Low Phase Noise, High Bandwidth Frequency Synthesis Techniques

by

Scott Edward Meninger

Bachelor of Science in Electrical Engineering and Computer Science
Boston University, June 1996
Master of Science in Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 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

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2005

Author .....
Department of Electrical Engineering and Computer Science
May 12, 2005

Certified by .....
Michael H. Perrott
Assistant Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Low Phase Noise, High Bandwidth Frequency Synthesis Techniques

by

Scott Edward Meninger

Submitted to the Department of Electrical Engineering and Computer Science
on May 12, 2005, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

A quantization noise reduction technique is proposed that allows fractional-N frequency synthesizers to achieve high closed loop bandwidth and low output phase noise simultaneously. Quantization induced phase noise is the bottleneck in state-of-the-art synthesizer design, and results in a noise-bandwidth tradeoff that typically limits closed loop synthesizer bandwidths to be <100kHz for adequate phase noise performance to be achieved. Using the proposed technique, quantization noise is reduced to the point where intrinsic noise sources (VCO, charge-pump, reference and PFD noise) ultimately limit noise performance.

An analytical model that draws an analogy between fractional-N frequency synthesizers and MASH $\Sigma\Delta$ digital-to-analog converters is proposed. Calculated performance of a synthesizer implementing the proposed quantization noise reduction techniques shows excellent agreement with simulation results of a behavioral model. Behavioral modeling techniques that progressively incorporate non-ideal circuit behavior based on SPICE level simulations are proposed.

The critical circuits used to build the proposed synthesizer are presented. These include a divider retiming circuit that avoids meta-stability related to synchronizing an asynchronous signal, a timing mismatch compensation block used by a dual divider path PFD, and a unit element current source design for reduced output phase noise.

Measurement results of a prototype $0.18\mu m$ CMOS synthesizer show that quantization noise is suppressed by 29dB when the proposed synthesizer architecture is compared to 2^{nd} order $\Sigma\Delta$ frequency synthesizer. The 1MHz closed loop bandwidth allows the synthesizer to be modulated by up to 1Mb/s GMSK data for use as a transmitter with 1.8GHz and 900MHz outputs. The analytical model is used to back extract on-chip mismatch parameters that are not directly measurable. This represents a new analysis technique that is useful in the characterization of fractional-N frequency synthesizers.

Thesis Supervisor: Michael H. Perrott
Title: Assistant Professor

This thesis is dedicated to the example and inspiration of my parents, Donald T. and Mary J. Meninger. Dad, I miss you every day. You will always be my best man.

"While I'm still confused and uncertain, it's on a much higher plane, d'you see, and at least I know I'm bewildered about the really fundamental and important facts of the universe." Treatle nodded. "I hadn't looked at it like that," he said, "But you're absolutely right. He's really pushed back the boundaries of ignorance".

They both savored the strange warm glow of being much more ignorant than ordinary people, who were only ignorant of ordinary things.

– *Terry Pratchett, Equal Rites*

Acknowledgments

"Give a man a fire and he's warm for a day, but set fire to him and he's warm for the rest of his life."

– *Terry Pratchett, Jingo*

In many ways, the grad school experience is like being set on fire. There's an initial burst of enthusiasm after which you feel like you're running around desperately asking people to put you out. After various periods of flaming up, someone dumps a bucket of water on you, and you wind up feeling a bit burnt out, but happy that the pain has stopped. And along the way you've learned a few valuable lessons....

....Of course, another way of looking at it is that grad school is in no way like being lit on fire. In fact, one could argue that grad school is the exact opposite of being lit on fire. And one thing you definitely learn in grad school is to be very, very careful about general statements and assumptions, so I'll just say that grad school is very much like being lit on fire, while in no way like being lit on fire.

So, on to more important matters ...

There are countless numbers of people to thank for the help, influence, and inspiration they have given me over the course of my PhD experience. In advance, I apologize if anyone is not included in this acknowledgment section who should be.

I'd like to start by thanking my thesis advisor Professor Mike Perrott. Mike has been a constant motivating force and positive influence on the work that is presented in this thesis. I have learned a phenomenal amount over the last four years, and owe much of my education to Mike's personal tutelage.

I owe my committee, Professors Anantha Chandrakasan and Jacob White a debt of gratitude for taking the time to meet with me during the course of this work, and for reading this thesis.

I would not have made it through the PhD experience without the help and support of numerous grad students working in MTL. They have aided me technically, as well as in the more important aspect of having some fun. Anh Pham, John Fiorenza, Nisha Checka, Albert Jerng, Andrew Chen, Mark Spaeth, and Tod Sepke all were sources of good technical discussions as well as good guys to grab a beer or go ski with. Tod also was very helpful (and patient) in helping me get a handle on noise processes and simulation.

The members of the High Speed Circuits and Systems Group have coped with my .... personality over the last four years, and so I owe them a huge debt of gratitude. Ethan Crain has been a great lab-mate to design with and work out with. Assuming I'm not trying to keep up with him running, that is. I'm not sure who owes who a beer at this point, but I look forward to being able to enjoy one with him in a more relaxed setting soon. Charlotte Lau has borne the brunt of what will be referred to as my "humor". Her patience and help (and lack of litigation to this point), are much appreciated. I'll miss our trips to ABP for lunch, and the recognition that comes with walking around with the queen of MIT. Belal Helal and I have had far too many late night discussions that have increased our graduation dates as well as our understanding. Belal's thoroughness has been an example. Matt Park is responsible for the excellent work on the data-weighted averager circuit used by my test chip. There are not many undergrads who I would trust with my something going into my thesis chip, but Matt was a clear exception. I'm looking forward to seeing the work he'll produce in the future. I also owe thanks to Min Park, Chun-Ming Hsu, Shawn Kuo, and Muyi Ogunnika, who have helped me out in numerous ways over the last four years.

Our group administrators, Rosa Fuentes and Valerie DiNardo, have been a great help getting the lab set up and putting through my very disorganized reimbursements.

The MTL staff has done a great job helping the students make our way through the grad school experience. Debb Hodges Pabon, in particular, has always made herself

available to us and done a fantastic job coordinating the numerous MTL gatherings.

I'd like to thank Marilyn Pierce in the EECS grad office for her help, guidance, and patience over the years that have comprised my eclectic grad school schedule.

The folks at National Semiconductor helped make my tapeout process as easy as it could be. In particular, Sangamesh Buddhiraju's CAD help was critical in getting the National setup working at MIT, and his excellent front-end work saved us countless hours of frustration. The National-MIT program wouldn't even exist if it wasn't for the work of Peter Holloway, who I've been lucky enough to work for and with over the years. Pete is truly one of the great guys I've met in the course of my travels, and I look forward to our paths crossing again.

I'd like to thank MTL CICS and MARCO for funding my research. I'd also like to thank the folks at Teradyne, especially Kurt Ware and Ron Sartchev, for funding me as a Teradyne fellow for a year. It was encouraging along the way to see that my research was sparking real-world interest.

I'd also like to thank Duke Xanthopoulos for helping me out with the initial CAD setup, and setting up a great situation for me at Cavium. I'm really looking forward to getting started.

My friends have been very supportive, in what can only be called their unique and "special" way. The boys of DOAIHPS have helped me enjoy what little free time I've had over the last four years, and I'm looking forward to finally having some time to waste with them now. Ken, Chris, Rufus, and Rob, the first $1.35 (or, $1.85 now...) at York is on me. (Hopefully you won't read this until *after* York...)

My family has been a strong influence on my life. I'd like to thank Steven, Mark, Cheryl, Donna, D.A., Kathy, Laureen, Dorene, Ellen, and John for the impact they've had in my getting to this point. Another way to look at this, of course, is that they are at least partly to blame. This is the way the youngest of eleven learns to think. I'm sure the survival skills I learned growing up in such a large family have helped me

out in un-accounted for ways over the years. Figuring out ways to reduce quantization noise by 29dB is nothing compared to trying to figure out, as an 8 year old, how to buy 21 Christmas presents with $20. Now that is tough math.

I'd like to thank my extended family, the Rhoads and Garlands, for their support over the many years they've known me.

I thank my wife, Stacy Meninger, for helping shape me to be the man I am today.

Finally, as long as this thesis is, if I spent every page trying to thank my parents for what they've done for me over the years, I'd run out of space. The example they've set and the support they've shown me have been overwhelming. So thank you most of all, Mom and Dad.

Contents

List of Figures

List of Tables

Chapter 1

Area of Focus: Fractional-N Synthesis

Frequency synthesis is an essential technique employed in RF systems to achieve local oscillator (LO) generation or direct modulation transmission [2]. Fractional-N synthesis offers the advantage over integer-N based systems of decoupling the choice of synthesizer resolution from its bandwidth. Fast settling, high resolution synthesis becomes possible, giving greater design flexibility at the system level. Fractional-N synthesis can be separated into two categories: classical fractional-N synthesis and $\Sigma\Delta$ fractional-N synthesis.

In this chapter we will explore the tradeoffs associated with the different types of fractional-N synthesis. We will show that the key constraint limiting the bandwidth of modern synthesizers centers around management of fractional-N quantization noise. Quantization noise is introduced to the synthesizer as a by-product of the fractional-N dithering process, and so is unavoidable. After examining existing techniques aimed at lowering the magnitude of quantization noise, we propose a synthesizer architecture that overcomes the core limitation of this prior work, namely achieving a high quality gain match between the quantization noise and a cancellation signal.

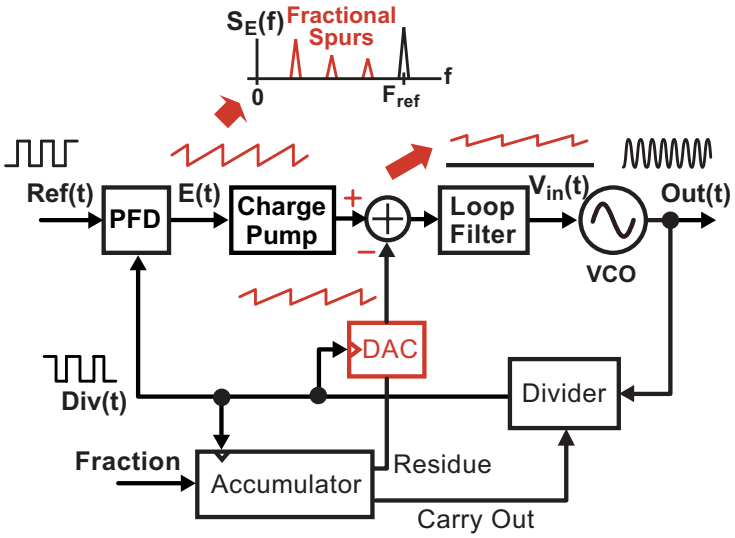


Figure 1-1 Classic Fractional-N Synthesis with Phase Interpolation

1.1 The Issue of Fractional-N Quantization Noise

The classical approach to fractional-N synthesizer design employs dithering and phase interpolation, as depicted in Figure 1-1. [3] An accumulator carry out signal is used to dither the control input to a multi-modulus divider such that a fractional average divide value is obtained from a divider that supports integer values. A digital to analog converter (DAC) is used in conjunction with a phase accumulation register to cancel out periodicities in the phase error signal, E(t). These periodicities represent quantization noise introduced by the fractional-N dithering process. The main performance limitation of this approach centers around the difficulty in creating a precise match between the noise cancellation DAC output and the phase error signal.

In $\Sigma\Delta$ fractional-N synthesis, the most popular technique used today to generate fractional divide values [4–9], the spurious performance is improved through $\Sigma\Delta$ modulation of the divider control. The quantization noise introduced by dithering the divide value is therefore whitened and shaped to high frequencies, such that it is substantially filtered by the synthesizer dynamics. In order to obtain sufficient randomization to reduce spurs to negligible levels, $\Sigma\Delta$ modulators of order 3 or higher (often employing LSB dithering) are often employed, necessitating a higher order loop filter to counteract increased noise slope. The shaped quantization noise can easily

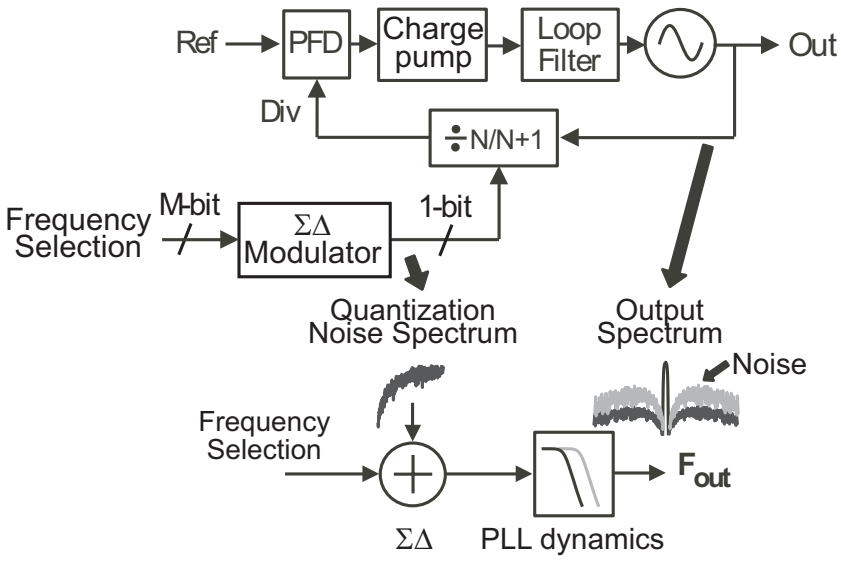


Figure 1-2 ΣΔ Fractional-N Synthesizer and Noise-Bandwidth Tradeoff

dominate at high offset frequencies, introducing a noise-bandwidth tradeoff which translates to low closed loop bandwidths for low phase-noise synthesizers. Depicted in Figure 1-2, this tradeoff is the bottleneck preventing high bandwidth fractional-N synthesis, and therefore presents an obstacle to achieving the central goal behind fractional-N synthesis, which is to increase synthesizer bandwidth. Bandwidths reported in the literature for ΣΔ synthesizers are typically below 100kHz for high performance applications [8, 10–14].

1.2 Prior Work Aimed at Reducing Fractional-N Quantization Noise

Two approaches have emerged to improve the noise-bandwidth tradeoff. The first involves reducing the quantization step-size of the divide value dithering action through the use of multiple VCO (or divider) phases [15–17]. Introducing multiple VCO phases is the ideal means by which to reduce the quantization step-size. To explain, phase is the system parameter on which a PLL operates, so choosing one of a number of finely spaced VCO phases to clock the divider directly translates to a reduced

quantization interval. In a similar manner, multiple divider output phases may be created, though it is still necessary to space these phases by some fraction of a VCO period, since the VCO period is the fundamental quantization step-size in the system.

In practice, the number of VCO or divider phases that can be accurately generated is limited. Phase resolution is often set by a gate delay which, for high frequency outputs, can be a significant fraction of the VCO period. Additionally, to generate multiple phases, either ring oscillators must be used, which have inherently poorer phase noise performance than LC oscillators, or a DLL with all of the associated overhead must be employed [17]. Mismatch between the phases occurs and must be carefully dealt with.

Very recently, a modification has been proposed to the selection logic used in the multi-phase approach of [15]. In this case, very high order $\Sigma\Delta$ modulation is applied to the phase selection mux in order to make each tap equi-probable in a histogram sense [18]. For the simulated synthesizer proposed in [18], a 7^{th} order $\Sigma\Delta$ modulator was used to randomize the selection of output phase for a 16 phase divider. Noise shaping of this order requires large numbers of extra poles in the loop filter to counteract the increased noise slope of the high order modulator. Additionally, the multi-phase approach is ultimately limited in its ability to reduce broadband phase noise by limitations in creating the delay.

The second approach to reduce the noise-bandwidth tradeoff uses a DAC to cancel the error signal [19, 20]. This method builds on the idea behind phase interpolation, but introduces $\Sigma\Delta$ design techniques to reduce the impact of DAC nonlinearity. The main limitations with this architecture center around achieving a good match between the DAC output and phase error signal. Matching these signals is difficult because they are processed by separate circuits whose outputs are summed. An additional limitation of this technique is that a high resolution DAC is required to achieve high levels of noise cancellation. In [19], a 4 bit coarse DAC and 4 bit fine DAC were used, resulting in 16dB improvement in broadband phase noise and -45dBc fractional spur levels. A similar approach proposed in [20] results in 15dB noise suppression and -60dBc fractional spurs for a 5-bit cancellation DAC.

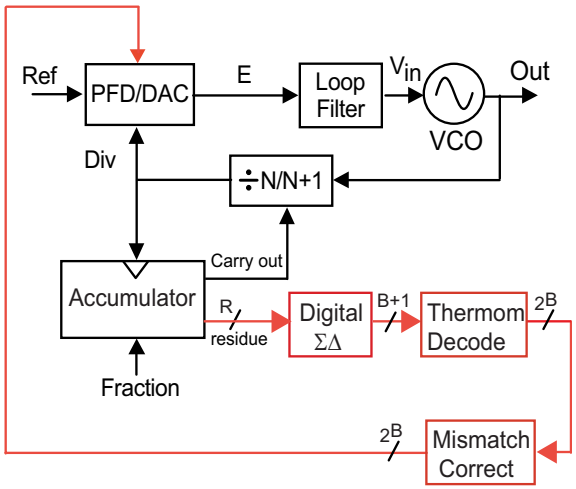


Figure 1-3 Proposed Synthesizer Architecture

An alternative approach that utilizes a DAC to reduce quantization-induced phase noise is proposed in [21]. The separate phase detector and DAC circuit elements are replaced by a hybrid structure, which we will refer to as a PFD/DAC. By embedding the two functions into one circuit, an intrinsically better gain match between the phase error and DAC cancellation signals is obtained. However, the architecture presented in [21] does not address the issue of mismatch between unit elements of the DAC, or between the timing signals in the phase detector, which will result in incomplete phase error signal cancellation and spurious feed-through.

While all of the proposed techniques described above succeed in reducing broadband phase noise levels by effectively reducing the phase quantization step-size, their performance is limited by mismatch between the quantization noise and the cancellation signal.

1.3 Proposed Quantization Noise Reduction Technique

To reduce fractional-N quantization noise, we propose the architecture shown in Figure 1-3. The proposed synthesizer leverages advances in noise-shaping DAC design to ease the requirements on the cancellation DAC used by the traditional fractional-N

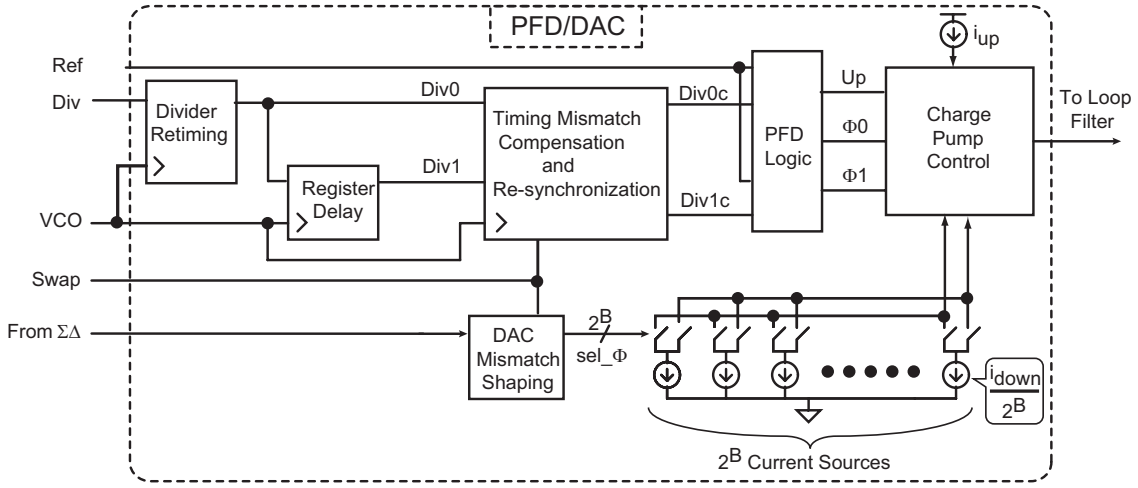


Figure 1-4 Proposed Mismatch Compensated PFD/DAC

approach. It utilizes a PFD/DAC structure [21] to obtain a good intrinsic gain match between the phase error and DAC cancellation signal. However, this work makes the key contribution of introducing techniques to minimize the impact that PFD/DAC mismatch sources have on phase noise performance. Indeed, matching issues create the bottleneck in previous approaches since they result in error feed-through that is manifested in the phase noise spectrum as large spurs, or increased broadband phase noise. The proposed architecture incorporates several digital signal processing techniques to reduce the impact of non-idealities that occur in the PFD/DAC such as unit element mismatch, timing mismatch, and any residual gain mismatch occurring between the PFD/DAC output and phase error signal.

The key issue with prior fractional-N synthesizer implementations is that the cancellation DAC output and phase error signal are poorly matched. This error is a direct result of the fact that separate circuits have been used to implement the two required blocks. Figure 1-4 depicts the proposed PFD/DAC structure, which greatly reduces mismatch between the two signal paths. The proposed PFD/DAC differs from a prior implementation of a hybrid phase detector/cancellation DAC scheme [21] in that it compensates for mismatches within the PFD/DAC structure itself. As will be discussed, mismatch between magnitudes of the phase error and cancellation signal and timing mismatch between signal paths in processing phase information is

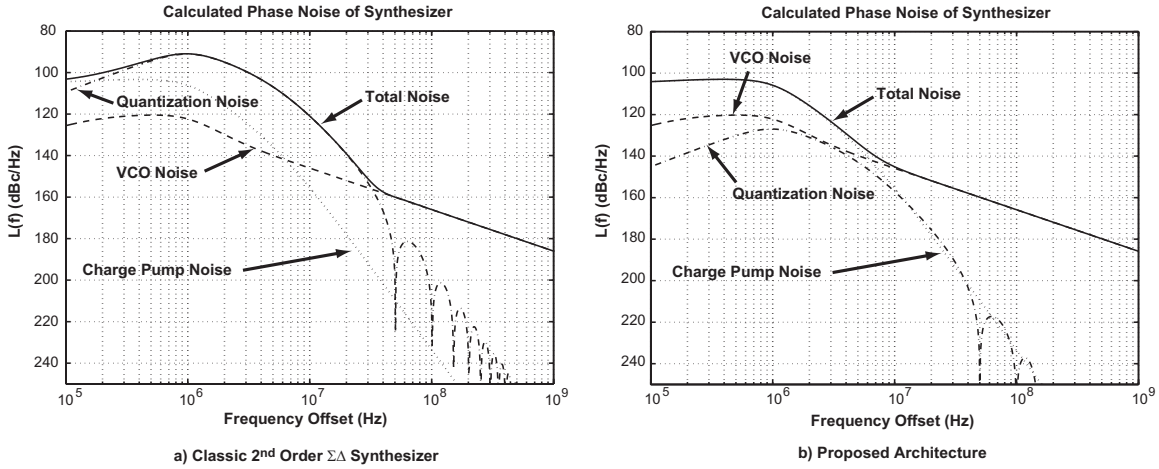


Figure 1-5 Calculated Phase Noise

a key consideration for achieving a high quality overall gain match.

1.4 Thesis Scope and Contributions

The mismatch compensated PFD/DAC synthesizer improves phase noise performance by reducing fractional-N quantization noise. If *quantization* noise can be lowered significantly, it is possible that it will not have a noticeable impact on overall phase noise performance, and that *intrinsic* noise sources should become the area of design focus for continued bandwidth extension. By intrinsic noise, we mean noise sources that are inherent to any PLL, such as charge-pump device noise, reference jitter, divider jitter, and VCO phase noise.

This thesis proposes techniques which allow low noise, high bandwidth, and fine resolution to be simultaneously achieved by a frequency synthesizer. Both intrinsic noise sources, as well as fractional-N quantization-induced phase noise are investigated. Our primary focus is reducing the quantization noise impact on output phase noise by using the proposed mismatch compensated PFD/DAC synthesizer architecture.

Figure 1-5 shows a calculation of output phase noise for a 1MHz bandwidth fractional-N synthesizer, with each noise source's contribution shown individually. The left plot in the figure clearly shows that, for the case of a 2^{nd} order $\Sigma\Delta$ synthe-

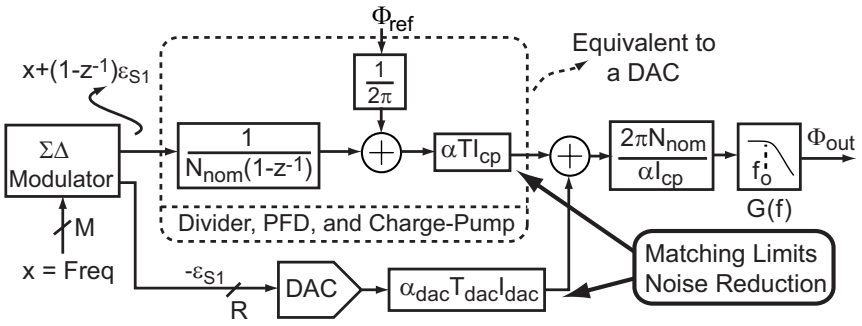


Figure 1-6 New Model View of Fractional-N Synthesizer as a MASH ΣΔ DAC

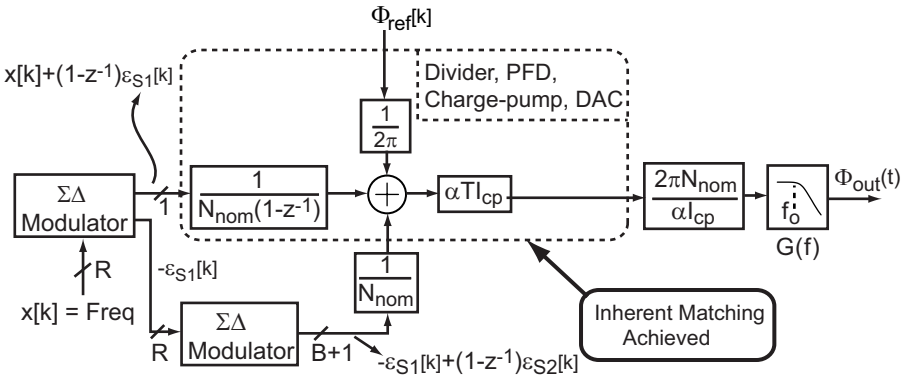


Figure 1-7 PFD/DAC Synthesizer Model

sizer, quantization induced phase noise dominates over a broad frequency range. The right plot shows that, if quantization noise can be reduced by 40dB, the synthesizer noise profile is determined by the *intrinsic* noise sources. Because quantization noise is removed from consideration with respect to output phase noise, the fractional-N synthesizer looks, from a noise standpoint, like an integer-N synthesizer!

1.4.1 System Analytical Noise Modeling

Our exploration into ways to reduce the impact of fractional-N quantization noise will begin with an examination of the model used for phase noise analysis proposed in [1]. In Chapter 3, we propose re-formatting this noise model to the form depicted in Figure 1-6. The new model view suggests that phase interpolation is both a general case of fractional-N synthesis that is a superset of the synthesizer architectures described in this thesis, as well as being directly analogous to a ΣΔ DAC. The previously de-

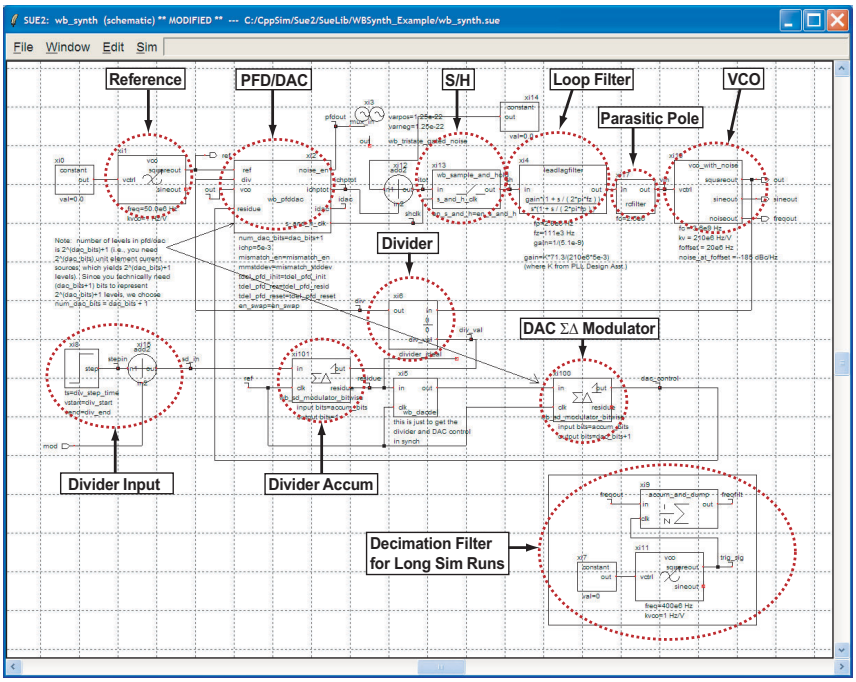


Figure 1-8 Behavioral Model for PFD/DAC Synthesizer

scribed mismatch between the feed-forward cancellation path and the quantization error signal can been seen directly in the figure.

To eliminate the systematic mismatch problem associated with classical phase interpolation based synthesis, we propose the PFD/DAC synthesizer architecture, which can be represented by the model in Figure 1-7. Unlike the approach described in [21], the proposed mismatch compensated PFD/DAC synthesizer introduces dynamic element matching techniques to mitigate circuit mismatch internal to its structure. We will present details of the mismatch compensated PFD/DAC structure in Chapter 4.

1.4.2 Behavioral Modeling and Simulation

Fractional-N synthesizers are difficult systems to simulate because the system dynamics have dominant poles typically on the order of kHz to MHz, while the synthesizer output frequency is usually in the GHz range. Simultaneously capturing transient effects occurring in a fraction of the VCO period and dynamic settling of the loop filter require prohibitively long simulations if performed at the transistor level [22–24].

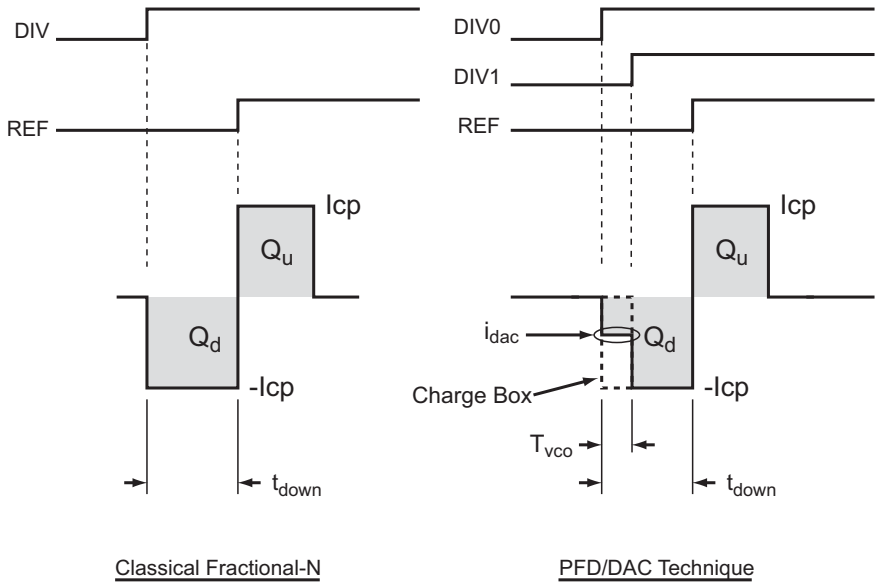


Figure 1-9 PFD/DAC Technique Charge-box

For this reason, we use the C++ behavioral simulator CppSim described in [23] and freely available at http://www-mtl.mit.edu/researchgroups/perrottgroup/tools.html.

In Chapter 5 we propose behavioral modeling techniques used to capture non-ideal circuit characteristics from transistor level simulations and include them in behavioral simulations. Figure 1-8 depicts the behavioral model used for simulation. We will show good agreement between the results of behavioral simulation and the analytical model, motivating the use of behavioral simulation in complex mixed signal systems such as frequency synthesizers. Perhaps more importantly, we demonstrate that key internal system parameters that are not directly measurable can be back extracted using the analytical model, a powerful new analysis technique that the PLL designer can use for system characterization.

1.4.3 Circuit Design

The PFD/DAC technique uses a one VCO period wide charge-box controlled by a DAC to achieve cancellation of the fractional-N quantization noise [25]. A snapshot of PFD/DAC operation is compared to classical fractional-N operation in Figure 1-9. As will be explained in detail in Chapter 4, the phase detector architecture chosen

for the PFD/DAC results in steady-state operation where the reference signal, REF, always occurs after the divider, DIV, and produces a constant amount of positive charge Q_u.

Because the reference operates at a constant frequency, its edge location, ignoring random jitter, will always occur at the same location in time. What is meant here is that the reference signal has a constant period. By contrast, the divider period, and therefore its edge location, will vary over time due to the fractional-N dithering process. The varying divider edge location causes the negative current pulse-width, t_{down}, to vary over time because t_{down} is determined by the difference between the divider edge and reference edge. In a classical fractional-N synthesizer, *all* of the charge-pump negative current is delivered at once, and so an instantaneous error charge equal to $Q_u - Q_d$ is produced. It is important to note that the *average* error charge occurring over several periods will equal zero when the synthesizer is operating in steady-state, but the *instantaneous* error charge on a period-by-period basis will be non-zero. This error charge is due to the fractional-N dithering process and results in fractional spurs.

The PFD/DAC technique controls the value of current delivered in a one VCO period wide window to compensate for the instantaneous phase errors introduced by the fractional-N dithering process. The divider output and a one VCO period delayed version of the divider output create the timing window used to create the charge-box. As negative pulse-width, t_{down}, changes period-by-period due to dithering, the value of i_{dac} is changed to compensate such that $Q_d = Q_u$ *every* period, and instantaneous charge errors do not occur. The DAC current is referenced to a single VCO period because the fractional-N dithering introduces phase error that is referenced to a single VCO period, and therefore changes t_{down} by some fraction of a VCO period. The fractional-N dithering accumulator contains information about the instantaneous phase error magnitude in its residue [26]. The residue is therefore used to control the DAC output level. A detailed explanation of PFD/DAC operation is given in Chapter 4.

