of 3.6504 GHz with spurs at 3.6500 and 3.6508GHz), as illustrated in Figure 7-16. In addition, snapshots of the measured spurs when the VCO frequency is 3.651 and 3.6504 GHz are shown in Figure 7-17. Note that although spurs shown in Figure 7-17 are slightly better than the claimed numbers (-53 and -42 dBc), the claimed numbers are the worst numbers ever seen at these two carrier frequencies.

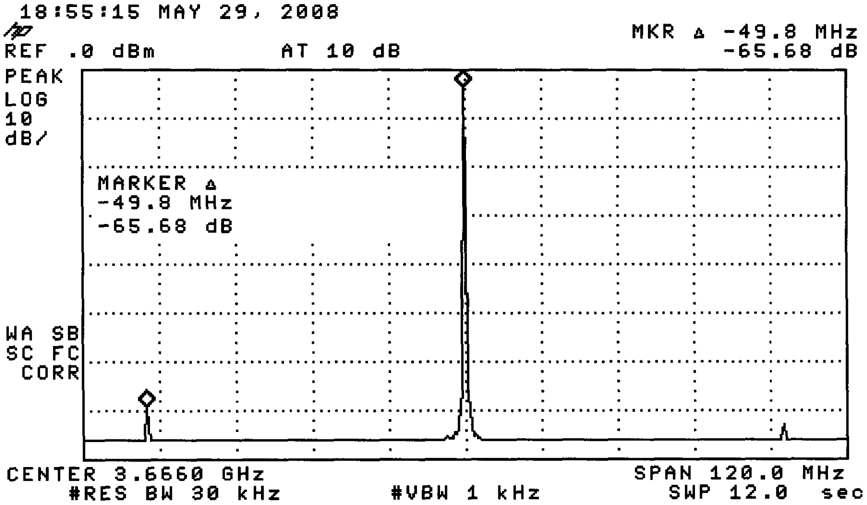


Figure 7-14: Measured reference spur when the VCO frequency is 3.67 GHz.

The phase noise performance at 3.67 GHz with a lower reference clock is also measured, as illustrated in Figure 7-18. The lowest frequency supported in the prototype is 30.5 MHz due to a limitation on the divider range. At this reference frequency, the PLL bandwidth scales to about 300 kHz in proportion to the reference clock, and proper adjustment of the open-loop gain of the PLL is required to maintain the stability. Although the in-band noise becomes higher, the phase noise at 400 kHz can

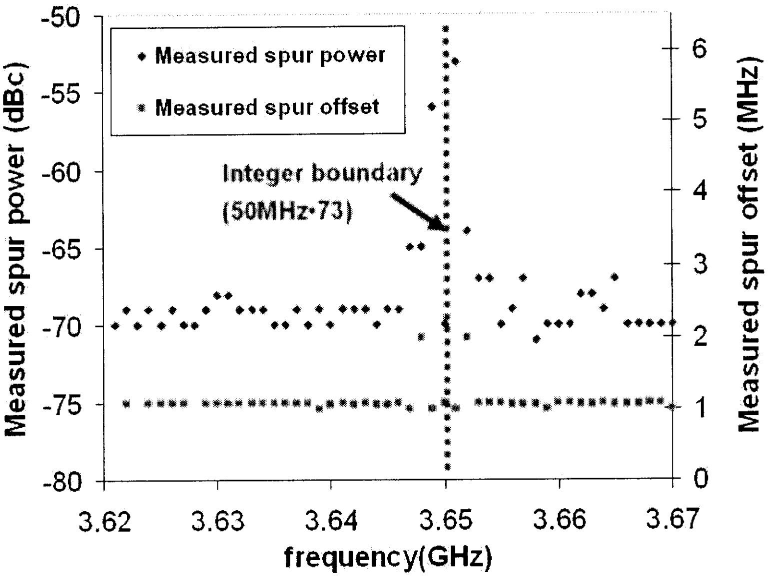


Figure 7-15: Measured worst-case fractional spurs over a 50-MHz range with 1-MHz increments.

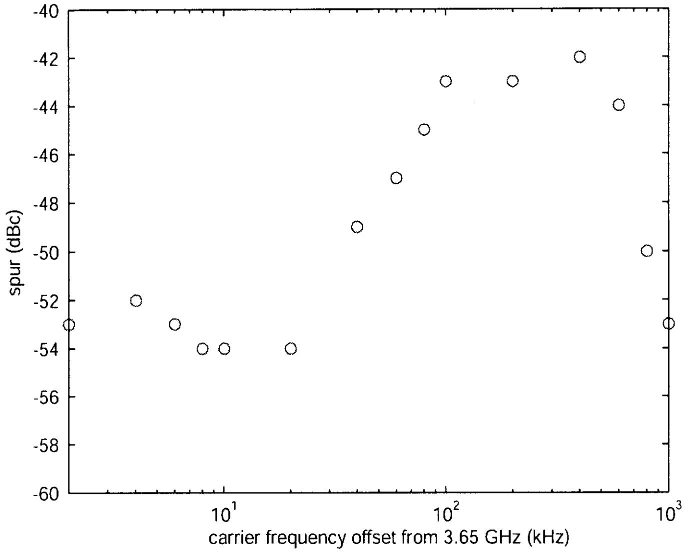


Figure 7-16: Measured worst-case fractional spurs when the carrier frequency is less than 1 MHz away from 3.65 GHz.

still achieve -106 dBc/Hz.

Figure 7-19 reveals the phase noise performance at 4.1 GHz with a 50-MHz reference clock. The measured DCO open-loop phase noise at this frequency is -113 and

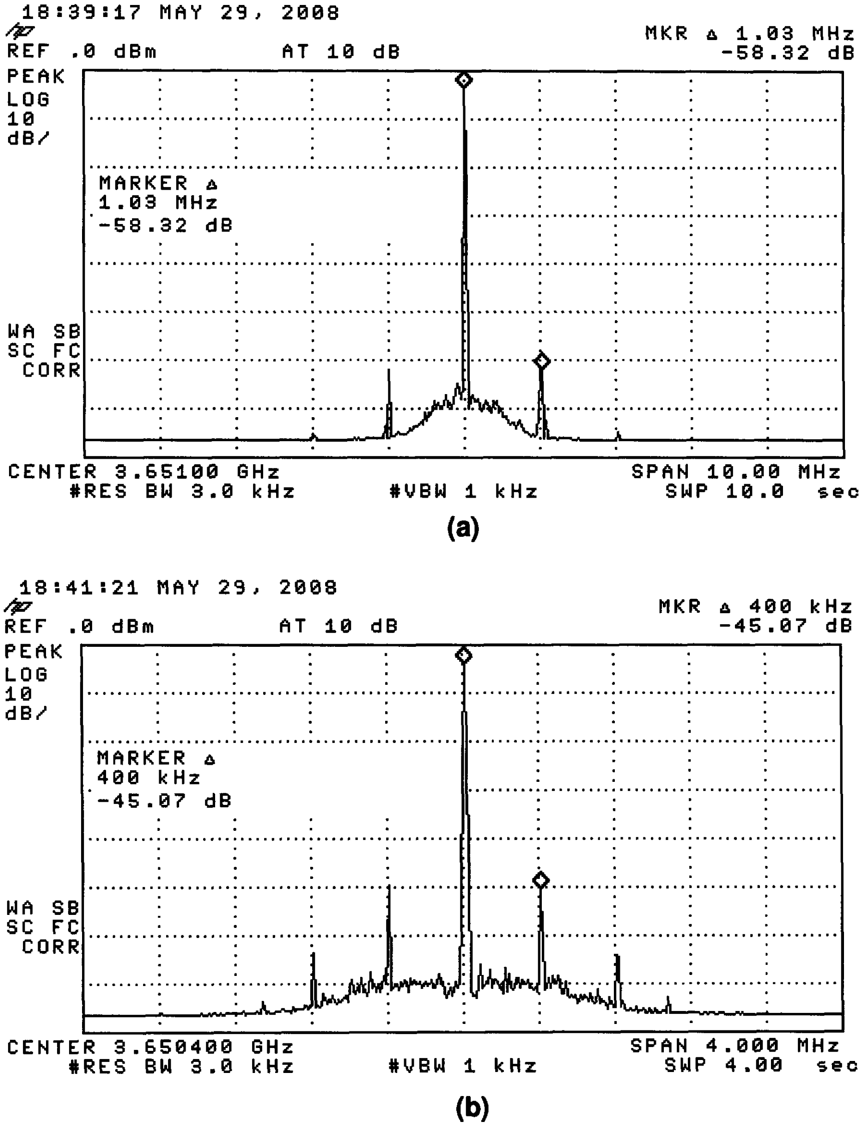


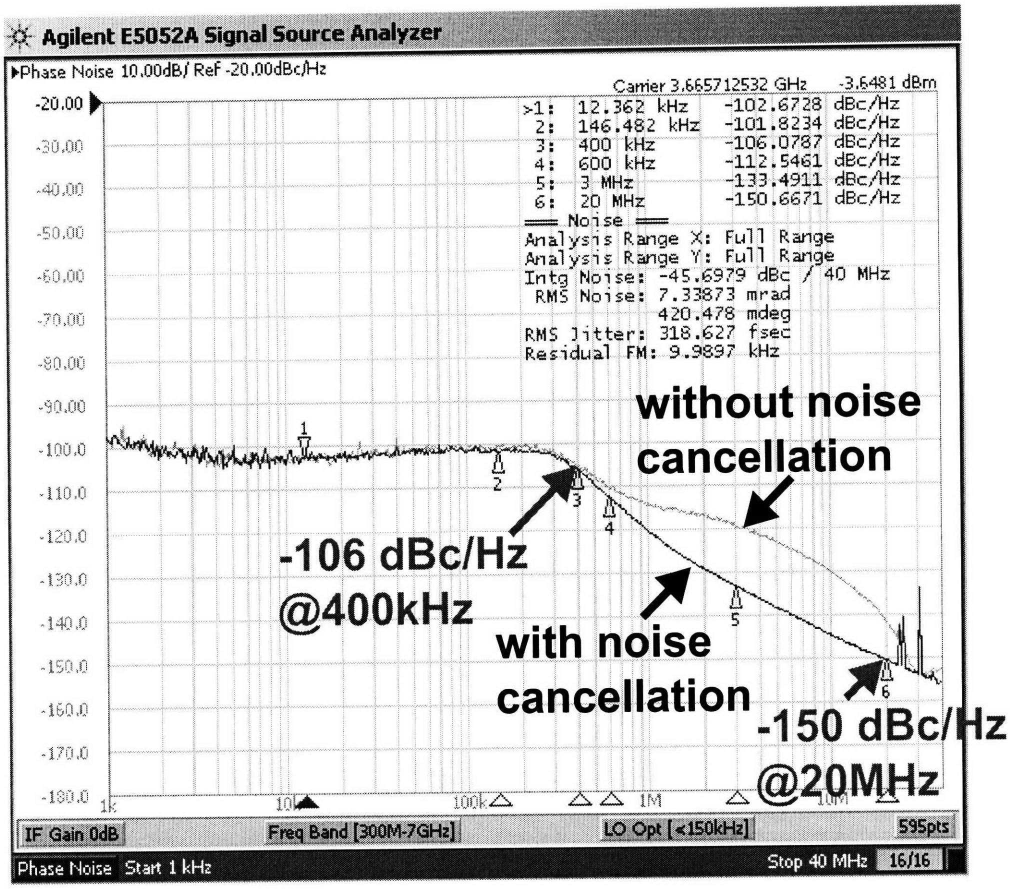
Figure 7-17: Measured fractional spur when the VCO frequency is (a) 3.651 GHz (b) 3.6504 GHz.

-150 dBc/Hz at 400 kHz and 20 MHz offsets, respectively.

7.4 Locking Time

Finally, a settling time of 20 μs for 10-ppm accuracy is measured when a frequency step of 20 MHz is applied to the synthesizer, as illustrated in Figure 7-20.

If the time period assigned to the coarse-tuning phase is extended on purpose,



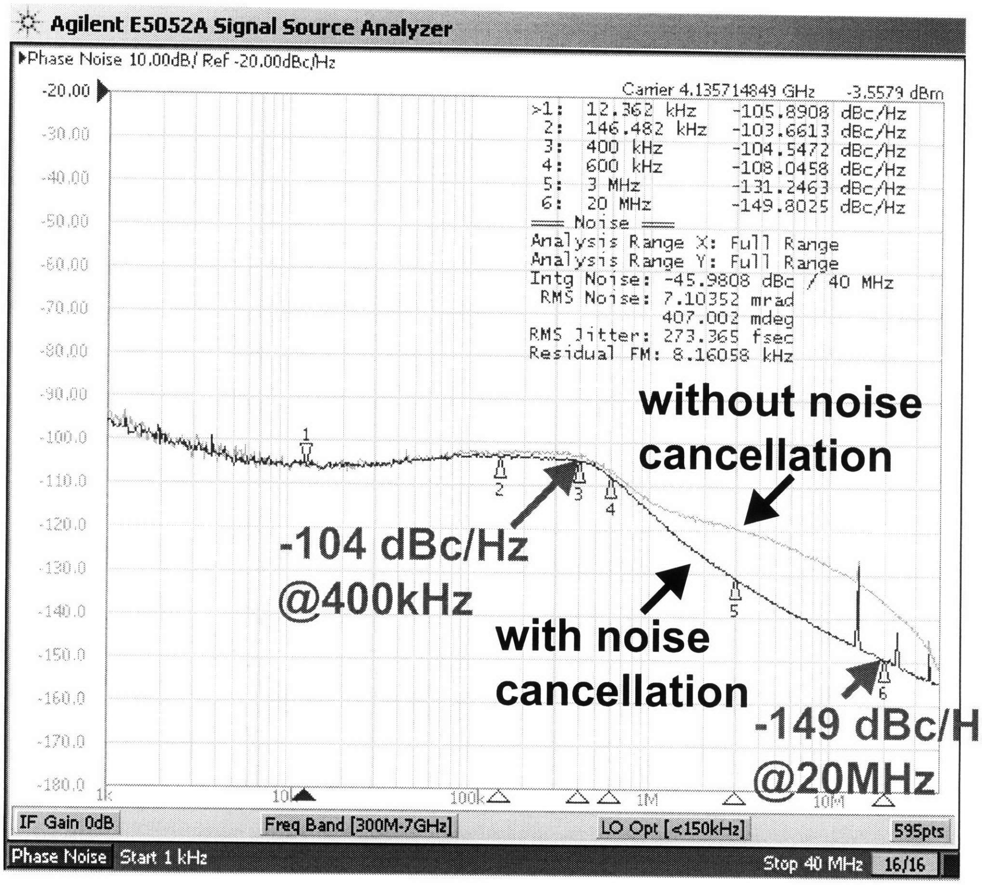
```
Carrier 3.665712532 GHz     -3.6481 dBm
>1:  12.362 kHz     -102.6728  dBc/Hz
 2:  146.482 kHz    -101.8234  dBc/Hz
 3:  400 kHz        -106.0787  dBc/Hz
 4:  600 kHz        -112.5461  dBc/Hz
 5:  3 MHz          -133.4911  dBc/Hz
 6:  20 MHz         -150.6671  dBc/Hz
=== Noise ===
Analysis Range X: Full Range
Analysis Range Y: Full Range
Intg Noise: -45.6979 dBc / 40 MHz
 RMS Noise: 7.33873 mrad
            420.478 mdeg
RMS Jitter: 318.627 fsec
Residual FM: 9.9897 kHz
Start 1 kHz
Stop 40 MHz
```

Figure 7-18: Measured phase noise at 3.67 GHz with a 30.5-MHz reference clock.

it can be seen that the frequency toggles between two frequencies after the coarse tuning is completed, as illustrated in Figure 7-21, due to the insufficient resolution and lack of a $\Delta\Sigma$ modulator on the coarse-tuning path. This frequency step is about 100 kHz, which is close to the calculated value in Section 5.4.