A key merit of the PFD/DAC technique is that, as the VCO period changes with

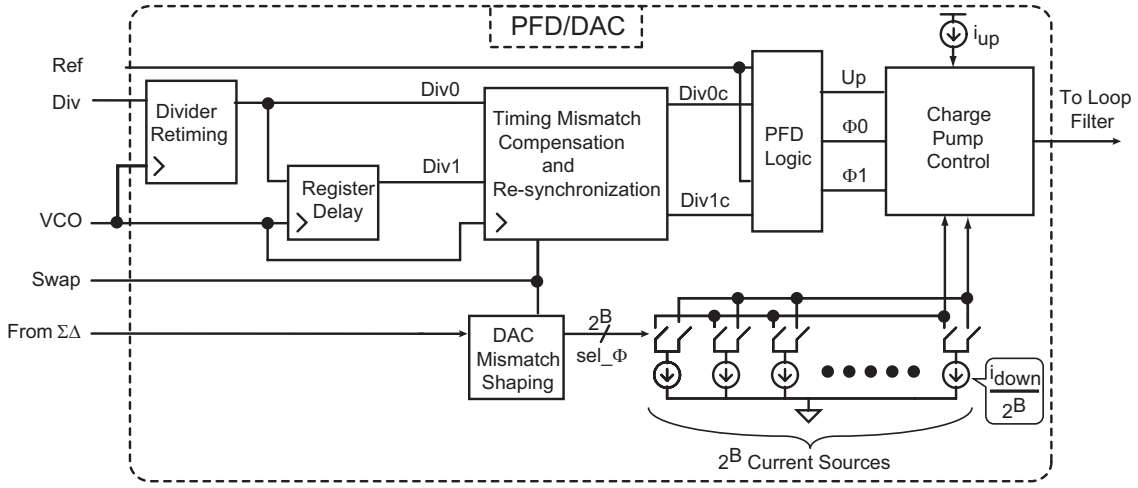


Figure 1-10 Proposed Mismatch Compensated PFD/DAC

different divide values, the width of the charge-box changes in the same manner. The charge-box therefore tracks the instantaneous phase errors over time, resulting in a high-quality gain match between the quantization error signal (the changing divider edge location and value of t_{down}) and the cancellation signal (the changing magnitude of i_{dac}).

Accurate generation of the charge-box is key to achieving accurate cancellation of the quantization noise. In Chapter 6 we propose circuit design techniques that enable accurate generation and control of the charge-box.

The charge-box is a single VCO period wide, and since the prototype synthesizer is designed to operate with a 3.6GHz VCO, operation of PFD/DAC circuitry with fast edges is critical. Because the PFD/DAC approach effectively removes quantization noise from overall synthesizer noise performance, other noise sources within the system, such as charge-pump device noise, become dominant. Therefore, low noise circuit design techniques are also proposed in Chapter 6.

Figure 1-10 shows the proposed mismatch compensated PFD/DAC architecture. Three key circuit blocks are introduced to the basic PFD/DAC structure [21] to compensate for error sources. First, a divider retiming block is used to re-synchronize the divider to the VCO. This establishes the first timing edge, Div0, used to create the charge-box.

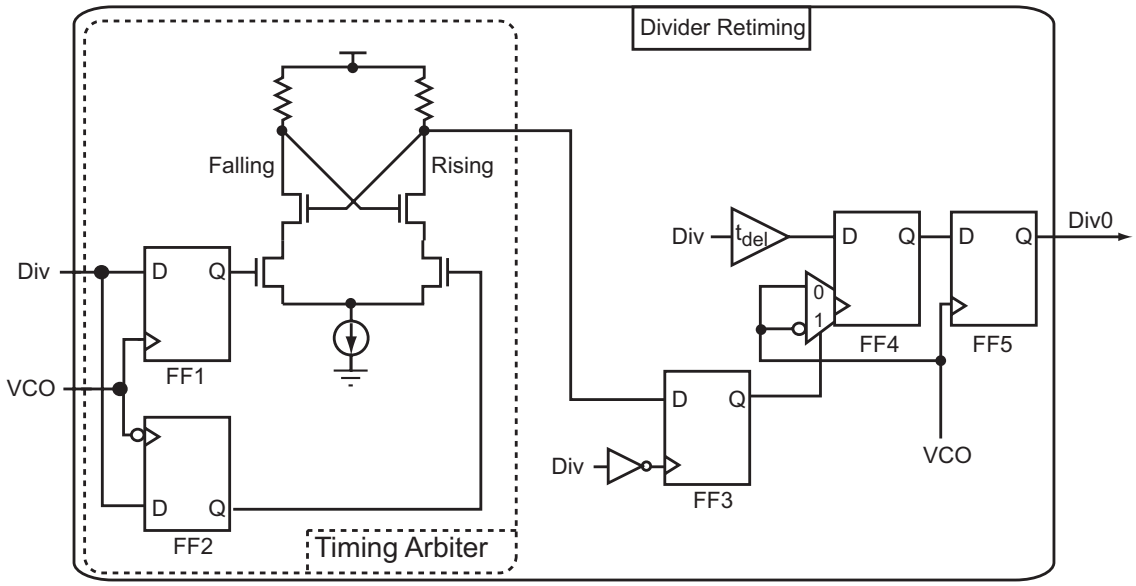


Figure 1-11 Proposed Divider Retiming Circuit

The second edge, Div1, used to create the charge-box is generated by a register delay. Mismatch between the paths seen by Div0 and Div1 is compensated by a timing mismatch block, which dynamically matches the paths, resulting in compensated signals, Div0c and Div1c that are, on average, exactly one VCO period apart. The PFD logic processes Div0c and Div1c and generates two output phase signals, Φ_0 and Φ_1, which combine with the charge-pump circuitry to create the charge-box.

The DAC unit element current sources used to generate the charge-box will have some mismatch between them. This mismatch represents a gain error, and can severely limit noise cancellation. A DAC mismatch shaping block is introduced to dynamically match the unit elements, and improve overall noise performance.

As a preview of the circuit details discussed in Chapter 6, we present our proposed divider retiming circuit in Figure 1-4. For proper operation and timing window duration, the divider output must be synchronized to the VCO. However, high speed multi-modulus dividers used in fractional-N synthesizers are typically asynchronous in nature due to speed and power considerations [27]. The divider output must therefore be synchronized to the VCO *before* the flip-flop delay used to create the PFD/DAC cancellation window.

The proposed approach differs from previous divider retiming techniques [19, 28],

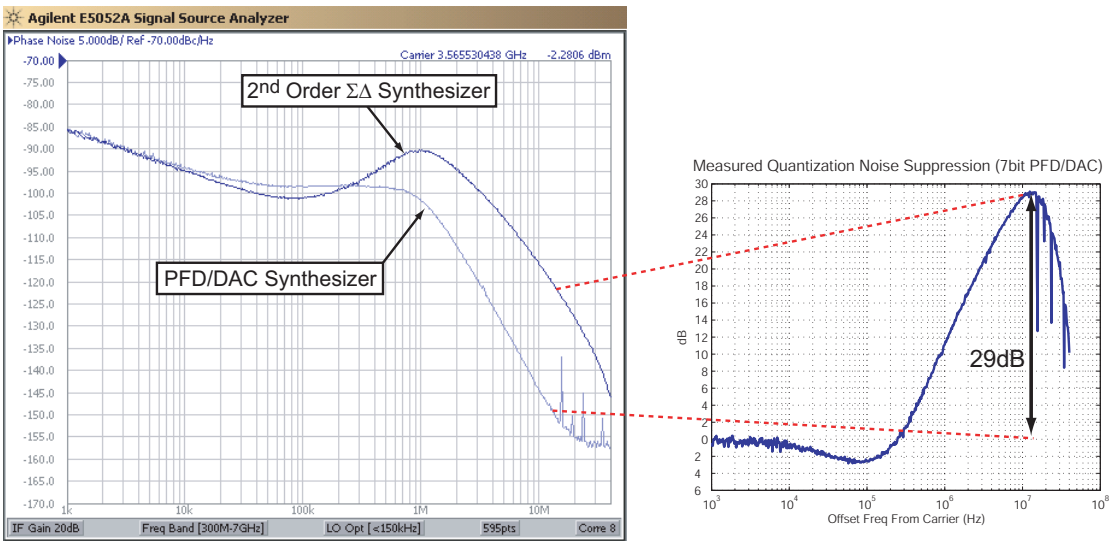


Figure 1-12 Measured Phase Noise Comparison of PFD/DAC Synthesizer Vs. 2^{nd} Order $\Sigma\Delta$ Synthesizer

in that meta-stability at the re-timing flip-flop is determined directly by a high speed arbiter circuit. A simple finite state machine then chooses whether to clock the synchronization flip-flop on the VCO rising or falling edge. A detailed explanation of operation of the re-timing circuit, as well as other high speed, low noise circuit techniques, is presented in Chapter 6.

1.4.4 Measured Results

In chapter 7 we present measured results for a 0.18um CMOS IC that implements the proposed mismatch compensated PFD/DAC synthesizer architecture, and compare it with state-of-the-art $\Sigma\Delta$ fractional-N frequency synthesis. Figure 1-12 shows measured results demonstrating $29dB$ broadband quantization noise suppression using the proposed mismatch compensated PFD/DAC synthesizer. This high level of noise suppression results in both excellent phase noise performance (-154dBc/Hz at 20MHz offset), and very high (1MHz) synthesizer bandwidth.

When configured as a dual band transmitter, the prototype synthesizer achieves data rates of up to 1Mb/s for a GMSK (Gaussian Minimum Shift Keying) data signal centered at either 1.8GHz or 900MHz.

1.4.5 Thesis Outline

In summary, this thesis presents techniques to dramatically reduce quantization noise in fractional-N synthesizers using a proposed mismatch compensated PFD/DAC architecture. We will present the approach and supporting material in the seven remaining chapters. This thesis is divided into multiple chapters. Chapter 2 provides background for modern fractional-N synthesis techniques. In Chapter 3, a new view of the analytical noise model used for fractional-N synthesizer design is proposed. Chapter 4 builds on this model and presents the proposed architecture for reducing quantization noise. In Chapter 5, behavioral modeling techniques for fractional-N synthesis are presented. Emphasis is given to methods for incorporating circuit nonidealities determined from SPICE level simulations into the behavioral model. Chapter 6 focuses on proposed circuit design techniques for high speed and low noise that enable the PFD/DAC approach. Measured results for a prototype PFD/DAC synthesizer IC are presented and discussed in Chapter 7. Limitations of the prototype system are particularly emphasized. Finally, in Chapter 8 we draw conclusions about the effectiveness of the PFD/DAC technique, and propose areas of future work to further extend synthesizer bandwidth. Our focus will be interpreting the measured results presented in Chapter 7, and examining other recent techniques in the literature. Since quantization noise can be reduced by the PFD/DAC technique to the point where intrinsic noise sources dominate, we will see that reduction of intrinsic noise sources now becomes the focus of future work.

Chapter 2

Frequency Synthesis Background

In this chapter, we present some background into the history and evolution of frequency synthesizer design. Particular attention is paid to the performance tradeoffs that result from the choice of synthesizer architecture. To motivate the need for frequency synthesis, we begin with a discussion of possible applications.

2.1 Motivation for Fractional-N Synthesis

As the information age progresses, an increased emphasis is placed on data transmission and reception technologies. Of particular interest is the wireless application space [6, 29–32]. As more personal communication devices (personal digital assistants (PDAs), cell phones, sensor networks, laptop computers, desktop computer local area networks (LANs)) become wireless, increased demands are placed on integrated circuit transceiver designs to achieve higher bandwidths, generate less noise, and be flexible enough to accommodate multiple transmission standards.

2.1.1 Mixer-based Transceivers

One of the biggest challenges in the transceiver space is design of the programmable frequency source required for up-conversion and down conversion of data. Figure 2-1 depicts a typical transceiver architecture used in a cell phone. A transmit/receive

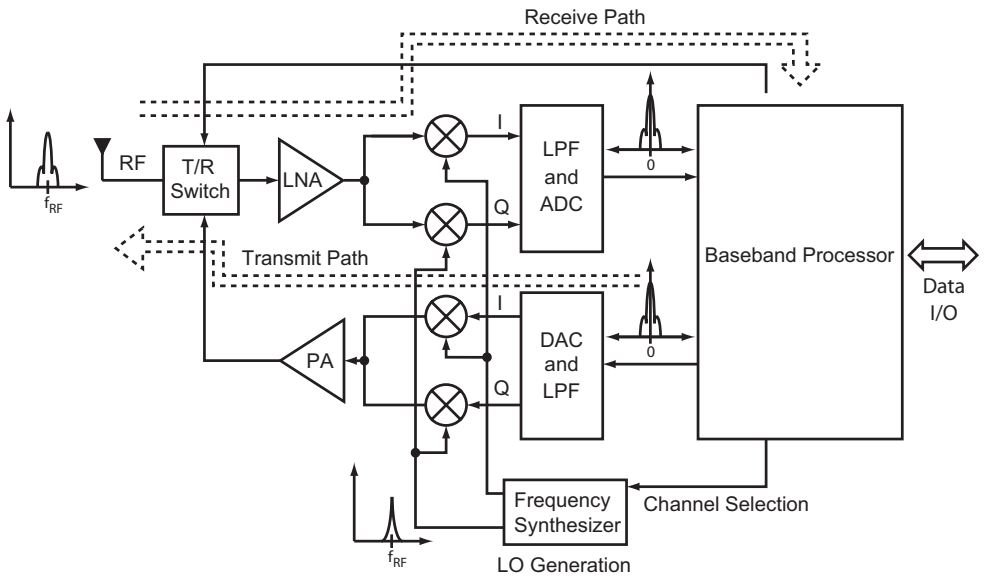


Figure 2-1 Typical RF Transceiver

(T/R) switch selects whether the transmitter is in transmit or receive mode by connecting either the transmit path circuitry or receive path circuitry to the antenna. The unused circuitry is typically shut down to save power.

In receive mode, the antenna output is amplified by a low noise amplifier (LNA), and then mixed with a local oscillator (LO). The mixing process translates the modulated high frequency signal to a low frequency. In the direct conversion transceiver architecture shown in Figure 2-1, the LO frequency is selected to be the same as the carrier frequency of the received signal, f_{RF}, and so the data is mixed down to DC [33]. The received signal is filtered and converted to a digital signal by analog-to-digital converters (ADCs), and then sent to the baseband processor, which further filters and decodes the data to be used by the system. Typically, the data is de-modulated in quadrature, so parallel I and Q channels are converted by the ADC circuitry.

In transmit mode, the process is reversed. The low frequency baseband data is converted from a digital to analog signal by digital-to-analog converters (DACs) and filtered. The analog data stream is mixed with the programmable LO signal to generate a modulated carrier centered at f_{RF}, as shown in Figure 2-2. A power amplifier amplifies the signal to a level appropriate for transmission, and is sent to the antenna via the T/R switch.

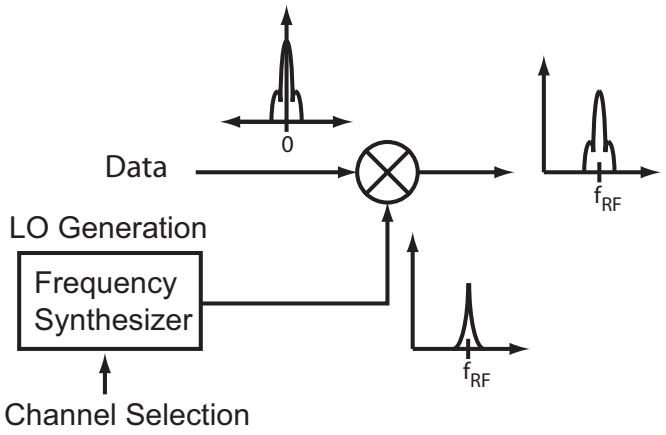


Figure 2-2 Mixer Up-conversion Operation

The key performance requirements on the LO are threefold. First, it must be programmable at a fine enough resolution to accommodate all channels inside the given band of interest. As an example, for a GSM cell-phone, the LO must span a range from 890MHz to 915MHz in 200kHz increments. Second, the LO has to be able to jump from one channel to another fast enough to accommodate a channel hopping specification. This requirement translates into a high LO bandwidth, and is particularly important in modulation schemes where the output signal is required to hop between channels as part of the modulation. Finally, the LO must satisfy a spectral phase noise mask. This requirement is to ensure that the LO does not corrupt the data, and that the transmitted signal does not act as an overpowering interferer on adjacent channels.

2.1.2 Direct Modulation Transmission

A different approach to data transmission is to directly modulate the frequency synthesizer itself, as shown in Figure 2-3. In this case, the digital input to the frequency synthesizer consists of a constant component to set the channel, plus a varying component due to a filtered digital data signal. The output of the synthesizer is connected to the PA through a buffer. Direct modulation transmission therefore offers the possibility of eliminating the mixers and DAC circuitry shown in Figure 2-1.

The main limitation of direct modulated synthesizers is that the data path passes

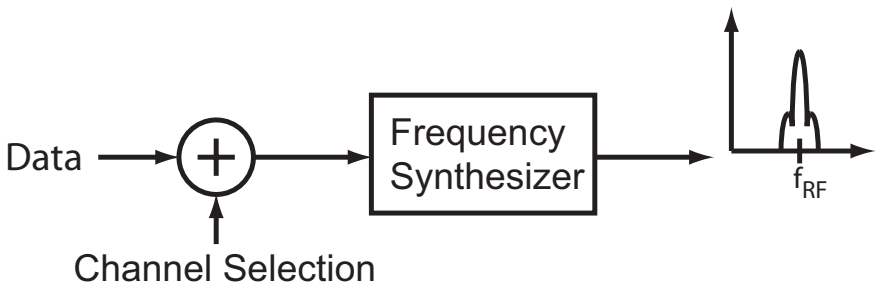


Figure 2-3 Direct Modulation Transmitter

through the frequency synthesizer. The low-pass nature of the synthesizer dynamics will filter the data, requiring either a high synthesizer bandwidth, or some pre-filtering to overcome a low synthesizer bandwidth [7, 34]. As with a mixer based system, we still require good noise performance and fine frequency resolution. Simultaneously achieving fine frequency resolution, high bandwidth, and low noise is an extremely challenging task, since these constraints often run in direct conflict with one another.

2.2 Frequency Synthesis

In this section, we discuss the various frequency synthesizer architectures available for use in a practical system. We also present the drawbacks of each approach, and propose a new technique which de-couples the primary performance tradeoff in modern frequency synthesizer design - namely, a noise-bandwidth tradeoff.

2.2.1 Integer-N Synthesis

Figure 2-4 depicts the most basic synthesizer type, the integer-N frequency synthesizer [3]. To achieve lock, the PLL compares the divider phase (Div) to the reference phase (Ref) via the phase/frequency detector (PFD), produces an error signal, E(t), which is scaled by a charge-pump and filtered by the loop filter, and servos the VCO output with the filtered control signal. The output frequency is determined as:

$$F_{out} = N * F_{ref} \tag{2.1}$$

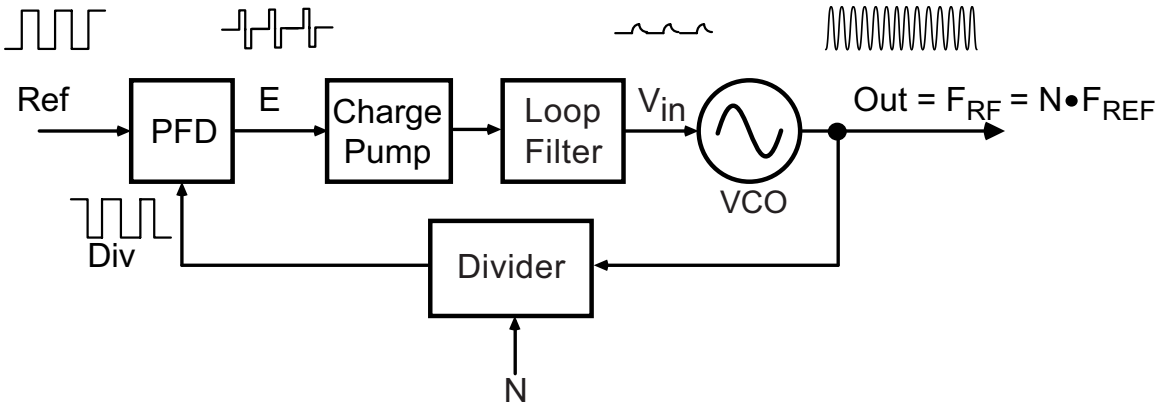


Figure 2-4 Integer-N Synthesizer

where N is the divider value and F_{ref} is the reference frequency. For reasons of stability, the closed loop bandwidth of the PLL is chosen to be at least ten times lower than the reference frequency [35]. Since the output resolution of the synthesizer is an integer multiple of F_{ref}, fine frequency resolution requires a low reference frequency and therefore low PLL bandwidth, and correspondingly slow dynamic response. This resolution-bandwidth tradeoff limits the use of integer-N synthesizers in high performance RF systems.

2.2.2 Fractional-N Synthesis

In order to break the resolution-bandwidth tradeoff that exists in integer-N synthesizers, fractional-N synthesis has been introduced [3]. Depicted in Figure 2-5, the fractional-N synthesizer uses a dithering modulator to dynamically vary the divide value. In classical fractional-N synthesis, the dithering modulator is a simple digital accumulator. The accumulator input represents the fractional portion of the divide value. Each reference period, the accumulator increments by a fraction of the full-scale accumulator range. This can be seen in Figure 2-5, where the accumulator residue (the value of its LSBs), increases each period. When the accumulator wraps, the carry out is used to increase the divide value from N to N+1. An *average* divide

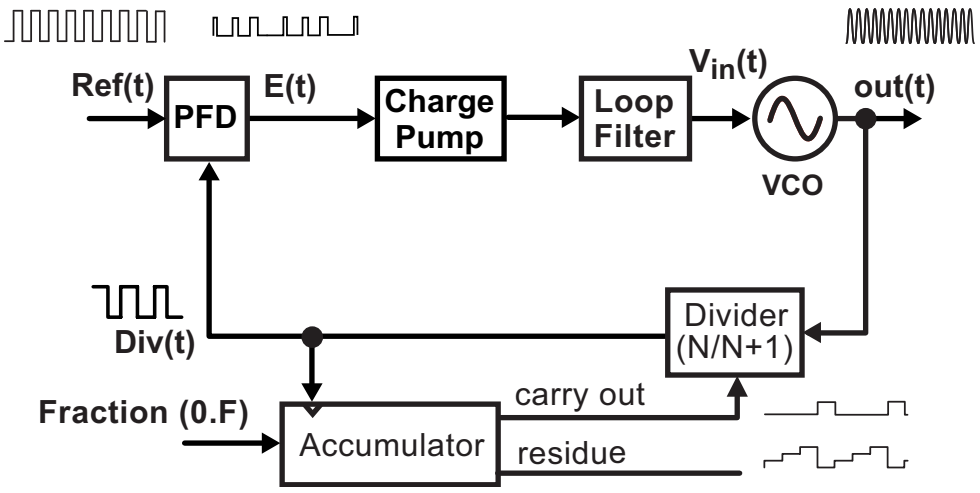


Figure 2-5 Classical Fractional-N Synthesizer

value that is fractional is thereby obtained, with the result that

$$F_{out} = N.F * F_{ref} \tag{2.2}$$

where N is an integer as before, and F is a fraction. The reference frequency therefore can be chosen to be much higher than the desired frequency resolution, and PLL bandwidth can be increased accordingly. Breaking the bandwidth-resolution tradeoff comes at a cost, however, namely the introduction of quantization noise into the system.

To see what is meant by quantization noise, we examine Figure 2-6, keeping in mind the accumulator operation. In the example of Figure 2-6, the integer value, N, is four and the fractional value, F, is one-fourth. This means that the divide sequence is a repeating {4,4,4,5}, giving an average divide value of 4.25. We see that while the *average* divide value is as desired, the *instantaneous* phase error output from the PFD (waveform E(t) in Figure 2-6), increases during the accumulation. Instantaneous values of the continuous waveform E(t) are represented by the discrete time sequence $\epsilon[k]$. When the accumulator wraps, the phase error is reset to zero.

As the figure illustrates, the quantization of the divider output to integer multiples of the VCO period is the source of the phase error signal. This quantization noise source severely limits performance for several reasons. First, the noise is highly

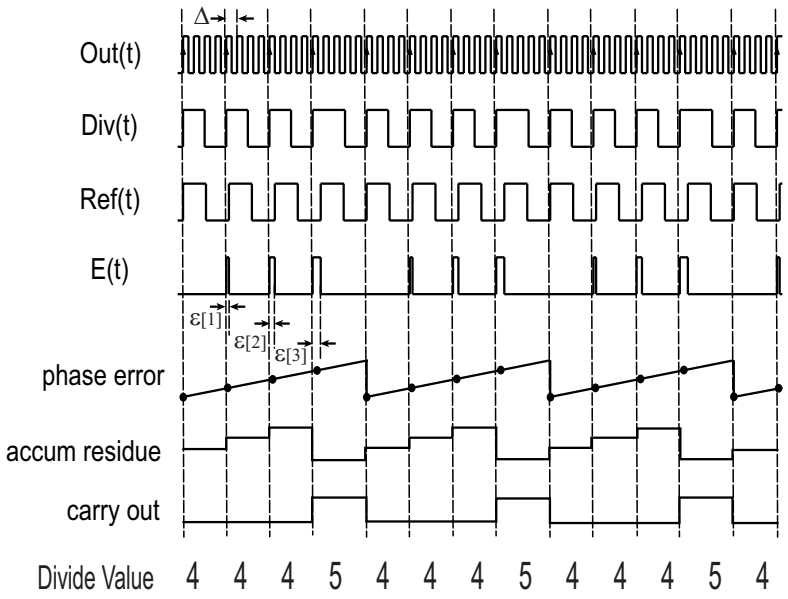


Figure 2-6 Quantization Noise in Fractional-N Synthesis for N.F = 4.25

periodic. The loop filter cannot completely attenuate the amplitude of the noise, and therefore the periodic noise input to the VCO results in a periodic modulation of the VCO output. The resulting frequency modulation appears in the phase noise spectrum of the VCO as *fractional spurs*. Second, the periodicity of the error signal occurs at a frequency lower than the reference frequency, and at a fundamental frequency equal to:

$$F_{spur_{fund}} = 0.F * F_{ref} \tag{2.3}$$

and therefore can appear inside the synthesizer bandwidth, where it will not be filtered by the PLL dynamics. Furthermore, the pulsed nature of the error signal means that spurs also occur at harmonics of the fundamental spur frequency. Spurious tones are highly undesirable in any RF application, especially mixer-based systems where undesirable mixing products can be generated, so minimizing the impact of spurious tones is very important.

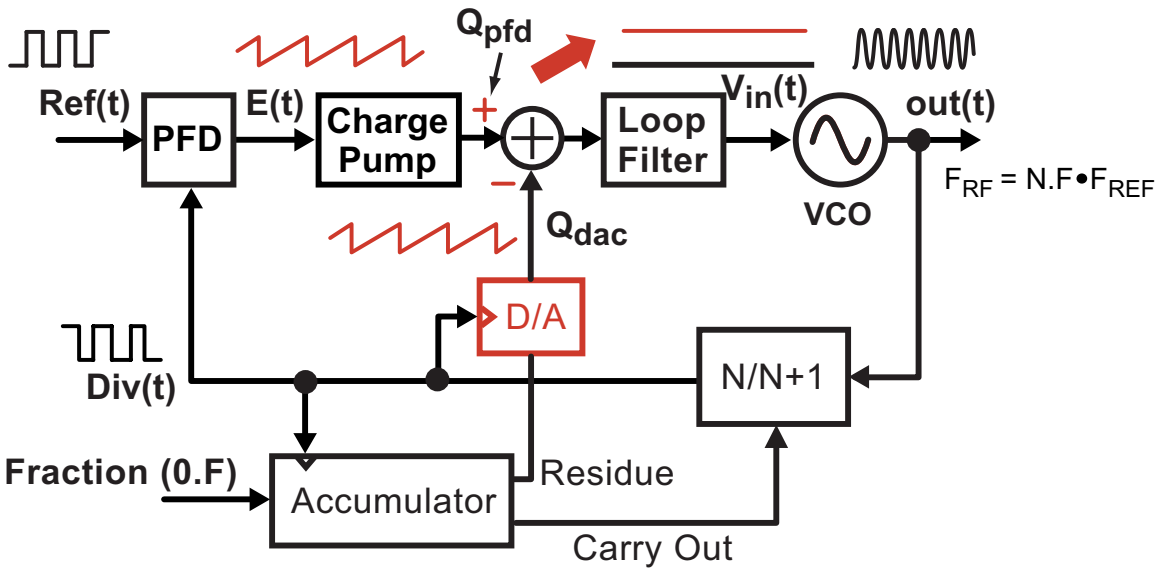


Figure 2-7 Classical Fractional-N Synthesizer With Phase Interpolation

2.2.3 Phase Interpolation Based Fractional-N Synthesis

Classical fractional-N synthesis uses a technique called phase interpolation to cancel the quantization noise, as depicted in Figure 2-7 [26]. As the accumulator counts, the value stored in the LSBs represent the magnitude of the quantization error. This is apparent in Figure 2-6, where we see that as the accumulator count increases, so does the area of the error pulse. The waveform labeled "phase error" in Figure 2-6, represents the area enclosed by the error signal E, and clearly shows that we can relate this ramp waveform to the increasing residue stored in the LSBs of the accumulator, since they both have the same wave-shape. By using the LSBs to control a cancellation DAC, the quantization error can, ideally, be canceled.

The limitation of phase interpolation based fractional-N synthesis is that the cancellation is not ideal. At a basic level, we see that the summation of the DAC cancellation output and scaled error signal in Figure 2-7 is done at the charge-pump output. The fact that the cancellation is the result of a feed-forward process is enough to suggest that any mismatch in the gains between the two paths will result in incomplete cancellation. Since the phase error signal and cancellation signal are generated by separate circuits, the possibility for mismatch is very high. In fact, the resulting

spur cancellation levels in classical phase-interpolation based synthesis is poor, and is the main reason this technique is not used in high performance systems.

The phase error signal is a charge packet weighted by the phase error in time, and the charge-pump output in magnitude. The phase error signal varies between zero and one VCO period, T_{vco}, and the charge-pump output is I_{cp}. The error charge packet produced at the charge-pump output, Q_{PFD}, is therefore:

$$Q_{pfd} = I_{cp} T_{vco} \epsilon[k]; \qquad 0 \leq \epsilon[k] \leq 1 \tag{2.4}$$

where ϵ is a weighting function representing the instantaneous quantization error, and corresponds to a fraction of a VCO period in time. It should be noted that $\epsilon[k]$ varies over time, as shown in Figure: 2-6. The output charge from the cancellation DAC is:

$$Q_{dac} = I_{dac} T_{dac} \epsilon_{dac}[k]. \tag{2.5}$$

where I_{dac} is the DAC full-scale current, T_{dac} is the DAC on-time, and $\epsilon_{dac}[k]$ is a fraction of the DAC full-scale output. In order to cancel the quantization noise, we require:

$$I_{dac} T_{dac} \epsilon_{dac}[k] = I_{cp} T_{vco} \epsilon[k] \tag{2.6}$$

Because the VCO output frequency is usually very high (on the order of GHz), T_{vco} is very small, while T_{dac} is typically many VCO cycles to allow the DAC output to properly settle. This places a burden on the DAC resolution, which must compensate for long T_{dac} by having a very finely resolved I_{dac} to achieve the same degree of resolution in charge. Even if the quantization noise introduced by the fractional-N dithering process is completely canceled by the DAC, the DAC itself has noise related to its own finite resolution. DAC quantization noise must be made small enough such that it does not become the dominant noise source over any frequency range, increasing demand on the already stringent DAC resolution requirements.

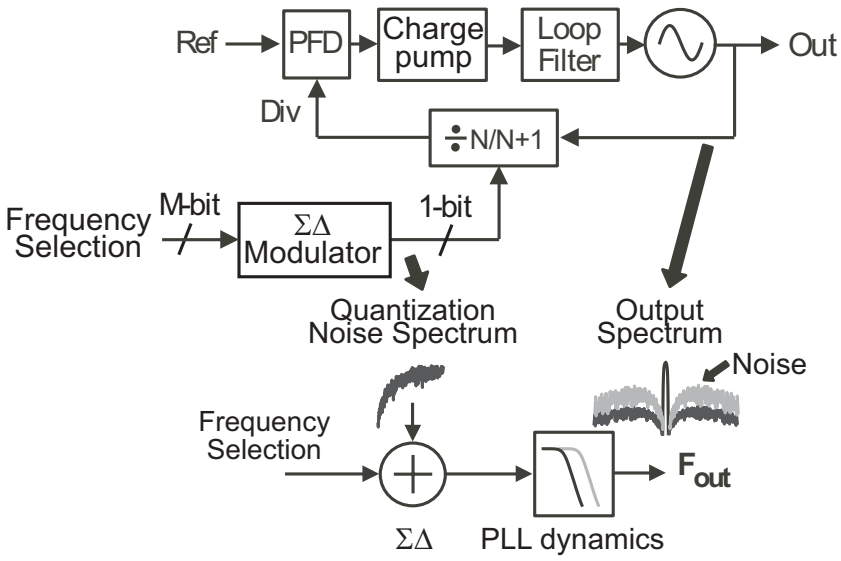


Figure 2-8 ΣΔ Fractional-N Synthesizer and Noise/Bandwidth Tradeoff

2.2.4 ΣΔ Fractional-N Synthesis

More recently, the simple digital accumulator divider control of classical fractional-N synthesis has been replaced by a high order digital ΣΔ modulator [4]. In fact, as [4] shows, an accumulator is a first order ΣΔ modulator. First order modulators contain large spurious components in their outputs [36], consistent with our discussion of the periodic component in a fractional-N synthesizer. Unlike first order modulators, high order ΣΔ modulators exhibit a spectrum that is not primarily composed of spurs, but rather appears as though a white noise spectrum has been shaped by a high-pass characteristic. The cancellation DAC used in phase-interpolation synthesis is not used in ΣΔ synthesis, so attenuation of the quantization noise magnitude appearing at the synthesizer output is achieved solely by the filtering action of the PLL dynamics on the noise. Figure 2-8 depicts a ΣΔ frequency synthesizer, and the noise-bandwidth tradeoff that accompanies use of this technique.

Figure 2-8 shows that the shaped noise becomes dominant at intermediate offset frequencies around the carrier. As PLL bandwidth is increased, this noise becomes even more pronounced. The resulting noise-bandwidth tradeoff is the bottleneck to achieving high bandwidths in state-of-the art ΣΔ synthesis, and the subject of much

research activity [15–17, 19, 20, 34].

2.3 Summary

In this chapter we have reviewed the evolution of fractional-N synthesis from integer-N synthesis, and motivated the need for techniques to reduce the impact of fractional-N quantization noise. In particular, we have shown that state-of-the art $\Sigma\Delta$ fractional-N synthesis suffers from a noise-bandwidth tradeoff that ultimately limits $\Sigma\Delta$ synthesizers to low bandwidths. In Chapter 4 we propose a synthesizer architecture capable of dramatically reducing the impact of quantization noise on synthesizer output performance by actively canceling it. The proposed architecture thereby breaks the noise-bandwidth tradeoff present in $\Sigma\Delta$ synthesizers, and offers the possibility of not only excellent resolution and noise performance, but also high closed loop synthesizer bandwidth and improved transient response.

Chapter 3

Fractional-N Synthesizer Noise Modeling

In this chapter we derive a new block diagram model view of fractional-N synthesis based on the noise model presented in [1]. Using this model we will see that we can represent ΣΔ synthesis as a subset of phase interpolation synthesis. Also, by drawing an analogy between the new model and a MASH ΣΔ DAC, we find that we can leverage ΣΔ DAC dynamic element matching techniques to improve the performance of phase interpolation based synthesis, resulting in the proposed technique that will be explained in Chapter 4.

3.1 Basics of Noise Modeling of Fractional-N Synthesizers

In [1], a noise model is developed for the analysis of ΣΔ fractional-N synthesizers. The model, shown in Figure 3-1 is useful to determine the impact of various system noise sources on overall synthesizer behavior. In the model, T is the reference period, α the charge-pump gain, I_{cp} the full-scale charge-pump current, H(f) the loop filter dynamic characteristic, K_v the VCO gain, and N_{nom} the nominal (average) divide value. In [1] it is also shown that the combination of charge-pump noise and reference jitter is

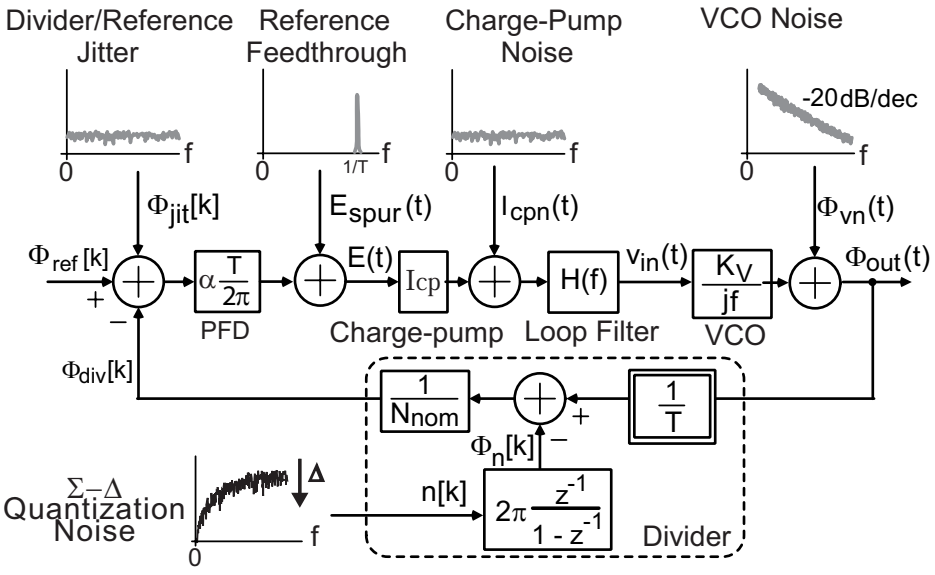


Figure 3-1 Noise Model for a ΣΔ Synthesizer from [1]

low-pass filtered by the PLL dynamics and is dominant at low frequencies, the VCO noise is high-pass filtered and dominant at high frequencies, and the quantization noise is low-pass filtered and dominant at intermediate offset frequencies.

A synthesizer optimally designed for noise performance will therefore have its bandwidth chosen such that quantization noise is less than PFD and VCO noise [37]. This concept is demonstrated in the calculated phase noise plots of Figure 3-2, where reference referred noise and charge-pump referred noise have been lumped together into one noise source, detector noise. The plots were generated using the PLL Design Assistant CAD tool [38, 39], which implements the analytical model that we use as a basis for calculations.

The target performance for this example synthesizer is to obtain -150dBc/Hz noise at 20MHz offset frequency. The left plot is for a ΣΔ synthesizer with a 1MHz bandwidth, and second order ΣΔ modulator. The quantization noise dominates over a very broad frequency range, to the point where the VCO noise is not even a factor. In the right plot, the synthesizer bandwidth is changed to 100kHz, and we see that ΣΔ quantization noise is reduced. There is an optimal tradeoff of noise performance between each element of the system and overall bandwidth. Namely, the -150dBc/Hz noise at 20MHz specification is achieved by having the VCO noise be the dominant

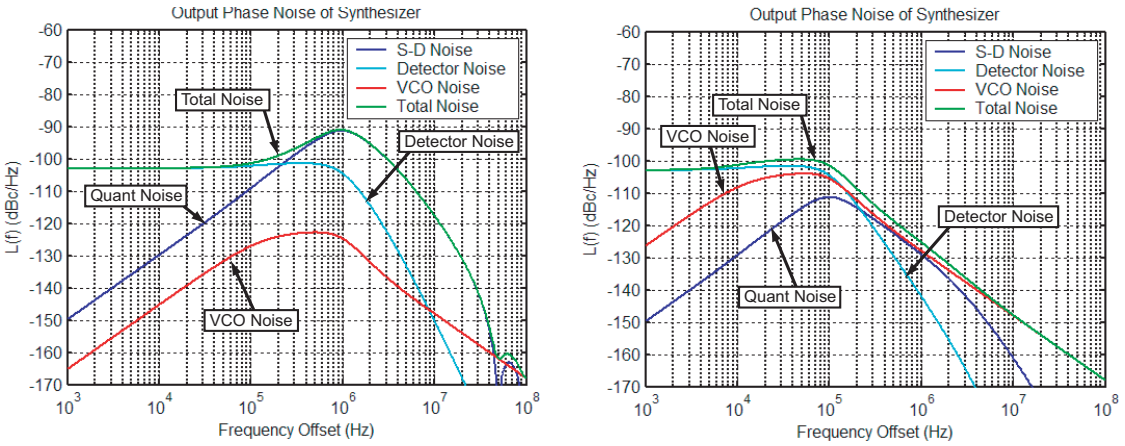


Figure 3-2 PLL Filtering of System Noise Sources

source at that frequency. Figure 3-2 clearly demonstrates the noise-bandwidth trade-off encountered with $\Sigma\Delta$ fractional-N synthesizers discussed in Chapter 1.

In this thesis, we focus on techniques to extend the bandwidth of fractional-N frequency synthesizers. As Figure 3-2 shows, a primary area of concern is the impact of quantization noise produced from the fractional-N dithering process. If the quantization noise magnitude in the left plot of Figure 3-2 could be reduced by 36dB, then it would not be dominant over any frequency range, and a 1MHz bandwidth could be achieved while meeting the 20MHz phase noise specification. In the case where the quantization noise does not dominate the synthesizer phase noise over any frequency range, performance is determined by detector noise and VCO noise. For this reason, we segregate synthesizer noise sources into two categories: *intrinsic noise* and *quantization noise*. Intrinsic noise sources are those that will be present in any PLL, namely the detector noise and VCO noise. Quantization noise is particular to fractional-N synthesizers, and is the noise introduced by the fractional-N dithering process. The primary area of focus of this thesis is to develop techniques to reduce the magnitude of quantization noise present in the system. In addition, we will explore circuit techniques to reduce the impact of intrinsic noise sources.

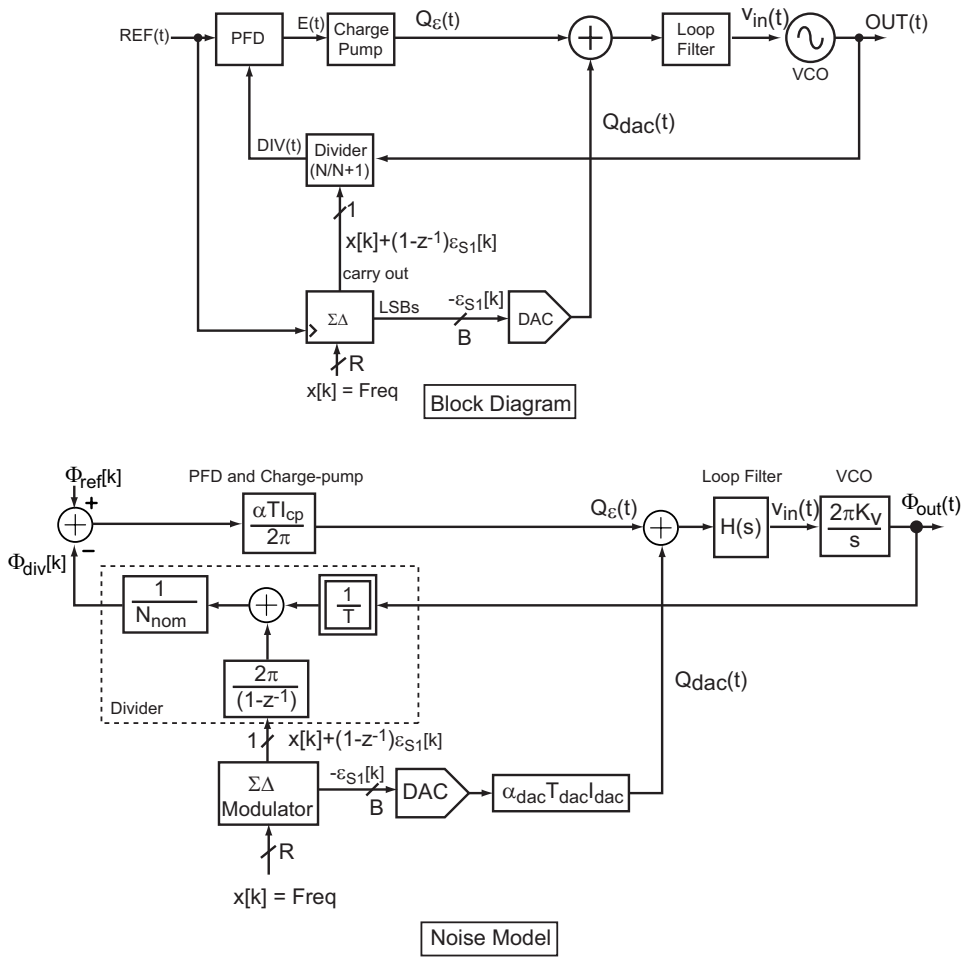


Figure 3-3 PI Synthesizer System Block Diagram and Noise Model

3.2 A New Analytical Model View of Fractional-N Synthesizers

The noise model presented in section 3.1 is in the format traditionally used to explain fractional-N synthesizers. Namely, the signal flow in the block diagram is from left to right, with the reference being viewed as the input phase source. In this section, we reformat the noise model to a view where the input port is the divider control, and the reference phase is just that, a reference. In this context, a fractional-N synthesizer can be understood as a DAC, where the analog output value being controlled is phase. In keeping with our emphasis on reducing fractional-N quantization noise, we will ignore other noise sources when reformulating the model. Equations for calculating

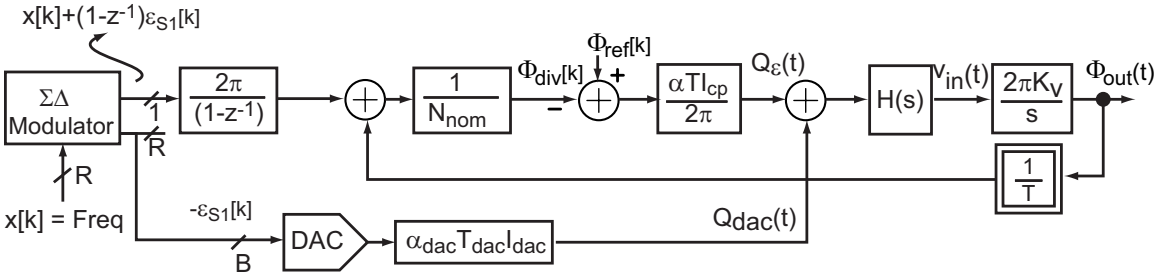


Figure 3-4 Base PI Noise Model

the impact of VCO and detector noise on overall synthesizer noise performance can be found in [1].

3.2.1 Phase Interpolation Fractional-N Synthesis

We start the derivation of the new model with phase interpolation (PI) based fractional-N synthesis. The system block diagram of a PI synthesizer and its noise model, derived from the noise analysis block diagram of Figure 3-1, is shown in Figure 3-3. The input to the system is a frequency value, x[k], where k is a discrete time index. As shown in [4], the accumulator used in PI synthesis is equivalent to a first order $\Sigma\Delta$ modulator. For this reason, a $\Sigma\Delta$ modulator has been substituted for the accumulator in Figure 3-3. Also, to keep all operations in the s-domain, the VCO block has had numerator and denominator multiplied by 2π.

Since the accumulator is determined to be a first order $\Sigma\Delta$ modulator, we can represent its output as [36]

$$x[k] + (1 - z^{-1})\epsilon_{S1}[k] \tag{3.1}$$

where ϵ_{S1} represents the quantization noise, and $(1 - z^{-1})\epsilon_{S1}[k]$ is a periodic, pulsed waveform as depicted in Figure 2-6. As discussed in section 2.2.2, the accumulator LSBs contain information about the magnitude of the quantization noise, so we represent this in the diagram by the term $-\epsilon_{S1}[k]$. PI fractional-N synthesis uses this information to control a DAC which then, ideally, cancels the error.

Before beginning the derivation of the new model, we first manipulate the noise model of Figure 3-3 so that the divider input is on the left, as shown in Figure 3-4.

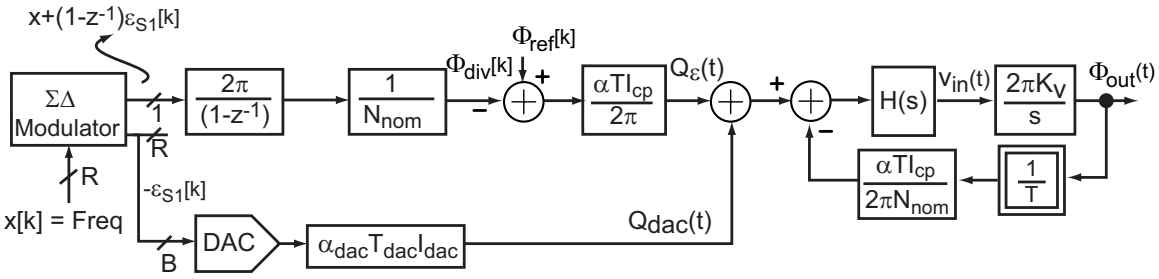


Figure 3-5 PI Noise Model: Step 1

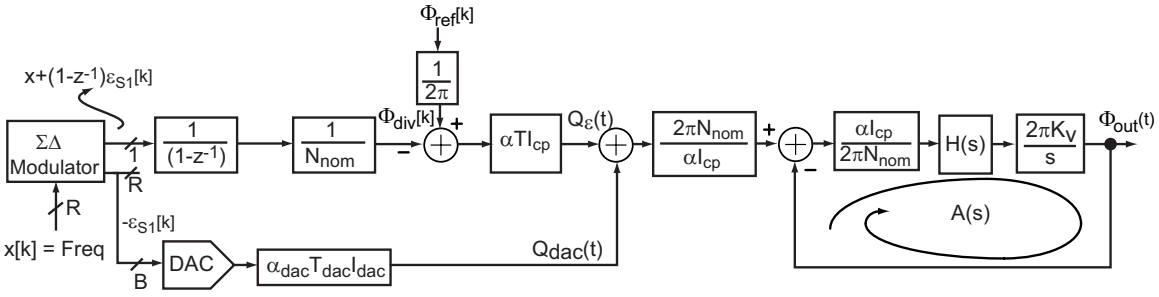


Figure 3-6 PI Noise Model: Step 2

This is consistent with our stated desire to emphasize that the divider control is the synthesizer input.

The first step in deriving the new model is simple block diagram manipulation, depicted in Figure 3-5. The summing junction for the feedback path has been moved forward in order to separate the PLL closed loop dynamics from the quantization noise and DAC cancellation paths. The next step is shown in Figure 3-6. The dynamics have been formatted into a form where they can be expressed as

$$G(s) = \frac{A(s)}{1 + A(s)} \tag{3.2}$$

where A(s) is the loop shown in the picture, and G(s) represents the closed loop PLL transfer function, consistent with the nomenclature in [1].

The final form of the new model is shown in Figure 3-7. The divider, PFD, and charge-pump terms perform the DAC function, expressed by the dashed box. The feed-forward DAC sums at the charge-pump output. Ideally, the DAC charge will

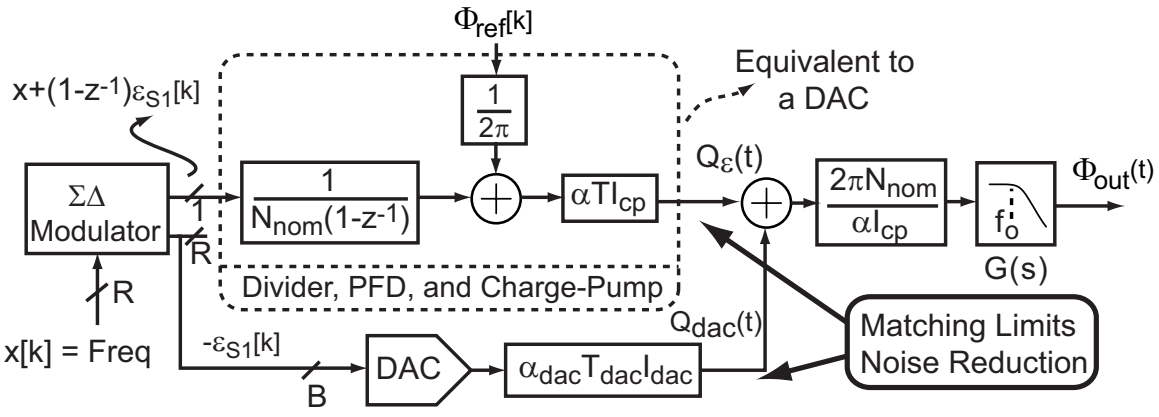


Figure 3-7 PI Noise Model: Step 3

cancel the error charge perfectly:

$$Q_e(t) + Q_{dac}(t) = 0. \tag{3.3}$$

In order for equation 3.3 to hold true, we require

$$\frac{\alpha T I_{cp}}{N_{nom}} \epsilon_{S1}[k] + \alpha_{dac} T_{dac} I_{dac} \epsilon_{S1}[k] = 0 \qquad ; 0 \leq \epsilon_{S1}[k] \leq 1 \tag{3.4}$$

This may be simplified to

$$\alpha T_{vco} I_{cp} \epsilon_{S1}[k] + \alpha_{dac} T_{dac} I_{dac} \epsilon_{S1}[k] = 0 \qquad ; 0 \leq \epsilon_{S1}[k] \leq 1 \tag{3.5}$$

Equation 3.5 is the key relationship of importance in PI fractional-N synthesis. The quantization step in the synthesizer is a VCO period, T_{vco}. This is intuitive because the divider counts VCO periods to produce its output, and it is constrained to counting integer numbers of periods. The divider is therefore the quantizer in the system. The term $\epsilon_{S1}[k]T_{vco}$ represents the instantaneous value of the phase error, and is constrained to be between 0 and 1 VCO period.

If the error signal is to be used by the system, it must be converted from a purely time based signal into an electrical signal. Transduction is performed by the charge-pump, which weights the phase error signal $\epsilon_{S1}[k]T_{vco}$ by multiplying it by I_{cp}. The

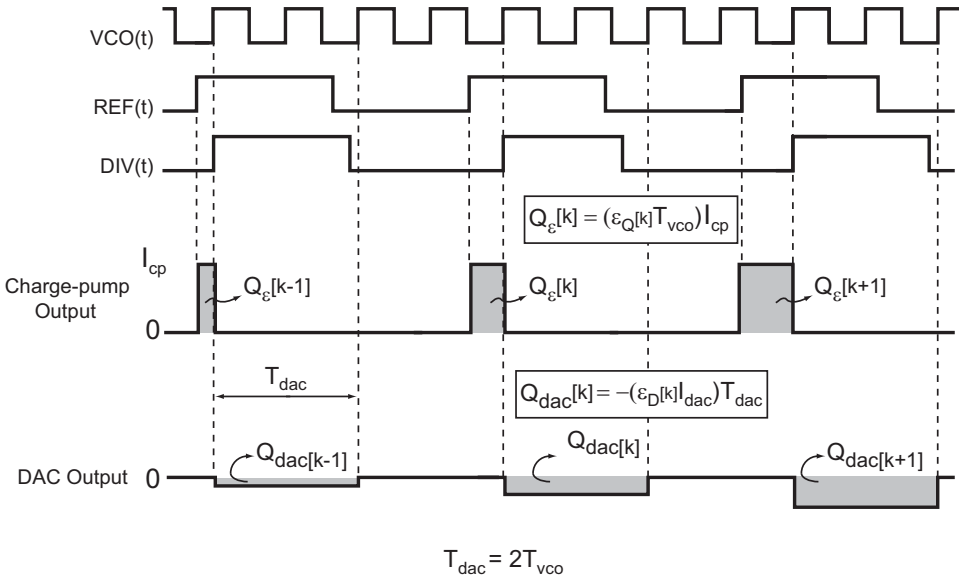


Figure 3-8 Charge Cancellation in PI Synthesis

resultant error charge is then converted by the loop filter into a voltage which can be used to control the VCO.

The DAC is employed to precisely cancel the error charge. In practice, it is very difficult to achieve perfect cancellation for several reasons. First, the cancellation charge should be referenced to a single VCO period and full-scale charge-pump current for maximum effect. This is not the case in classical PI synthesis. Because T_{vco} is very small (typically 100's of ps to 10ns), T_{dac} is usually set to be many VCO periods so that the DAC output can settle [19, 20]. This tradeoff requires the DAC to have even more resolution to counteract a large T_{dac}.

The increased demand on DAC resolution for large T_{dac} is depicted in Figure 3-8. While the phase error charge is resolved in *time*, the cancellation charge is resolved in *magnitude*. In the example presented in the figure, the DAC on-time has been set to two VCO periods so that the DAC has time to settle. The instantaneous values of the quantization error signal $\epsilon_{S1}[k]$ are represented by $\epsilon_Q[k]$, and the instantaneous cancellation charge packets produced by the DAC are weighted by $\epsilon_D[k]$, where k is the time index. In order for the cancellation charge to offset the error charge, we require equation 3.5 to be satisfied. For the charge packets in Figure 3-8, this

translates to:

$$\alpha T_{vco} I_{cp} \epsilon_Q[k] + \alpha_{dac} T_{dac} I_{dac} \epsilon_D[k] = 0 \tag{3.6}$$

If we assume that the PFD gain, α, and DAC gain, α_{dac}, are equal, equation 3.6 can be simplified to:

$$T_{vco} I_{vco} \epsilon_Q[k] = T_{dac} I_{dac} \epsilon_D[k]. \tag{3.7}$$

We then solve equation 3.7 for $\epsilon_D[k] I_{dac}$:

$$\epsilon_D[k] I_{dac} = \frac{T_{vco}}{T_{dac}} \epsilon_Q[k] I_{cp}. \tag{3.8}$$

If the DAC is on for 2^B VCO periods, ($2^B T_{vco} = T_{dac}$), equation 3.8 simplifies to:

$$\epsilon_D[k] I_{dac} = \frac{1}{2^B} I_{cp} \epsilon_Q[k]. \tag{3.9}$$

The final assumption we make is that the full-scale DAC current, I_{dac} equals the charge-pump current, I_{cp}. We then arrive at the final result:

$$\epsilon_D[k] = \frac{\epsilon_Q[k]}{2^B}. \tag{3.10}$$

Equation 3.10 is an intuitive result and states that as the DAC is kept on longer so that its output can settle, more levels are required to properly cancel the quantization noise. This is clearly shown in Figure 3-8, where, as T_{dac} increases, $\epsilon_D[k] I_{dac}$ must get correspondingly smaller to keep the error charge and cancellation charge equal.

A second problem with PI synthesis is that any difference between the feed-forward DAC path and error signal results in a gain error, and incomplete cancellation. Generating well matched, low magnitude current levels is very difficult to achieve in the design of high speed, high resolution DACs. Any differential non-linearity in DAC output appears as a gain mismatch in the DAC transfer function. This is equivalent to saying that $\epsilon_Q[k] \neq \epsilon_D[k]$ in equation 3.9. Gain mismatch is the limiting factor in classical PI synthesis. Recent approaches have used separate cancellation DAC paths, and large T_{dac}, and are therefore limited in their ability to cancel the quantization

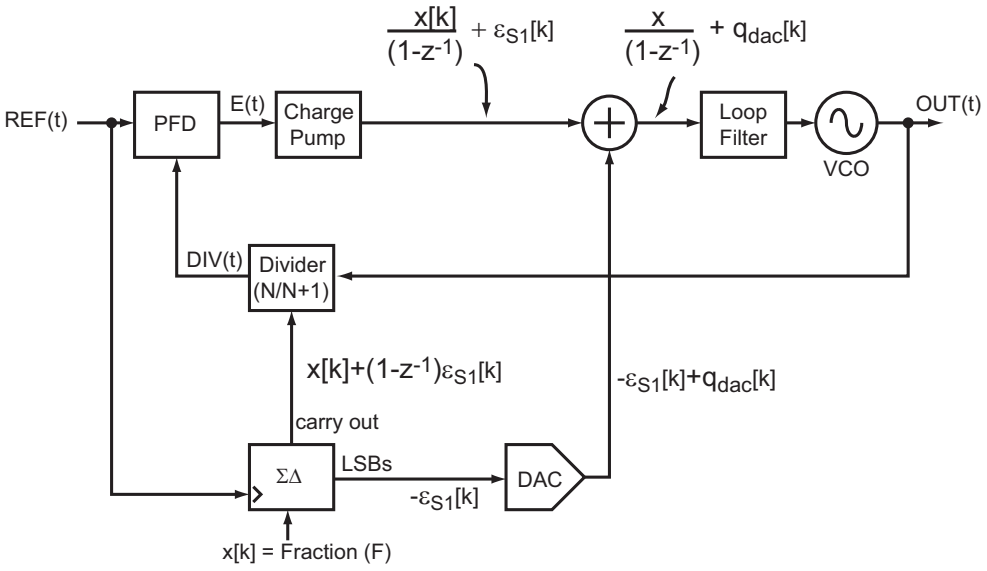


Figure 3-9 PI Synthesizer Block Diagram Including DAC Quantization Noise

noise [19, 20].

A third non-ideality present in classical PI synthesis is finite DAC resolution. In practice, the number of bits in the residue of the accumulator that make up the quantization error, $\epsilon_{S1}[k]$, will be large, perhaps 16 to 20 bits. A practical implementation of the cancellation DAC will not be able to achieve the same resolution, so there will be some quantization error due to the DAC resolution itself. Figure 3-9 illustrates the effect of finite DAC resolution. In the figure, we have taken some liberty with notation. The charge-pump and DAC outputs are continuous functions of time, but contain components due to discrete time processes $\epsilon_{S1}[k]$ and $q_{dac}[k]$. Because we emphasize noise behavior in Figure 3-9, we represent the instantaneous values of the continuous signals in the figure.

Even if the quantization error due to fractional-N dithering, $\epsilon_{s1}[k]$, is completely canceled, the DAC introduces its own quantization noise, $q_{dac}[k]$, into the system. In classical PI synthesis, the DAC is not noise shaped, and so DAC quantization noise can become dominant in-band if the DAC does not have enough resolution. This issue exacerbates the DAC resolution issues already discussed.

Finally, as Figure 3-8 shows, while the cancellation charge may offset the error charge in magnitude, the *shape* of the waveforms are different. This suggests that PI

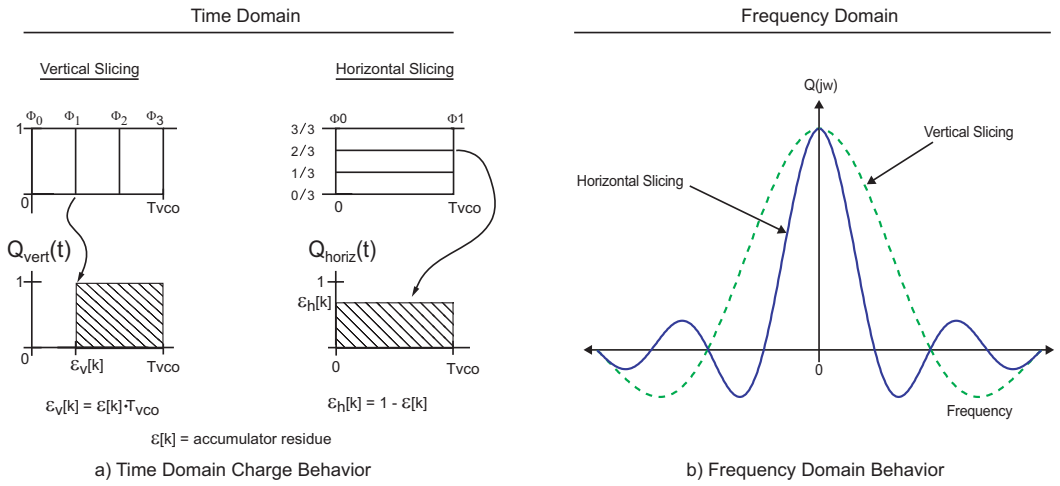


Figure 3-10 Vertical Vs. Horizontal Resolution

synthesis will achieve a very good DC match between the phase error and cancellation signals, but some frequency dependent gain error will result. Ideally, the cancellation charge would be a vertically resolved waveform like the error charge, but this is impractical for several reasons. First, generating a finely resolved VCO period is difficult for high speed VCOs. Practical limitations result in at most eight to twelve VCO phases being generated [15, 17]. This is equivalent to a three or four bit DAC being used in a PI synthesizer. Second, any mismatch in output phase for a multi-phase VCO translates into a gain error between the cancellation signal and phase error signal. It is very difficult to match the phases well, and so the ultimate ability of a vertically resolved cancellation scheme to cancel quantization noise is limited [15, 17].

While vertical slicing of the cancellation charge is limited in resolution for practical reasons, it is the preferred approach because it could theoretically cancel the error charge in both magnitude and shape. Figure 3-10 illustrates the systematic frequency dependent gain error that results from the shape mismatch between the horizontally and vertically sliced waveforms. The vertically resolved waveform represents the quantization error charge, and the horizontally resolved waveform the DAC cancellation charge in PI synthesis. Time domain behavior, as depicted in the left plot, is represented by the number of steps being resolved for the two techniques, and the corresponding charge transferred during the resolved VCO period. To simplify

analysis, the charge-pump magnitude is normalized to one. A look at the Fourier transforms for each approach shows that at DC the difference between the spectra is zero, as expected since the shaded regions in the time domain plot have the same area. As frequency increases, the difference between the spectra increases and will be manifested in the phase noise power spectrum by imperfect fractional spur cancellation. This behavior places a limit on the ability of the horizontally resolved system to exactly cancel the phase error waveform in the absence of a correction scheme.

Because the phase error changes with time, $\epsilon_v[k]$ and $\epsilon_h[k]$ are time indexed. The area enclosed by the shaded regions therefore vary with time, as do the zero crossing of the sinc waveforms that correspond to frequency domain behavior. The sinc waveforms in Figure 3-10 represent the spectra for the particular value of ϵ creating the time domain waveforms in the left plot.

The Fourier transform for the charge enclosed inside the dashed box for the vertical slicing case is a function of both frequency, f, and discrete time index, k:

$$Q_{vert}(j\omega, k) = \frac{1}{j\omega}(e^{-j\omega\epsilon_v[k]} - e^{-j\omega T_{vco}}) \tag{3.11}$$

and for the horizontal case is:

$$Q_{horiz}(j\omega, k) = \frac{\epsilon_h[k]}{j\omega}(1 - e^{-j\omega T_{vco}}) \tag{3.12}$$

By expanding these expressions using Taylor series, keeping up to the second order terms, and subtracting to obtain the error, we arrive at:

$$Q_{err}(j\omega, k) \approx -\frac{j\omega}{2}\epsilon[k](1 - \epsilon[k]) \tag{3.13}$$

where $\epsilon[k]$ corresponds to the instantaneous accumulator residue. There is therefore a systematic mismatch between the error charge and cancellation charge which fundamentally limits performance [25].