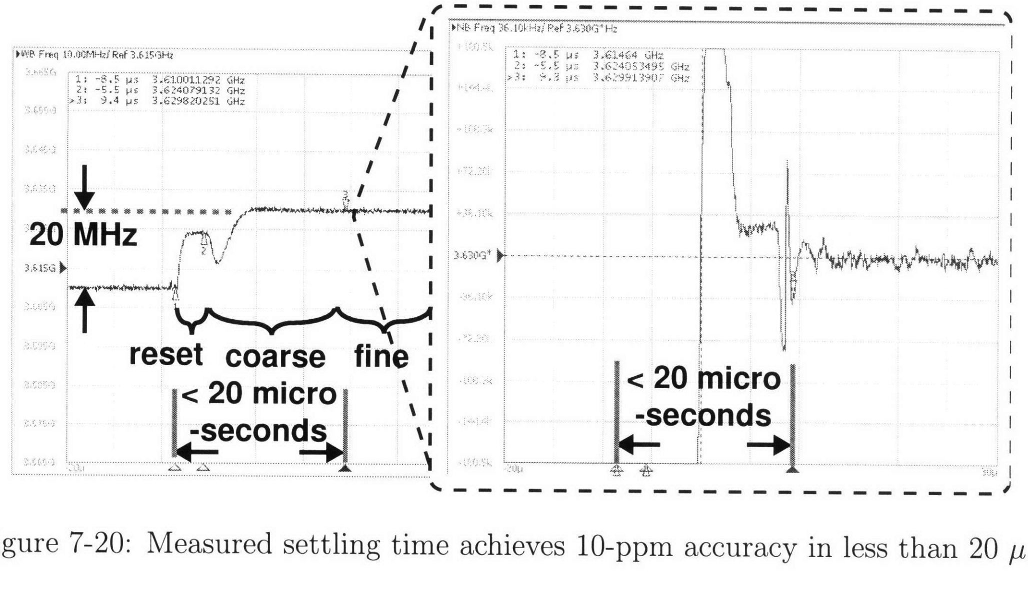
7.5 Comparison

Table 7.2 displays a comparison of the synthesizer to other recently published digital frequency synthesizers. Note that the original reported phase noises are normalized to 3.6 GHz by adding $20log(3.6GHz/(carrier))$ in this table for a fair comparison. According to this table, we can conclude that the proposed synthesizer achieves excellent phase noise, especially within the loop bandwidth, as well as jitter performance.



```
Carrier 4.135714849 GHz     -3.5579 dBm
>1:  12.362 kHz     -105.8908 dBc/Hz
 2:  146.482 kHz    -103.6613 dBc/Hz
 3:  400 kHz        -104.5472 dBc/Hz
 4:  600 kHz        -108.0458 dBc/Hz
 5:  3 MHz          -131.2463 dBc/Hz
 6:  20 MHz         -149.8025 dBc/Hz
=== Noise ===
Analysis Range X: Full Range
Analysis Range Y: Full Range
Intg Noise: -45.9808 dBc / 40 MHz
 RMS Noise: 7.10352 mrad
            407.002 mdeg
RMS Jitter: 273.365 fsec
Residual FM: 8.16058 kHz
```

Figure 7-19: Measured phase noise at 4.135 GHz with a 50-MHz reference clock.



```
 1: -8.5 µs  3.610011292 GHz
 2: -5.5 µs  3.624079132 GHz
>3:  9.4 µs  3.629820251 GHz
```

```
 1: -8.5 µs  3.61464 GHz
 2: -5.5 µs  3.624053495 GHz
>3:  9.3 µs  3.629913907 GHz
```

Figure 7-20: Measured settling time achieves 10-ppm accuracy in less than 20 μs.

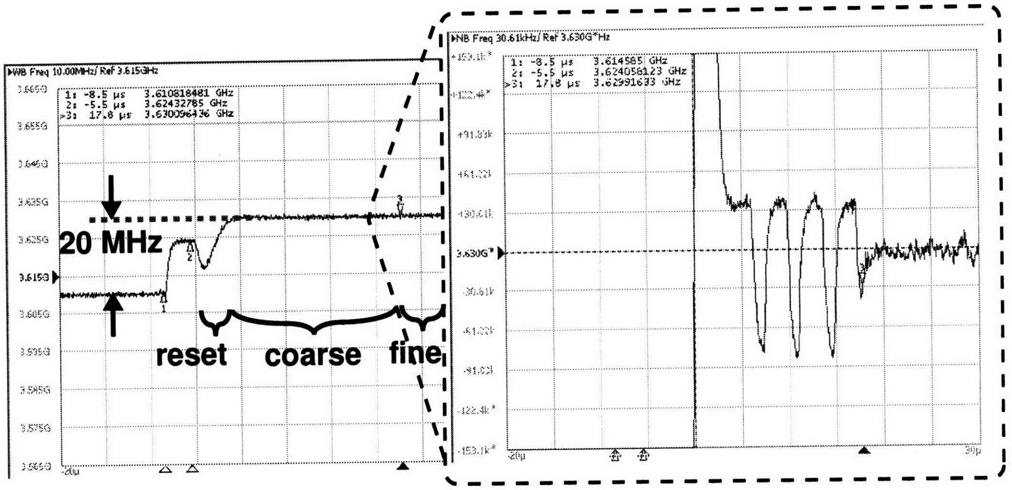


Figure 7-21: Frequency toggles between two levels after the coarse-tuning is performed.

Table 7.2: Comparison Among Published Digital Synthesizers

	[9]	[10]	[11]	[12]	[13]	This Work
Technology(μm)	0.09	0.13	0.13	0.13	0.065	0.13
Reference Freq.(MHz)	26	40	185.5	26	25	50
Carrier(GHz)	0.9	2	2.2	3.6	3	3.67
Bandwidth(kHz)	40	3000	142	50	1200	500
Phase Noise(in-band) at	-81 30kHz	-96.9 400kHz	-76 30kHz	-78 10kHz	-100	-108 400kHz
Phase Noise(400kHz)	-110	-96.9	-92	-116	N/A	-108
Phase Noise(20MHz)	-153	-135	N/A	-152	N/A	-150
Jitter(ps)	N/A	N/A	N/A	N/A	N/A	0.2
Reference Spur(dBc)	-92	N/A	N/A	-84	N/A	-65
Fractional Spur(dBc)	N/A	-42	N/A	under phase noise level	-45	-42
Locking Time(μs)	10	N/A	N/A	N/A	N/A	20
Power	50.4mW	25mW	14mW	40mA	9.5mW	46.7mW
Active Area(mm^2)	1.5	0.8	0.7	0.86	0.4	0.95

First, this table reveals that the in-band noise floor of the proposed digital synthesizer is much better than other solutions because we leverage a low-noise high-resolution GRO TDC. Phase noise at 400 kHz is 2 dB and 8 dB worse than that in [9] and [12], respectively. Our open-loop VCO phase noise that is -115 dBc/Hz at 400 kHz offset is actually close to that of [9] and [12], but our 500-kHz bandwidth causes slight peaking around the loop bandwidth, resulting in a higher closed-loop phase noise at 400 kHz offset.

As for the phase noise at 20 MHz offset, although our result only achieves -150 dBc/Hz, which is 2 and 3 dB worse than that of [9] and [12], respectively, it is clear that the performance is limited by the intrinsic VCO noise instead of the proposed noise cancellation technique. Therefore, this performance can be improved with a more carefuly designed VCO. Measured results of an improved design are shown in Section 7.6.

The integrated noise (jitter) of the proposed approach is less than 300 fs. Again, since jitter is not usually reported for a frequency synthesizer, we estimate the jitter of [9] to be around 1.5 ns. Therefore, the jitter of the proposed wide-bandwidth synthesizer is about five times better than that of a narrow-bandwidth synthesizer.

In addition, although our chip is designed with a relatively higher reference frequency (50 MHz) than that of other chips, it is demonstrated that the chip can also use a 30.5-MHz reference clock, and the phase noise at 400 kHz offset degrades by 2 dB at 3.67 GHz. In this case, the in-band noise floor is still better than those in other works. The phase noise at 20 MHz offset does not change significantly when the reference clock is decreased.

Table 7.3 compares the performance of this chip with earlier reported analog PLL utilizing phase noise cancellation techniques. Again, originally reported phase noise numbers are normalized to 3.6 GHz. From this table, the noise performance of our chip is better than others. Although the phase noise at 20 MHz offset is 5 dB better in [24], an off-chip VCO was used in that work.

In addition, the calibration time of this work is three and 100 times better than [26] and [25], respectively.

As mentioned earlier, the main advantage of the digital PLL is the resulting area savings, as revealed in Table 7.3. The die area of this chip, which includes the digital loop filter, is smaller than those in all the other works. Although the die area of [25] is close to ours, an off-chip filter was used in that work. Note that even though the proposed synthesizer uses two DACs, the area of the digital loop filter plus these two

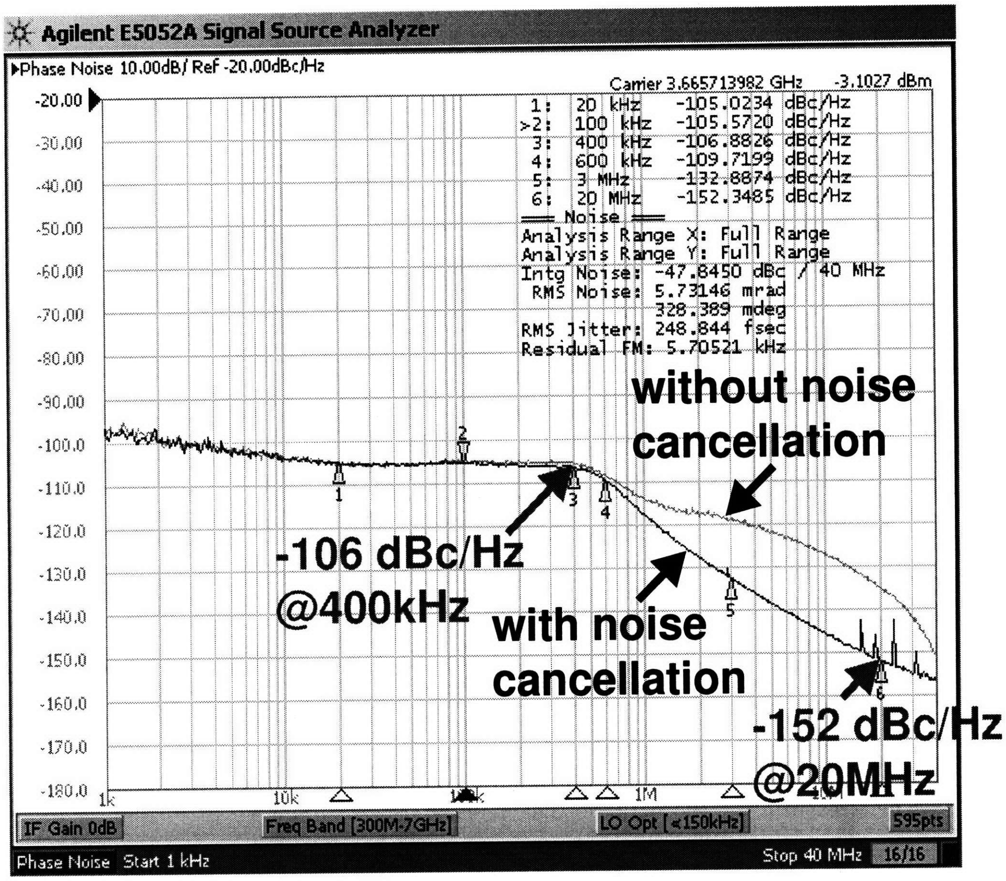
Table 7.3: Comparison Among Published Analog Noise Cancellation Synthesizers

	[22]	[23]	[24]	[25]	[26]	This Work
Technology(μm)	0.18	0.18	0.18	0.18	0.18	0.13
Reference Freq.(MHz)	48	35	50	14.3	12	50
Carrier(GHz)	2.4	2.1	3.6	1.8	2.4	3.67
Bandwidth(kHz)	460	700	1000	400	730	500
Phase Noise(100kHz)	N/A	-100	-97.8	N/A	-97.3	-106
Phase Noise(1MHz)	N/A	N/A	N/A	-111.8	N/A	-120
Phase Noise(3MHz)	-117.5	N/A	N/A	N/A	-120.3	-131
Phase Noise(20MHz)	N/A	N/A	-154.8	N/A	N/A	-150
On-Chip VCO?	Yes	Yes	No	Yes	Yes	Yes
On-Chip Filter?	No	Yes	No	No	Yes	Yes
Reference Spur(dBc)	N/A	N/A	-74	-75	-53	-65
Fractional Spur(dBc)	-54	-60	-45	N/A	-39	-42
Locking Time(μs)	N/A	7	N/A	N/A	N/A	20
Calibration Time(μs)	None	None	None	1000	35	10
Power(mW)	88	28	181	29	54.5	46.7
Active Area(mm^2)	N/A	N/A	2.7	N/A	N/A	0.95
Die Area(mm^2)	6.72	3.4	7.29	2	4.84	1.96

DACs is still only about one-fourth of the area of the analog loop filter in [26].

7.6 Improved phase noise at 20 MHz offset

As mentioned earlier, the phase noise at 20 MHz is 2-3 dB worse than the other two works. The performance is limited by the VCO itself instead of the proposed quantization noise cancellation technique. The VCO is therefore redesigned to improve the phase noise at high frequency offsets. Figure 7-22 shows the measured phase noise with the new chip. Although the new chip indeed achieves -152 dBc/Hz at 20 MHz offset, the flicker noise of the new VCO is worse, resulting in higher in-band noise (-106 dBc/Hz at 400 kHz offset) and integrated jitter (250 fs).



```
Carrier 3.665713982 GHz    -3.1027 dBm
 1:  20 kHz    -105.0234 dBc/Hz
>2:  100 kHz   -105.5720 dBc/Hz
 3:  400 kHz   -106.8826 dBc/Hz
 4:  600 kHz   -109.7199 dBc/Hz
 5:  3 MHz     -132.8874 dBc/Hz
 6:  20 MHz    -152.3485 dBc/Hz
══ Noise ══
Analysis Range X: Full Range
Analysis Range Y: Full Range
Intg Noise: -47.8450 dBc / 40 MHz
 RMS Noise: 5.73146 mrad
            328.389 mdeg
RMS Jitter: 248.844 fsec
Residual FM: 5.70521 kHz
```

Figure 7-22: A modified chip improves phase noise at 20 MHz offset.

7.7 Summary

By combining the proposed techniques, we demonstrate a 3.6-GHz, 500-kHz bandwidth digital $\Delta\Sigma$ frequency synthesizer architecture to achieve excellent in-band and out-of-band phase noise. The prototype is implemented in a 0.13-μm CMOS process and its active area occupies 0.95 mm^2. Operating under 1.5-V, the core parts excluding the VCO output buffer dissipate 26 mA. Measured phase noise at 3.67 GHz achieves -108 and -150 dBc/Hz at 400 kHz and 20 MHz, respectively. Integrated phase noise at this carrier frequency yields 204 fs of jitter (measured from 1 kHz to 40 MHz).

Chapter 8

Proposed Techniques for Digital Phase Control

Although the fractional-N technique is usually only used in frequency synthesis, in this chapter we demonstrate that there is another interesting application of this technique. It is shown that with slight modification, the fractional-N synthesizer can provide a digitally controlled delay and thus can serve as a phase shifter in a delay-locked loop (DLL) for high-speed chip-to-chip communications.

8.1 Background

The application of the DLL in the high-speed data link interface has become popular recently. In the case where the reference clock is transmitted together with the data, as illustrated in Figure 8-1, the clock frequency is usually perfectly matched to the data rate. However, there is usually a phase mismatch between the received clock and data due to the different propagation delays on the printed circuit board. Therefore, a variable delay controlled by a feedback loop is necessary to realign the clock edge to the center of the data sequence automatically in order to minimize the bit-error rate.

People often use an analog phase interpolator as the variable delay because it provides an infinite delay range [16][17][18][19], which is not available in a delay-line-based DLL [60]. Figure 8-1 shows an example of the implementation of an analog phase interpolator. A quadrature generator first produces the I and Q reference clocks that are 90 degrees apart from each other. Two differential-pair-based variable gain amplifiers modulate the amplitudes of the I and Q clocks by adjusting the bias current of each differential amplifier. By summing the weighted I and Q clocks, any phase between them can be interpolated with a constant output amplitude, as shown in the vector diagram in Figure 8-1. A phase outside the first quadrature can also be interpolated by changing the polarity of the I and/or Q clocks in order to cover the whole 2π range. Furthermore, since the phases of $2\pi + \theta$ and θ can be regarded as the same, this technique can provide an infinite phase/delay range for a DLL.

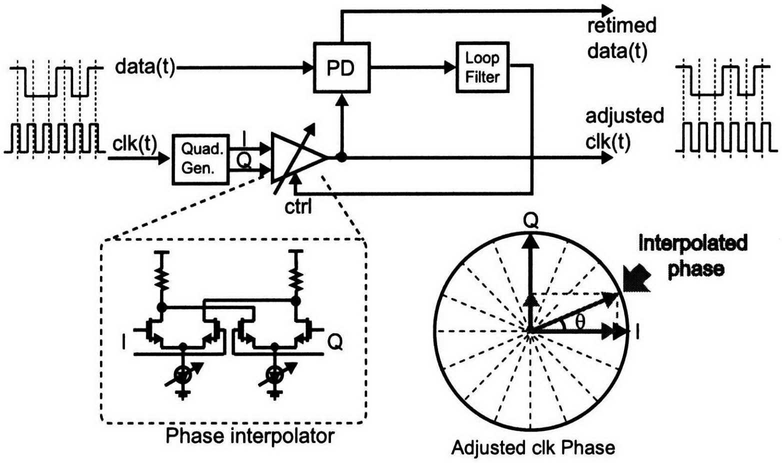


Figure 8-1: DLL with an analog phase interpolator.

Nevertheless, the magnitudes of the I and Q clocks must be accurately controlled for a desired phase. Given the interpolated phase θ, the required amplitude of the I and Q clocks are $|cos(\theta)|$ and $|sin(\theta)|$, respectively, assuming the amplitude of the interpolated clock is one. When the phase interpolator is controlled in an analog way, it usually requires complicated analog circuits with good matching, which are not always available in modern digital processes. In addition, when a linear transfer function between the interpolated phase and input control signal is necessary, compensation in the analog domain is also required [17].

To leverage the digital calculation capability provided by sub-micron processes, several works have demonstrated the possibility of controlling the phase interpolator with a digital loop filter instead of an analog filter [61], as shown in Figure 8-2. In this case, current-steering digital-to-analog converters (DAC) [62] replace the variable current sources in Figure 8-1 but cause two problems. First, due to the finite resolution of the DACs, the phase interpolator can only generate a finite number of output phases. Second, uniform distributed output phases require non-uniform distributed DAC levels, which are difficult to implement. Therefore, uniform distributed DAC levels are used instead in practice, making unequal phase steps unavoidable [61]. In a DLL-based CDR circuit [17], a small frequency error may exist between the reference clock and data rate such that the phase interpolator has to rotate its output phase constantly to track that of the data. In this situation, the phase rotator needs to visit all of the unequal phase steps, and thus its jitter performance is degraded.

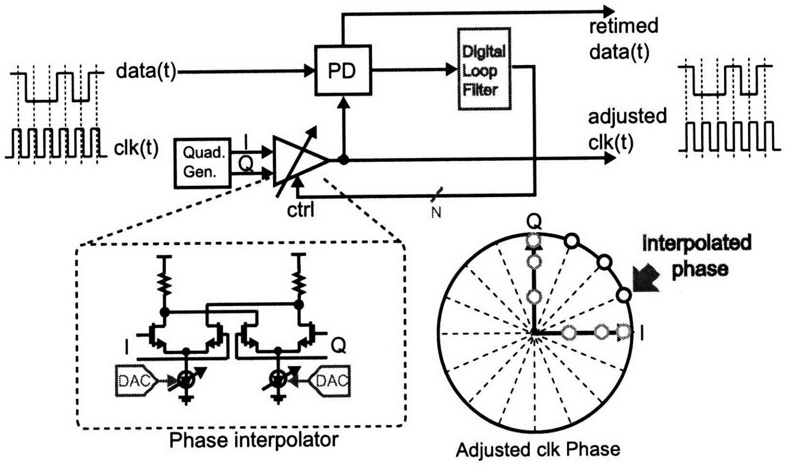


Figure 8-2: Phase interpolator controlled by current DACs.

We propose to use a simple voltage-controlled oscillator (VCO) instead of a phase interpolator to achieve the phase shifting functionality within a DLL [15]. By implementing the VCO as a standard ring oscillator, this approach offers a very simple, highly digital implementation that has the ability to achieve very fine phase shifts and infinite phase range. By applying feedback to the VCO in the form of a fractional-N synthesizer, the phase resolution can be digitally controlled and is less sensitive to the process, temperature, and voltage (PVT) variations than conventional structures based on phase interpolators.

In the following section, we first provide details of the proposed DLL architecture that includes a second-order digital $\Delta\Sigma$ modulator structure. This modulator structure allows a high clock rate with a low-power and compact implementation. Section 8.3 provides the circuit implementation. Finally, we present the measurement results in Section 8.4.

8.2 Proposed DLL Architecture

The proposed DLL architecture consists of two parts: a synthesizer-based phase shifter and a bang-bang detector, as depicted in Figure 8-3. The details of the synthesizer-based phase shifter and the proposed $\Delta\Sigma$ modulator for our DLL architecture are first explained in Section 8.2.1 and 8.2.2, respectively. In Section 8.2.3, the proposed bang-bang architecture is described.

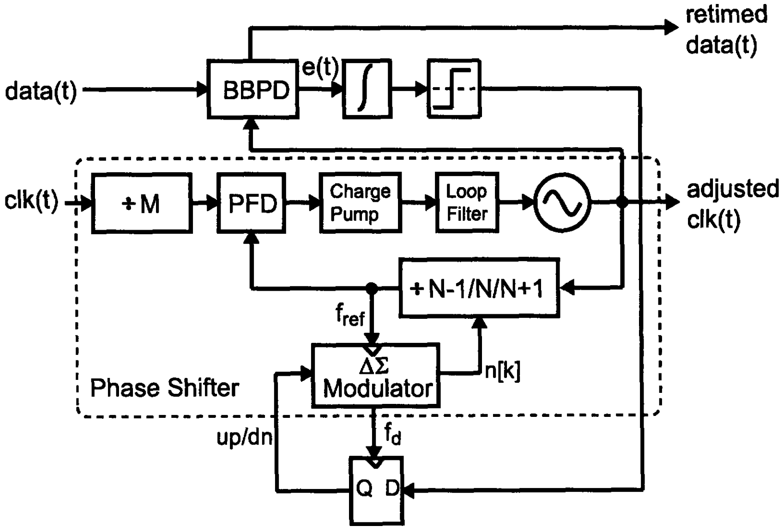


Figure 8-3: Proposed DLL with a synthesizer-based phase shifter.

8.2.1 Synthesizer-based Phase Shifter

We begin by discussing the application of a VCO as a phase shifter. As shown in Figure 8-4, a VCO can be modeled as an integrator with the VCO phase being regarded as the output. The input voltage $V_{ctrl}(t)$ is multiplied by the VCO gain K_v and integrated to become the phase $\Phi_{out}(t)$. Because of the integration function, if a positive or negative rectangular pulse with a height of ΔV and a width of T_p is fed to the VCO, the VCO phase increases or decreases by $\Delta V \cdot k_v \cdot T_p \cdot 2\pi$ at each time increment. Through proper adjustment of these parameters, very fine phase resolution can be achieved with an infinite phase range due to the fact that the VCO phase range is unlimited. Within a DLL application, the phase would be appropriately shifted to a desired value according to the control signal of the DLL. Being an ideal phase integrator without an output limit, a VCO is a potential candidate to be a phase shifter within a DLL, especially when it is implemented in the form of a ring oscillator, which usually occupies smaller chip area than an LC oscillator at the expense of its higher phase noise. However, the results of this work show that a ring oscillator can still provide a reasonable jitter performance for high-speed data-link applications. Besides, its wide tuning range also makes it attractive for multi-rate applications.

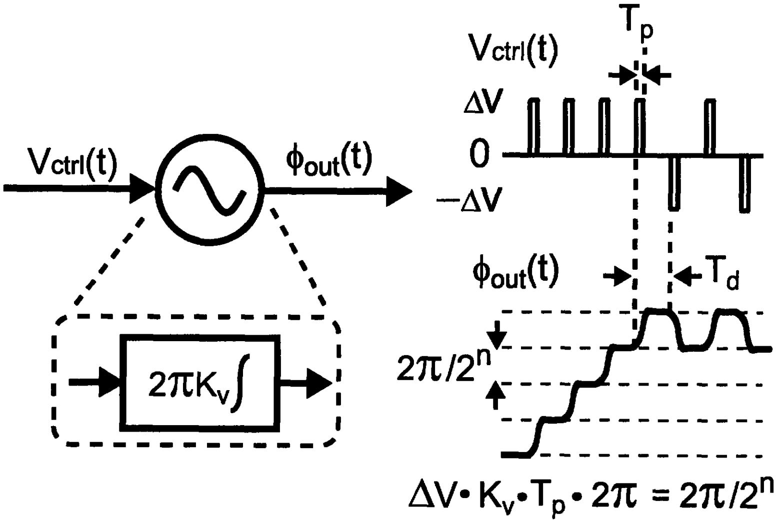


Figure 8-4: VCO-based phase shifter.

When considering the VCO as a standalone element, it is quite difficult to accurately control ΔV, K_v, and T_p and to set the nominal oscillation frequency of the VCO such that it is locked to the received clock of the DLL. However, by placing the

VCO within a $\Delta\Sigma$ fractional-N frequency synthesizer, we can accurately control the VCO with digital precision. Figure 8-5 illustrates this concept, with discrete-time impulses of value of Δf or $-\Delta f$ being fed into the $\Delta\Sigma$ modulator input of a fractional-N synthesizer. The effect of the digital pulse is the same as that of the analog pulses directly fed to the standalone VCO in Figure 8-4. However, with the help of the feedback, the variation of ΔV, K_v, and T_p are automatically calibrated by the PLL. Thus, a precise phase step $2\pi \cdot \Delta f \cdot T$ can be provided by the synthesizer-based phase shifter, where T is the reference period of the synthesizer, and Δf is equal to $\Delta V \cdot K_v$. Notice that if T_p in Figure 8-4 is equal to T in Figure 8-5, the phase step provided by the standalone VCO and the synthesizer-based phase shifter are equal.

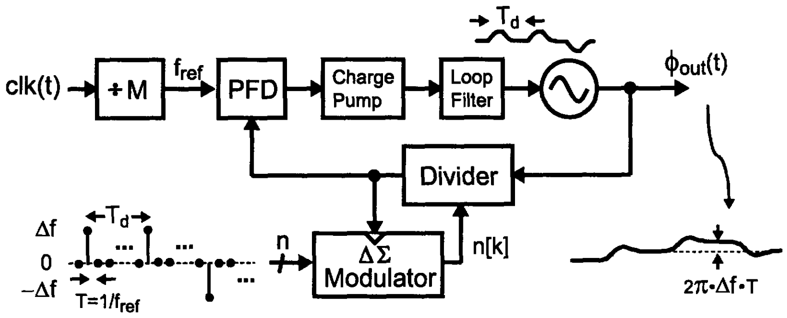


Figure 8-5: Proposed synthesizer-based phase shifter.