Figure 3-11 qualitatively shows the error plot for equation 3.13 as a function of both frequency (normalized to the VCO frequency) and discrete index k. The figure

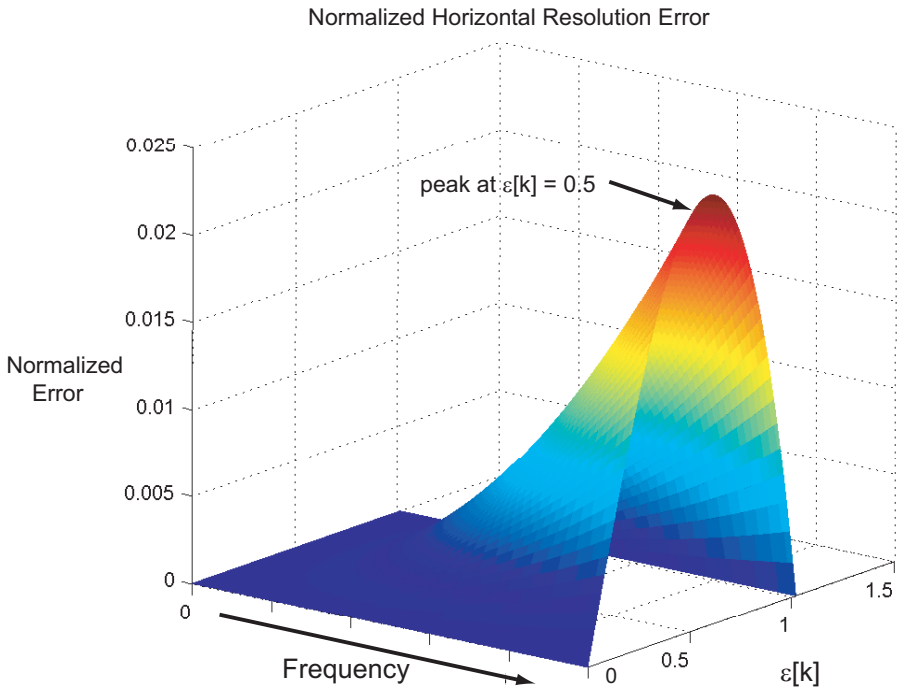


Figure 3-11 Shape Mismatch Error as a Function of Frequency and Discrete Index k

shows that there is an increasing error with frequency for a given value of $\epsilon[k]$. The PLL dynamics will act to filter any error feed-through at frequencies above the loop bandwidth, reducing the error impact at high frequencies. We also see that, at a given offset frequency, the error has an inverted parabolic shape as a function of $\epsilon[k]$. This is intuitive because at the two extremes, $\epsilon[k] = 1$ or $\epsilon[k] = 0$, there is no difference between vertical slicing and horizontal slicing, and the horizontal slicing error goes to zero. Similarly, the largest error occurs when $\epsilon[k] = 0.5$, corresponding to the biggest mismatch in shape between the vertically resolved signal and horizontally resolved signal.

In section 4.2.5 we will use equation 3.13 to propose a digital compensation scheme for shape mismatch between the cancellation DAC charge and quantization error charge.

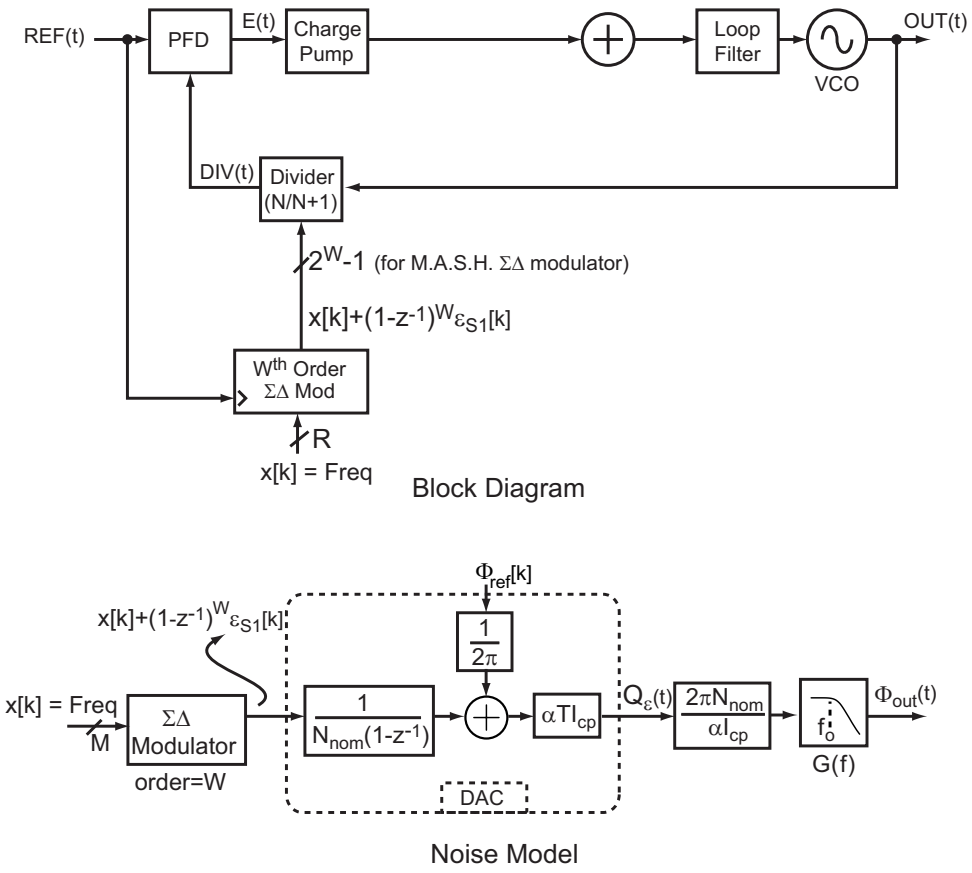


Figure 3-12 ΣΔ Synthesizer Block Diagram and Noise Model

3.2.2 ΣΔ Fractional-N Synthesis

We now apply the new model view for fractional-N synthesis to the ΣΔ synthesizer, resulting in the noise model presented in Figure 3-12. There are two key differences between a ΣΔ synthesizer and PI synthesizer that the model emphasizes. First, the ΣΔ synthesizer employs a ΣΔ modulator of order two or higher, whereas the PI synthesizer, by definition, uses a first order modulator. This implies that the divider control of a PI synthesizer will be one bit, since first order modulation is constrained to a one bit output [36]. By contrast, the high order modulators of ΣΔ synthesizers can have multi-bit outputs, requiring multi-bit divider control. The number of bits required for divider control depends on the ΣΔ modulator architecture chosen. If a multi-stage, noise shaping (MASH) architecture is used, then the number of divider control bits is 2^{W-1}, where W is the order of the ΣΔ modulator [36].

The second difference between PI synthesis and ΣΔ synthesis can be understood by comparing the model views in Figures 3-7 and 3-12. ΣΔ synthesizers do not take advantage of the opportunity to use the feed-forward path employed by PI synthesizers to cancel the fractional-N quantization noise! Attenuation of the quantization noise in ΣΔ frequency synthesis is accomplished solely via the attenuation provided by the synthesizer low-pass dynamics, giving rise to the noise-bandwidth tradeoff described in section 2.2.4.

A key point to make is that PI fractional-N synthesis is a more general fractional-N synthesizer architecture, of which ΣΔ synthesizers are a subset. This is because ΣΔ frequency synthesis can be understood as PI frequency synthesis without the cancellation DAC, and with a higher order ΣΔ modulator for divider control. In this context, the reason for using a higher order modulator is simply because first order modulators have a significant periodic component in their output, and therefore do not have output spectra that look like shaped random noise. This reason alone does not necessarily justify elimination of the cancellation DAC, as we will shortly see.

3.2.3 Similarity Between a Fractional-N Synthesizer and ΣΔ MASH DAC

The last point to make before introducing our proposed solution to the problem of how to reduce the impact of quantization noise on fractional-N phase noise performance has to do with the general form of the model proposed in Figure 3-7. Figure 3-13 compares Figure 3-7 with the noise model for a second order ΣΔ MASH DAC [36]. The two systems are very analogous, and so we can think of a fractional-N synthesizer as a MASH ΣΔ DAC, where the analog output is phase (or frequency) rather than voltage or current. There are two subtle differences between the ΣΔ DAC and synthesizer, which merit some discussion.

First, the differentiator (the $(1 - z^{-}1)$ block) present in the cancellation path in the MASH DAC is not present in the synthesizer. This is because the one-bit ΣΔ modulator output passes through an integrator in the synthesizer and does not in the

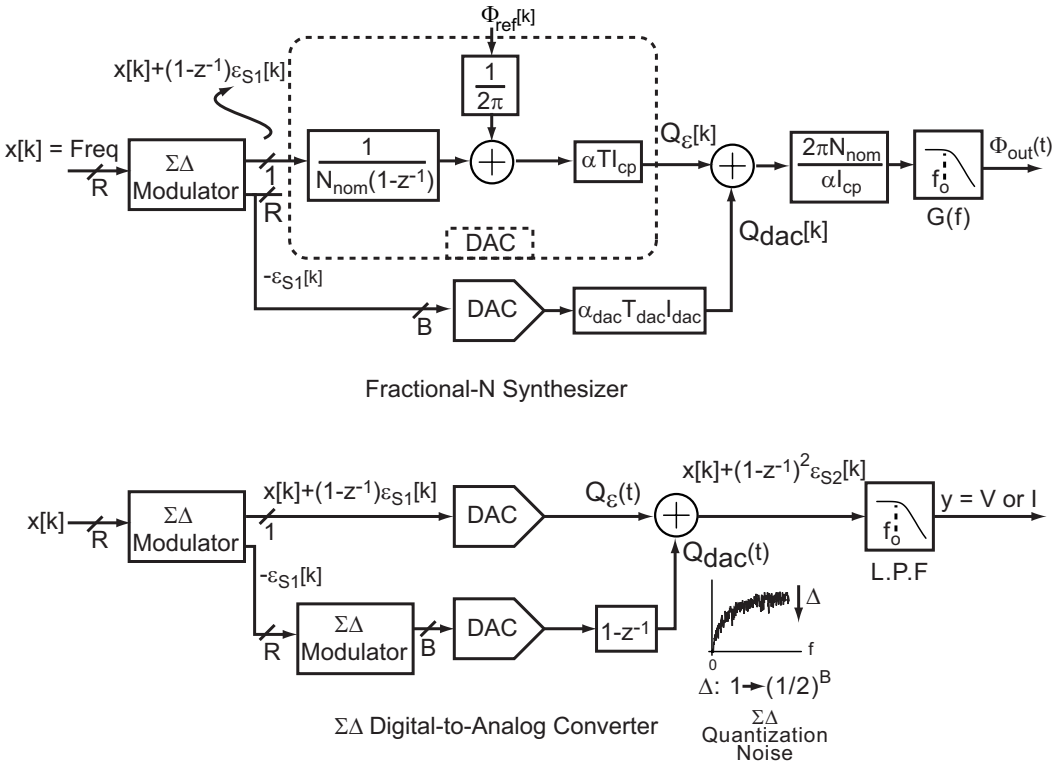


Figure 3-13 Comparison of Fractional-N Synthesizer and MASH ΣΔ DAC

ΣΔ DAC. The synthesizer integrator function is provided by the divider, which has a *frequency* input, but outputs a divider *phase* for use by the PFD [37]. Therefore, in order to cancel the quantization error in the synthesizer, the differentiator is not required.

The second difference is that the MASH ΣΔ DAC uses a *noise shaped* cancellation DAC to cancel the quantization noise of the first stage ΣΔ modulator. In the case of a classical PI synthesizer, the cancellation DAC is **not** noise shaped, and as discussed in section 3.2.1, the residual quantization noise introduced by the DAC after cancellation may dominate at frequencies near the carrier (in-band) unless a large resolution DAC is employed. Finite resolution in the cancellation DAC has the same effect for the ΣΔ MASH DAC as for the PI synthesizer. Namely, even if ideal cancellation is achieved, the DAC still introduces its own quantization noise into the system. In Figure 3-13, the quantization noise of the cancellation DAC is represented by ϵ_{S2}, and has a noise

power spectral density

$$S_{\epsilon_{S2}}(f) = \frac{1}{12}\left((1-z^{-1})^2\Delta\right)^2 \tag{3.14}$$

where Δ is the first stage quantization step-size, and is equal to one. Equation 3.14 shows that, in a 2^{nd} order MASH $\Sigma\Delta$ DAC, if ideal cancellation of the first stage quantization noise is achieved, the residual quantization noise, which is the quantization noise of the cancellation DAC, is 2^{nd} order shaped. The residual noise after cancellation in a PI synthesizer will not be 2^{nd} order shaped since the cancellation DAC is not noise shaped.

Further, if a multi-bit (B-bit in the figure) cancellation DAC is used, the quantization noise magnitude of the second stage is reduced by a factor of 2^B compared to the noise in the first stage [36]. For the MASH $\Sigma\Delta$ DAC this results in:

$$S_{\epsilon_{S2}}(f) = \frac{1}{12}\left((1-z^{-1})^2\frac{\Delta}{2^B}\right)^2 \tag{3.15}$$

This reduction of quantization noise is precisely what we wish to achieve for fractional-N synthesis!

In a MASH $\Sigma\Delta$ DAC, the critical circuit blocks are the two DAC structures. Because the cancellation is feed-forward in nature, matching between the DACs is essential for satisfactory cancellation to result. Dynamic element matching techniques are used to match the two DACs to a high degree [36]. Having recognized the similarity between fractional-N synthesizers and MASH $\Sigma\Delta$ DACs, we can borrow ideas and techniques from $\Sigma\Delta$ DAC design and apply them to fractional-N synthesizers.

3.3 Summary

In this chapter we have proposed a re-formulation of the model used for noise analysis of fractional-N synthesizers. The new model view explicitly demonstrates the limitation of existing synthesizer architectures to manage fractional-N quantization noise by framing synthesizer analysis in the context of $\Sigma\Delta$ MASH DAC design. Using the proposed model, the feed-forward gain mismatch and waveform shape mismatch

associated with phase interpolation based synthesis has been explained. The proposed model view of a $\Sigma\Delta$ synthesizer indicates that this architecture misses out on an opportunity to cancel the quantization noise, and instead relies purely on the synthesizer dynamics to filter quantization noise, giving rise to an undesirable noise-bandwidth tradeoff. In the next chapter, we propose a new synthesizer architecture capable of achieving high bandwidth and low noise fractional-N frequency synthesis simultaneously.

Chapter 4

Proposed Quantization Noise Reduction Technique

In the previous chapter, we arrived at a new understanding of fractional-N synthesis as a MASH $\Sigma\Delta$ DAC, and made several key observations. Namely, PI synthesis is a general case of fractional-N synthesis, a $\Sigma\Delta$ synthesizer misses out on an opportunity to reduce quantization noise through a cancellation path, the cancellation DAC should be noise shaped as in a MASH $\Sigma\Delta$ DAC, and perhaps we can borrow concepts from $\Sigma\Delta$ DAC literature to mitigate mismatch between the feed-forward cancellation path and the quantization error path. Also, as discussed in section 3.2.1, we must keep in mind the fact that the error charge is referenced to a single VCO period, so any cancellation DAC should also have its output occur over a single VCO period to maximize noise cancellation for a given DAC resolution.

In this chapter, we present an architecture that eliminates the systematic mismatch between the quantization error signal and cancellation DAC signal. Appearing first in [21], this approach offers a first step towards our final goal of reducing quantization-induced phase noise to the point where it does not impact synthesizer performance. As will be shown in section 4.2, additional measures must be taken to account for mismatch sources internal to the architecture in [21] in order to truly gain the potential benefit it offers.

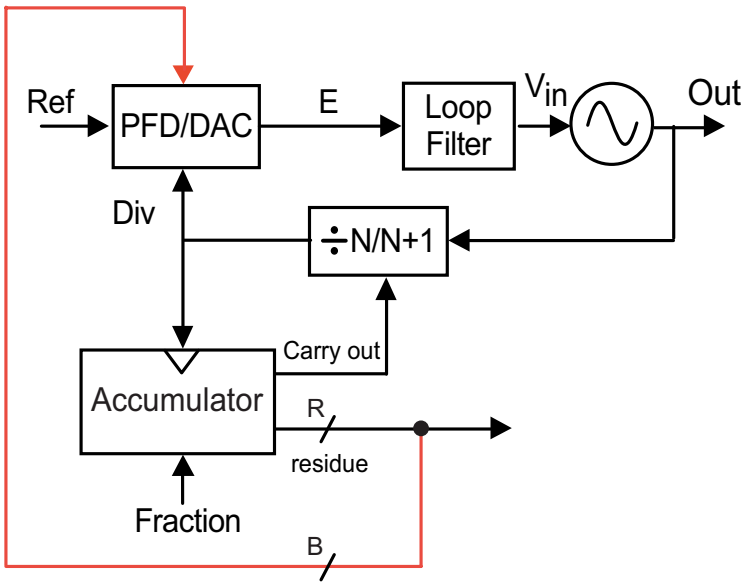


Figure 4-1 PFD/DAC Synthesizer Architecture

4.1 The PFD/DAC Approach

The fundamental challenge in PI synthesis is to match the feed-forward cancellation charge to the quantization error charge. The two signal paths go through independent domain transformations. The error signal is integrated from frequency to phase via the divider, is converted to a pulse width modulated signal by the PFD, and is converted to charge by the charge-pump. By contrast, the cancellation signal goes through a direct digital to analog transformation via the cancellation DAC. It is no wonder that there will be significant mismatch when the two signals are summed at the charge-pump output!

Our proposed method to eliminate this systematic mismatch begins with the synthesizer architecture depicted in Figure 4-1 [21]. In the figure, the PFD, charge-pump, and cancellation DAC have been combined into one circuit. The key concept is that by processing the error signal and cancellation signal in the same circuitry self-alignment is achieved, and the systematic feed-forward mismatch is eliminated.

The PFD/DAC architecture is presented in Figure 4-2. The DAC is controlled by a subset of the residue bits available from the accumulator. These bits are decoded so that the thermometer decoder outputs control a unit current source. For a B-bit

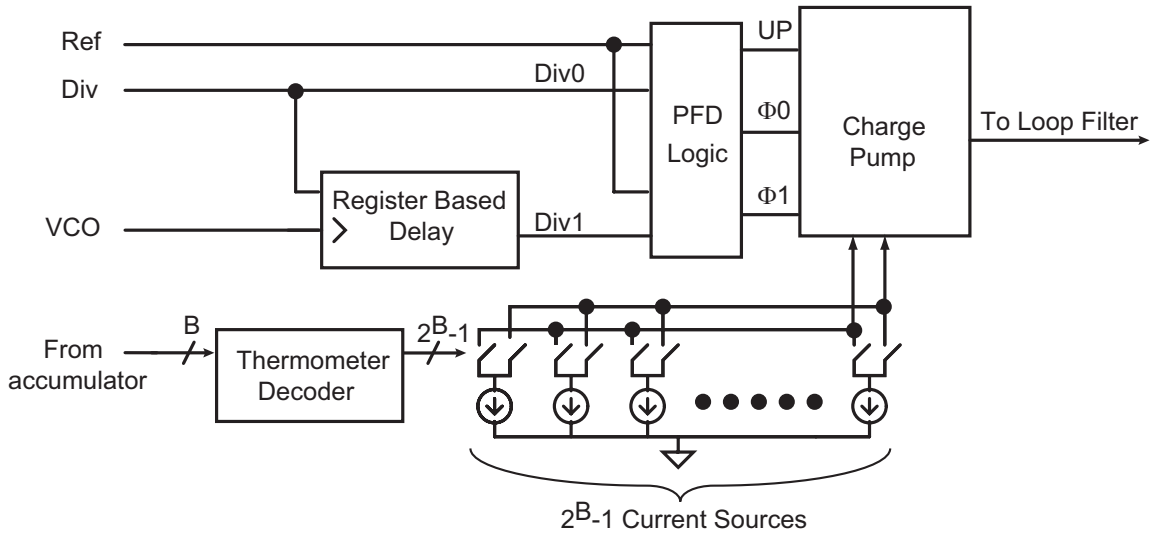


Figure 4-2 PFD/DAC Architecture

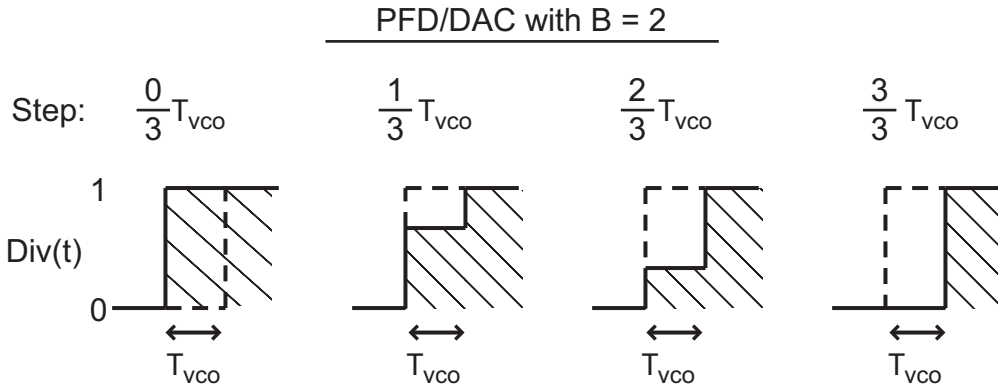


Figure 4-3 PFD/DAC Operation: The Charge Box

word from the accumulator, there will be $2^B - 1$ unit current sources. Two phase comparison paths are created by clocking the divider through a register delay cell. The register creates a divider phase that is one VCO period later than the DIV signal. The two divider phases, Φ_0 and Φ_1 in the figure, are then processed by the phase detector circuitry.

Operation of the PFD/DAC is qualitatively explained in Figure 4-3 for a 2 bit PFD/DAC. The LSBs of the accumulator control how much current is delivered within a one VCO time window created by the register based delay. The fractional-N dithering quantization error determines the *phase* of the divider output, and therefore

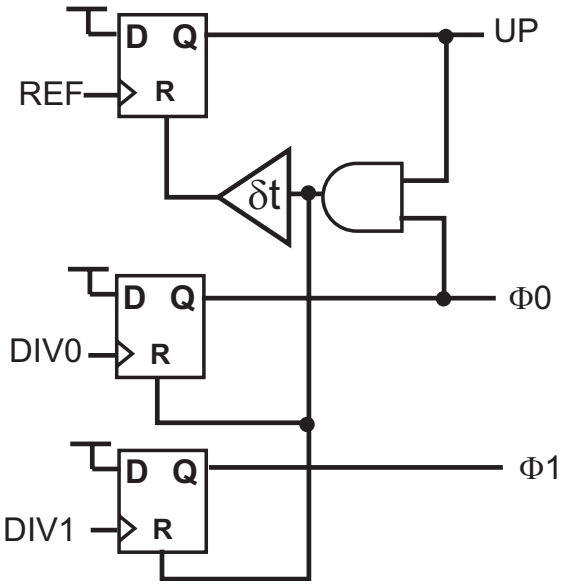


Figure 4-4 PFD Logic Architecture for a Practical Implementation of the PFD/DAC

the phase of the PFD output. The LSBs of the accumulator output control the *magnitude* of the current delivered in the one VCO period window. By processing the error signal and cancellation signal in the same circuitry, the systematic gain mismatch between them is eliminated. Furthermore, the cancellation charge is referenced to a single VCO period. This is a key advantage of the PFD/DAC approach because the VCO period changes with the varying divide value, and requires a matching change in the width of the cancellation window. The PFD/DAC self-aligns the width of the cancellation window to one VCO period and therefore allows the maximum possible amount of quantization noise reduction to be obtained.

4.1.1 The PFD/DAC Approach: Constant Charge Delivery

To provide a more detailed explanation of PFD/DAC operation, we examine a practical implementation of charge balance in a real synthesizer. We begin by choosing a PFD architecture. For reasons that will be explained in Chapter 5, we choose the offset reset tri-state PFD, depicted in Figure 4-4. Recall that Div1 is a one VCO period delayed version of Div0.

Charge balance is depicted in Figure 4-5. As the accumulator residue increases

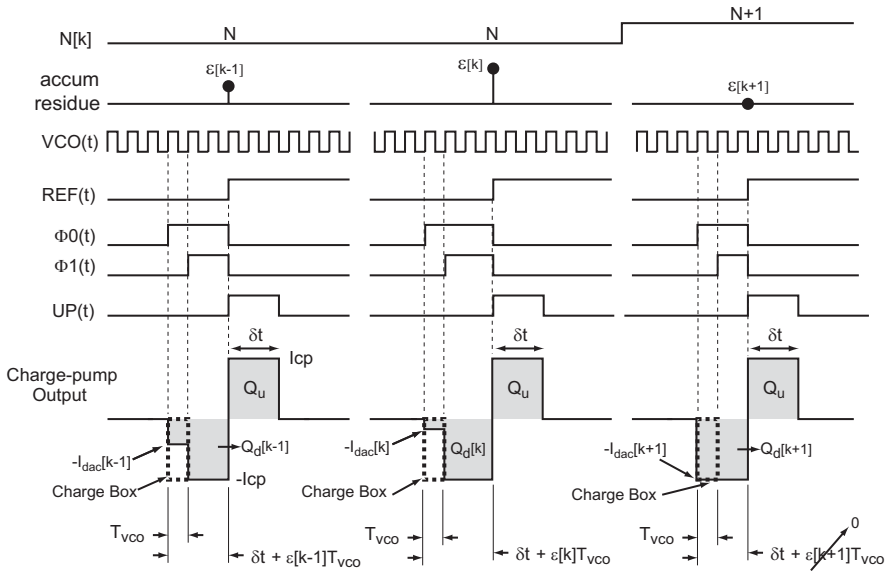


Figure 4-5 PFD/DAC Operation: A Practical Example

each period, so does the phase error. The magnitude of the quantization error term is represented by $\epsilon[k]$. Q_u and $Q_d[k]$ represent positive and negative charge packets delivered by the PFD/DAC to the loop filter. N[k] is the instantaneous divide value and varies between N and N+1, according to the fractional-N dithering process. While Q_u is a constant width and constant magnitude charge packet, $Q_d[k]$ will vary in shape during normal operation.

The offset PFD architecture results in a steady-state condition where the REF signal occurs after the divider signals, and produces an UP pulse of width δt seconds. This creates a constant positive charge packet, Q_u. The key feature of the PFD/DAC technique is to control delivery of negative current in such a way that it balances the positive current every period.

As the accumulator residue increases, the width of the negative current pulse changes proportionally because phase error, represented by $\epsilon[k]$, changes. This is represented in the figure by the changing width of the down current pulse, which is equal to

$$\delta t + \epsilon[k]T_{vco} \quad . \tag{4.1}$$

To ensure that the changing pulse-width does not upset charge balance by generating

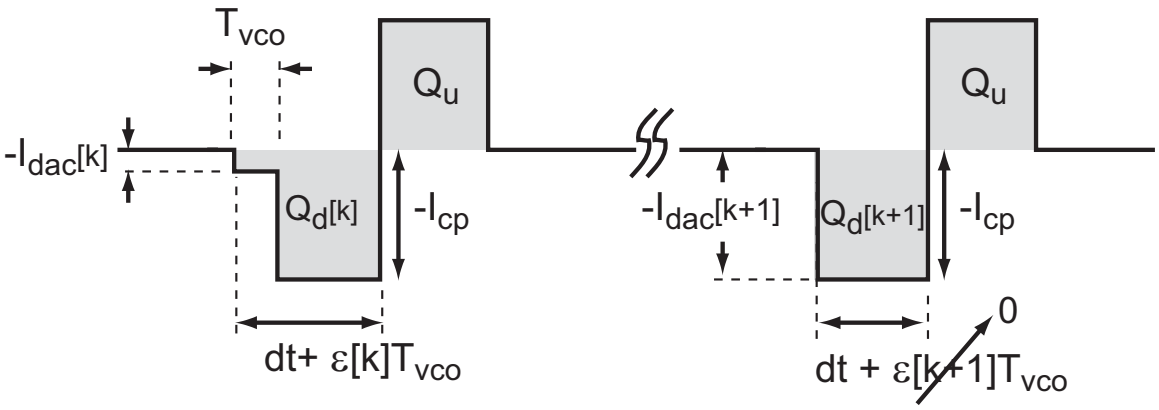


Figure 4-6 Charge Balance In the PFD/DAC Using the Two Right-most Charge-packets in Figure 4-5

a value of $Q_d[k]$ that is unequal to Q_u, the magnitude of negative current delivered by the charge-pump, i_{dac}, is DAC controlled during the time window corresponding to the charge-box. The key differences between this approach and classical PI synthesis are first that the PFD/DAC technique performs the noise cancellation operation by controlling the charge in a *one* VCO period window, and second, that the cancellation signal (the variable charge control in the charge-box) and the quantization noise signal (the variable pulse-width of the negative current) are produced at the same time and in the same circuitry. An inherent gain match is therefore made between the quantization noise signal and cancellation signal. Additionally, as the VCO period changes with changing divider values, the charge-box tracks the change and compensates accordingly.

The operation of the PFD/DAC in Figure 4-5 can be summarized as follows: *As the phase error varies because of the fractional-N dithering process, the PFD/DAC changes the magnitude of current delivered in a one VCO period window so that a constant amount of negative charge is delivered each period, which balances a constant positive charge, resulting in no net charge transfer in steady-state.*

The negative charge has a constant magnitude but varying shape, and balances a constant magnitude and constant shape positive charge. Overall, no net charge is delivered to the loop filter, and fractional-N quantization noise is eliminated. Of course, this is only true in a DC sense, because, while the total charge does equal

zero, the charge-pump output takes on non-zero values during operation, and therefore moves the loop filter voltage. We propose two methods to mitigate the impact of this error in section 4.2.5.

We now derive how much current the PFD/DAC should deliver during operation to cancel the quantization error. We start with the assumption that the PFD/DAC is operating as desired, and all of the negative charge packets in Figure 4-5 are equal and exactly cancel the positive charge packets. We then examine the last two negative charge packets in more detail as depicted in Figure 4-6.

For our assumption of constant net negative charge to hold, we require both charge packets to be equal. As shown in Figure 4-6, the difference in the widths of the negative current pulse will vary according to the phase error introduced by the fractional-N dithering process, $\epsilon[k]T_{vco}$. The constant charge balance equation for the negative charge packets in the figure is:

$$(T_d + \epsilon[k]T_{vco} - T_{vco})(-I_{cp}) + (-T_{vco}I_{dac}[k]) = (T_d + \epsilon[k+1])(-I_{cp}) \quad . \tag{4.2}$$

We can simplify equation 4.2 by making the substitution $\epsilon[k+1] = 0$, with the result

$$(T_d + \epsilon[k]T_{vco} - T_{vco})(-I_{cp}) + (-T_{vco}I_{dac}[k]) = -T_d I_{cp} \quad . \tag{4.3}$$

Solving for $I_{dac}[k]$, we conclude

$$I_{dac}[k] = (1 - \epsilon[k])I_{cp}. \tag{4.4}$$

Equations 4.2 and 4.4 have a simple and intuitive explanation. As the phase error due to fractional-N dithering increases according to $\epsilon[k]T_{vco}$, the DAC current decreases according to $(1 - \epsilon[k])I_{cp}$ so that the net negative charge remains constant.

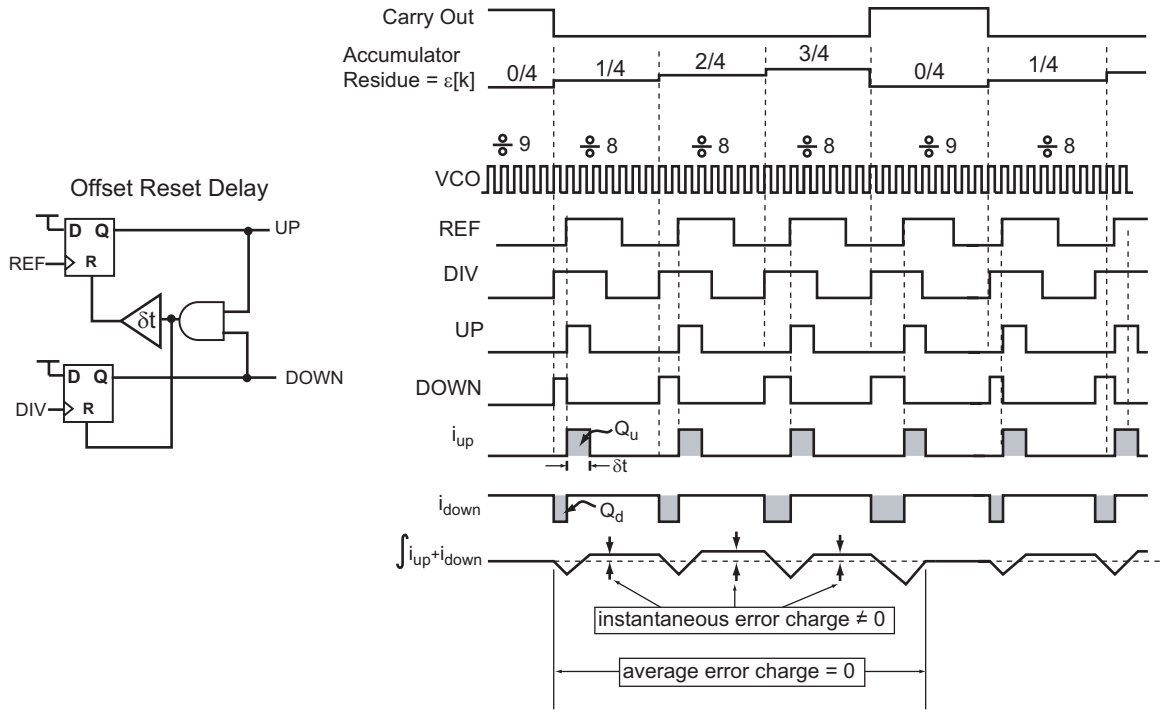


Figure 4-7 Charge Balance in a Classical Fractional-N Synthesizer for N=8.25

4.1.2 Comparison of Charge Balance in a Classical Fractional-N Synthesizer with the PFD/DAC Synthesizer

To reinforce the merit of the PFD/DAC technique, we step through an example of charge balance for a divide value of 8.25. We compare charge balance in a classical fractional-N synthesizer to the PFD/DAC synthesizer.

Figure 4-7 depicts the steady-state waveforms associated with a classical fractional-N synthesizer with N=8.25. The offset tri-state PFD previously described is used for comparison purposes. In the case of the classical fractional-N synthesizer, only one divider phase is employed.

In steady-state, the divider is dithered between N=8 and N=9 such that, on average, the divide value is 8.25. This corresponds to a repeating {8, 8, 8, 9} divider pattern. As the accumulator residue increases, so does the value of $\epsilon[k]$, and, correspondingly, the width of the phase error signal associated with the divider output. We note that the pulse-width of i_{down} grows until the residue is reset and a carry out is produced, at which point the width of i_{down} is reset to its initial value.