It is worthwhile to highlight the difference between our phase shifter and the conventional application of the $\Delta\Sigma$ synthesizers with the help of Figure 8-6. In a $\Delta\Sigma$ frequency synthesizer, a n-bit resolution digital input can generate a n-bit resolution output frequency. In our application, the $\Delta\Sigma$ technique is used to generate a fractional output *phase* instead of frequency. If a pulse with a magnitude of one is fed into the divider, the output phase increases by 2π because one more VCO cycle must be swallowed by the divider. By decreasing the pulse height, a finer phase step can be obtained from the VCO. A phase resolution of $2\pi/2^n$ can be achieved by simply setting the number of fractional bits in the $\Delta\Sigma$ modulator to n. Thus, the resolution can be accurately and finely controlled and is independent of the PVT variations. For example, when an eight-bit $\Delta\Sigma$ modulator is used, the phase resolution is 1.4 degrees, which is equivalent to 1.2 ps for a 3.2-GHz clock. Compared to a phase interpolator, both the linearity and resolution of the proposed phase shifter are improved. Notice that the control voltage of the VCO looks like a filtered rectangular pulse. Since it returns to a constant value in the end of each cycle, the output phase increases while the frequency does not in the end of each cycle.

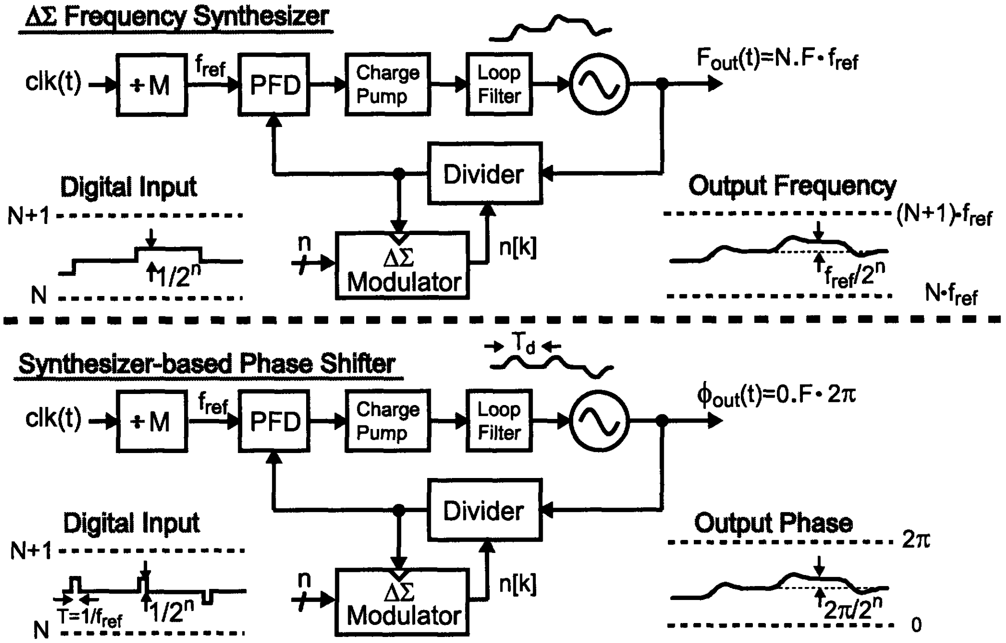


Figure 8-6: Comparison of the $\Delta\Sigma$ synthesizer and proposed phase shifter.

As with the $\Delta\Sigma$ synthesizer, the PLL response should be designed properly to jointly minimize the VCO phase noise and quantization noise by adjusting the loop bandwidth. Besides, after one pulse is applied, there should be enough time T_d before the second pulse can be applied such that the VCO phase can settle properly. To meet the requirement, T_d should be larger than reciprocal of the PLL bandwidth.

To generate the control pulses, an up/down counter and a differentiator are added in front of the $\Delta\Sigma$ modulator, as illustrated in Figure 8-7. The input to the phase shifter is a binary *up/dn* signal ($+1/2^n$, $-1/2^n$) updated at the rate of $1/T_d$. The up/dn counter, functioning as an accumulator here, integrates the *up/dn* signal, and the counter output is regarded as a n-bit fractional *Phase Control Word* (PCW) that determines the VCO phase. The full scale and LSB of the PCW represent one

VCO period (2π) and the unit phase step ($2\pi/2^n$), respectively. Therefore, the step size ($2\pi/2^n$) is determined only by the number of bits of the hardware. A divide-by-R divider provides a clock rate of $1/T_d$ to the up/dn counter, where $1/T_d = f_{ref}/R$. Notice that operating the up/dn counter at a speed slower than the reference clock by choosing a proper value of R allows the VCO phase to settle properly before the PCW is updated again (In our final implementation: $1/T_d \sim 1$ MHz, $f_{ref} = 533$ MHz, R=512). By differentiating the PCW at the rate of f_{ref}, a pulse signal with a magnitude of $1/2^n$ is obtained every T_d and used to shift the VCO phase.

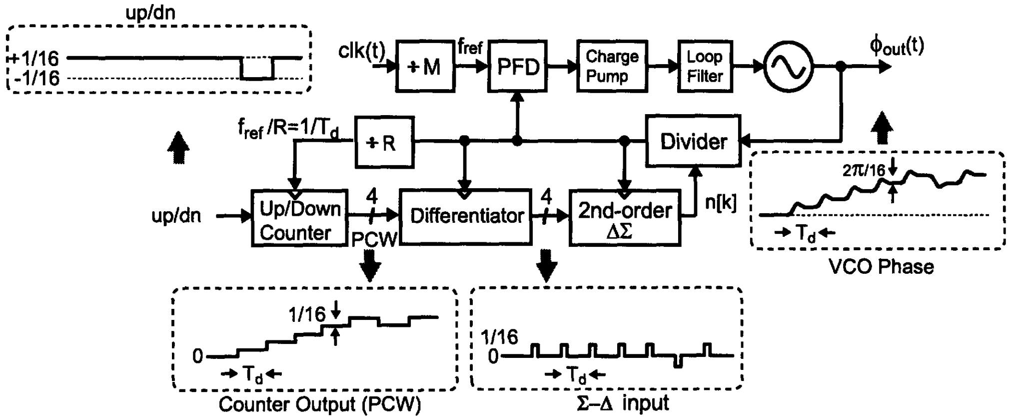


Figure 8-7: Synthesizer-based phase shifter including circuits to generate control pulses.

The proposed phase shifter can also be understood graphically with the help of Figure 8-8. The left circle represents the PCW, the output of a four-bit up/down counter in this example, and the right circle represents the VCO phase. The VCO output phase is zero in the beginning in this example, and we want to shift the VCO phase close to for instance $2\pi \cdot 19/64$. Since the initial VCO phase is far away from the targeted value, the phase shifter rotates the VCO phase counterclockwise step by step each cycle by setting the up/dn signal to $+1/2^n$. Once the VCO phase is close to the targeted phase, the phase shifter rotates back and forth around the targeted phase, according to the sign of the phase error between the current VCO phase and the targeted phase. By increasing the number of bits of the circuits, the phase step and thus the phase error can be decreased in order to obtain less DLL jitter, as shown in

Figure 8-9. We can minimize the phase step until the final jitter is dominated by the VCO intrinsic jitter and $\Delta\Sigma$ quantization noise. The compromise of using a smaller phase step is the longer locking time because more cycles are needed to rotate the VCO phase to the targeted value. However, it becomes more clear in Section 8.2.2 that we can control the phase directly by setting the PCW. Therefore, a binary search algorithm can be applied in initial locking in order to reduce the overall DLL settling time [60]. Furthermore, a better phase detector is necessary to detect the smaller phase error when n increases.

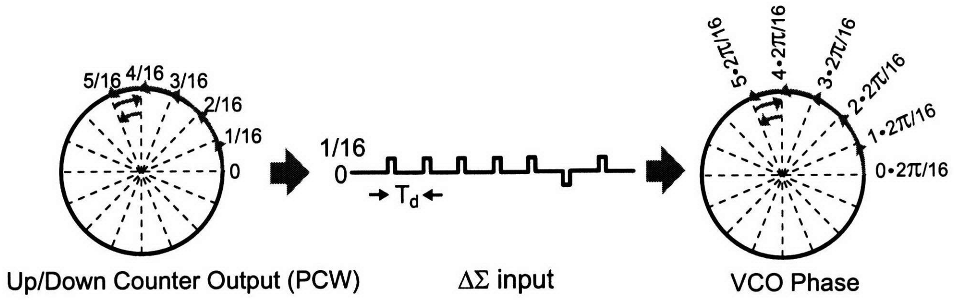


Figure 8-8: Phase-shifting operation without up/down counter overflow.

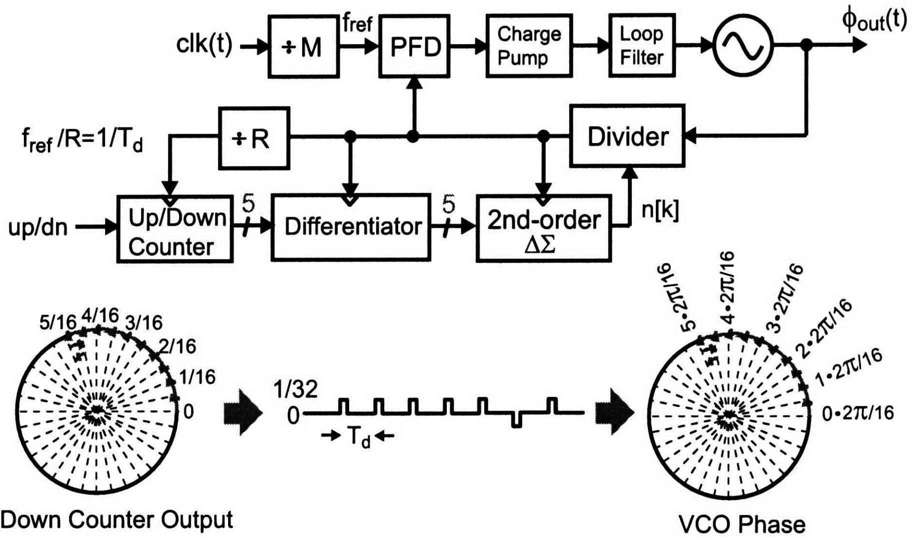


Figure 8-9: Improving resolution by increasing the number of bits of the hardware.

The problem encountered by the architecture in Figure 8-7 is that the up/down counter output may overflow. The differentiator can sense this change and generate a large negative pulse, as shown in Figure 8-10. Due to this negative pulse, the VCO phase rotates clockwise by 15 steps instead of keeping rotating counterclockwise by one step. As a result, the phase shifter can provide only a phase range of 2π, instead of an infinite phase range. Although the phases of $2\pi+\theta$ and θ can be regarded as the same, the transition in the wrong direction degrades the jitter performance whenever overflow occurs. In the case when an offset frequency exists between the reference clock and data rate, the overflow issue is unacceptable because it occurs frequently due to the constantly rotating VCO phase.

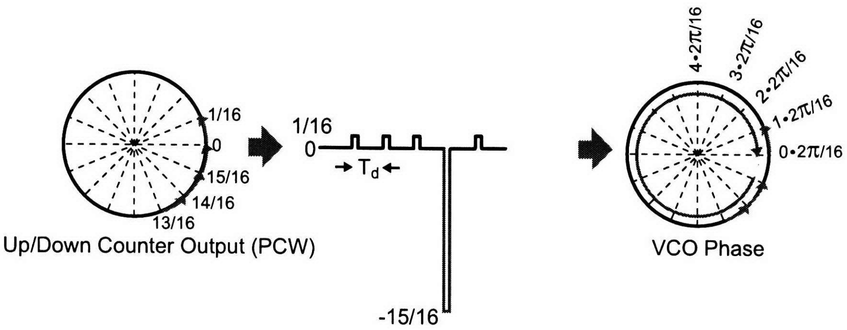


Figure 8-10: Phase-shifting operation with up/down counter overflow.

We can solve the problem easily by using an overflow detector to generate a pulse of +1. By adding this pulse to the undesired negative pulse, as shown in Figure 8-11, we can get a net pulse of -1/16. Thus, the VCO phase can keep rotating counterclockwise by one step even when overflow occurs. Since the output phase can now keep increasing or decreasing, the phase shifter can provide an infinite phase range as needed.

8.2.2 $\Delta\Sigma$ Modulator

The order of the $\Delta\Sigma$ modulator in Figure 8-7 must be at least two, because the quantization noise of a first-order modulator is too large for this application. However, we can simplify the modulator to a first-order one by exchanging the positions of the modulator and differentiator in Figure 8-7, obtaining a new modulator architecture as shown in Figure 8-12. The original second-order modulator can be replaced by a

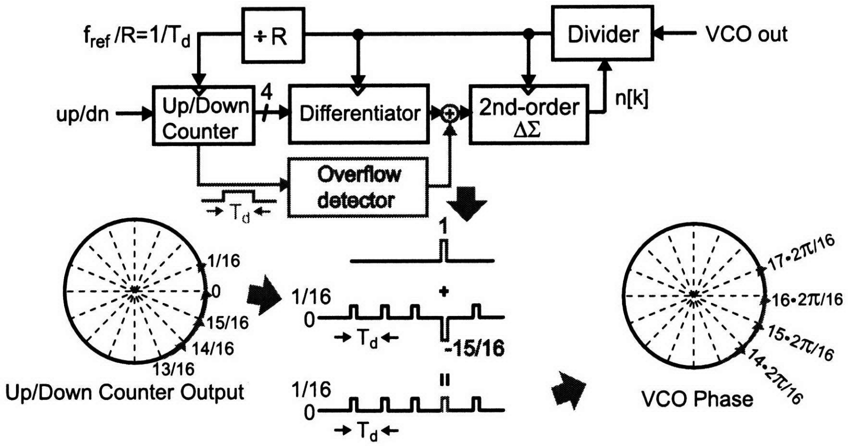


Figure 8-11: An overflow detector can remove the undesired negative pulse.

first-order one because now the first-order-shaped quantization noise is differentiated, still resulting in a second-order-shaped noise at the output (i.e., $n[k]$). A first-order modulator can be simply implemented in the form of an accumulator. Therefore, by modifying the structure slightly, we reduce the circuit complexity without increasing noise.

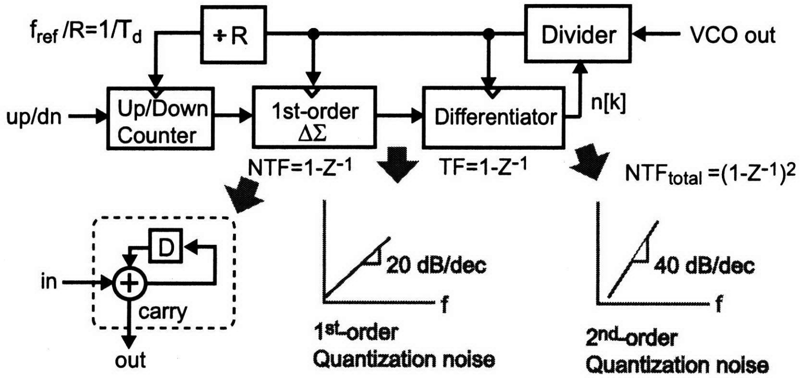


Figure 8-12: Modified ΔΣ architecture with less circuit complexity.

The resulting architecture in the last paragraph is actually a special case of a more general second-order modulator. Setting R equal to 1 and replacing the up/dn counter with a real accumulator give us the general second-order modulator, which accepts multi-level inputs. The central part of this modulator is a first-order ΔΣ modulator, whose signal transfer function (STF) and noise transfer function (NTF) are Z^{-1} and $1 - Z^{-1}$, respectively. A digital differentiator, whose transfer function is $1 - Z^{-1}$, is then added after the modulator to get a cascaded NTF of $(1 - Z^{-1})^2$, which is equivalent to that of a second-order $\Delta\Sigma$ modulator. However, this results in a cascaded STF of $Z^{-1}(1 - Z^{-1})$. This STF is undesirable but can be easily fixed by adding a digital accumulator, whose transfer function is $1/(1 - Z^{-1})$, before the first-order $\Delta\Sigma$ modulator, so that the overall cascaded STF becomes Z^{-1}. Thus, both the STF and NTF of the proposed second-order $\Delta\Sigma$ modulator are the same as those of a standard topology.

Up to now the architecture in Figure 8-12 still needs an accumulator (i.e., the first-order modulator) running at the highest speed (i.e., f_{ref}). The advantage of the proposed $\Delta\Sigma$ modulator is not clear until we demonstrate how this architecture can be simplified for our DLL application by applying a multi-rate clock, as illustrated in Figure 8-13. Notice that *up/dn* is being updated at a rate of approximately $f_d = 1$ MHz while the output is being updated at $f_{ref} = 533$ MHz. To connect these different sample rates, the $\Delta\Sigma$ modulator must be progressively clocked from low to high frequencies. We achieve this goal by cascading three first-order $\Delta\Sigma$ modulators with different resolutions and clock rates. By using this approach, the bit-number decreases while the clock rate increases, and thus only a small portion of the overall modulator circuit operates at the highest frequency (i.e., 533 MHz). As a result, the power consumption and design complexity is reduced at the expense of slightly larger area. By gradually changing the clock rate through the structure, the metastability and synchronization problems are also avoided.

The output of the differentiator is a three-value signal (1, 0, -1), since the two-value (1, 0) first-order modulator output is differentiated. As mentioned before, we can solve the overflow problem by adding the overflow signals to the output. While the overflow signals are propagated to the output, they are automatically realigned to the main signals by D-flip-flops (DFF) in each clock domain. Even with the extra circuits, it can be shown that the output $n[k]$ is still a three-value signal, and thus the divider needs only three division ratios. The differentiator in the modulator is actually implemented as a DFF plus an encoder, as illustrated in Figure 8-14.

Although conceptually we need an adder and a subtractor to fix the overflow issue, as shown in Figure 8-13, they can be simplified to only two XOR gates, as illustrated in Figure 8-14.

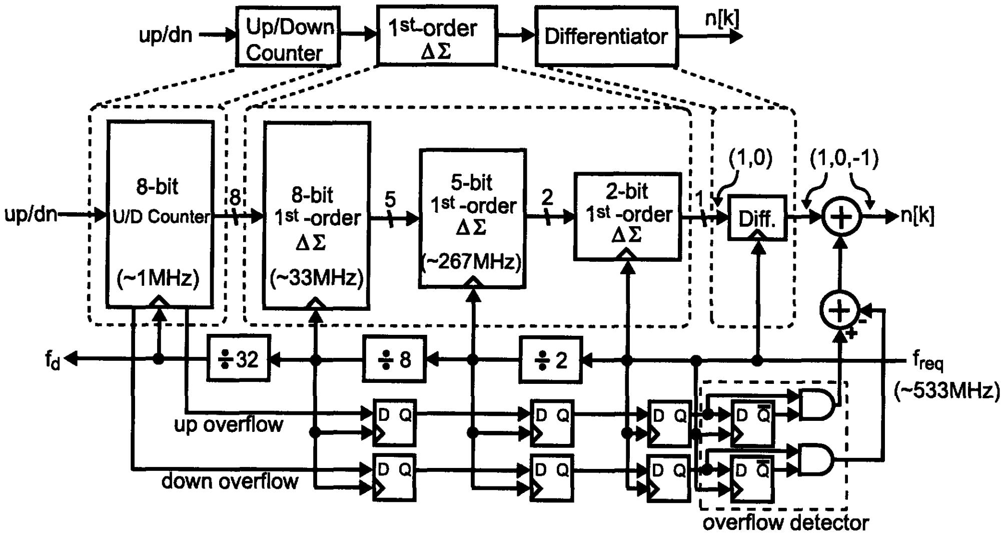
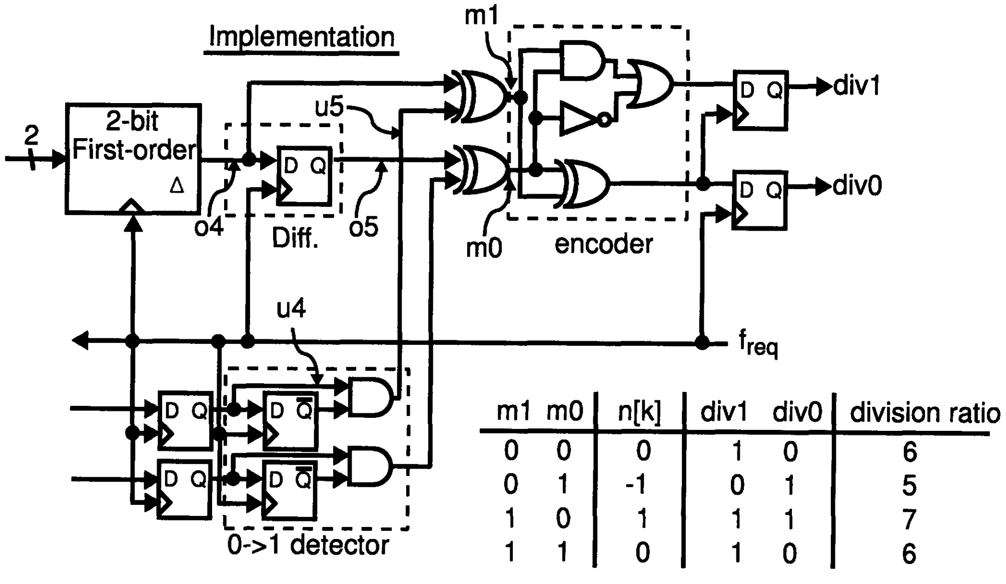


Figure 8-13: Multi-rate implementation of the proposed $\Delta\Sigma$ architecture.



m1	m0	n[k]	div1	div0	division ratio
0	0	0	1	0	6
0	1	-1	0	1	5
1	0	1	1	1	7
1	1	0	1	0	6

Figure 8-14: Simple implementation of the differentiator and adders.

In designing the multi-rate, first-order $\Delta\Sigma$ modulator, the bit-lengths of the lower-frequency stages are chosen to be higher than those of the higher-frequency stages in order to ensure that the total quantization noise is dominated by the last (highest frequency) stage. Again, we rely on the behavior simulation with *CppSim* to find the minimum bit-number as well as clock rate at each stage of the modulator without sacrificing the worst-case jitter performance [56].

It is worthwhile to provide another point of view from which to study the proposed phase shifter and modulator with the help of the synthesizer model proposed in [34]. The proposed modulator is plugged into the synthesizer model, as shown in Figure 8-15. To simplify the analysis, we assume here that the first-order modulator consists of only one stage instead of a multi-rate architecture. Thus, the PCW is first-order modulated, differentiated, and accumulated as well as filtered by the PLL loop filter. The result is then added together with the filtered detector and VCO noises to generate the final phase $\Phi_{out}(t)$.

One should notice that $\Phi_n[k]$ can be expressed as a delayed version of the PCW, with a gain factor of 2π, plus a first-order shaped quantization noise because the effect of the differentiator is cancelled by the PLL. Since the out-of-band quantization noise is filtered by the PLL, the VCO phase $\Phi_{out}(t)$ is therefore determined by the PCW plus the residual quantization noise as well as the VCO noise. Its DC value of $2\pi \cdot PCW/2^n$ explains why the up/dn counter output can be regarded as the *Phase Control Word.*

Figure 8-15 also gives examples of the waveforms within the $\Delta\Sigma$ modulator with two different PCW values. According to the waveform of $\Phi_n[k]$, we can also see that the phase shifter provides two phase values, zero and 2π, and a phase between them is interpolated with a first-order $\Delta\Sigma$ modulation. For example, when the PCW is 64, which is one-fourth of 256, the phase shifter outputs a phase of 2π during one-fourth of the period and a phase of zero in the rest of the period. Through averaging, a phase of $\pi/2$ is obtained. To conclude, we can also regard the proposed synthesizer-based phase shifter as a digital phase interpolator that can interpolate a phase between zero and 2π with a phase resolution of $2\pi/2^n$.

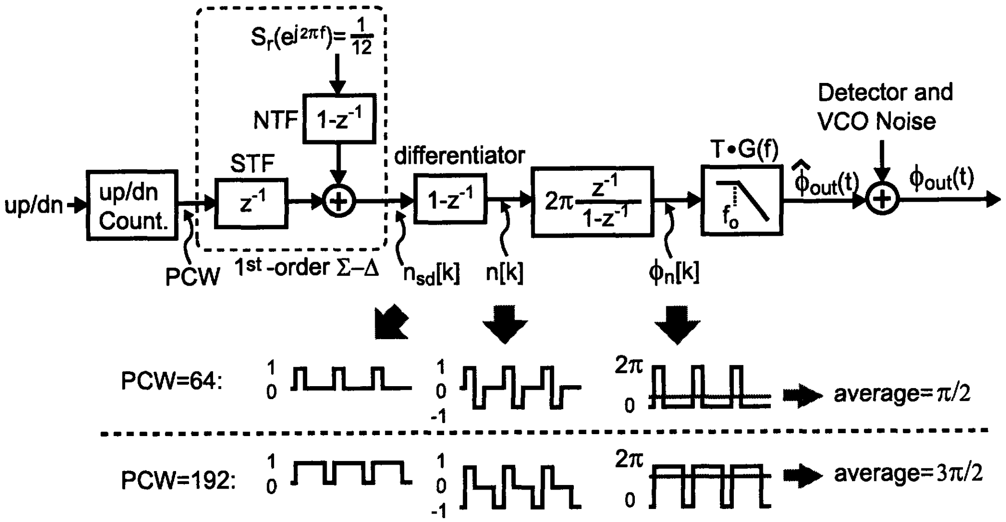


Figure 8-15: The synthesizer noise model and phase interpolation operation.

8.2.3 Bang-bang Detector

We leverage the bang-bang detector to compare the phases of the VCO and the received data. Figure 8-16 shows a typical bang-bang phase detector (BBPD) and a timing diagram to explain its operation [53]. The output of a bang-bang detector is a high-speed three-value signal. Whenever there is a data transition, the bang-bang detector outputs either a positive or negative pulse according to the phase difference between the clock and data. When there is no data transition, the output remains zero. Thus, the bang-bang detector outputs a series of positive (negative) pulses when the data edges lead (lag) the clock edges.

Since the bang-bang detector updates its output at a rate equal to the data rate, which is 3.2 Gb/s in our case, while we need a low-rate control signal, which is about 1 MHz in our case, to shift the VCO phase, an efficient interface is necessary between the bang-bang detector and phase shifter. As illustrated in Figure 8-17, we feed the bang-bang output $e(t)$ into a saturating integrator, which allows the detector output to be averaged and converted from a three-value signal (1,0,-1) to a two-value signal (1,-1). For example, when the VCO edge begins to lead the data edge, $i(t)$ keeps on increasing and saturates to 1 after several periods. The transition region of $i(t)$ is reduced by the limiter whose output is then sampled by a DFF with a period of T_d.

The sampled signal *up/dn* that updates every T_d controls the phase shifter. Notice that the gain of the integrator should be chosen properly, since an extremely large gain degrades the filtering function, while an extremely small gain contributes a long delay, which may increase the limit cycle of the output phase.

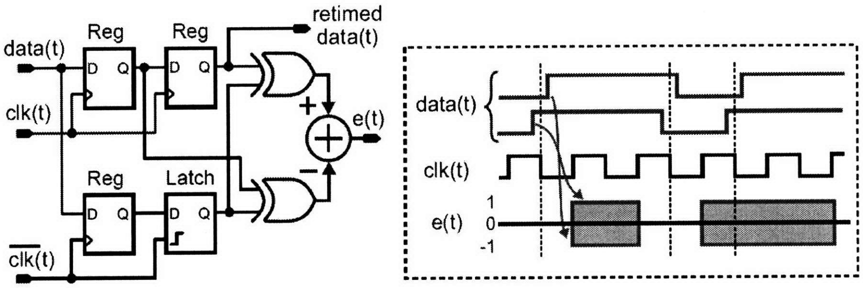


Figure 8-16: Conventional bang-bang detector architecture.

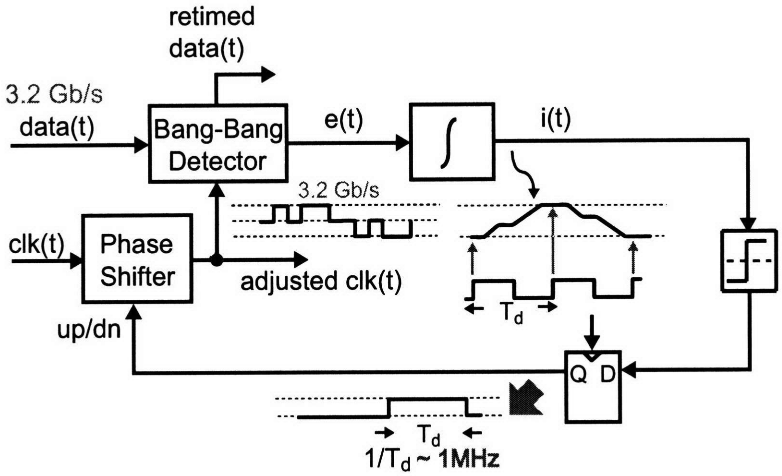


Figure 8-17: Proposed bang-bang detector architecture.

8.3 Circuit Implementation

The DLL prototype is designed for a 3.2-Gb/s application with an input clock frequency of 1.6 GHz. Aside from achieving fine resolution and infinite range in the phase adjustment, the proposed DLL structure also allows us to easily multiply the incoming clock. As illustrated in Figure 8-3, the output clock of the DLL structure is multiplied by the ratio N/M - we have chosen N=6 and M=3 in the prototype, so that the input clock frequency is multiplied by two. An eight-bit $\Delta\Sigma$ modulator is chosen to provide a phase resolution of 1.4 degrees, which is equivalent to 1.2 ps for a 3.2-GHz clock. The nominal $\Delta\Sigma$ clock f_{ref} is 533 MHz, generated from the VCO clock after being divided by N=6. The reason to choose N=6 is to allow f_{ref} to be slow enough that we can implement the $\Delta\Sigma$ modulator with the full-swing logic in the 0.18-μm process we use. The VCO phase is updated at a rate of $f_d \sim 1$ MHz generated by dividing f_{ref} by 512. The bandwidth of the PLL is chosen to be 4 MHz to jointly minimize the impact of the VCO phase noise and $\Delta\Sigma$ quantization noise. The behavior of this system is verified with the *CppSim* behavior-level simulator [56].