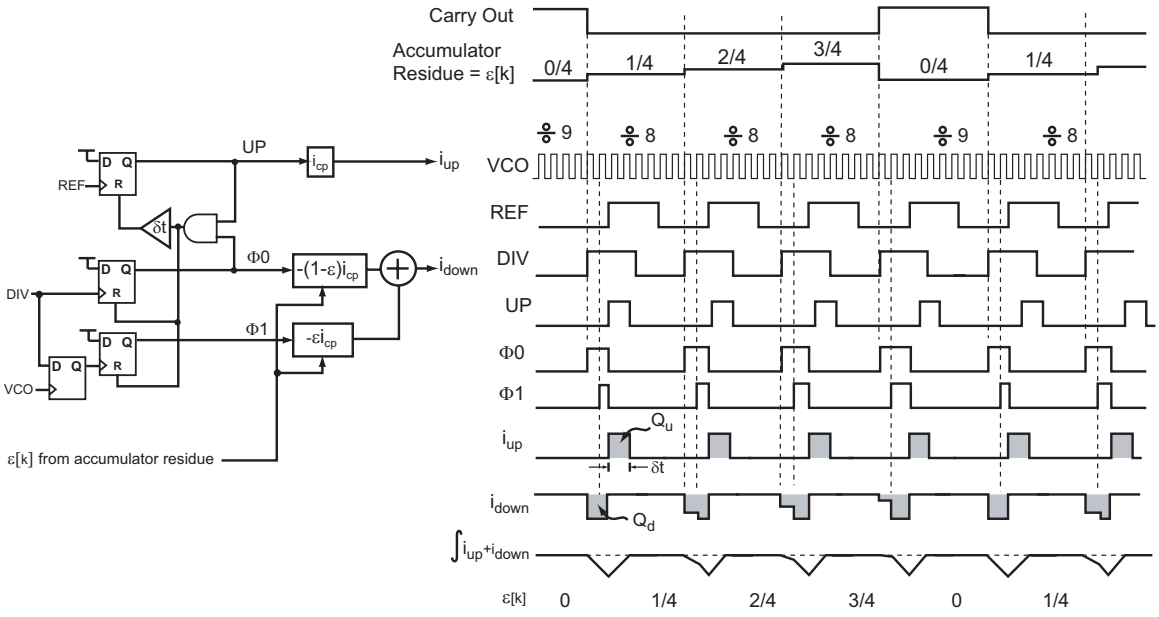


Figure 4-8 Charge Balance in a PFD/DAC Fractional-N Synthesizer for N=8.25

Also shown in the figure are the current waveforms associated with the UP and DOWN PFD outputs. UP controls the positive charge-pump output current, and DOWN controls the negative charge-pump output current. We look at the total charge delivered each period to observe how the PLL operates in steady-state. The bottom waveform, which is the time integrated value of the positive and negative charge-pump currents, corresponds to the charge that controls PLL operation.

For steady-state conditions to exist, the average charge delivered over some time-frame must be zero. For the classical fractional-N synthesizer represented by the waveforms in Figure 4-7, the average charge delivered equals zero every *four* reference periods. instantaneous error charges are produced by the dithering process on a period-by-period basis, but the *average* error charge over four periods is zero. It is the instantaneous error charges that produce fractional spurs. As the figure clearly shows, error charge is delivered periodically at a rate proportional to the fractional portion of the divide value. This results in a VCO control voltage that has periodic components varying about some desired DC value and results in fractional spurs.

The PFD/DAC synthesizer compensates for the instantaneous errors by controlling the negative charge-pump current in a one VCO period wide window. This is

because the quantization phase error signal is referenced to a single VCO period according to $\epsilon[k]T_{vco}$. Because charge is the control parameter used by the PLL, the increasing phase error $\epsilon[k]T_{vco}$ can be offset by decreasing current in a proportional manner.

The PFD/DAC accomplishes such compensation as depicted in Figure 4-8. By using two divider phases that are separated by a single VCO period, in combination with the information about the magnitude of the phase error embedded in the accumulator residue, a charge-box is created. The current controlled in the charge-box compensates for increasing phase error by delivering decreasing compensation current. The net result, as depicted in the figure, is that no net error charge is delivered to the loop filter to move the VCO control voltage on a period-by-period basis. Elimination of the instantaneous error charges present in Figure 4-7 result in improved spurious performance and reduction of broadband quantization noise.

We do note that, while the *net* error charge is zero every period (to within the resolution of the charge-box) positive and negative current are delivered every period. This implies that there will be some periodic information present in the PFD/DAC output, but it will be concentrated at the reference frequency, which is much higher than the fractional spur frequency. The resulting reference spur will be attenuated by the PLL dynamics. There is also some residual fractional information present in the output current because the shape of the charge box varies periodically according to the fractional value. However, the residual spurious information is much smaller for the PFD/DAC than the classical fractional-N synthesizer.

We will propose techniques later in this chapter that remove the residual spurious information in the charge-pump output waveforms shown in Figure 4-8.

4.1.3 An Alternative Explanation of the PFD/DAC Approach

Here, we present an alternative explanation of PFD/DAC operation that develops the charge balance relationship in a slightly different way. It focuses on the idea that, using the PFD/DAC technique, spurious energy is moved from the fractional spur frequency to the reference frequency. We begin with a re-examination of classical

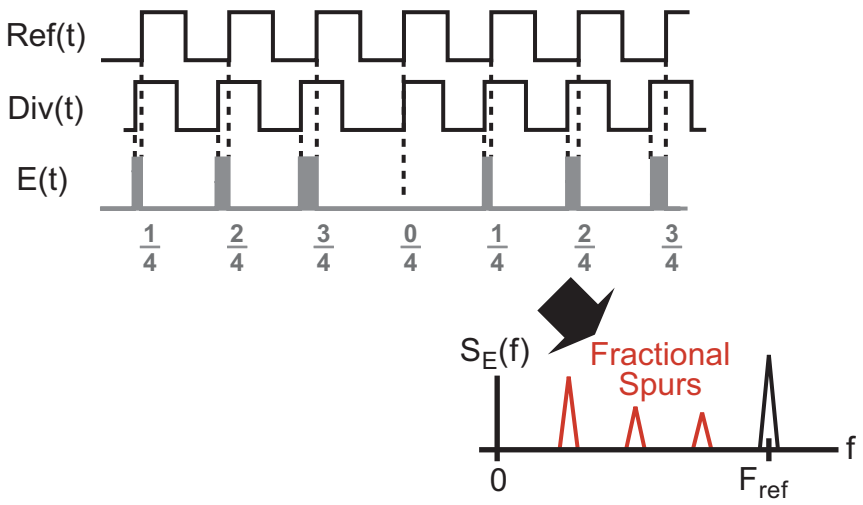


Figure 4-9 Fractional-N Spurs in Classical Fractional-N Synthesis

fractional-N synthesis, as depicted in Figure 4-9. The key signals for analysis are the reference, the divider output, and the PFD error pulse output, E. The example waveforms shown are for a fractional divide value of 1/4. The accumulator residue, $\epsilon[k]$, therefore goes through a repeating {1/4, 2/4, 3/4, 0/4} sequence. Note that, for this explanation, we ignore the constant negative pulses that would accompany positive ones that are shown.

When the accumulator residue is {1/4, 2/4, 3/4}, the accumulator carry out is low, and the divider divides by N. When the accumulator wraps, its residue is at 0/4, and a carry out is produced, causing the divider to divide by N+1.

As has been discussed, the instantaneous phase error is proportional to the accumulator residue, as evidenced by the changing pulse-width of E(t) in the figure. In fact, the accumulator residue represents the instantaneous phase error in fractions of a VCO period. This can be understood by normalizing the area enclosed by E(t) to a VCO period. 1/4 corresponds to one quarter of the area that is enclosed by a one VCO wide pulse, 2/4 corresponds to the area enclosed by a pulse one half the width of the VCO period, etc.

The figure shows that the periodic, pulsed nature of the error signal gives rise to fractional spurs in the system. In addition to low frequency fractional spurs, a spur appears at the reference frequency because phase comparisons are performed by the

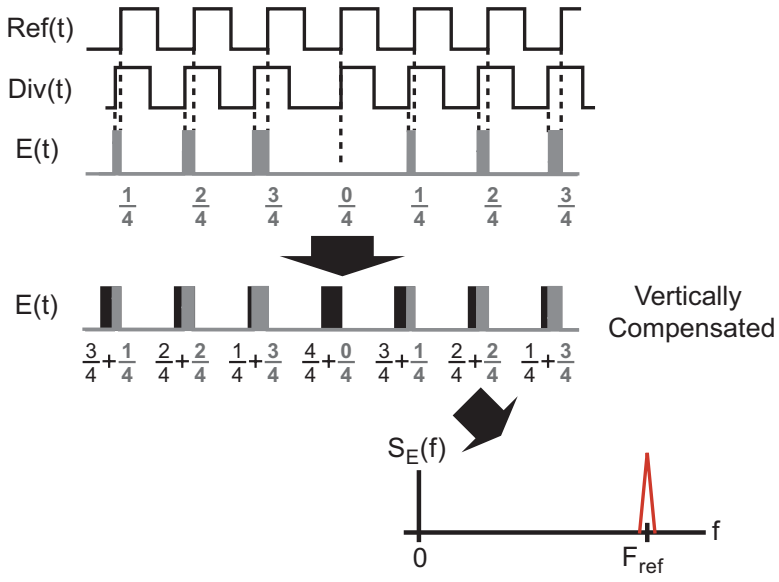


Figure 4-10 Vertical Compensation of fractional spurs

PFD at the reference frequency.

Figure 4-10 presents a vertical scheme to compensate for fractional spurs. Fractional spurs occur because E(t) has a time varying, pulsed nature. Vertical compensation eliminates fractional spurs by adding an additional signal to E(t) such that the resulting E(t) exhibits constant area pulses. In Figure 4-10, the added signal is represented by the black boxes. The compensation signal is scaled such that the pulse height is the same as the phase error signal, and the total pulse area of the compensated E(t) will always equal one. The result is that spurious energy in the compensated system occurs at the reference frequency only. This is a good tradeoff, because the reference spur occurs outside the closed loop bandwidth and is attenuated by the PLL low-pas transfer function.

As has been discussed, veritical compensation requires the use of multi-phase VCOs or dividers, and is limited in its ability to compensate fractional spurs by the ability to generate multiple phases of GHz frequency signals with low mismatch [15–17].

An alternative to vertical compensation is depicted in Figure 4-11. Horizontal compensation creates constant area pulses by controlling a variable height, but constant width, pulse that is added to the phase error signal. In this case, the fractional

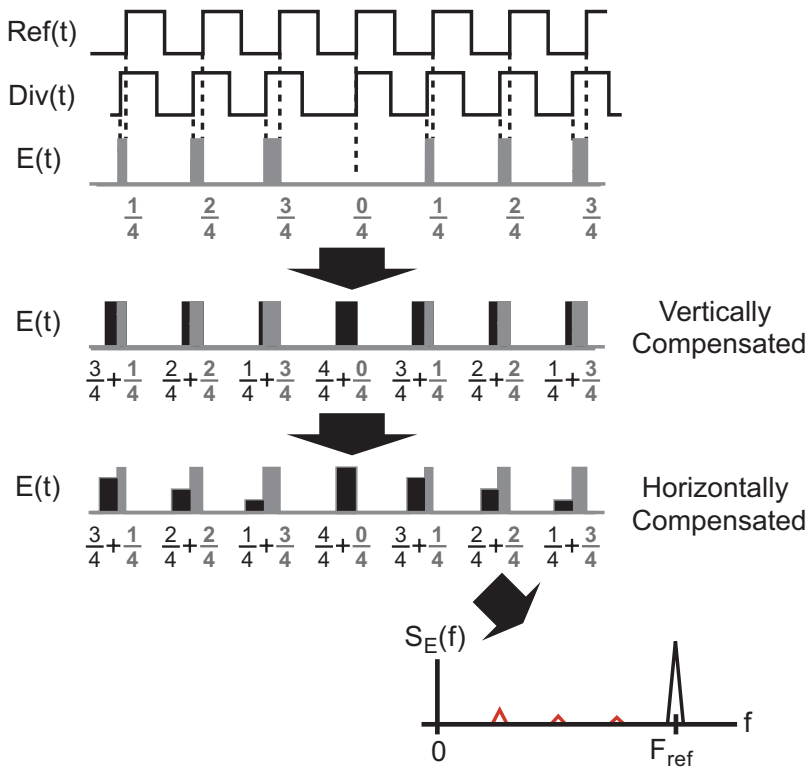


Figure 4-11 Horizontal Compensation of fractional spurs

spurs are not completely eliminated because the shape of the overall horizontally compensated waveform varies periodically. While most of the spurious energy using this technique appears at the reference frequency, there is a residual component at the fractional spur frequencies because of the periodically changing wave-shape.

Figure 4-12 illustrates the PFD/DAC approach to implement horizontal compensation. Two error signals are used to create the overall horizontally compensated waveform. The first error signal, $E_1(t)$, is the result of a PFD comparison between the reference and a delayed version of the divider. The second error signal, $E_2(t)$, is created by comparing the divider signal to the reference via a second PFD. The two error signals are scaled according to the accumulator residue, $\epsilon[k]$. $E_1(t)$ is scaled by $\epsilon[k]$, and $E_2(t)$ is scaled by $(1 - \epsilon[k])$. These waveforms are summed and sent to the loop filter.

Figure 4-13 focuses on the scaled error pulses. As the figure shows, summation of the scaled error pulses results in the same waveform depicted in Figure 4-11. There

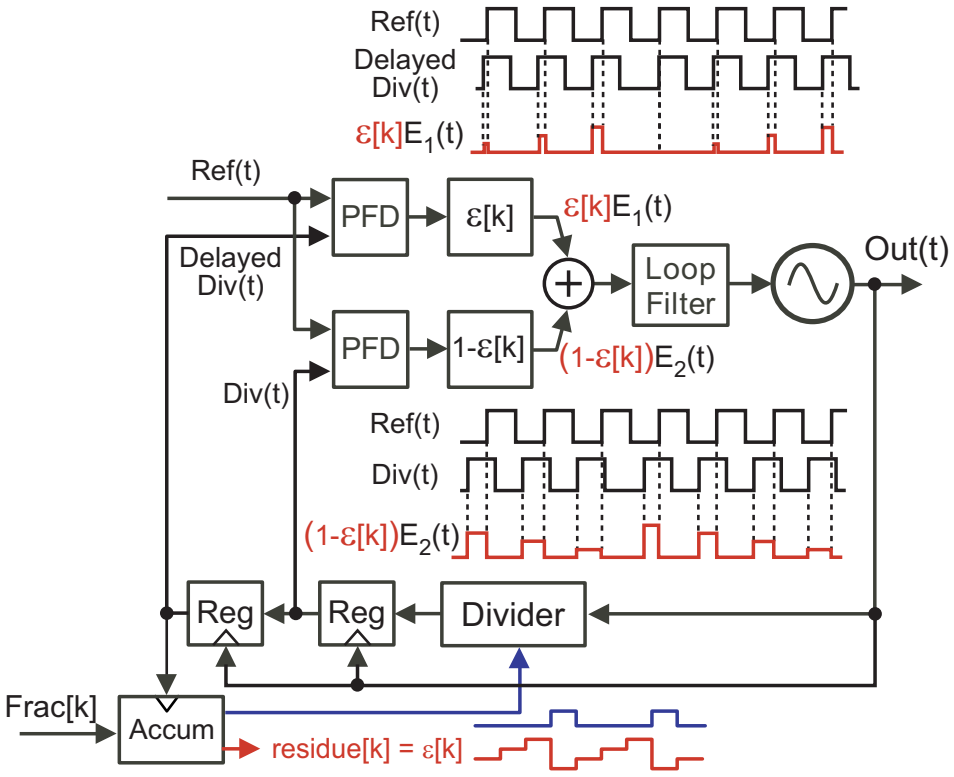


Figure 4-12 Implementation of the Horizontal Compensation Scheme

is a subtle difference revealed in the figure. The proposed implementation of the horizontal compensation scheme results in the same ultimate wave-shape as depicted in Figure 4-11, but the two signals added to generate it differ from the two signals added together in Figure 4-11. However, the resulting waveform is the same, as represented by the equivalence of the two E(t) waveforms shown in Figure 4-13.

An actual implementation of the horizontal compensation technique is depicted in Figure 4-14. This architecture corresponds to the PFD/DAC approach [21]. Charge-pumps are used to weight the PFD outputs and perform the scaling function on the error signals. This is intuitive because, as we have emphasized in this thesis, the error *charge* is the control variable of interest in the system. The PFD outputs are time signals that control the amount of time the charge-pumps deliver charge to the loop filter. The amount of current delivered by each charge-pump is controlled according to the residue. Therefore, the charge-pumps are acting in the manner of a DAC rather than as a traditional charge-pump, which only outputs one value of current. Because

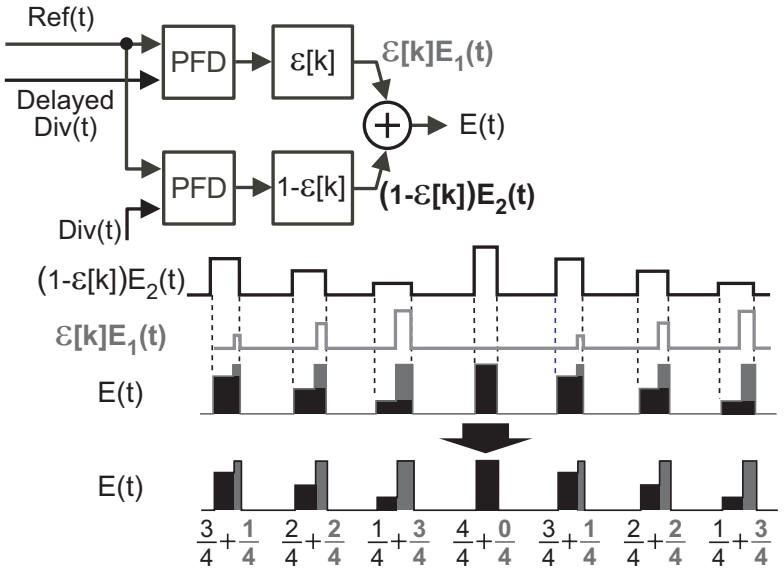


Figure 4-13 Horizontal Compensation Error Signals

the error signal and compensation signal are processed in the same circuitry at the same time, there is an inherent gain match between them.

4.1.4 Model for the PFD/DAC Synthesizer

Having established the method of operation of the PFD/DAC approach, we present a noise model based on the analysis of Chapter 3. Figure 4-15 presents the model for a PFD/DAC synthesizer. Because the DAC control has been combined with the PFD circuitry, they both see the same gain, namely $\alpha T_{vco} I_{cp}$, where we have made the simplification that

$$\frac{T}{N_{nom}} = T_{vco} \quad . \tag{4.5}$$

Inherent matching between the quantization noise signal and cancellation signal is achieved. A B-bit subset of the R-bit accumulator residue is used to control the PFD/DAC. The resulting quantization noise due to the finite DAC resolution is represented by ϵ_{S2}, and has a step-size

$$\Delta_{S2} = \frac{\Delta_{S1}}{2^B} = \frac{1}{2^B} \quad . \tag{4.6}$$

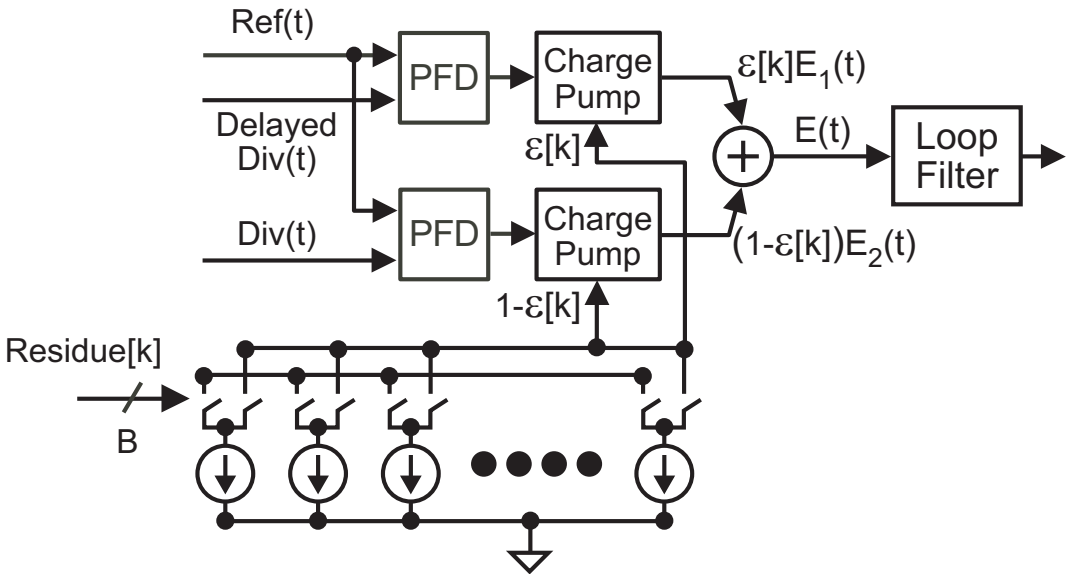


Figure 4-14 PFD/DAC Implementation of the Horizontal Cancellation Scheme

Equation 4.6 shows that, if perfect cancellation of the fractional-N quantization noise ϵ_{S1} is achieved, we still must consider the impact of the finite resolution of the cancellation signal. If this noise source is not noise shaped, it may become dominant at low frequencies.

4.1.5 The Issue of Mismatch

As with any real circuit, there are several sources of mismatch possible within the PFD/DAC, and these must be managed to maximize performance. The ability of the PFD/DAC to properly cancel quantization noise is determined by the quality of the charge-box generated by the system, and the degree to which charge within the charge-box is accurately controlled. Since charge is created by controlling a DAC current in a one VCO window in time, two immediate sources of error are magnitude error between the different DAC output levels, and timing error in accurately creating a one VCO wide window.

Additionally, DAC quantization noise and the more subtle effect of waveform shape mismatch between the quantization noise signal and cancellation signal create additional sources of noise in the system. The basic PFD/DAC synthesizer architec-

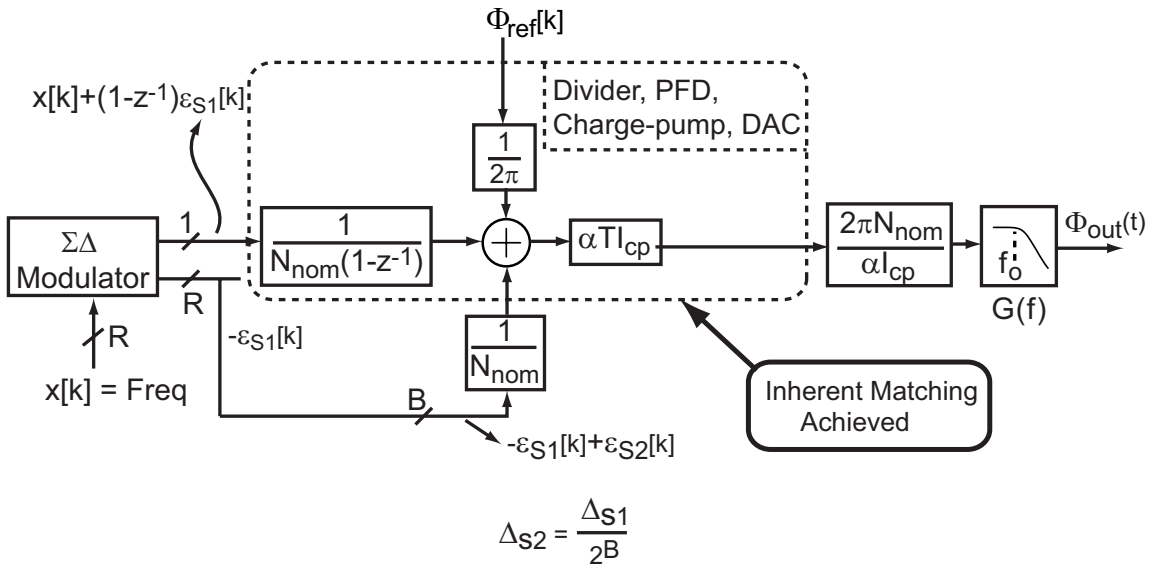


Figure 4-15 Model for PFD/DAC Synthesizer

ture presented thus far does not account for any of these error sources and therefore will have a limited ability to improve noise performance.

In the next section, we propose dynamic matching techniques that allow the PFD/DAC synthesizer to achieve excellent noise suppression in the face of internal mismatch, as well as a way to reduce the impact of finite cancellation DAC resolution and minimize the impact of shape mismatch between the quantization noise signal and cancellation DAC output.

4.2 Proposed Solution: A Mismatch Compensated PFD/DAC Synthesizer Architecture

We now propose an architecture that evolves from the basic PFD/DAC architecture and is capable of mitigating the effects of mismatch sources internal to its structure and inherent to its operation. In order to emphasize the ability of the proposed architecture to answer each of the concerns raised so far, we will proceed to the final proposed solution in several steps.

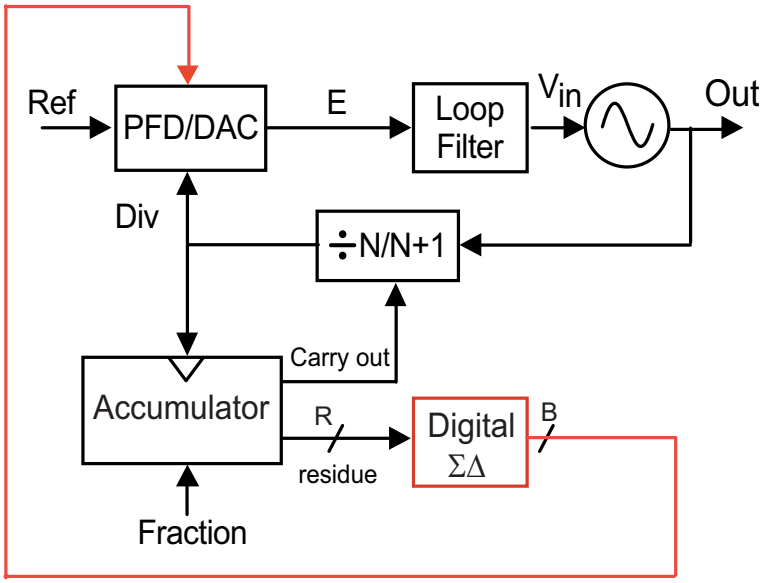


Figure 4-16 Using a Noise Shaped PFD/DAC

4.2.1 Using a Noise Shaped Cancellation DAC for Improved In-band Noise

We conclude from the discussion of section 4.1.4 that it is desirable to noise-shape the cancellation DAC's quantization noise. This is accomplished by processing the residue of the divider control accumulator with a $\Sigma\Delta$ modulator, as depicted in Figure 4-16. Now the cancellation DAC quantization noise will have a shaped profile, and is no longer a concern at low offset frequencies from the carrier. Because the DAC is multi-bit, the magnitude of the quantization noise is as described by equation 4.6.

In an ideal system, the quantization noise of the divider control $\Sigma\Delta$ is perfectly canceled, and effectively "replaced" by the quantization noise of the cancellation DAC, which is smaller in magnitude because of the multi-bit DAC. The quantization noise suppression achieved by a noise shaped PFD/DAC synthesizer, when compared to a 2^{nd} order $\Sigma\Delta$ synthesizer, is directly related to the resolution of the PFD/DAC according to equation 4.6, and can be expressed in dB as:

$$Quantization\ Noise\ Suppression = 6.02 \cdot B \quad dB, \tag{4.7}$$

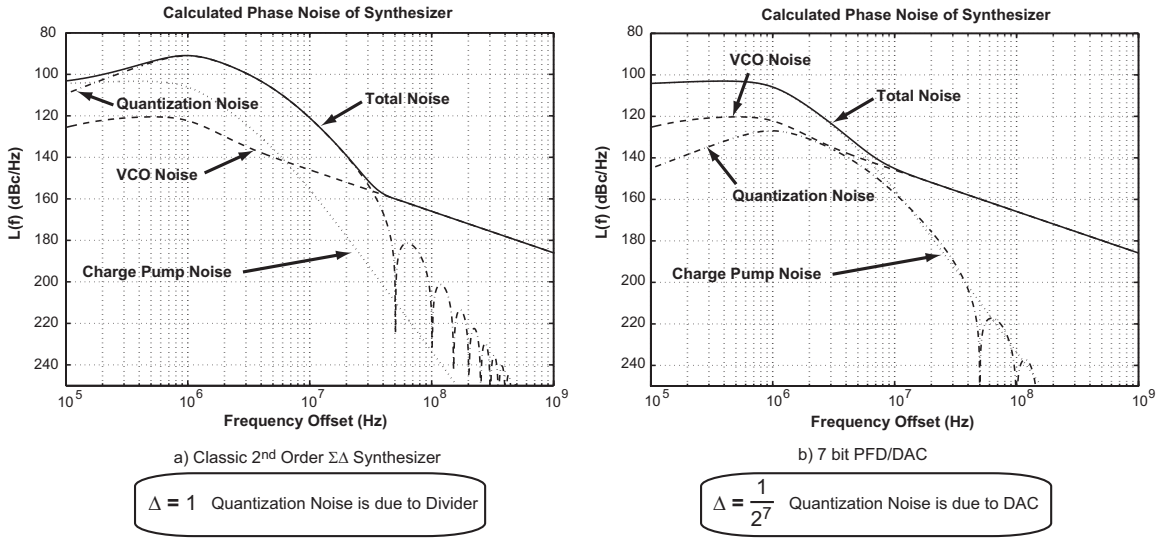


Figure 4-17 Quantization Noise Suppression

where a B-bit PFD/DAC is used.

The reason we compare the PFD/DAC synthesizer to a 2^{nd} order $\Sigma\Delta$ modulator is because they both employ two first order modulators, and have the same order noise shaping at the synthesizer output. In a $\Sigma\Delta$ synthesizer, the order of noise shaping is reduced by one order because of the integration function performed by the divider [1]. A 2^{nd} order $\Sigma\Delta$ synthesizer therefore has a 1^{st} order quantization noise slope at the synthesizer output. The PFD/DAC completely cancels the quantization noise of the divider control $\Sigma\Delta$ modulator, and introduces its own noise, which is 1^{st} order shaped. If we were to use the MASH $\Sigma\Delta$ DAC to describe this situation, we would say that the PFD/DAC synthesizer is a 1-1 MASH, meaning it is equivalent to a second order $\Sigma\Delta$ consisting of two first order stages. [36].

The calculated noise suppression for a 7-bit PFD/DAC synthesizer is shown in Figure 4-17. The left plot represents a classical $2nd$ order $\Sigma\Delta$ synthesizer with a 1MHz bandwidth. Since no cancellation path is employed, the magnitude of the quantization step-size is $\Delta = 1$, and quantization noise dominates the phase noise profile from 10kHz to 30MHz. In the right plot, representing a 7-bit PFD/DAC synthesizer, quantization noise appearing at the output is 42dB lower and does not dominate the phase noise profile over any frequency range. Reducing the quantization

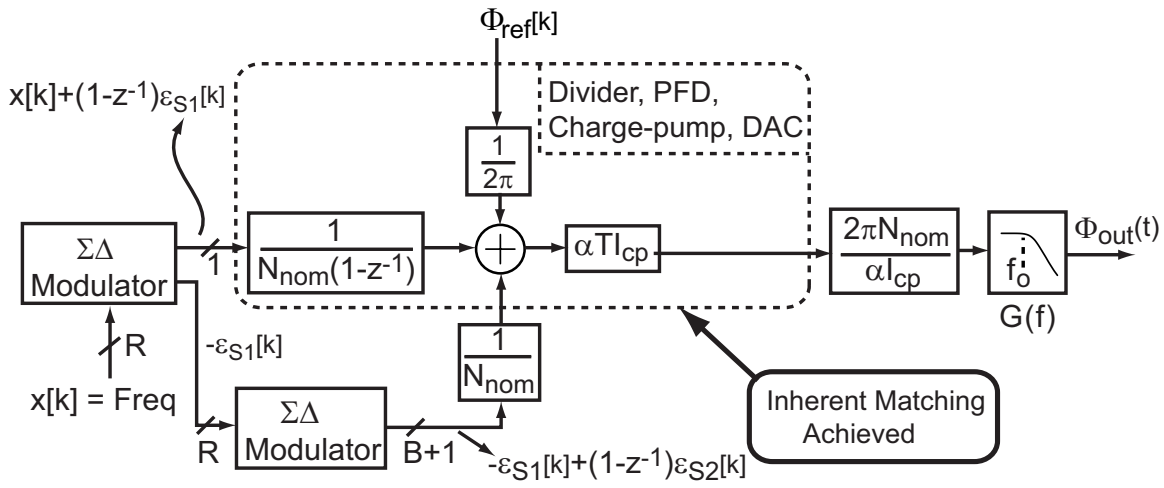


Figure 4-18 Model for Noise Shaped PFD/DAC Synthesizer

noise by 42dB has effectively eliminated it from consideration when designing the synthesizer, allowing the designer to concentrate on reducing intrinsic noise sources.

Figure 4-18 presents the model for a noise shaped PFD/DAC synthesizer. The only change from Figure 4-15 is the addition of the $\Sigma\Delta$ modulator in the cancellation path.

Finally, Figure 4-19 shows the implementation of the noise shaped PFD/DAC, including the two $\Sigma\Delta$ modulators. One key point to understand in the figure is the difference in bit-widths of the various signals in the digital data path.

In Figure 4-19, the divider control $\Sigma\Delta$ has an R-bit input and 1-bit output. This $\Sigma\Delta$ is a simple digital accumulator, and its output is the carry out of the accumulator. The divider control modulator LSBs represent the quantization error residue, and is R-bits wide. For a B-bit PFD/DAC, the B MSBs of the R-bit residue are added to the one-bit output of the second $\Sigma\Delta$ modulator. The DAC control $\Sigma\Delta$ modulator takes as input the R-B residue LSBs of the R-bit residue.