Figure 8-18 shows a simplified schematic of the circuits. In order to achieve a compact design, we use a ring oscillator similar to that proposed in [63]. A divider based on [5] is designed to provide a divide ratio from five to seven. Besides, the XOR phase-frequency detector (PFD) in [64] is used. Due to the limit on the speed of this 0.18-μm process, the current-mode logic is used in the divider, XOR PFD, and the BBPD. A differential-to-single-ended charge pump and an on-chip loop filter are used as shown in the figure. A source follower is inserted in the loop filter to shift the nominal voltage to the center of the VCO control range. MOS capacitors are used in the loop filter to reduce the area. In addition, a standard bang-bang detector [53] is used and followed by two differential-to-single-ended converters. The integrator is composed of a current pump and a capacitor; an inverter following the integrator is used as a limiter.

As shown in Figure 8-18, only simple analog circuits are required in the proposed DLL architecture, without the need for good matching between any of their elements. The overall architecture is primarily digital and well suited for more advanced CMOS processes. All of the digital blocks are primary DFFs. The area and power could be dramatically reduced with a more advanced process since the full-swing logic rather than the current-mode logic could be used in the divider, PFD, and bang-bang detector.

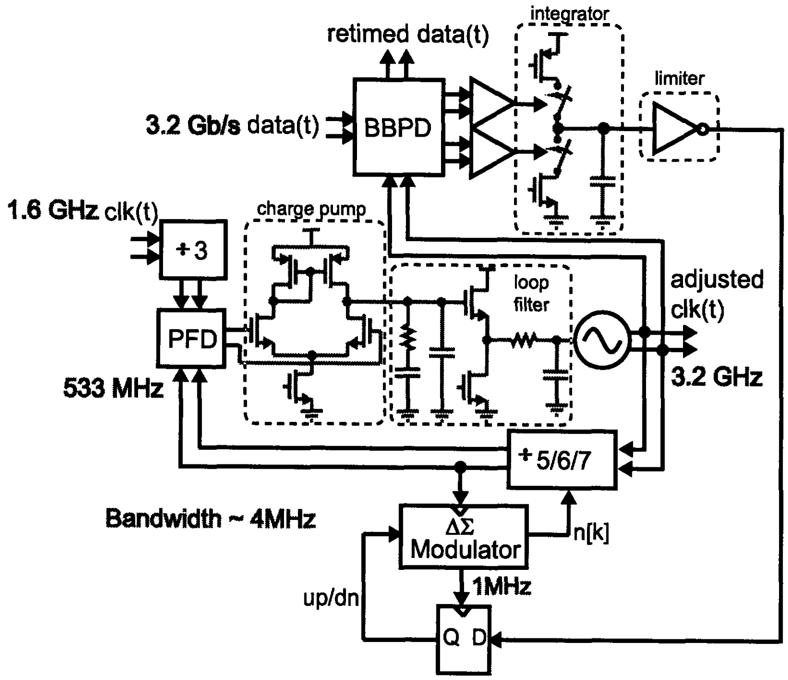


Figure 8-18: Schematic of the DLL.

8.4 Results

The prototype chip is fabricated in a 0.18-μm CMOS process. The die photograph is shown in Figure 8-19, and its active area is $600\mu m \times 700\mu m$. It is packaged and mounted on a printed circuit board for measurement. The chip operates at 1.8 V, and the DLL, excluding the input and output buffers, dissipates 55 mA.

The synthesizer is first set to an integer-N mode, and measured phase noise and K_v of the VCO are -118 dBc/Hz at 20 MHz offset and 140 MHz/V, respectively. The measured single-ended recovered clock and data jitter under different conditions are summarized in Table 8.1. Setting the synthesizer in the integer-N mode indicates the intrinsic jitter performance, whereas the synthesizer is set to the fractional-N mode to test the actual DLL performance. Note that the clock jitter increases when the data output driver is turned on due to the coupling between the clock and data output drivers through the shared bias circuits.

Figure 8-20 illustrates the eye-diagram of the recovered data and clock when the input data is a 3.2-Gb/s PRBS $2^{31}-1$ sequence, and reveals 4.8-ps singled-

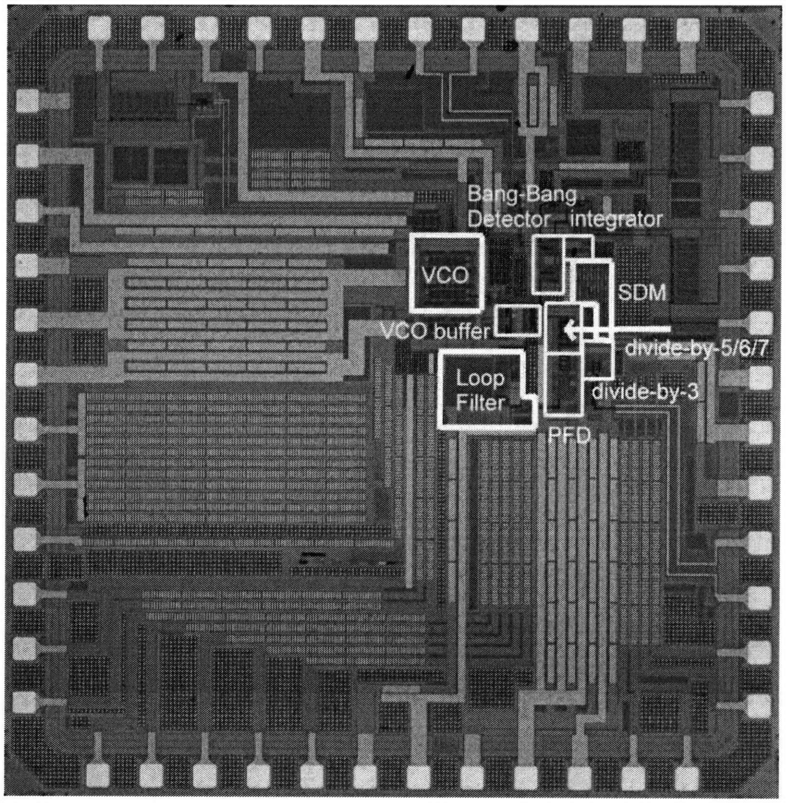


Figure 8-19: Die photo of the DLL chip.

Table 8.1: Measured Single-ended RMS Clock/Data Jitter

Testing Condition	3.2 Gb/s $2^{31}-1$ PRBS	3.2 Gb/s $2^{7}-1$ PRBS	1.6 Gb/s $2^{7}-1$ PRBS
Integer-N PLL with data output driver off	3.4/-	3.4/-	3.1/-
Integer-N PLL with data output driver on	4.3/30.1	4.1/30.2	4.3/4.7
DLL in synchronous mode	4.8/30.5	4.7/29.8	4.7/5.2
DLL in asynchronous mode	4.8/30.0	4.6/30.7	4.6/5.0

ended clock jitter and 30-ps singled-ended data jitter. A separate differential clock measurement reveals jitter less than 3.6 ps, which means part of the 4.8-ps single-ended clock jitter is due to the common-mode noise. The high data jitter is due to the intersymbol interference that is likely introduced by the BBPD and output buffer having inadequately high bandwidth. To verify this fact, Figure 8-21 shows that the output data jitter is reduced to 5.2 ps with a 1.6 Gb/s PRBS 2^7-1 input sequence.

Table 8.2: Measured Differential Clock Jitter

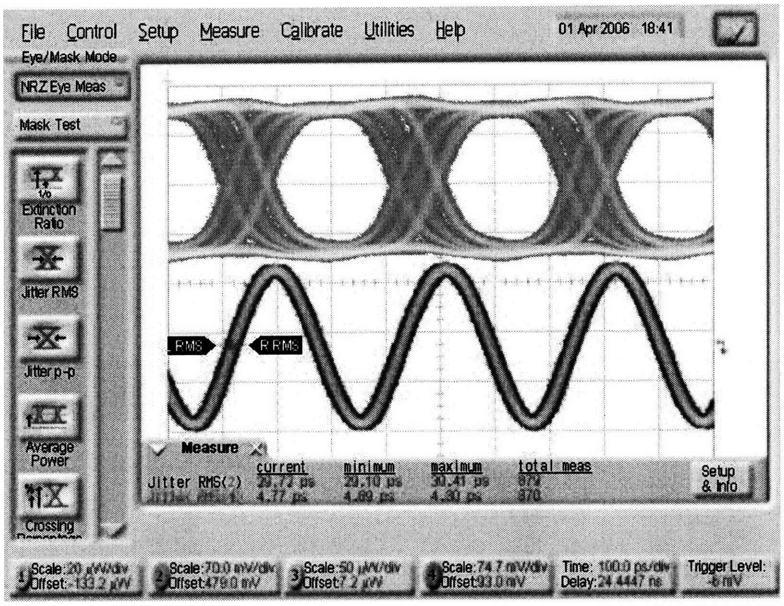
Testing Condition	3.2 Gb/s $2^{31}-1$ PRBS
Integer-N PLL with data output driver off	2.4
Integer-N PLL with data output driver on	3.5
DLL in synchronous mode	3.7

Note that the bit-error rate of the DLL is less than 10^{-12} in all of the measurements.

The DLL is also tested in the asynchronous mode by introducing a frequency offset between the input data and clock. In this condition, the phase difference between the data and clock increases linearly so that the DLL must constantly rotate its output phase. As Table 8.1 reveals, the resulting jitter with a frequency offset of 3 kHz is very close to that obtained in the synchronous mode. It implies that the successive phase steps within 2π are very close to each other, and hence the very good linearity of the synthesizer-based phase shifter. The bit-error rate also remains less than 10^{-12} in this measurement. Notice that the maximum frequency offset Δf that the prototype DLL can tolerate can be derived to be 3.9 kHz, which is 1.2 ppm of the 3.2-GHz/s data rate.

8.5 Summary

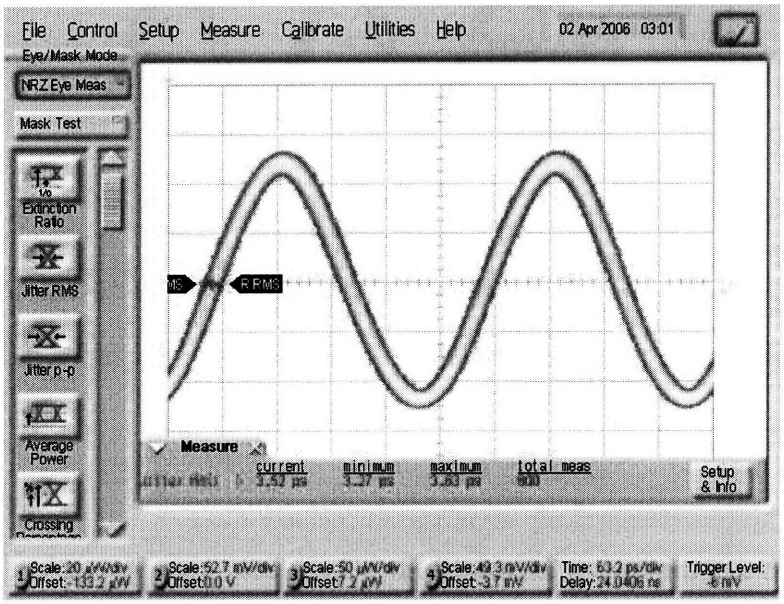
A 3.2-Gb/s DLL in a 0.18-μm CMOS for chip-to-chip communications is presented. By leveraging the fractional-N synthesizer technique, this architecture provides a digitally-controlled phase adjustment with fine resolution and an infinite range that is less sensitive to the PVT variations than conventional techniques. A new $\Delta\Sigma$ modulator enables a compact and low-power implementation of this architecture. A simple bang-bang detector is used for phase detection. The prototype operates at a 1.8-V supply voltage with a current consumption of 55 mA. The phase resolution and differential rms clock jitter are 1.4 degrees and 3.6 ps, respectively.



```
                 current    minimum    maximum    total meas
Jitter RMS(2)    29.72 ps   29.10 ps   30.41 ps   874
                 4.77 ps    4.69 ps    4.30 ps    370
```

Scale:20 µW/div Offset:-133.2 µW
Scale:70.0 mV/div Offset:479.0 mV
Scale:50 µW/div Offset:7.2 µW
Scale:74.7 mV/div Offset:93.0 mV
Time: 100.0 ps/div Delay:24.4447 ns
Trigger Level: -6 mV

(a)

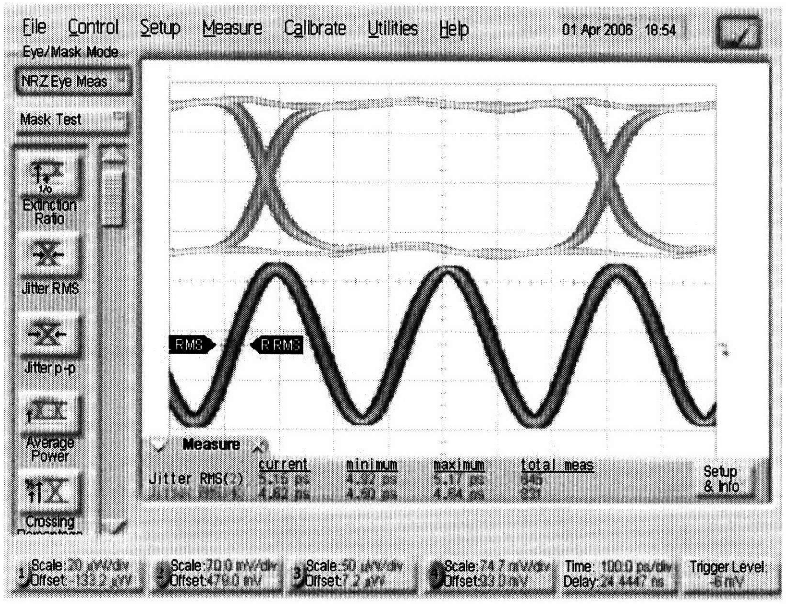


```
                 current    minimum    maximum    total meas
                 3.52 ps    3.27 ps    3.63 ps    800
```

Scale:20 µW/div Offset:-133.2 µW
Scale:52.7 mV/div Offset:0.0 V
Scale:50 µW/div Offset:7.2 µW
Scale:49.3 mV/div Offset:-3.7 mV
Time: 63.2 ps/div Delay:24.0406 ns
Trigger Level: -6 mV

(b)

Figure 8-20: Recovered eye-diagram with a 3.2-Gb/s input data (a) Single-ended data and clock (b) differential clock.



	current	minimum	maximum	total meas
Jitter RMS(2)	5.15 ps	4.92 ps	5.17 ps	845
Jitter RMS(4)	4.62 ps	4.60 ps	4.64 ps	831

1 Scale:20 µW/div Offset:-133.2 µW
2 Scale:70.0 mV/div Offset:479.0 mV
3 Scale:50 µW/div Offset:7.2 µW
4 Scale:74.7 mV/div Offset:93.0 mV
Time: 100.0 ps/div Delay:24.4447 ns
Trigger Level: -6 mV

Figure 8-21: Recovered eye-diagram with a 1.6-Gb/s input data.

Chapter 9

Conclusion

This chapter starts with a thesis summary, leading to the discussion of possible future research.

9.1 Thesis Summary

This thesis has proposed techniques aimed at extending the bandwidth of digital fractional-N synthesizers. In contrast to the more traditional analog filter implementation that is area-consuming, a digital frequency synthesizer achieves a low-cost solution by leveraging the high density of modern CMOS processes to implement the necessary filtering function in a digital way. Extending the PLL bandwidth enables a higher-data-rate phase/frequency modulation without the need for pre-emphasis or two-point modulation. Other advantages of having a wider-bandwidth include higher suppression of VCO thermal/flicker noise, which potentially leads to a much lower integrated jitter, and a shorter PLL settling time without the need for dynamically adjusting the PLL bandwidth.

We demonstrate that there are two key challenges to extend the bandwidth of a digital fractional-N PLL. The first challenge is the need for a high-resolution TDC since its quantization noise becomes the dominating noise source at low frequency offsets in a wide-bandwidth PLL after VCO noise is highly suppressed. The second challenge is the need for the divider quantization noise cancellation because a wide bandwidth also allows more quantization noise to go through.

To solve the first problem, we leverage a recently invented GRO TDC [3]. Implemented in a 0.13-μm process, this TDC achieves 6-ps *raw* resolution using a multi-path ring oscillator. Furthermore, GRO TDC improves its $effective$ resolution by first-order noise-shaping. Therefore, we can move the TDC quantization noise to higher frequencies and leverage PLL filtering to attenuate this undesired noise. By doing so, <-100 dBc/Hz phase noise within the bandwidth has been demonstrated in a 3.6-GHz digital PLL, where the low-frequency performance is limited by the flicker noise of the gated ring oscillator.

To solve the second problem, we have proposed an all-digital quantization noise cancellation scheme. Although the idea is borrowed from an analog PLL, implementing the noise cancellation in a digital PLL turns out to be much simpler. Unlike the analog PLL, in a digital PLL, scaling of the accumulated quantization noise can be performed purely with a digital multiplier since a DAC is no longer necessary in the digital implementation. The scale factor is simply set by a digital correlation circuit that consists of a multiplier, an accumulator, and an IIR filter in contrast to the need for operational amplifiers and analog integrators in previous works. With proper design, the correlation loop can settle in 10 μs without impacting the phase noise performance. This locking time is shorter than the 35 μs reported in the previous work [26]. The results show that the proposed noise cancellation technique achieves at least 15 dB of noise suppression. This amount is sufficient to reduce the quantization noise below the VCO intrinsic noise level, which is -150 dBc/Hz at 20 MHz offset in this implementation, when a third-order MASH modulator is used.

While the TDC and digital noise cancellation circuits play the key roles in achieving low noise with a high bandwidth, the DCO and frequency divider circuits present their own challenges in striving for an elegant implementation of the overall digital synthesizer. As mentioned earlier, we consider the case of using a combination of a DAC and hybrid VCO to implement the DCO. We have proposed an efficient passive DAC implementation that requires minimal analog content. As for the divider, current digital PLL structures commonly use a synchronous structure with the argument that it has excellent jitter characteristics. Unfortunately, such structures also have relatively high power consumption due to the fact that many elements must be clocked at the highest frequency in the system (i.e., the VCO frequency). Instead, we have proposed an asynchronous divider structure that has low power consumption while still maintaining excellent noise performance by avoiding the divide-value-dependent delay in an asynchronous divider. In addition, this divider also lowers the required TDC range by a factor of four. We also say a few words about the hybrid VCO structure that is used.

A Type-II second-order PLL is used with a third-order $\Delta\Sigma$ modulator. A coarse/fine tuning scheme is applied. The fine-tuning filter is equivalent to an analog lead-lag filter. The first pole is created by a digital IIR filter. Another pole provided by the fine-tuning DAC helps to suppress the $\Delta\Sigma$ noise at high frequency offsets. In addition, the feedforward signal of the coarse-tuning filter is fed to the $\Delta\Sigma$ modulator instead of the VCO so that the coarse-tuning DAC bandwidth can be narrowed dramatically to reduce its negative impact on the overall PLL noise. Furthermore, a systematic way to determine the loop filter parameters is derived.

We also present the noise modeling of this synthesizer as well as a behavior model based on *CppSim*. The development of these two models allows us to verify the performance of the proposed system before the chip is implemented.

By combining the proposed techniques, we demonstrate a 3.6-GHz, 500-kHz bandwidth digital $\Delta\Sigma$ frequency synthesizer architecture that achieves excellent in-band and out-of-band phase noise. The prototype is implemented in a 0.13-μm CMOS process and its active area occupies 0.95 mm^2. Operating under 1.5 V, the core parts excluding the VCO output buffer dissipate 26 mA. Measured phase noise at 3.67 GHz achieves -108 dBc/Hz and -150 dBc/Hz at 400 kHz and 20 MHz offsets, respectively. Integrated phase noise at this carrier frequency yields 204 fs of jitter (measured from 1 kHz to 40 MHz).

In addition, a 3.2-Gb/s DLL in 0.18-μm CMOS for chip-to-chip communications is also presented. By leveraging the fractional-N synthesizer technique, this architecture provides a digitally-controlled phase adjustment with fine resolution and an infinite range. The provided delay is less sensitive to the PVT variations than conventional techniques. A new $\Delta\Sigma$ modulator enables a compact and low-power implementation of this architecture. A simple bang-bang detector is used for phase detection. The prototype operates at a 1.8-V supply voltage with a current consumption of 55 mA. The phase resolution and differential rms clock jitter are 1.4 degrees and 3.6 ps, respectively.

To conclude, a summary of the contributions of this thesis is as follows.

1. A 3.6-GHz low-noise, 500-kHz bandwidth digital $\Delta\Sigma$ frequency synthesizer architecture that achieves <-100 dBc/Hz in-band phase noise is presented. Its integrated jitter from 1 kHz to 40 MHz achieves 204 fs at 3.67 GHz. The bandwidth and jitter are ten times and five times better than those in [9], respectively.

2. An all-digital quantization noise cancellation technique achieving phase noise of -150 dBc/Hz at 20 MHz offset is presented. The proposed technique does not need any analog components and can be implemented with digital standard cells. Settling time of the calibration loop is three times better than that in [26].

3. A passive 10-bit 50-MHz digital-to-analog converter structure is used as an efficient interface between the digital loop filter and a conventional LC oscillator.

4. A 1.5-mW asynchronous divider structure is designed to reduce the TDC range by a factor of four as well as to avoid the divide-value-dependent delay variation without the need for re-timing the divider output.

5. A digitally-controlled phase shifter based on the fractional-N technique is introduced. It achieves a 1.4-degrees-resolution infinite-range delay that is less sensitive to the PVT variations. Its application to a DLL for 3.2-Gb/s chip-to-chip communications is demonstrated.

6. A digital $\Delta\Sigma$ modulator architecture and a bang-bang detector that efficiently support the proposed phase shifter are presented.

9.2 Future Research

Although the proposed synthesizer demonstrates excellent noise performance, there are still a few things that can be improved in the future.

In the current implementation, the noise at 20 MHz offset is -150 dBc/Hz. Although it is 2-3 dB higher than the state-of-the-art performance [9][12], it is very clear that our noise at high frequency offsets is limited by the VCO itself instead of the proposed synthesizer structure. Therefore, it is very possible to improve it with a more careful VCO design (see Section 7.6). Reducing the VCO power dissipation is also very attractive because the VCO dominates the power consumption of the prototype.

Also, more noise at intermediate frequency offsets is observed than expected. It is suspected that the source of it is digital switching noise that couples to the VCO through the coarse-tuning DAC. Putting more bypass capacitors in the analog components should be able to improve it.

The prototype synthesizer also exhibits larger fractional spurs than desired. Although ideally less fractional spur should occur when a high-order modulator is used, spurs can still be generated through intermodulation between VCO output frequency and harmonics of the 50-MHz reference. We suspect that the intermodulation occurs due to coupling through ground or power supply since improvement in spurs is observed when power supplies of the reference and VCO output buffers are reduced.

As for the reference spur, one of the sources is the clock feedthrough inside the proposed DAC. One option to improve this is to have another RC lowpass filter following each DAC. In principle, having a pole at 5 MHz should attenuate the reference spur by 20 dB.

If a more complete solution is desired, a method to set the four-bit MIM capacitor array should be investigated. In addition, an easy method for correctly adjusting the open-loop gain must be pursued.

The digital circuits, including the loop filter and noise cancellation, consumes 6.8 mW in the prototype. This power is not optimized in the design phase and should receive more attention in the future implementation.

Furthermore, there are two modified architectures worth pursuing:

(1) An alternative potential way to implement the digital noise cancellation is illustrated in Figure 9-1. Instead of scaling the quantization noise as shown in Figure 2-14, the TDC output is scaled in this architecture. Therefore, the TDC output is automatically normalized to the VCO period, so the TDC gain calibration is self-achieved. One potential problem of this architecture is that the open-loop gain now becomes proportional to the scale factor. Therefore, since the scale factor may slightly fluctuate around the average value in the steady state, the open-loop gain changes and may have some negative impact on the phase noise performance. One way to avoid this problem is to narrow the bandwidth of the correlation circuit, but it results in a longer calibration time. The other way is to freeze the scale factor once the calibration is done. However, it is not feasible for the case where the synthesizer needs to be active all the time because TDC gain may change due to the voltage and temperature variations. In this kind of application, the correlation function therefore needs to be turned on and off frequently.

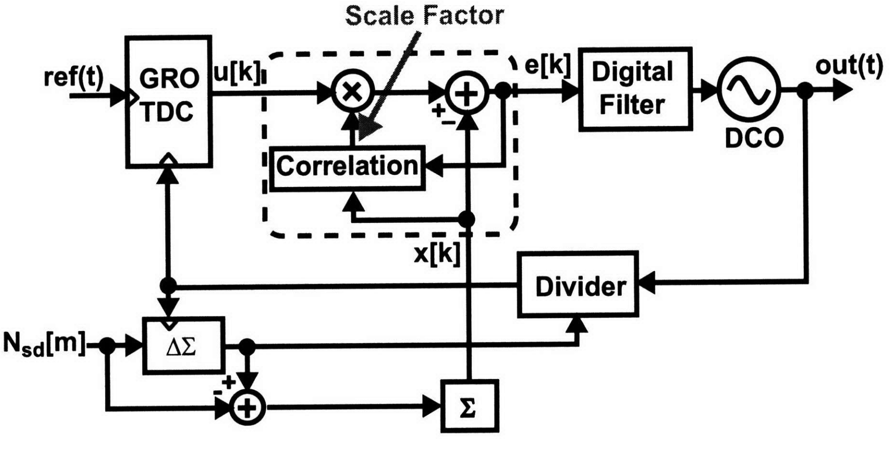


Figure 9-1: The other potential implementation of the digital noise cancellation.

(2) A problem of the proposed PLL architecture is that it may suffer from thermal drift for the case where the PLL needs to operate continuously. To explain that, since the coarse-tuning DAC is frozen in steady state, and the fine-tuning varactor only

provides a small tuning range (K_v=5MHz/V), if the VCO frequency drifts due to the temperature or supply voltage change, the PLL may lose locked. To avoid this issue, another interesting PLL architecture is illustrated in Figure 9-2. The filter is split into two parts: an integration path and a feed-forward path. The integration path is fed to the coarse-tuning varactor; the feed-forward path is fed to the fine-tuning varactor. Since the coarse-tuning varactor can provide a wider frequency range, the PLL has a smaller chance to lose lock even with a serious drift in the VCO frequency. Note that a $\Delta\Sigma$ modulator is needed before the coarse-tuning varactor since the integration path keeps on operating in this case. The challenge of this architecture is the requirement of a linear VCO tuning curve because the control voltages of the coarse and fine tuning varactors can be different. The fine-tuning voltage should be biased at $V_{DD}/2$, but the coarse-tuning voltage needs to cover the whole supply range. Therefore, if the VCO gain is not linear over the whole supply range, the integration and feed-forward gains become functions of the control voltages, making the PLL transfer function difficult to control.

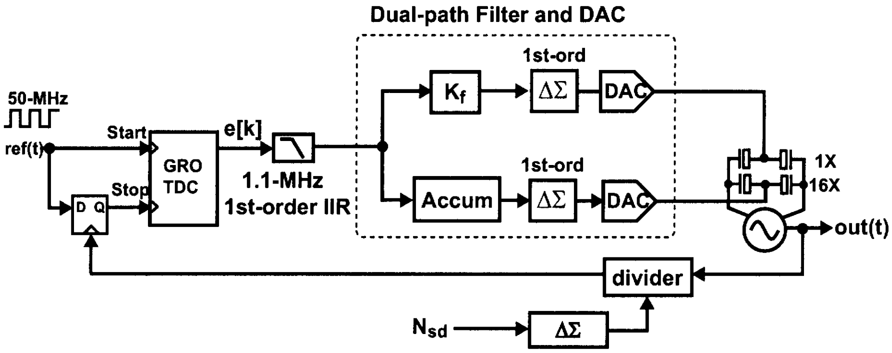


Figure 9-2: A dual-path PLL architecture.

As for the proposed DLL, it will be very interesting to implement this solution in a more advanced process, such as 45-nm CMOS. It is expected that the area and power can be dramatically reduced compared to the prototype chip. Also, many of the digital PLL techniques can be applied to the proposed phase shifter to achieve a more digital implementation. For instance, the proposed phase shifter can be implemented in the form of an all-digital PLL, and it is also possible to replace the bang-bang detector with a TDC.

Bibliography

[1] R. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P. Balsara, "1.3 V 20 ps time-to-digital converter for frequency synthesis in 90-nm CMOS," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 3, pp. 220-224, March 2006.