The B-bit signal input to the adder takes on values of 0 to $2^B - 1$. For B=7, this corresponds to a range of 0 to 127. The $\Sigma\Delta$ modulator processing the R-B residue LSBs generates a 1-bit output which takes on values of 0 or 1. Therefore, the output of the adder will take on values of 0 to 2^B, which is 0 to 128 for a 7-bit PFD/DAC. Basically, the $\Sigma\Delta$ process introduces one extra level to the noise shaped PFD/DAC

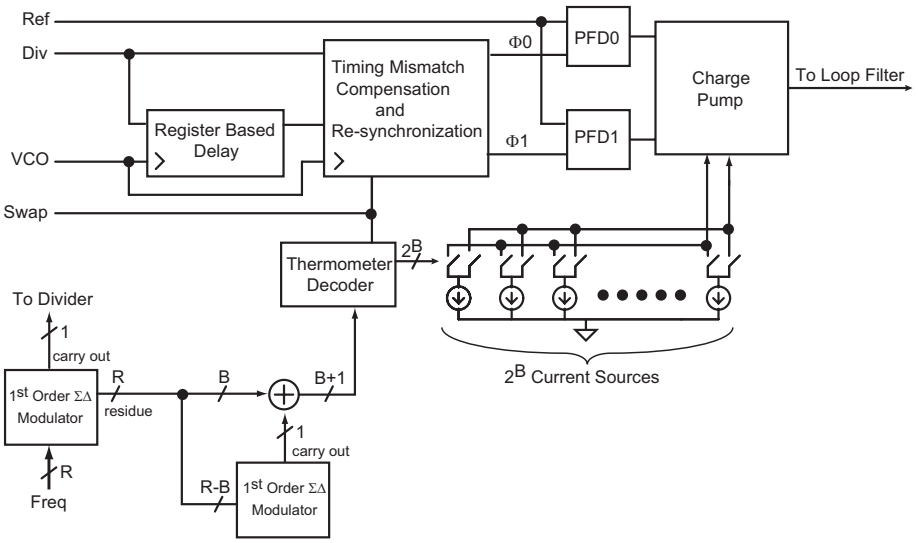


Figure 4-19 Noise Shaped PFD/DAC Architecture

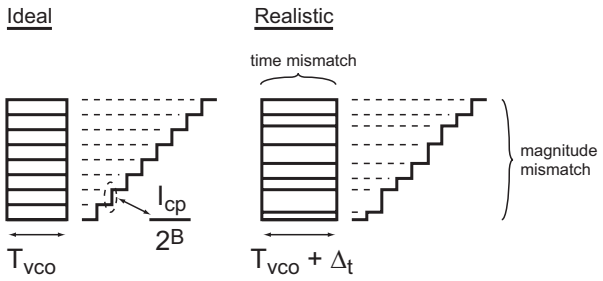


Figure 4-20 Non-ideal Charge-box

when compared to a PFD/DAC that is not noise shaped. An extra unit element is used to produce this extra level. So, while a non noise shaped PFD/DAC employs $2^B - 1$ unit elements, a noise shaped PFD/DAC utilizes 2^B unit elements.

4.2.2 Non-idealities Within the Charge-box

The PFD/DAC architecture offers the advantage over a classical PI synthesizer in that the systematic mismatch between the fractional-N quantization error and the DAC cancellation signal is eliminated. However, mismatch internal to the PFD/DAC structure proposed in [21] can lead to a gain mismatch between these two signals, and therefore incomplete quantization noise cancellation. Figure 4-20 compares a

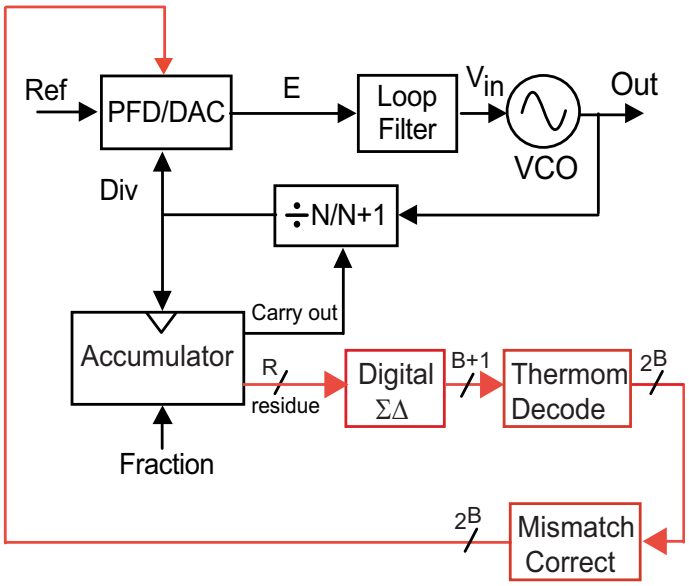


Figure 4-21 Dynamic Element Matching to Reduce Unit Element Mismatch

charge-box created by an ideal PFD/DAC with one created by realistic circuitry. As the figure clearly shows, any mismatch between levels in the DAC, or any timing mismatch between the two divider phases that create the charge-box will lead to errors in cancellation. In sections 4.2.3 and 4.2.4 we propose techniques to transform the gain error resulting from charge-box non-ideality into broadband noise sources. Gain errors result in large fractional spurs, whereas broadband noise sources are filtered by the synthesizer dynamics, and may be inconsequential if properly managed.

4.2.3 PFD/DAC Unit Element Mismatch and Compensation

Mismatch between the unit elements comprising the PFD/DAC will result in incomplete cancellation of the quantization noise. To minimize the impact of this mismatch, we propose the architecture shown in Figure 4-21. The unit elements are dynamically matched by selecting different combinations of unit elements at different times to generate a desired output level. In this way, the mismatch between the unit elements is averaged out, and the mismatch is converted from a gain error into a broadband, shaped noise source [36]. To accomplish this, the 2^B outputs of the thermometer decoder are processed by a mismatch shaping block. Of the different dynamic element

matching algorithms, we choose data weight averaging (DWA), because it produces a shaped mismatch noise profile [36, 40, 41].

Impact of the DAC mismatch noise on overall performance is best observed through detailed behavioral simulations. Deriving an analytical expression for the dynamic element matching DWA technique is beyond the scope of this thesis. In [36] some discussion into the topic of DWA noise shaping is presented, but with the conclusion that the dynamic matching process is not well defined in analytical terms. This is due to the fact that unit element mismatch is a single distribution that is sampled by the noise shaped digital $\Sigma\Delta$ modulator controlling the DAC. Dependence on the particular output code sequence of the digital $\Sigma\Delta$ modulator as related to its input code complicates the analysis process. For this reason, we focus on behavioral simulation results to examine the impact of unit element mismatch noise on overall performance.

4.2.4 PFD/DAC Internal Timing Mismatch and Compensation

Timing mismatch between the two divider phases Φ_0 and Φ_1 in Figure 4-2 result in a non-ideal charge-box as depicted in Figure 4-20. To mitigate this error, we propose a timing mismatch compensation technique, included in the mismatch compensated PFD/DAC depicted in Figure 4-22. The unit element dynamic element matching is included in the figure as the DAC mismatch shaping block for completeness.

A precise one-VCO period between the two divider phases Div0 and Div1 must exist to properly establish the charge-box used to cancel the quantization noise. Mismatch in physical layout, loading, and device gradients results in a propagation delay difference between the signal paths for Div0 and Div1. The value of time resolved by the charge-box Figure 4-20 therefore will equal $T_{vco} + \Delta_t$ in practice. This timing mismatch results in a gain error, incomplete quantization noise cancellation, and fractional spur feed-through.

We can apply dynamic element matching techniques to correct for timing mismatch. As shown in Figure 4-23, use of re-timing flip-flops limits the skew between

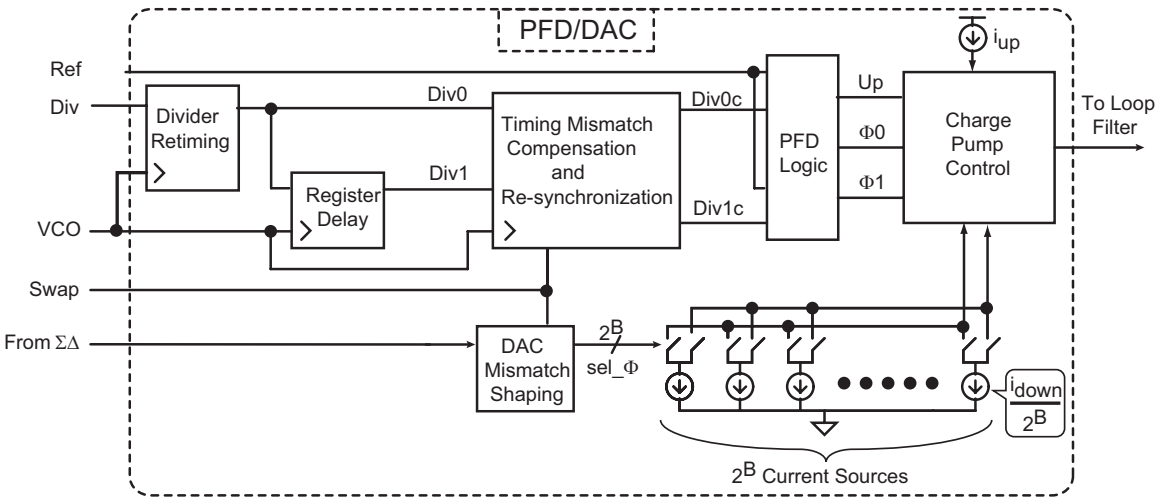


Figure 4-22 Mismatch Compensated PFD/DAC

the two phase paths to differences between the flip-flop clk-to-q times and PFD circuit paths. The muxes are toggled by a phase swap signal so that the two phase paths see each PFD, on average, the same amount of time. The current steering control bits from the $\Sigma\Delta$ modulator are selectively inverted to maintain correct functionality.

The consequences of introducing the phase swapping process can most easily be understood via a straightforward time analysis. From Figure 4-23, we see that the re-timing flip-flops eliminate the differential timing error Δ_{t1}. We lump all of remaining mismatch between the two paths into a variable, Δ_{t2}, which is referenced to the output of one of the flip-flops. In the example shown, Δ_{t2} is referenced to the upper flop. Following through the time evolution of the phase paths, we see that the phase path Div0 experiences an average delay of:

$$t_{del_{Div0}} = (1 - D) \cdot 0 + D \cdot \Delta_{t2} \tag{4.8}$$

while the phase path Div1 encounters:

$$t_{del_{Div1}} = (1 - D) \cdot \Delta_{t2} + D \cdot 0 \tag{4.9}$$

where D is the duty cycle of the swap control. Clearly, if the duty cycle is set to be 0.5, each path will see the same average delay, and the timing mismatch is eliminated.

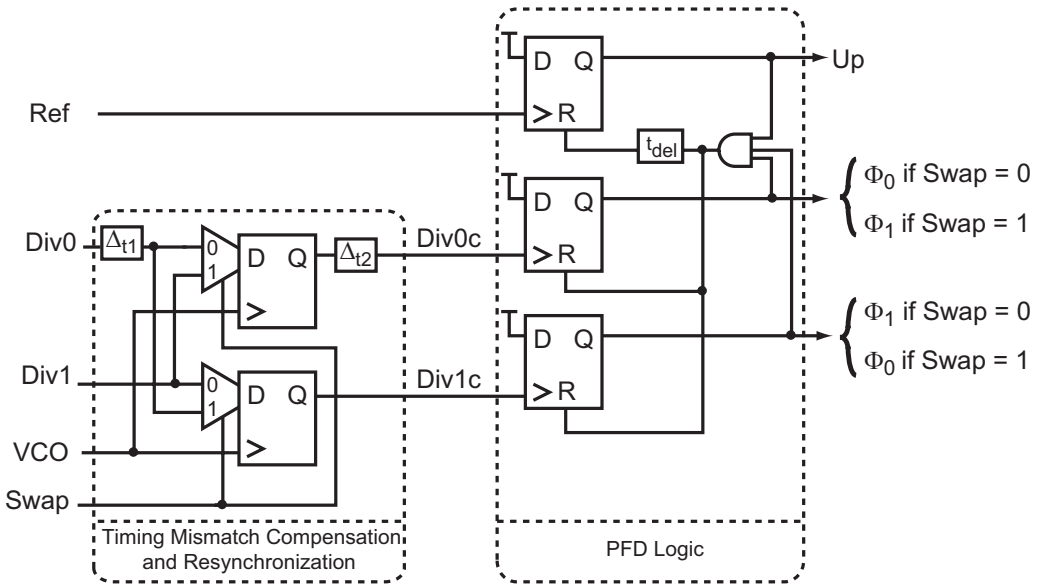


Figure 4-23 Dynamic Phase Matching Block

We denote the output phase path associated with Div0 as Φ_0 and the output phase path associated with Div1 as Φ_1. Because the swap path dynamically alters which of the two output flip-flops are associated with Div0 or Div1, Φ_0 and $\Phi 1$ also vary, as denoted in the figure.

Two constraints are placed on the characteristics of the phase swapping control signal. First, it must have an average value very near 0.5 to ensure that both phase signals see the same average propagation times. Second, the swap signal must contain little or no spurious energy. For this reason, two immediate possibilities arise for control of the swapping operation: a pseudo random linear feedback shift register (LFSR) or a single bit output $\Sigma\Delta$ modulator with a sufficient order to ensure that the output spectrum is random. We have found through simulation that the LFSR is the better solution. It may also be possible to control the swap signal in a manner that will result in a shaped noise profile. Noise shaped phase swapping control is left as a topic for future work.

Figure 4-24 presents the noise model for the PFD/DAC synthesizer employing phase swapping. As the synthesizer progresses through different instantaneous phase errors, the divider edge location, and therefore the location of the charge-box, is changing in time. The noise power introduced by the time mismatch is therefore

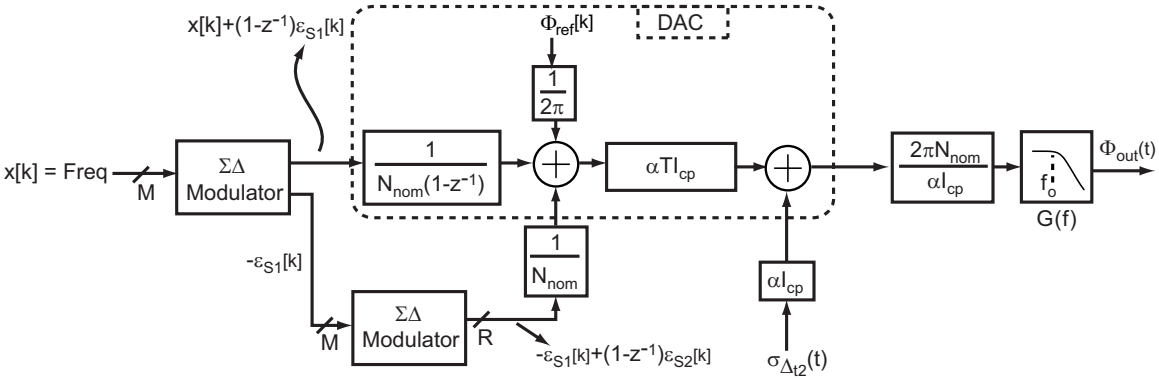


Figure 4-24 Model with Timing Mismatch

assumed to have a uniform distribution in time, and a variance $\sigma^2_{\Delta_t} = \Delta^2_{t2}/12$. As Figure 4-24 shows, the output noise due to timing mismatch after the swapping operation is

$$S_{\Phi_{out}|\Delta_t} = \frac{1}{T} \cdot \frac{\Delta^2_{t2}}{12} \cdot (2\pi N_{nom})^2 \cdot |G(f)|^2 \tag{4.10}$$

where the $1/T$ factor is present because a discrete time noise process is being filtered by a continuous time filter [1]. Equation 4.10 allows us to analytically determine how much timing mismatch can be tolerated to achieve a desired output phase noise performance.

4.2.5 Shape Mismatch Between the Error Signal and Cancellation Signal

The final mismatch source to be compensated in the PFD/DAC synthesizer is the shape mismatch between the cancellation signal and phase error signal. As discussed in section 3.2.1, the phase error is a vertically resolved signal (in phase), while the cancellation signal is horizontally resolved signal (in magnitude). Therefore, a frequency dependent gain error will result. In this section we propose two possible solutions to this problem.

A first approach is to utilize the fact that we can predict the error value, as derived in equation 3.13. Repeated here for convenience, equation 3.13 states that the shape

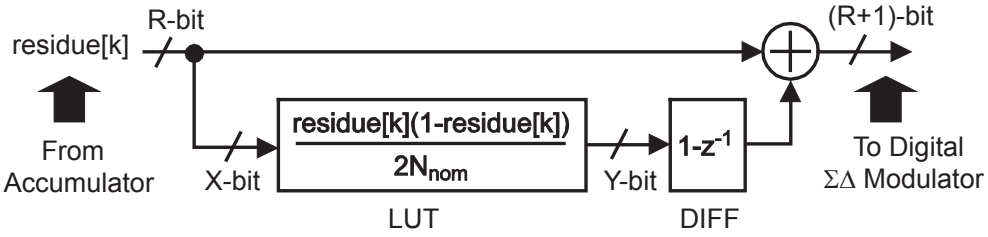


Figure 4-25 Digital Gain Compensation Block

mismatch error charge present at the charge-pump output will be:

$$Q_{err}(j\omega, k) \approx -\frac{j\omega}{2}\epsilon[k](1-\epsilon[k]) \tag{4.11}$$

where both the VCO period and charge-pump current have been normalized to one, and a Taylor series approximation has been used to simplify the result.

Having arrived at a simple closed form approximation for the expected error, we can build a digital gain compensation block to correct for it. Figure 4-25 depicts the implementation of the gain compensation block. Some portion of the accumulator residue bits are used to address a look up table (LUT), which may be implemented as read-only memory (ROM) or random-access memory (RAM). The output of the LUT is differentiated and summed with the residue. This is then sent to the digital $\Sigma\Delta$ modulator controlling the PFD/DAC.

There are two points to note about the digital gain compensation block. The first point is that, in practice, the LUT has finite input and output resolution. In Chapter 5, the impact of changing the values of X and Y, the LUT input and output resolutions, respectively, will be presented. It will be shown that only 1Kbit of ROM (with R=20, X=6, and Y=4) is required to achieve an 18dB improvement in fractional spur rejection. This additional rejection, coupled with the improved gain match due to the PFD/DAC, achieves overall fractional spur levels of <-90dBc in detailed behavioral level simulations.

The second point relates to the variable N_{nom} used to calculate the compensated accumulator output. The factor of $1/N_{nom}$, where N_{nom} is the nominal divide value, stems from the fact that the compensator uses a discrete time differentiator that is

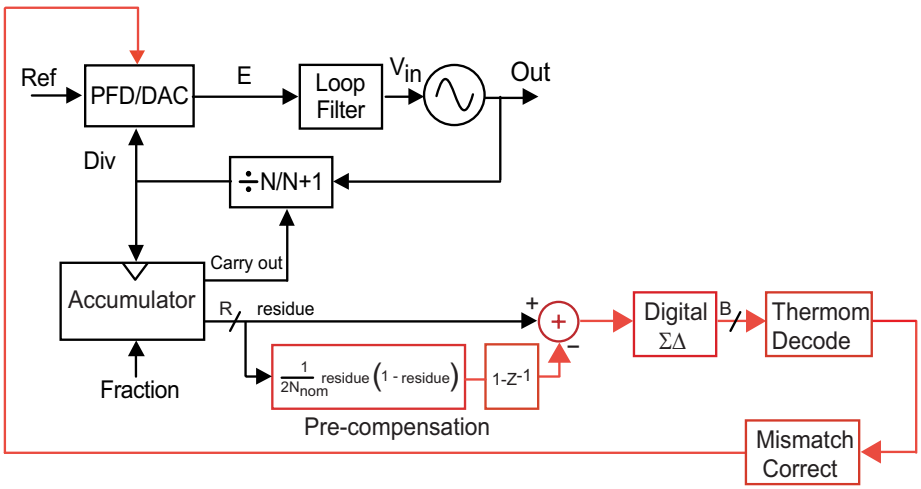


Figure 4-26 Digitally Compensated PFD/DAC Synthesizer

clocked at the reference (or divider) frequency whereas the phase error is referenced to the VCO period. Ideally, this re-normalization factor needs to vary with the instantaneous divide value, requiring a full digital divider. Simulations have shown that using a static $1/N_{nom}$ value yields good results. As long as the approximation error is less than the desired level of compensation, the approximation is acceptable. For a 20dB improvement, the VCO frequency can change by $\pm 5\%$ from the nominal value while maintaining a valid approximation. If the synthesizer is employed in a system with requirements exceeding this range, multiple LUTs can be employed to keep the approximation error acceptable, or a full digital divider can be implemented in the compensation block. Since a 1Kbit LUT occupies very little on-chip area, multiple LUTs appears to be the best solution. The compact nature of memory in modern processes coupled with the small memory size required to achieve high levels of compensation translates into a very low area penalty for a large degree of design flexibility. Figure 4-26 depicts the PFD/DAC synthesizer with digital compensation for shape mismatch included.

A second technique to combat shape mismatch is to perform a sample-and-hold operation before the VCO input. The simplest way to explain the impact of using a sample-and-hold is presented in Figure 4-27, where we revisit the charge-box.

Typically, the loop filter between the charge-pump and VCO will be of the form

$$H(s) = \frac{t_z s + 1}{Cs(t_p s + 1)} \quad , \tag{4.12}$$

where t_z is the filter zero time constant, t_p the filter pole time constant, and C is a capacitance that establishes the filter gain. H(s) can be thought of as an integrator followed by a lead-lag filter. In Figure 4-27 we plot the integrator output waveform in addition to the charge-box output.

The PFD/DAC technique ensures that the net negative charge balances the positive charge by controlling the magnitude of current in a one VCO period window. This means that the slope of the integrator varies with PFD/DAC output current during the time of the charge-box. What is most important to recognize is that the *net* voltage excursion is zero. If an ideal sample-and-hold is used to sample the integrator output after the PFD/DAC has delivered all of its charge, the VCO will never see a change in voltage and *all* spurs will be eliminated (this includes any reference spur, which is normally present in any PLL due to the phase detection operation). In Figure 4-27, *Samp* is a control signal that causes the integrated charge to be sampled. The sampled-and-held output is then used to control the VCO. The net voltage excursion experienced by the loop filter is zero, and the sample operation is done *after* the loop filter voltage returns to its starting value, therefore the sample-and-held voltage is always at a constant value (V_c in the figure). Figure 4-28 depicts the modified PFD/DAC synthesizer to include a S/H.

4.3 Summary

In this chapter, we have proposed a new synthesizer architecture, the mismatch compensated PFD/DAC synthesizer, that is capable of dramatically reducing the impact of fractional-N quantization noise on overall synthesizer phase noise performance. The architecture has been derived in a progressive manner to emphasize its ability to address the non-idealities that limit the ability of prior work to cancel quantization

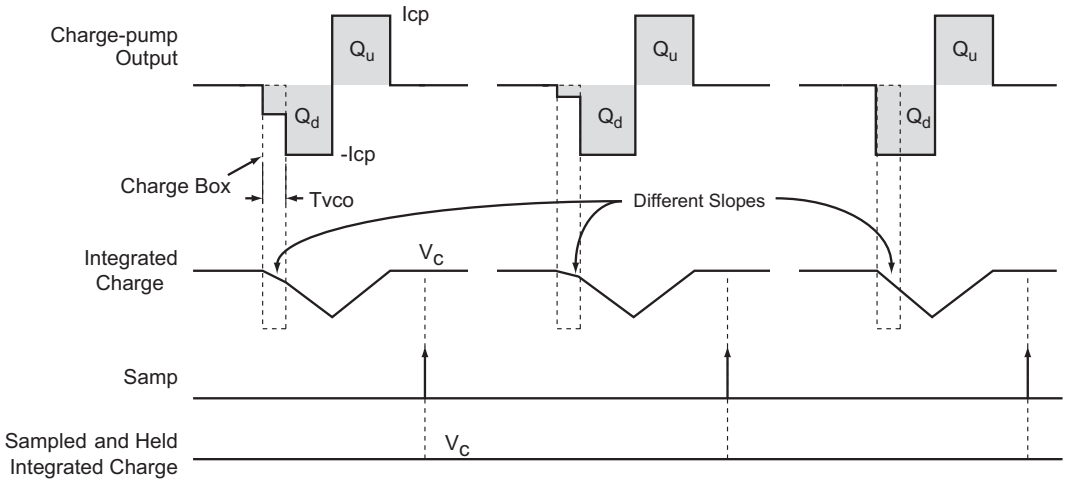


Figure 4-27 Using a Sample-and-Hold to Eliminate Shape Mismatch

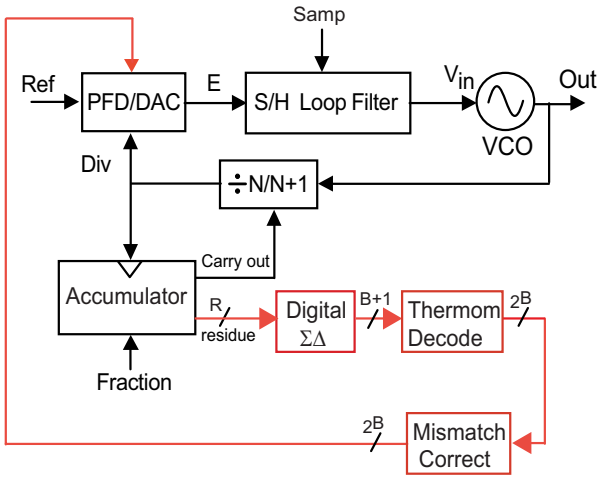


Figure 4-28 PFD/DAC Synthesizer with Sample-and-Hold to Eliminate Shape Mismatch

noise.

Systematic non-idealities associated with the PFD/DAC approach, such as magnitude and timing mismatch in the charge-box, have been included in the development of the model. Finally, we have proposed two techniques to overcome the shape mismatch that results from the PFD/DAC approach. The first is a simple digital pre-compensation scheme, while the second is the analog approach of implementing a sample-and-hold. In Chapter 5 we will present behavioral modeling techniques and simulation results of the proposed mismatch compensated PFD/DAC architecture.

Chapter 5

Behavioral Simulation of Fractional-N Synthesizers

In this chapter, we propose techniques that can be used to construct detailed behavioral simulations of fractional-N frequency synthesizers. Using CppSim, a C++ behavioral simulator that employs a simulation methodology explained in [23] and that is available at [42], we build an evolving model of the proposed mismatch compensated PFD/DAC synthesizer. As SPICE level simulations reveal non-idealities in individual circuit blocks, we update the model to include these non-idealities and examine their influence on synthesizer performance. More importantly, the behavioral model allows us to quickly evaluate architectural changes that help alleviate the impact of these non-idealities and therefore eases the circuit design process.

First we will develop a base model for the synthesizer using ideal circuit blocks, and then add non-idealities as the model progresses. As a target for performance, we desire a 1MHz bandwidth synthesizer that exhibits $< -150dBc/Hz$ phase noise at a 20MHz offset, and <-100dBc/Hz in-band phase noise for a 3.6GHz carrier. This performance is equivalent to -162dBc/Hz noise @ 20MHz and -112dBc/Hz in-band for a 900MHz carrier and would meet the stringent GSM transmit spectral mask. A 3.6GHz output frequency is chosen so that the quantization noise step-size, $\alpha T_{vco} I_{cp}$, is referenced to a very small T_{vco}. To generate dual band output carriers (1.8GHZ or 900MHz), the 3.6GHz output is simply divided by two or four.

5.1 Setting the PLL Dynamics and Preliminary Noise Analysis Using the PLL Design Assistant

We begin design of the PFD/DAC synthesizer by setting the PLL dynamics and performing a preliminary noise analysis. We use the PLL Design Assistant (PDA) tool described in [39]. Our desired specifications are:

- 1MHz closed loop synthesizer bandwidth
- 3.6GHz output frequency
- -150dBc/Hz phase noise at a 20MHz offset from the carrier
- <-100dBc/Hz in-band noise
- Minimal residual spurs present in the output spectrum. For practical purposes, we will aim for spurs no larger than -80dBc/Hz

Figure 5-1 shows the PDA interface. A closed loop bandwidth of 1MHz is desired with a type II PLL (meaning there are two integrators in the PLL loop) and a Butterworth filter response. A 50MHz reference frequency is used to generate an output frequency of 3.6GHz. Phase detector noise, which represents the sum of charge-pump noise, reference jitter, and divider jitter is set to -110dBc/Hz. The VCO noise is entered as -154dBc/Hz at a 20MHz offset. It should be noted that both the detector noise specification of -110dBc/Hz and the VCO specification represent state-of-the art performance. Circuit techniques to achieve these specifications will be discussed in chapter 6.

As a starting point for comparison, we assume a state-of-the-art 2^{nd} order MASH $\Sigma\Delta$ synthesizer. The quantization noise shaping transfer function for a 2^{nd} order synthesizer is

$$(1 - 2z^{-1} + z^{-2})\Delta \tag{5.1}$$

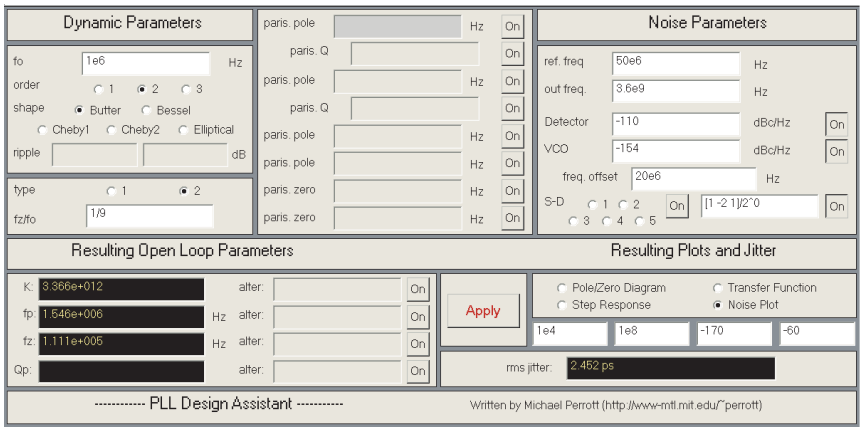


Figure 5-1 PLL Design Assistant (PDA) Design Parameters for 2^{nd} Order $\Sigma\Delta$ Synthesizer

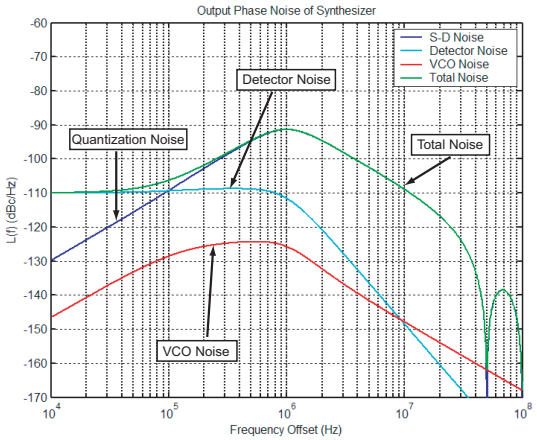


Figure 5-2 PLL Design Assistant Calculated Phase Noise for 2^{nd} Order $\Sigma\Delta$ Synthesizer

where $\Delta = 1$ is the quantization step-size for s classical 2^{nd} order $\Sigma\Delta$ synthesizer, for reasons discussed in chapter 3. The quantization noise is entered into the PDA parameter window as

$$(1 - 2z^{-1} + z^{-2})/2^0. \tag{5.2}$$

The scale factor is present so that we can examine the effect of lowering the quantization step-size using the proposed PFD/DAC approach. Figure 5-1 shows that the PDA returns required values of the loop filter poles and zero, as well as necessary loop gain to achieve the desired bandwidth and filter response.

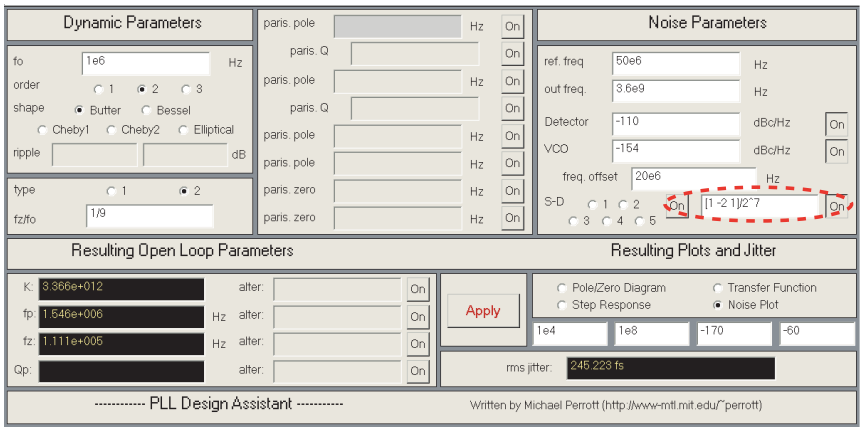


Figure 5-3 PLL Design Assistant (PDA) Design Parameters for 7-bit PFD/DAC Synthesizer

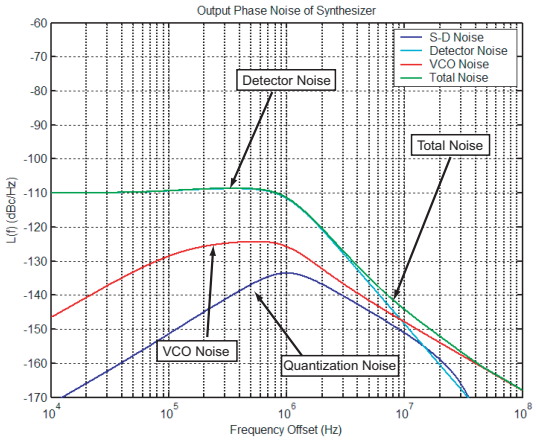


Figure 5-4 PLL Design Assistant (PDA) Calculated Phase Noise for 7-bit PFD/DAC Synthesizer

Figure 5-2 presents the phase noise response of the synthesizer given the parameters specified in Figure 5-1. We see that the quantization noise dominates over a wide frequency range, and the total noise in the system exceeds the desired specification. One possibility to attenuate the quantization noise is to lower the PLL bandwidth, but the 1MHz bandwidth specification would then be violated. This highlights the noise-bandwidth tradeoff associated with $\Sigma\Delta$ fractional-N synthesis.

Now we analyze a PFD/DAC synthesizer that reduces the quantization step-size by a factor of 2^B, where B is the number of bits in the PFD/DAC. Figure 5-3 shows the PDA parameter window set up for a 7-bit PFD/DAC. Figure 5-4 shows that

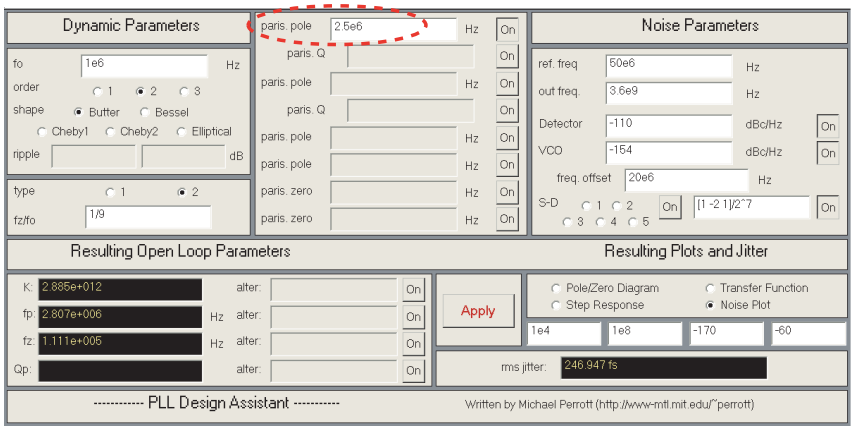


Figure 5-5 PLL Design Assistant (PDA) Design Parameters for 7-bit PFD/DAC Synthesizer With Added Pole

the quantization noise is now reduced to the point where it does not dominate the phase noise profile over *any* bandwidth. By using the PFD/DAC architecture, the synthesizer meets all of the desired specifications! We do note that there is not much margin at 20MHz, where the -150dBc/Hz specification is exceeded only by a couple of dB. To gain margin, we add a pole at 2.5MHz to the loop dynamics. Figures 5-5 and 5-6 show that margin is added by using the additional pole. The pole will also have the effect of attenuating the reference spur that will appear in the output spectrum at 50MHz by an additional 26dB, reinforcing our desire to use the additional pole. Note that the required loop filter poles, zero, and loop gain have all changed to accommodate the addition of the 2.5MHz pole. Having arrived at a set of parameters which achieve the desired specifications, we build a behavioral model using CppSim.

5.2 PFD/DAC Synthesizer Base Behavioral Model

The starting system used for behavioral simulation is presented in Figure 5-7. Key blocks in the model are highlighted in the figure. In order to allow flexibility in testing, the starting model includes the capability to step the divider input to examine synthesizer step response. Additional features of the base model are :

- The PFD/DAC structure, which has been coded so that it has programmable

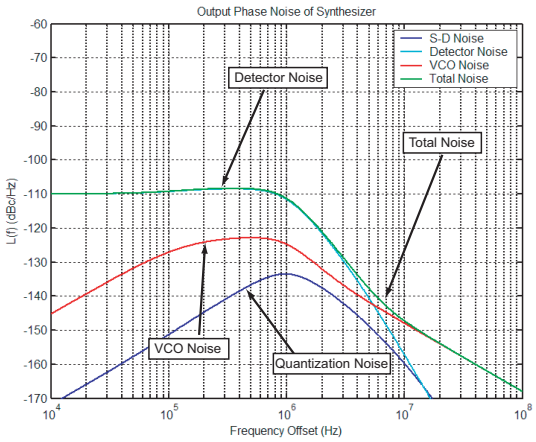


Figure 5-6 PLL Design Assistant (PDA) Calculated Phase Noise for 7-bit PFD/DAC Synthesizer With Added Pole

resolution. To explore mismatch issues, the unit elements may be mismatched by turning on a mismatch enable switch, and the standard deviation of the mismatch may be specified (a Gaussian mismatch profile is assumed). Also, phase swapping between the two phase paths may be enabled or disabled. The swapping is controlled by a 23-bit LFSR random number generator, and the magnitude of the residual timing error can be specified.

- A sample-and-hold can be enabled or disabled to examine the effect of shape mismatch between the error signal and cancellation pulse that was discussed in section 4.2.5.

- The loop filter can be adjusted according to the parameters from the PLL Design Assistant.

- The 2.5MHz pole added to improve noise margin is included.

- The VCO phase noise at a given offset may be specified. This is useful to both examine the effect of VCO noise by entering the same values as used by the PLL Design Assistant, or to lower the noise so much that it is effectively "off". In general, different noise sources can be isolated from each other during the simulation process.

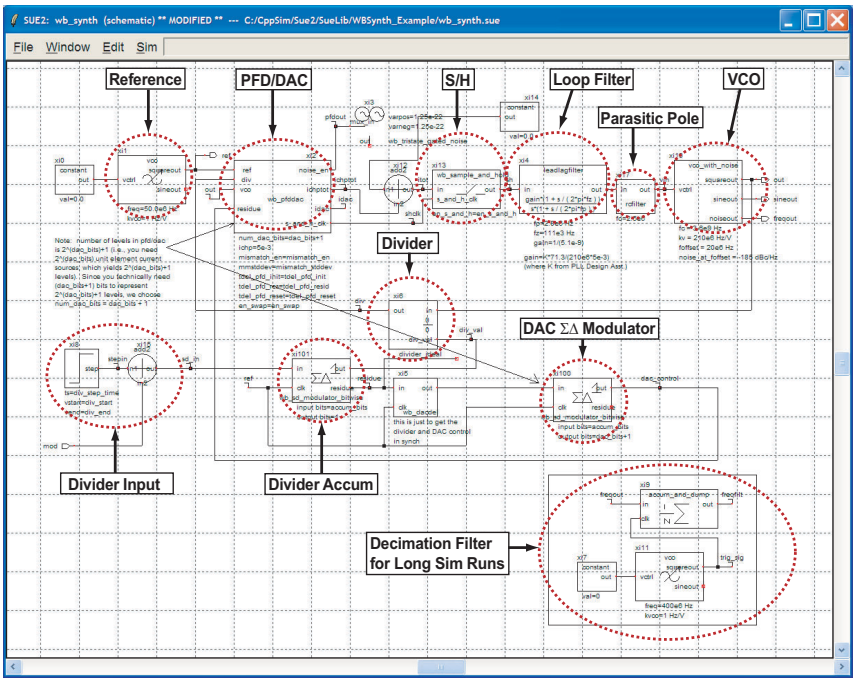


Figure 5-7 Behavioral Model for PFD/DAC Synthesizer

- Detector Noise is also modeled. In the behavioral simulations to follow, we assume that charge-pump noise dominates the detector noise. As with VCO noise, the magnitude of this noise can be "turned off" by entering a zero for its value.

- A filter and decimator for viewing the output of long simulations is included. This analysis technique is valuable when a long simulation is done to explore modulation schemes or very low frequency noise performance.

5.2.1 Loop Filter and Loop Gain Calculation

The PDA returns the values of $K = 2.885e12$, $fp = 2.807MHz$, and $fz = 111.1kHz$. These values assume a lead-lag loop filter, as described in [39], and depicted in Figure 5-8. A charge-pump and VCO are included in the figure to give context for the loop filter. In the behavioral model, fp is used for the filter pole, fz for the filter zero,

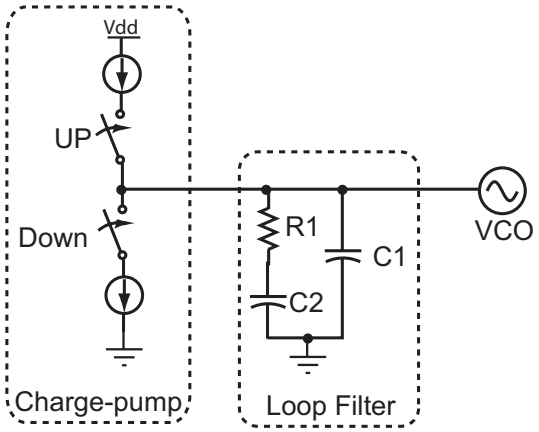


Figure 5-8 Lead-lag Loop Filter Configuration

and K represents the *total* loop gain, where:

$$K = \frac{\alpha I_{cp} K_v}{C_{tot} N_{nom}} \tag{5.3}$$

and K_v is the VCO gain, I_{cp} is the charge-pump current, α is the PFD gain, C_{tot} is the total loop filter capacitance ($C_{tot} = C_1 + C_2$), and N_{nom} is the nominal (average) divide value. Solving for C_{tot}, we find:

$$C_{tot} = \frac{\alpha I_{cp} K_v}{K N_{nom}} \quad . \tag{5.4}$$

Values for the parameters in equation 5.4 are chosen based on data sheet specifications (for the VCO), desired output frequency (for N_{nom}), PFD topology ($\alpha = 1$ for a tri-state PFD), and circuit Hspice simulation (for I_{cp}). The nominal set of values that will be used in the behavioral simulations presented in this chapter are summarized in table 5.1.

5.2.2 Baseline Noise Calculations

Before exploring non-idealities such as magnitude and timing mismatch in the PFD/DAC, we perform several baseline simulations to validate the analytical mode proposed in [1] and implemented by the PLL Design Assistant to predict phase noise performance. To add in detector and VCO noise, we use the noise model presented in [1].

Variable	Nominal Value	Comment
N_{nom}	71.31307	Output frequency = 3.565GHz
I_{cp}	5mA	
t_{on}	3ns	Steady-state on-time of positive and negative charge-pump current (Total charge-pump on-time = $2t_{on}$)
K_{vco}	210MHz/V	From ZComm VCO Datasheet
C_{tot}	5.1nF	Determined from equation 5.4
α	1	Tri-state PFD used [1]

Table 5.1 Nominal Parameter Values for Behavioral Simulations

5.2.3 Detector Phase Noise Calculation

The output phase noise due to charge-pump noise is:

$$S_{\Phi_{out}|\overline{i_n^2}}(f) = \overline{i_n^2}\left(\frac{2\pi N_{nom}}{I_{cp}}\right)^2 |G(f)|^2 \tag{5.5}$$

where G(f) represents the low-pass filter function of the closed loop PLL dynamics discussed in chapter 3. To calculate the required noise magnitude to get -110dBc/Hz low frequency phase noise we make three assumptions. The first is that, at frequencies below the loop bandwidth, $G(f) \approx 1$. The second is the assumption that the detector noise will be dominated by charge-pump noise, and that the reference jitter and re-timed divider jitter are insignificant. Our final assumption is that the charge-pump noise magnitudes are approximately equal for positive and negative current sources. This assumption is not generally true in practice. However, in the prototype synthesizer IC that will be presented in Chapters 6 and 7, the current sources were designed such that the positive and negative current source noise magnitudes will be approximately equal (based on SPICE simulation), validating the last assumption. Detector noise is calculated using equation 5.5, with one modification.

Since the charge-pump is not always on, its noise power is not always present at the loop filter. Finite charge-pump on-time is depicted in Figure 5-9. The on-time of the up current is much greater than the charge-box time, T_{vco}. For example, for the synthesizer we will examine, $T_{vco} = 277ps$ and $t_{on} = 3ns$, so there is more than

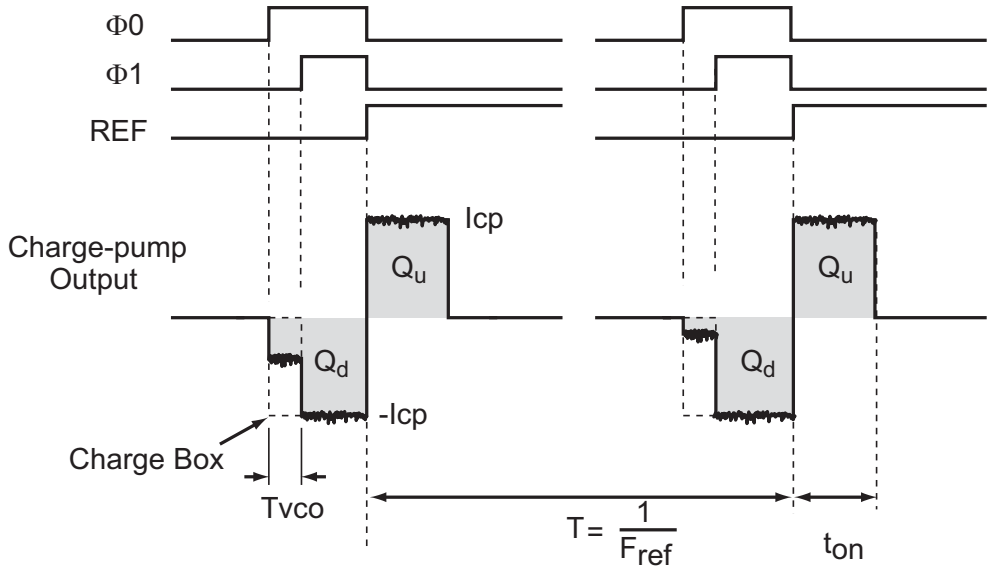


Figure 5-9 Charge Pump With Noisy Currents

a 10X difference between them. We can therefore approximate that t_{on} is the same for positive and negative currents in the figure. If we make the final assumption that the positive current sources and negative current sources are designed to have the same output current noise, we can modify equation 5.5 to account for finite on-time as follows:

$$S_{\Phi_{out}|\overline{i_n^2}}(f) = \overline{i_n^2}\left(\frac{2\pi N_{nom}}{I_{cp}}\right)^2 |G(f)|^2 \cdot D \quad , \tag{5.6}$$

where D is the duty cycle of the current on-time. For the case where the up and down current noises are the same, and the up and down current pulse on-times are approximately the same, D is expressed as

$$D = \frac{2t_{on}}{T} \quad , \tag{5.7}$$

where T is the reference clock period.

Solving equation 5.6 for $\overline{i_n^2}$ we find:

$$\overline{i_n^2} = \frac{1}{D}\left(\frac{I_{cp}}{2\pi N_{nom}}\right)^2 \cdot 10^{\frac{\text{spec in dBc/Hz}}{10}} \quad , \tag{5.8}$$

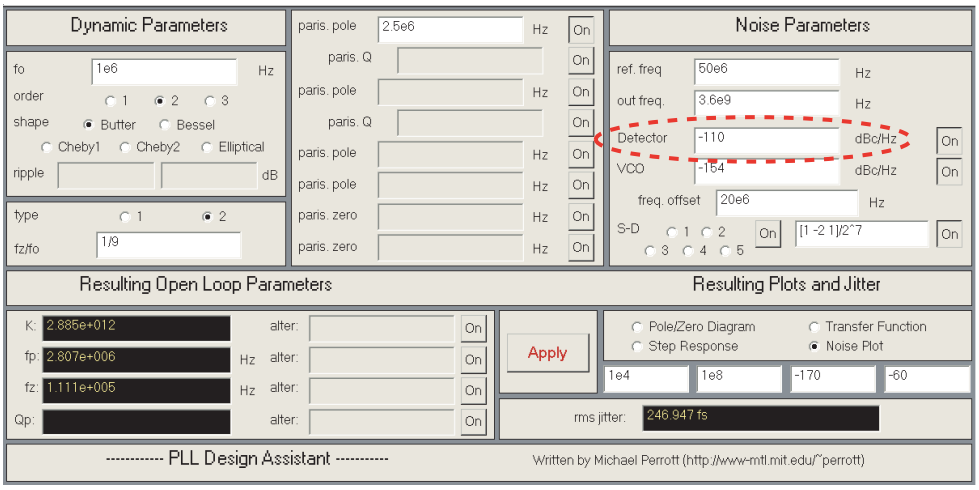


Figure 5-10 Detector Noise Inclusion in the PLL Design Assistant

which, for our parameter set as specified in Table 5.1, is

$$\overline{i_n^2} = \frac{20ns}{6ns}\left(\frac{5mA}{2\pi 71.31307}\right)^2 \cdot 10^{\frac{-110}{10}} = 4.151e^{-20}A^2/Hz \tag{5.9}$$

Detector noise is incorporated into the analytical model as shown in Figure 5-10. SPICE level simulations are run to verify that this noise specification is achievable. Circuit design of the unit elements that comprise the PFD/DAC will be presented in Chapter 6.

5.2.4 VCO Phase Noise Calculation

Calculation of VCO noise is also done based on the model in [1].

$$S_{\Phi_{out}|_{vco}}(f) = S_{\Phi_{vco}} \cdot |(1 - G(f))|^2 \tag{5.10}$$

Equation 5.10 shows that the VCO phase noise is simply high pass filtered by the loop dynamics. To incorporate the VCO noise into our model, we make use of the analysis presented in Figure 5-11.

The input to the VCO is a voltage that changes the VCO output *frequency*. The PLL operates on VCO *phase*, so the VCO appears as an integrator from the standpoint of the PLL loop. At the offset frequencies where VCO noise becomes a

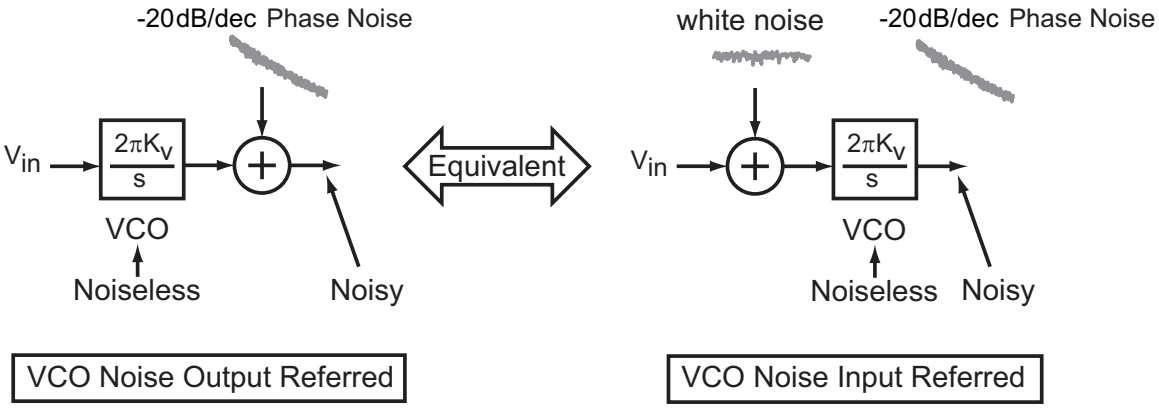


Figure 5-11 VCO Noise Modeling

concern in synthesizer design, the VCO has a phase noise characteristic that rolls off at -20dB/decade [33]. One possible way to include VCO noise is shown in the left of Figure 5-11. Output referring the phase noise means that a time varying delay must be introduced to the behavioral model to make the VCO period change in a way that appears to have a single-pole slope. While this is possible, an easier solution is to input-refer the VCO noise.

Since the VCO is an integrator in phase, the input referred power spectral density (PSD) of the VCO noise appears white, and a simple random number generator with the proper gain can be used to represent the noise.

The equation to solve to determine model parameters is a simple matching of phase noise at a given offset to the output VCO noise. The input referred noise PSD is due to a noise voltage, $\overline{v_{vcon}^2}$, and is described by

$$10log\left(\left(\frac{2\pi K_v}{2\pi f_{offset}}\right)^2 \overline{v_{vcon}^2}\right) = \text{VCO noise spec in dBc/Hz} \quad , \tag{5.11}$$

where f_{offset} is the offset frequency at which the noise is specified.

Equation 5.11 can easily be solved for $\overline{v_{vcon}^2}$. The VCO block in the behavioral simulation does this internally, so all that must be entered into the block is a frequency offset and phase noise specification at that offset. Table 5.2 summarizes the noise assumptions used for the baseline simulations. The value for VCO phase noise was obtained from the data sheet for the discrete VCO used in the prototype PFD/DAC

Variable	Nominal Value	Comment
t_{on}	3ns	Nominal charge-pump current pulse-width
$\overline{i_n^2}$	2.989e-20A^2/Hz	Total Output Referred Detector Noise Variance
f_{offset}	20MHz	Offset frequency for VCO noise specification
$S_{\Phi_{vcon}}(f)$	-154dBc/Hz	VCO Phase Noise at f_{offset}

Table 5.2 Nominal Noise Parameter Values for Behavioral Simulations

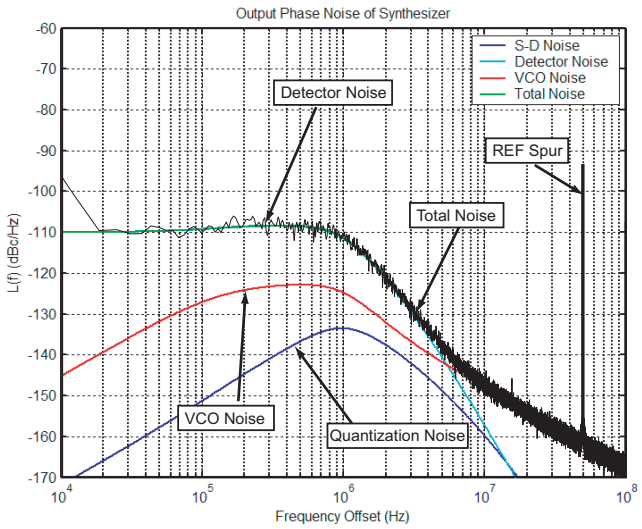


Figure 5-12 Baseline Behavioral Simulation Comparison With PLL Design Assistant Calculations

synthesizer that is presented in Chapter 7. This represents excellent VCO phase noise performance.

If the 3.6GHz VCO output is noiselessly divided down to 900MHz, the resulting -166dBc/Hz noise level at 20MHz offset betters the GSM requirement of -162dBc/Hz by 4dB. The 12dB difference between the noise specification referenced to 3.6GHz versus 900MHz is due to the fact that phase noise scales inversely with the period it is referenced to [33]. A 900MHz signal has a period four times longer than a 3.6GHz signal, and therefore, for the same system noise source magnitudes, exhibits 12dB lower phase noise in dBc/Hz.

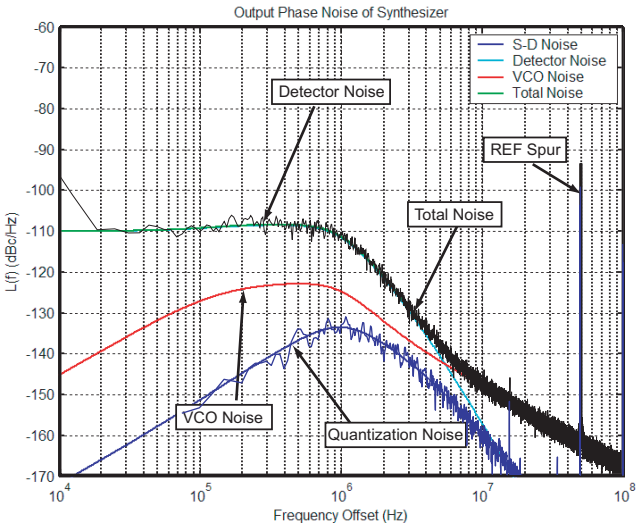


Figure 5-13 Separate CppSim Simulations of VCO, Detector and Quantization Noise, and Only Quantization Noise Overlayed with PDA Calculations

5.2.5 Baseline Phase Noise Simulation

Figure 5-12 presents results of the initial baseline behavioral simulation. The output of the CppSim simulation run is processed using Matlab and a phase noise power spectral density calculation script. The results of the CppSim simulation is overlayed on the PDA calculations. There is excellent agreement between the calculated noise response and simulated result, indicating that the noise magnitudes and parametric values of tables 4.1 and 4.2 are correct. The slight discrepancy at very low frequency can be attributed to a very small DC component left in the PSD calculation of the behavioral simulation results. Also note the reference spur at 50MHz, which is not included in the analytical PDA calculations. This excellent agreement between the independent analytical and behavioral models indicates the validity of both in describing the system.

In order to demonstrate the versatility of the behavioral modeling approach, a simulation is run with all noise sources turned off, so that the impact of quantization noise on the output phase noise spectrum can be observed. As Figure 5-13 clearly shows, there is also excellent agreement between the PDA calculated quantization noise impact on the output, and the CppSim simulated result. In the figure, the

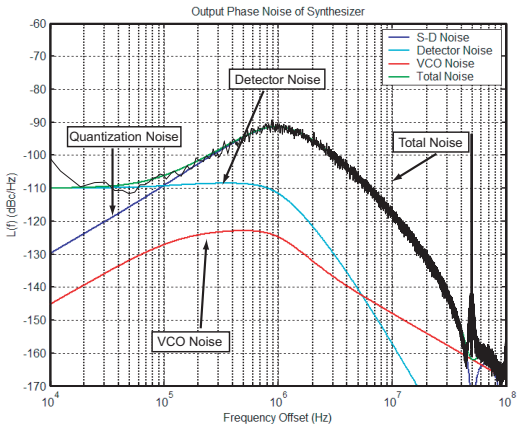


Figure 5-14 Baseline Simulation for a 2^{nd} order $\Sigma\Delta$ Synthesizer

results of both the first baseline CppSim simulation as well as the simulation with VCO noise and detector noise off are overlayed with the PDA calculations.

As a final cross-check between the analytical model and the behavioral model, a classical 2^{nd} order $\Sigma\Delta$ synthesizer is simulated to verify the PDA calculations of its noise performance. We see in Figure 5-14 that there is also excellent agreement between CppSim and the PDA for this synthesizer architecture.

Figure 5-15 presents an overlay of the simulated phase noise profiles of both the classical $\Sigma\Delta$ synthesizer, as well as the PFD/DAC synthesizer. The right plot shows the relative attenuation achieved by using the PFD/DAC approach. For the configuration chosen in the simulations, $> 32dB$ noise suppression is achieved. In terms of rms jitter, which corresponds to the total integrated phase noise, the $\Sigma\Delta$ synthesizer exhibits 2.478ps while the PFD/DAC synthesizer only 374fs, an improvement of more than 6X in rms jitter!

5.2.6 Baseline Dynamic Performance

A good way to evaluate the dynamic performance of the synthesizer is to examine the step response. For this reason, the PDA calculated step response and CppSim

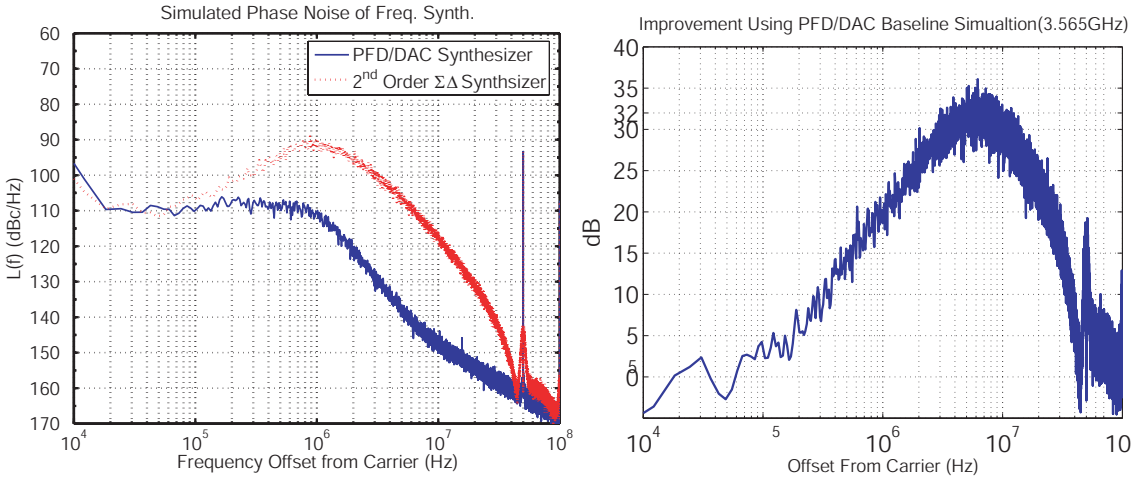


Figure 5-15 Baseline Improvement Using the PFD/DAC vs a 2^{nd} Order $\Sigma\Delta$ Synthesizer

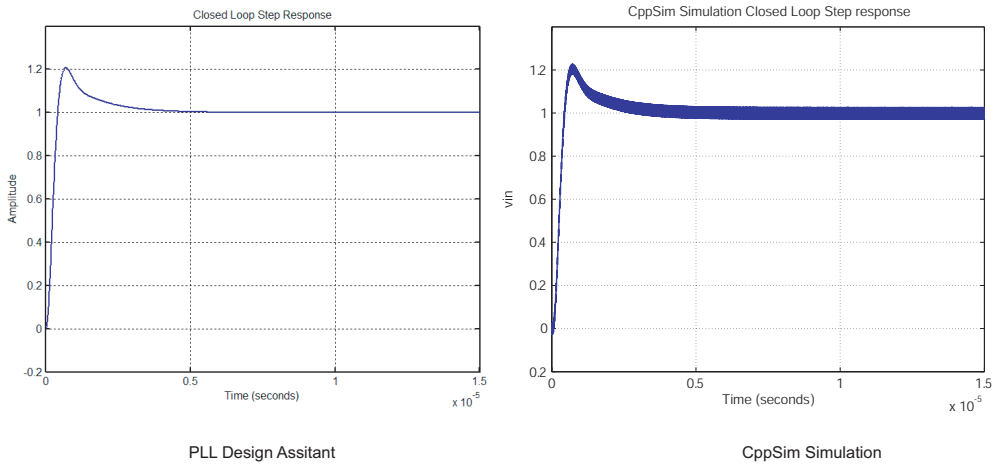


Figure 5-16 Comparison of PDA and CppSim Step Responses for Baseline PFD/DAC Synthesizer

simulated step response of the baseline PFD/DAC synthesizer are presented in Figure 5-16. Once again, we see excellent agreement between the calculated and simulated responses.

Figure 5-17 shows the PFD/DAC output plotted over several cycles as an eye diagram. The zoom-in is of the charge-box. As the phase error changes the location of the divider edge, the multi-level DAC acts to keep the negative charge constant. The current noise is observed as the noisy peak values of the PFD/DAC output once it has settled.

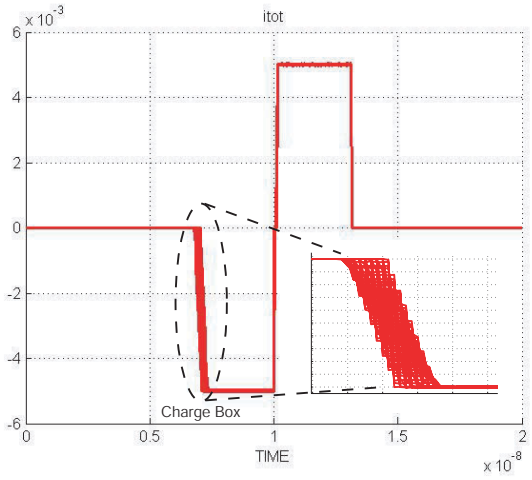


Figure 5-17 PFD/DAC Output Current

5.3 Behavioral Simulation of Non-Idealities and Proposed Compensation Techniques

Now that the baseline performance of the behavioral model has been verified against the calculated performance given by the PLL Design Assistant, we move forward into examining the impact of PFD/DAC non-idealities. Through behavioral simulation, we will verify that the analytical expressions derived from the model presented in chapter 3 accurately predict the impact of PFD/DAC internal mismatch on synthesizer output spectrum.

5.3.1 Compensation of Magnitude Mismatch in the Charge-Box

In section 4.2.3, we determined that magnitude mismatch in the PFD/DAC would result in a gain error. If left unchecked, this error will translate into incomplete quantization noise cancellation, and therefore fractional spurs.

In order to dynamically match the unit elements, the data weighted averaging (DWA) algorithm is used [36, 40]. The resultant mismatch noise has a shaped profile, and so will not be of concern at low frequency. Figure 5-18 presents the results of behavioral simulations done with unit element mismatch included in the PFD/DAC.

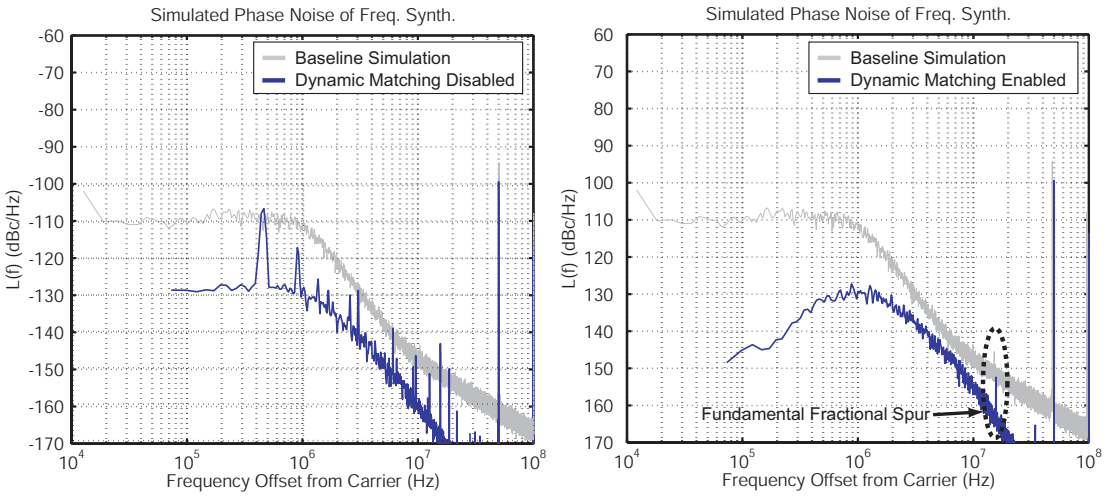


Figure 5-18 Simulation Showing the Impact of Unit Element Mismatch and Dynamic Element Matching

The VCO and detector noise sources are turned off, so that all that appears on the noise plot is the combination of the mismatch noise and the fractional-N quantization noise. The quantization noise is an inherent part of any fractional-N synthesizer, and so cannot be turned off, except in situations where an integer-N value is input to the divider.