[2] B. M. Helal, M. Z. Straayer, G.-Y. Wei, and M. H. Perrott, "A highly-digital MDLL-based clock multiplier that leverages a self-scrambling time-to-digital converter to achieve sub-picosecond jitter performance," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 855-763, April 2008.

[3] M. Z. Straayer and M. H. Perrott, "An efficient high-resolution 11-bit noise-shaping multipath gated ring oscillator TDC," *Proc. IEEE Symposium on VLSI Circuits*, June 2008, pp. 82-83.

[4] S. T. Lee, S. J. Fang, D. Allstot, A. Bellaouar, A. Fridi, and P. Fontaine, "A quad-band GSM-GPRS transmitter with digital auto-calibration," *IEEE J. Solid-State Circuits*, vol. 39, no. 12, pp. 2200-2214, Dec. 2004.

[5] C. Vaucher, I. Ferencic, M. Locher, S. Sedvallson, U. Voegeli, and Z. Wang, "A family of low-power truly modular programmable dividers in standard 0.35-μm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 35, no. 7, pp. 1039-1045, July 2000.

[6] B. Murmann and B. Boser, "A 12-bit 75-Ms/s pipelined ADC using open-loop residue amplification," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2040-2050, Dec 2003.

[7] B. Murmann, "Digitally assisted analog circuits," *IEEE Micro*, vol. 26, no. 2, pp. 38-47, March-April 2006.

[8] S. Chung, J. Holloway, and J. Dawson, "Open-loop digital predistortion using cartesian feedback for adaptive RF power amplifier linearization," in *Proc. IEEE/MTT-S International Microwave Symposium*, June 2007, pp. 1449-1452.

[9] R. Staszewski, J. Wallberg, S. Rezeq, C.-M. Hung, O. Eliezer, S. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, M.-C. Lee, P. Cruise, M. Entezari, K. Muhammad, and D. Leipold, "All-digital PLL and transmitter for mobile phones," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2469-2482, Dec. 2005.

[10] R. Tonietto, E. Zuretti, R. Castello, and I. Bietti, "A 3MHz bandwidth low noise RF all digital PLL with 12ps resolution time to digital converter," in *Proc. 32nd European Solid-State Circuits Conference*, Sept. 2006, pp.150-153.

[11] M. Ferriss and M. Flynn, "A 14mW fractional-N PLL modulator with an enhanced digital phase detector and frequency switching scheme," in *Proc. IEEE International Solid-State Circuits Conference*, Feb. 2007, pp. 352-353.

[12] H.-H. Chang, P.-Y.Wang, J.-H. C. Zhan, and B.-Y. Hsieh, "A fractional spur-free ADPLL with loop-gain calibration and phase-noise cancellation for GSM/GPRS/EDGE," in *Proc. IEEE International Solid-State Circuits Conference,* Feb. 2008, pp. 200-201.

[13] C. Weltin-Wu, E. Temporiti, D. Baldi, and F. Svelto, "A 3GHz fractional-N all-digital PLL with precise time-to-digital converter calibration and mismatch correction," in *Proc. IEEE International Solid-State Circuits Conference*, Feb. 2008, pp. 344-345.

[14] C.-M. Hsu, M. Z. Straayer, and M. H. Perrott, "A low-noise wide-BW 3.6GHz digital $\Delta\Sigma$ fractional-N frequency synthesizer with a noise-shaping time-to-digital converter and quantization noise cancellation," in *Proc. IEEE International Solid-State Circuits Conference,* Feb. 2008, pp. 340-341.

[15] C.-M. Hsu, C.-Y. Lau, and M. H. Perrott, "A delay-locked loop using a synthesizer-based phase shifter for 3.2 Gb/s chip-to-chip communication," in *Proc. 32nd European Solid-State Circuits Conference*, Sept. 2006, pp. 460-463.

[16] T. Lee, K. Donnelly, J. Ho, J. Zerbe, M. Johnson, and T. Ishikawa, "A 2.5 V CMOS delay-locked loop for 18 Mbit, 500 Megabyte/s DRAM," *IEEE J. Solid-State Circuits*, vol. 29, no. 12, pp. 1491-1496, Dec. 1994.

[17] A. Coban, M. Koroglu, and K. Ahmed, "A 2.5-3.125-Gb/s quad transceiver with second-order analog DLL-based CDRs," *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1940-1947, Sept. 2005.

[18] R. Kreienkamp, U. Langmann, C. Zimmermann, T. Aoyama, and H. Siedhoff, "A 10-Gb/s CMOS clock and data recovery circuit with an analog phase interpolator," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 736-743, March 2005.

[19] W. Rhee, B. Parker, and D. Friedman, "A semi-digital delay-locked loop using an analog-based finite state machine," *IEEE Trans. Circuits Syst. II*, vol. 51, no. 11, pp. 635-639, Nov. 2004.

[20] B. M. Helal, M. Z. Straayer, G.-Y. Wei, and M. H. Perrott, "A low jitter 1.6 GHz multiplying DLL utilizing a scrambling time-to-digital converter and digital correlation," in *Proc. IEEE Symposium on VLSI Circuits*, Jun. 2007, pp. 166-167.

[21] M. Z. Straayer, "Noise shaping techniques for analog and time to digital converters using voltage controlled oscillators," Ph.D. dissertation, Massachusetts Institute of Technology, 2008.

[22] S. Pamarti, L. Jansson, and I. Galton, "A wideband 2.4-GHz Delta-Sigma fractional-N PLL with 1-Mb/s in-loop modulation," *IEEE J. Solid-State Circuits*, vol. 39, no. 1, pp. 49-62, Jan. 2004.

[23] E. Temporiti, G. Albasini, I. Bietti, R. Castello, and M. Colombo, "A 700-kHz bandwidth Sigma-Delta fractional synthesizer with spurs compensation and linearization techniques for WCDMA applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1446-1454, Sept. 2004.

[24] S. Meninger and M. H. Perrott, "A 1-MHz bandwidth 3.6-GHz 0.18-μm CMOS fractional-N synthesizer utilizing a hybrid PFD/DAC structure for reduced broad-band phase noise," *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 966-980, April 2006.

[25] M. Gupta and B.-S. Song, "A 1.8-GHz spur-cancelled fractional-N frequency synthesizer with LMS-based DAC gain calibration," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2842-2851, Dec. 2006.

[26] A. Swaminathan, K. Wang, and I. Galton, "A wide-bandwidth 2.4 GHz ISM band fractional-N PLL with adaptive phase noise cancellation," *IEEE J. Solid-State Circuits*, vol. 42, no. 12, pp. 2639-2650, Dec. 2007.

[27] J. A. Crawford, *Advanced phase-lock technique*. Artech House, 2008.

[28] T. Riley, M. Copeland, and T. Kwasniewski, "Delta-Sigma modulation in fractional-N frequency synthesis," *IEEE J. Solid-State Circuits*, vol. 28, no. 5, pp. 553-559, May 1993.

[29] B. Miller and R. Conley, "A multiple modulator fractional divider," *IEEE Trans. Instrum. Meas.*, vol. 40, no. 3, pp. 578-583, June 1991.

[30] W. Rhee, B.-S. Song, and A. Ali, "A 1.1-GHz CMOS fractional-N frequency synthesizer with a 3-b third-order $\Delta\Sigma$ modulator," *IEEE J. Solid-State Circuits*, vol. 35, no. 10, pp. 1453-1460, Oct. 2000.

[31] T. Kenny, T. Riley, N. Filiol, and M. Copeland, "Design and realization of a digital ΔΣ modulator for fractional-N frequency synthesis," *IEEE Trans. Veh. Technol.*, vol. 48, no. 2, pp. 510-521, March 1999.

[32] M. H. Perrott, I. Tewksbury, and C. Sodini, "A 27-mw CMOS fractional-N synthesizer using digital compensation for 2.5-Mb/s GFSK modulation," *IEEE J. Solid-State Circuits*, vol. 32, no. 12, pp. 2048-2060, Dec. 1997.

[33] C. Durdodt, M. Friedrich, C. Grewing, M. Hammes, A. Hanke, S. Heinen, J. Oehm, D. Pham-Stabner, D. Seippel, D. Theil, S. Van Waasen, and E. Wagner, "A low-IF RX two-point ΔΣ modulation TX CMOS single-chip Bluetooth solution," *IEEE Trans. Microw. Theory Tech.*, vol. 49, no. 9, pp. 1531-1537, Sept. 2001.

[34] M. H. Perrott, M. Trott, and C. Sodini, "A modeling approach for ΔΣ fractional-N frequency synthesizers allowing straightforward noise analysis," *IEEE J. Solid-State Circuits*, vol. 37, no. 8, pp. 1028-1038, Aug. 2002.

[35] M. H. Perrott, "Digital phase-locked loops," in *ISSCC 2008 Tutorials*.

[36] S.-J. Lee, B. Kim, and K. Lee, "A novel high-speed ring oscillator for multiphase clock generation using negative skewed delay scheme," *IEEE J. Solid-State Circuits*, vol. 32, no. 2, pp. 289-291, Feb. 1997.

[37] E. Hegazi, H. Sjoland, and A. Abidi, "A filtering technique to lower LC oscillator phase noise," *IEEE J. Solid-State Circuits*, vol. 36, no. 12, pp. 1921-1930, Dec. 2001.

[38] M. H. Perrott, Y. Huang, R. T. Baird, B. W. Garlepp, D. Pastorello, E. T. King, Q. Yu, D. B. Kasha, P. Steiner, L. Zhang, J. Hein, and B. Del Signore, "A 2.5-Gb/s multi-rate 0.25-μm CMOS clock and data recovery circuit utilizing a hybrid analog/digital loop filter and all-digital referenceless frequency acquisition," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2930-2944, Dec. 2006.

[39] M. Kokubo, K. Hori, T. Ito, Y. Tazaki, and N. Takei, "A fast-frequency-switching PLL synthesizer LSI with a numerical phase comparator," in *Proc. IEEE International Solid-State Circuits Conference*, Feb. 1995, pp. 260-261.

[40] R. Staszewski, D. Leipold, K. Muhammad, and P. Balsara, "Digitally controlled oscillator (DCO)-based architecture for RF frequency synthesis in a deep-submicrometer CMOS process," *IEEE Trans. Circuits Syst. II*, vol. 50, no. 11, pp. 815-828, Nov. 2003.

[41] R. Radke, A. Eshraghi, and T. Fiez, "A 14-bit current-mode $\Delta\Sigma$ DAC based upon rotated data weighted averaging," *IEEE J. Solid-State Circuits*, vol. 35, no. 8, pp. 1074-1084, Aug. 2000.

[42] D. A. Johns and K. Martin, *Analog Integrated Circuit Design*. John Wiley & Sons, Inc., 1997.

[43] B. Razavi, *Design of analog CMOS integrated circuits*. McGraw-Hill, 2001.

[44] C.-M. Hsu, "Digital fractional-N synthesizer example achieving wide bandwidth and low noise." [Online]. Available: http://www.cppsim.com.

[45] P. Gray, P. Hurst, S. Lewis, and R. Meyer, *Analysis and design of analog integrated circuits*. Wiley, 2001.

[46] C.-A. Gobet and A. Knob, "Noise generated in switched capacitor networks," *Electronics Letters*, vol. 16, no. 19, pp. 734-735, Sept. 1980.

[47] R. Bunch and S. Raman, "Large-signal analysis of MOS varactors in CMOS – G_m LC VCOs," *IEEE J. Solid-State Circuits*, vol. 38, no. 8, pp. 1325-1332, Aug. 2003.

[48] A. Hajimiri, S. Limotyrakis, and T. Lee, "Jitter and phase noise in ring oscillators," *IEEE J. Solid-State Circuits*, vol. 34, no. 6, pp. 790-804, June 1999.

[49] A. Jerng and C. Sodini, "The impact of device type and sizing on phase noise mechanisms," *IEEE J. Solid-State Circuits*, vol. 40, no. 2, pp. 360-369, Feb. 2005.

[50] J. Craninckx and M. Steyaert, "A 1.8-GHz low-phase-noise CMOS VCO using optimized hollow spiral inductors," *IEEE J. Solid-State Circuits*, vol. 32, no. 5, pp. 736-744, May 1997.

[51] B. De Muer, M. Borremans, M. Steyaert, and G. Li Puma, "A 2-GHz low-phase-noise integrated LC-VCO set with flicker-noise upconversion minimization," *IEEE J. Solid-State Circuits*, vol. 35, no. 7, pp. 1034-1038, July 2000.

[52] J. Yuan and C. Svensson, "High-speed CMOS circuit technique," *IEEE J. Solid-State Circuits*, vol. 24, no. 1, pp. 62-70, Feb. 1989.

[53] M. H. Perrott. "High speed communication circuits and systems." [Online]. Available: http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-976High-Speed-Communication-Circuits-and-SystemsSpring2003/LectureNotes/index.htm

[54] M. H. Perrott, "PLL design using the PLL design assistant program." [Online]. Available: http://www.cppsim.com.

[55] C. Y. Lau and M. H. Perrott, "Fractional-N frequency synthesizer design at the transfer function level using a direct closed loop realization algorithm," in *Design Automation Conference*, Jun. 2003, pp. 526-531.

[56] M. H. Perrott, "Cppsim behavior simulator package." [Online]. Available: http://www.cppsim.com.

[57] E. A. Lee and D. Messerschmitt, *Digital Communication*. Kluwer Academic Publishers, 1994.

[58] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*. Prentice Hall, 2003.

[59] M. H. Perrott, "Fast and accurate behavioral simulation of fractional-N frequency synthesizers and other PLL/DLL circuits," in *Proc. 39th Design Automation Conference*, Jun. 2002, pp. 498-503.

[60] G.-K. Dehng, J.-M. Hsu, C.-Y. Yang, and S.-I. Liu, "Clock-deskew buffer using a SAR-controlled delay-locked loop," *IEEE J. Solid-State Circuits*, vol. 35, no. 8, pp. 1128-1136, Aug. 2000.

[61] C. Kromer, G. Sialm, C. Menolfi, M. Schmatz, F. Ellinger, and H. Jackel, "A 25-Gb/s CDR in 90-nm CMOS for high-density interconnects," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2921-2929, Dec. 2006.

[62] J. Deveugele and M. Steyaert, "A 10-bit 250-Ms/s binary-weighted current-steering DAC," *IEEE J. Solid-State Circuits*, vol. 41, no. 2, pp. 320-329, Feb. 2006.

[63] Y. Eken and J. Uyemura, "A 5.9-GHz voltage-controlled ring oscillator in 0.18-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 39, no. 1, pp. 230-233, Jan. 2004.

[64] M. Thamsirianunt and T. Kwasniewski, "A 1.2-μm CMOS implementation of a low-power 900-MHz mobile radio frequency synthesizer," in *Proc. IEEE Custom Integrated Circuits Conference*, May 1994, pp. 383-386.