Unit element mismatch is represented by a Gaussian mismatch vector of 2^B values, where B is the number of bits in the PFD/DAC. All simulations presented in this chapter use B=7. In the simulations presented in Figure 5-18, a standard deviation $\sigma = 10\%$ was used, a very conservative value given matching data of modern IC processes.

The unit element vector is multiplied by the mismatch vector to create the overall DAC output. Data weighted averaging is accomplished by performing a circular shift of the elements, once every reference clock period. The key to data weighted averaging is that the circular shift starting point moves with the value of the DAC control word (hence the name data weighted averaging).

Figure 5-18 reveals that, without DWA enabled, the unit element mismatch leads to a large amount of fractional spur feed-through. Also plotted as a reference point is the baseline synthesizer performance. Without employing dynamic element matching,

the spurs are significant and it is clear that performance suffers.

It should be pointed out that the absolute spur magnitude cannot be read directly from a phase noise plot. The spur power must be normalized to the carrier, and even so may not be exactly correct if the FFT used to calculate it does not have a frequency bin precisely at the spur frequency. As an example, the reference spur in Figure 5-18 appears at -99dBc/Hz, but is actually only -54dB below the carrier, which is expressed as -54dBc. The discrepancy between dBc/Hz and dBc exists because the PSD calculation is performed at a resolution other than 1Hz.

We can arrive at a reasonable approximation for the spur magnitude through simple calculation if we know some of the simulation parameters. Namely, we need to know the frequency bin resolution of the FFT used to calculate the PSD to translate from dBc/Hz to dBc. In the case of Figure 5-18, the frequency resolution per bin of the FFT is simply the sample rate of the output data, which is 400MHz because we are using the decimator output, divided by the number of bins in each FFT used to calculate the PSD, which is 2^{14}. The relationship is therefore simply

$$dBc = spur\ in\ dBc/Hz + 10log\left(\frac{F_{sample}}{\#\ of\ bins}\right) \quad . \tag{5.12}$$

For the plots in Figure 5-18, this translates to a reference spur of

$$spur\ in\ dBc/Hz + 44 = -99 + 44 = -54dBc \quad . \tag{5.13}$$

All of the spurs in the plot are therefore 44dB higher in dBc than they appear to be on the dBc/Hz axis. The largest spurs in the plot for no dynamic element matching are near -60dBc, much larger than the desired goal of -80dBc spurs.

The magnitude of the fractional spurs will depend on the fractional portion of the divide value according to equation 2.3. For the divide value N=71.31307 used in our simulations, the fundamental of the fractional spur occurs at 15.6535MHz. We check this against equation 2.3, repeated here for convenience.

$$F_{spur_{fund}} = 0.F * F_{ref} \tag{5.14}$$

For $F = 0.31307$ and $F_{ref} = 50MHz$, equation 5.14 predicts a fundamental fractional spur at 15.6535MHz, consistent with simulation results of the dynamically matched PFD/DAC in the right plot of Figure 5-18. This residual fractional spur is due to shape mismatch as described in Chapter 4. Spur compensation techniques such as introducing a sample-and-hold to the loop filter will be proposed in section 5.3.4.

Due to the pulsed nature of the error signal, harmonics of the spur will also appear. Non-linearities introduced by the unit element mismatch (the unit element mismatch can be thought of as a combination of integral non-linearity (INL) and differential non-linearity (DNL)) will cause tones to appear at other frequencies as well, as shown in the left plot of Figure 5-18. The worst case spurious performance can only be found by performing an exhaustive simulation set. Rather than perform all of these simulations, we will show that dynamic element matching removes most of these spurs, and then perform simulations on a subset of fractional values to determine worst case spurious performance.

With DWA enabled, this noise is converted into a broadband, shaped noise, that appears to have the same profile as the quantization noise spectrum. This last observation is intuitive when we consider that both the fractional-N quantization noise and the DWA mismatch noise are first order shaped. We desire that the shaped mismatch noise does not dramatically degrade the overall phase noise performance at intermediate offset frequencies. This has been determined by checking the performance of the synthesizer for several different mismatch vectors generated with different values of σ.

Other possibilities exist for dynamically matching the unit elements [36]. Mostly, these other schemes revolve around different ways to employ a shift register in a pseudo-random fashion. Data weight averaging was chosen for the PFD/DAC synthesizer because it offers a shaped broadband noise response.

5.3.2 Source of Unit Element Mismatch

In a real circuit, the value of σ chosen to represent the unit element mismatch will be derived from the physical size of the transistors used to construct the current sources.

Empirically, the current mismatch will be distributed according to a Gaussian profile, and have a standard deviation described by

$$\sigma = \frac{k}{\sqrt{WL}} \quad , \tag{5.15}$$

where W and L are the transistor width and length, respectively, and k is a proportionality constant related to the process parameters. Larger devices achieve better matching, an intuitive result since any deviations from desired dimensions will be a smaller percentage of the total area.

The physical dimensions of W and L stem from a variety of circuit tradeoffs. In a synthesizer, the transistors that make up the unit element current sources are typically sized based on noise constraints, a topic that will be discussed in more detail in chapter 6. For the purposes of behavioral modeling, we can assume that the devices will generally have large L (for low noise and good output impedance) and large W (for reasons of achieving higher voltage headroom), and that device mismatch standard deviation will be on the order of $\sigma < 0.1$, with a typical value being $\sigma = 0.05$. Unless specified otherwise, we will use $\sigma = 0.05$ ($\sigma = 5\%$) in all future behavioral simulations presented in this thesis where unit element mismatch is enabled.

5.3.3 Compensation of Timing Mismatch in the Charge Box

The two phase paths that generate the charge-box will have some timing mismatch between them. To compensate for this mismatch, we proposed a technique in Chapter 4 that dynamically matches the phase paths by swapping them according to the control of an LFSR random number generator.

We focus on the residual timing error, Δ_{t2}, that will exist at the output of the timing compensation and re-synchronization block in Figure 4-23. The swap process will transform this error from a gain mismatch into a broadband noise source, according

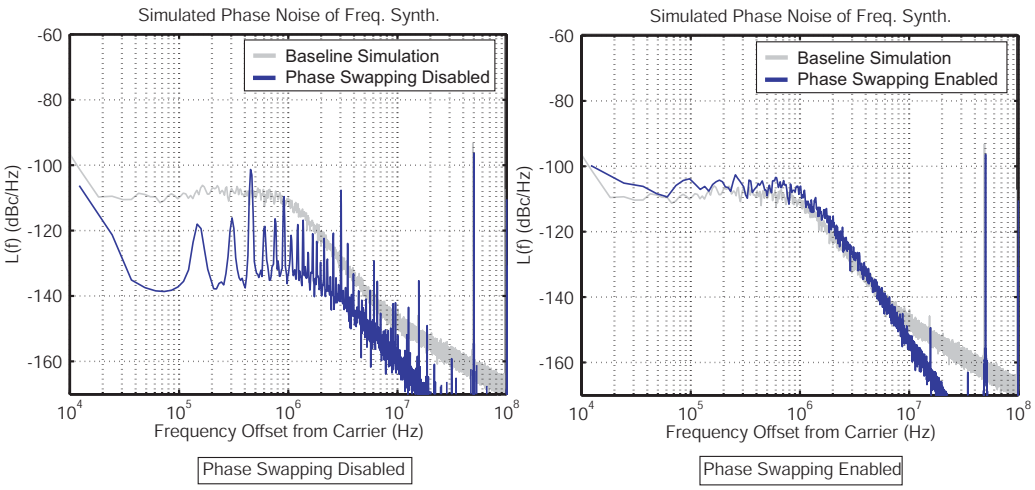


Figure 5-19 Simulations Showing the Impact of Timing Mismatch and Phase Swapping

to equation 4.10, repeated here for convenience.

$$S_{\Phi_{out}|\Delta_t} = \frac{1}{T} \cdot \frac{\Delta_{t2}^2}{12} \cdot (2\pi N_{nom})^2 \cdot |G(f)|^2 \tag{5.16}$$

For a 5ps timing mismatch, equation 5.16 predicts a low frequency mismatch noise of -107dBc/Hz. Figure 5-19 shows the result of a behavioral simulation with phase swapping disabled and enabled. Once again, VCO and detector noise are turned off to isolate the impact of mismatch noise on the output. Additionally, unit element mismatch is set to zero. We see that, as with unit element mismatch, timing mismatch creates a non-linearity that results in large fractional spurs. The swapping process, however, converts this noise into broadband noise that is filtered by the PLL dynamics. The resulting noise in the figure matches the predicted value of -107dBc/Hz, indicating the validity of equation 5.16.

The overlay of the baseline simulation phase noise indicates that the low frequency noise will be slightly degraded by the swapping process. However, the synthesizer will still meet the stated -100dBc/Hz goal, and spurious performance is dramatically improved.

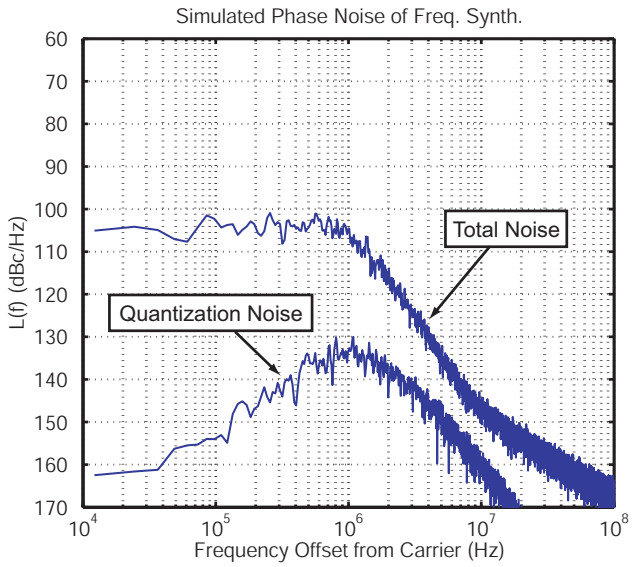


Figure 5-20 Elimination of Shape Mismatch by Using a S/H

5.3.4 Eliminating Shape Mismatch With a Sample-and-Hold

Shape mismatch between the vertically resolved phase error waveform and horizontally resolved DAC cancellation waveform was discussed in section 4.2.5. We propose that a sample-and-hold (S/H) be introduced to sample the loop filter voltage and eliminate shape mismatch related spurs. Of course, a real synthesizer will exhibit both magnitude and timing mismatch in addition to shape mismatch, and so both the DWA circuitry and the phase swapping circuitry must be enabled. A significant secondary benefit of using the sample and hold is that the reference spur due to the PFD action will be theoretically eliminated in addition to the shape mismatch induced fractional spurs.

In the baseline simulation of Figure 5-13 the reference spur magnitude is -93dBc/Hz which, using equation 5.12, corresponds to a spur of magnitude -52dBc at 50MHz. There is also a small, residual fractional spur due to shape mismatch at the fundamental fractional frequency of 15.6535MHz. It reads as -145dBc/Hz in Figure 5-13, which translates to a -104dBc fractional spur.

Two simulations were performed to show the effect of applying a S/H to the PFD/DAC synthesizer. Figure 5-20 shows the phase noise response due solely to the

quantization noise (VCO, detector, and mismatch noise off) as well as with all noise sources on. The fractional spurs are completely eliminated, as is the reference spur!

In a real implementation, the S/H will not be ideal, and some error charge will exist. Ideally, this error noise would be due only to charge-injection effects of implementing the S/H function, and would therefore result in a small reference spur. In Chapter 6 we propose a S/H architecture, and in Chapter 7 present measured results with the S/H enabled and disabled to compare performance.

5.3.5 A Digital Compensation Scheme for Reducing Shape Mismatch Spurs

In this section, we present simulation results for the digital compensation scheme proposed in section 4.2.5. The key idea behind this scheme is that the shape error charge can be approximated by equation 4.11, and we can use equation 4.11 to implement a digital compensation scheme to reduce the magnitude of this error [25]. Since the S/H approach offers a much higher potential benefit by completely eliminating shape mismatch, it will be employed in practice instead of the digital compensation scheme. However, it is useful to review the results of the digital compensation scheme we propose in [25] since they validate that it is possible to use a purely digital compensation scheme to reduce fractional spurs. The results presented in this section are for the synthesizer presented in [25], which has the same performance goals as the synthesizer we have presented thus far, but has a 55MHz reference and 5GHz output. The digital compensator is depicted in Figure 4-25 and the PFD/DAC synthesizer with digital compensation in Figure 4-26.

Figure 5-21 shows a simulation result for a particular divider input with and without the digital gain compensation enabled. The compensation can be implemented as a look-up table (LUT) and will therefore have finite input and output resolution. This is represented in the simulation results by denoting a synthesizer as being X/Y compensated, where X is the number of address bits to the LUT, and Y is the number of output bits. With 6/4 compensation, representing a 1Kbit LUT, the fractional

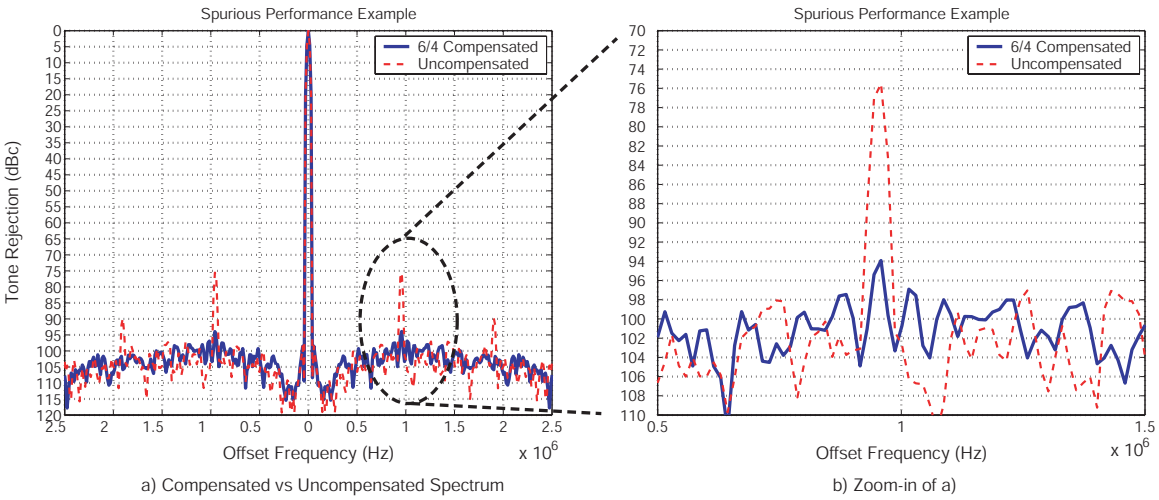


Figure 5-21 Output Spectrum with and without Digital Compensation

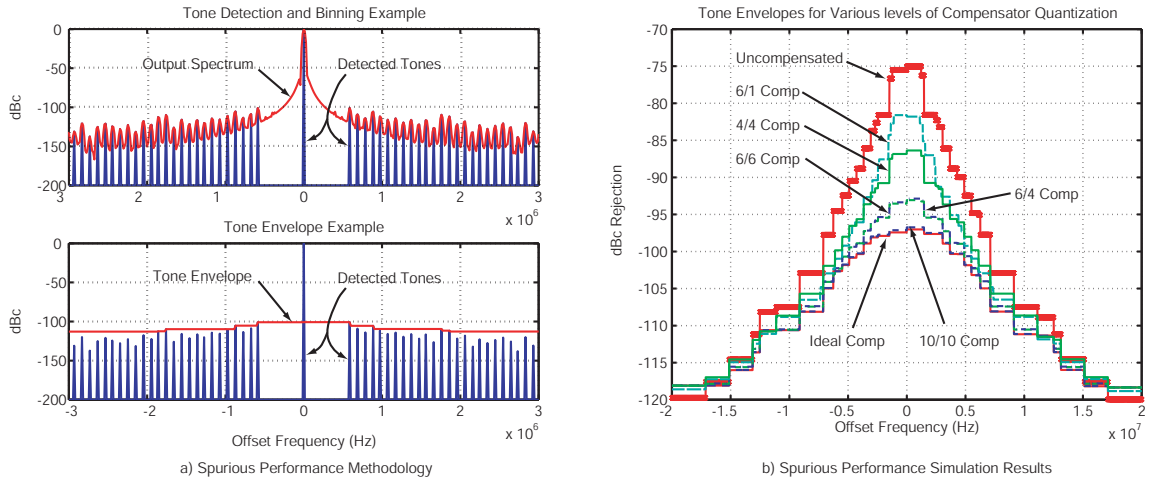


Figure 5-22 Spurious Performance Methodology and Simulation Results

spur is reduced by >15dB for this example!

In order to examine rejection over a broad range of fractional spur values, detailed behavioral level simulations are performed over a wide range of accumulator input values. The methodology used, as well as simulation results, is depicted in Figure 5-22. The upper trace on the left plot shows the output spectrum for a particular accumulator input. A tone-detection algorithm is used to detect any spurs in the spectrum. Simulations are run for 28 accumulator input codes, and the worst case spurs at each frequency are determined across simulation runs. The bottom left trace shows how, once worst case tones are determined, an envelope is used to represent

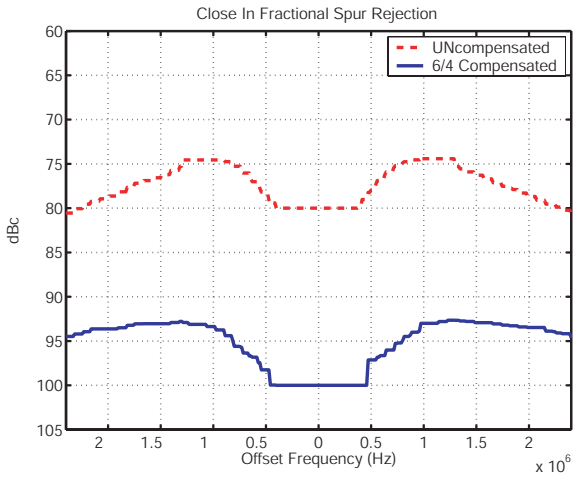


Figure 5-23 Close-In Spur Performance

the maximum spur levels. The right plot shows results from a number of simulation runs with various levels of quantization in the digital compensation LUT. With a 20 bit accumulator, it is extremely time intensive to simulate the entire 20 bit input space. Instead, to generate the plot, 28 simulations were done for each compensation level, with the accumulator input varied so that the fractional spurs generated would span a 20MHz bandwidth. The steps in the plot are due to stepping the accumulator input through a set of values chosen to generate fractional spurs across a range of offset frequencies. The worst case tones without any digital compensation are <-74dBc and occur at very low offset frequencies. In order to more closely examine the low frequency spur performance, an additional 186 finely spaced simulations are done with the accumulator input set to produce low frequency fractional spurs, as shown in Figure 5-23. (The envelope floor is set at -80dBc for the un-compensated synthesizer, and -100dBc for the compensated synthesizer in Figure 5-23) The figure shows that, once the 1MHz bandwidth is exceeded, filtering by the PLL dynamics helps reduce spur feed-through.

Returning to the results presented in the right plot of Figure 5-22, the -74dBc raw performance is seen to be very good, and may be attributed to the enhanced gain match obtained by the PFD/DAC as compared to prior art. Once compensation is enabled, the spur performance improves dramatically. "Ideal compensation"

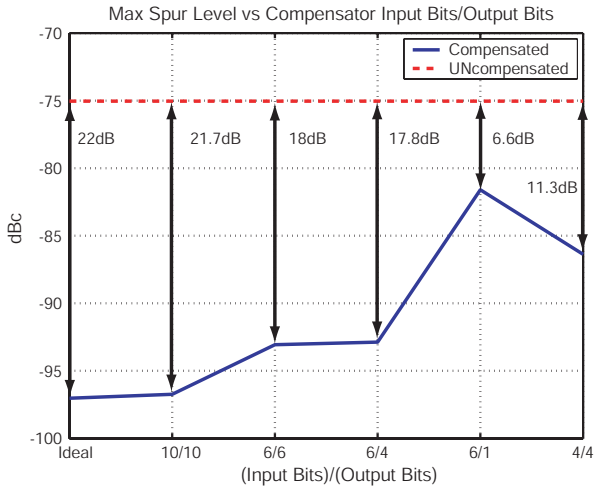


Figure 5-24 Maximum Spur Levels for Various Levels of Compensation

means that the resolution on the LUT matches that of the input accumulator, and serves as a limiting case. The synthesizer simulated is assumed to have a 20 bit input accumulator, corresponding to 20/20 compensation for the ideal case. With 10/10 compensation near ideal results are obtainable. However, 10/10 compensation requires a 1Mbit LUT, which is rather large. By contrast, with 6/4 compensation and 1Kbit LUT, all tones are kept below -92dBc, an 18dB improvement from the un-compensated case. Figures 5-22 and 5-23 demonstrate that using the compensation scheme results in improved performance across frequencies.

Finally, Figure 5-24 summarizes the envelope results presented in the right plot of Figure 5-22. Maximum fractional spur level is contrasted between the un-compensated synthesizer (dashed line) and compensated synthesizer for various combinations of LUT input and output resolution. Best possible performance is achieved with a very high resolution LUT and corresponds to 22dB improvement and a <-95dBc maximum spur. The 6/4 compensation level used in the example synthesizer results in 18dB improvement and <-92dBc maximum spur levels, a good performance enhancement with minimal added digital complexity.

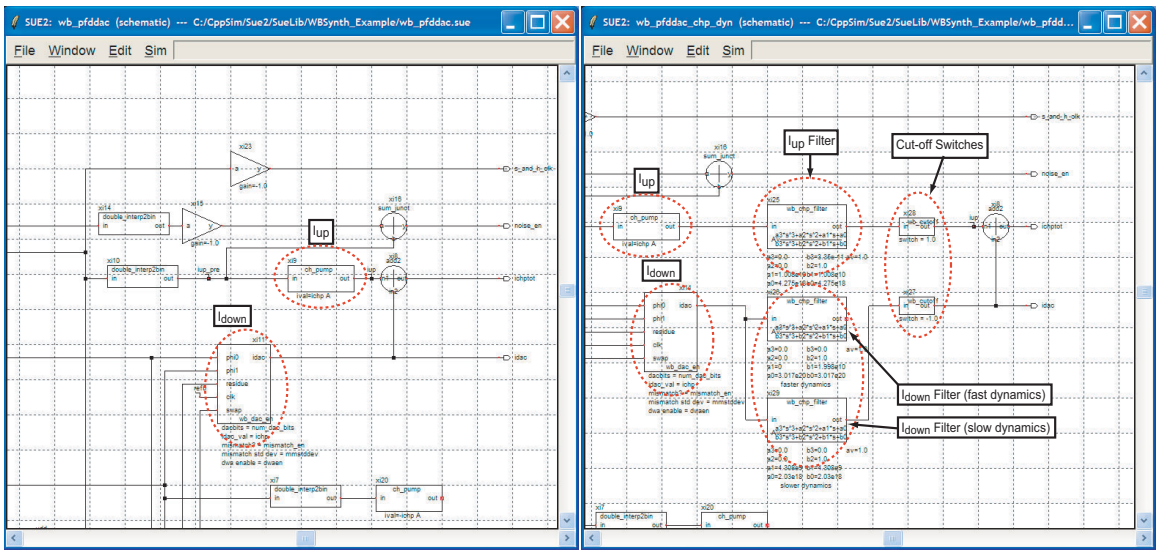


Figure 5-25 PFD/DAC With and Without Finite Settling Added

5.3.6 Impact of Finite PFD/DAC Settling

A real PFD/DAC will not settle instantaneously as has been assumed so far. In Figure 5-17 the PFD/DAC output current pulses are square-wave signals. In order to capture the effect of finite charge-pump settling we introduce a filter function to the PFD/DAC output. The filter dynamics are chosen by performing Hspice simulations on the PFD/DAC circuitry, and then matching the filter to the PFD/DAC response.

Figure 5-25 shows the methodology used to add PFD/DAC settling to the behavioral simulation. The left figure shows the PFD/DAC with no filtering, and the right figure with filtering added. Two filter choices have been included for the down current. The first filter represents charge-pump settling based on a SPICE simulation of the charge-pump using an early estimate of parasitic capacitances, and the second filter captures results of a SPICE level simulation performed on extracted layout. In this way we can evaluate pre-layout and post-layout performance.

One subtle point to note is that, since the filters are fed by ideal square-wave pulses, caution must be used in processing their outputs. The ideal square pulses will stimulate the filter on both rising and falling edges. In a real circuit, once the current is switched off, it will have zero magnitude. For an ideal filter fed by a negative pulse, the current will not go to zero, but rather will exhibit a negative step-response.

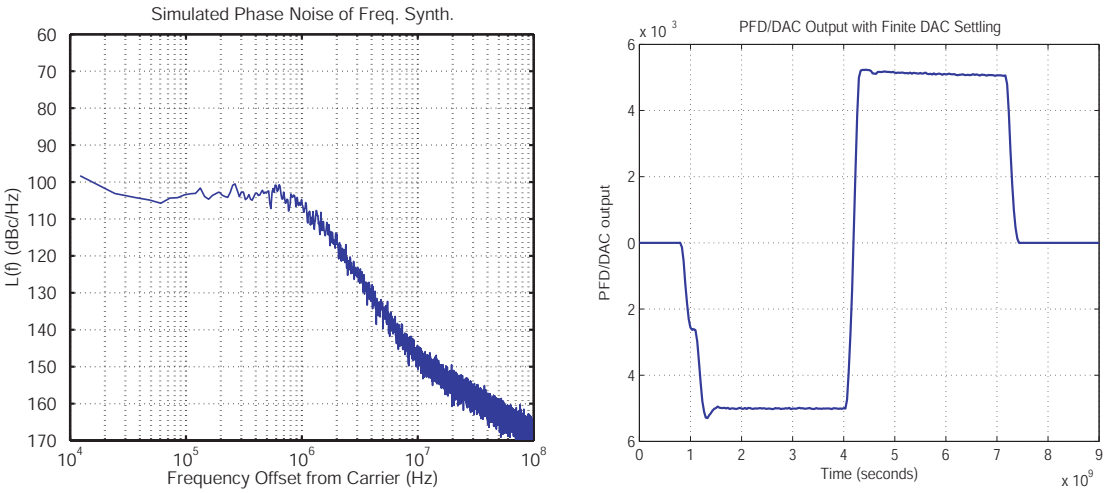


Figure 5-26 Finite Settling of the PFD/DAC Using Estimated Parasitics

For this reason, cut-off switch blocks have been added after the filters. The cut-off switches stop the up current from going negative and the down current from going positive. This effect can be thought of as a non-linearity captured by the simulation, since the current pulses will not have symmetric edges.

Figure 5-26 shows simulation results for the PFD/DAC synthesizer with finite PFD/DAC settling included. Note the different settling responses of the negative versus positive current pulses. This is because of the separate filter dynamics used to model the positive current sources versus the negative current sources. We can see that the spectrum has not changed at all from Figure 5-20, indicating that the settling dynamics of the PFD/DAC do not affect performance! This result implies very good charge-pump linearity in the face of finite charge-pump settling. The filter values chosen for I_{down} were based on estimated parasitic capacitances, and were somewhat optimistic. The filter values for I_{up} were based on parasitic extraction based simulations.

Figure 5-27 shows the system response to slower I_{down} settling. Filter values for this simulation were based on parasitic extraction simulations. We see that, even for this more severe case, the output phase noise is not affected by finite settling of the PFD/DAC output, indicating a system level robustness in the face of parasitic capacitances affecting charge-pump settling. Charge-pump non-linearity due to finite

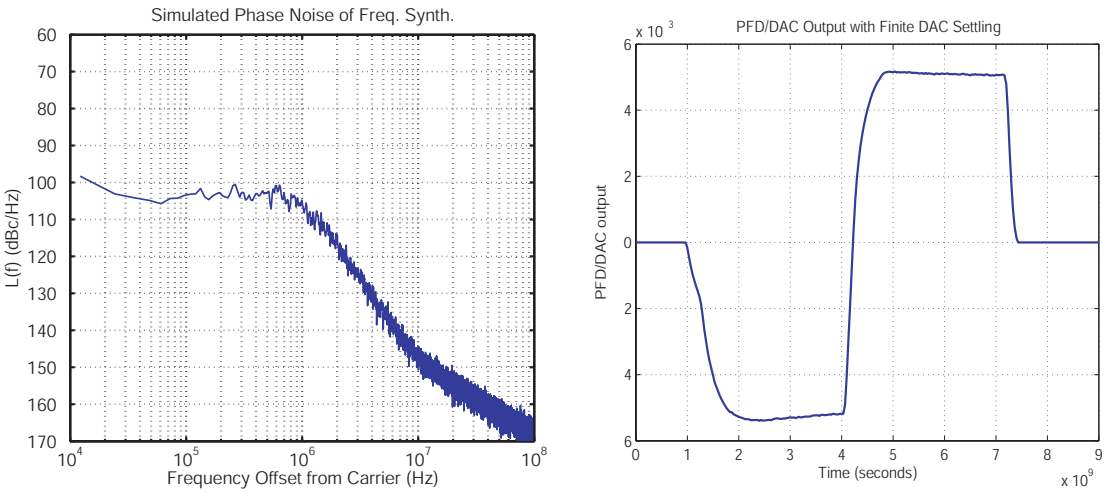


Figure 5-27 Finite Settling of the PFD/DAC Using Extracted Layout Parasitics

settling effects has been a concern in the past [26].

5.3.7 Impact of Finite Charge-Pump Output Impedance

The loop filter shown Figure 5-8 is a passive configuration. That is to say that the filter is composed entirely of passive components. A problem with a purely passive configuration is that the current sources that make up the PFD/DAC charge-pump will have finite output impedance, and will therefore not appear as ideal current sources. Finite output impedance means that PFD/DAC output current will be a function of the voltage at the output node according to

$$i_{up} = i_{up-nom} + \frac{V_{out} - V_{nom}}{r_{o-up}} \tag{5.17}$$

$$i_{down} = i_{down-nom} - \frac{V_{out} - V_{nom}}{r_{o-down}} \tag{5.18}$$

where V_{nom} is the loop filter output voltage that results in the desired value of output currents i_{up-nom} and $i_{down-nom}$. This problem is exacerbated by the fact that if the frequency synthesizer is modulated, then the voltage at the loop filter output will change. Any modulation of the output current with the VCO control voltage results in a non-linearity, which, if large enough in magnitude, can adversely affect synthesizer

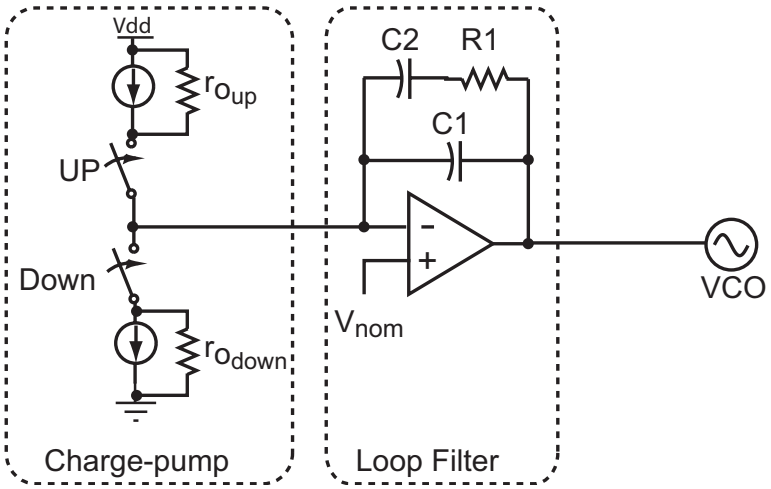


Figure 5-28 Active Loop Filter Configuration

performance by folding high frequency noise down to lower frequencies.

One possible solution to this problem is to maximize the output impedances of the current sources through circuit techniques such as cascoding, at the expense of voltage headroom. Another possibility is to use an active loop filter. Active loop filters employ an op-amp in feedback to create the desired filter response. Figure 5-28 depicts an active version of the lead-lag loop filter of Figure 5-8, with the finite current source output impedances included.

The op-amp sets the voltage at the charge-pump output to be V_{nom}. As the synthesizer operates and the VCO control voltage changes, the voltage across the charge-pump does not change, and there is no modulation of charge-pump output current. For this reason, active loop filters are very popular in synthesizer literature. The penalty paid for introducing the op-amp into the loop is that op-amp noise will now contribute to the total synthesizer output noise. This topic will be discussed in detail in Chapter 6. Because it is possible that the op-amp noise could be significant, we will investigate the use of passive loop filters, and therefore examine the impact of finite charge-pump output impedance on synthesizer performance.

The MOS I-V equation is used to model the charge-pump output current, and parameters for V_{nom}, I_{nom}, and r_o are included to set nominal performance. A feedback loop is closed around the circuit so that, as the voltage at the charge-pump output

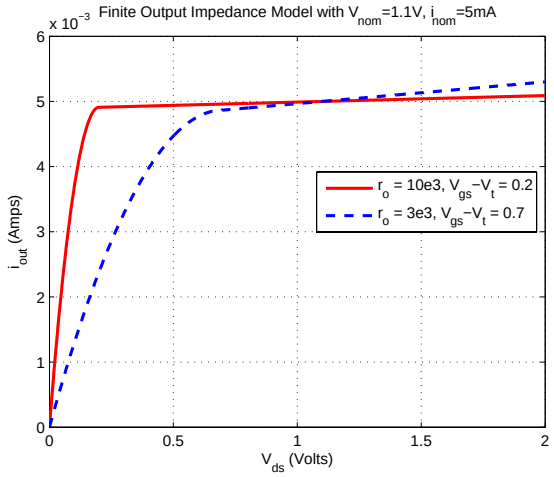


Figure 5-29 Simulation Results for Finite Output Impedance Current Source Model

is changed, the I-V equation modifies the charge-pump output current. Figure 5-29 shows simulation results of the finite output impedance current source model added to the PFD/DAC block. Results are shown for two values of $V_{gs} - V_t$ and r_o. It is also possible in the model to change I_{nom} and V_{nom}. The model captures operation of the current source in both saturation and triode regions.

Figure 5-30 shows synthesizer simulation results with finite charge-pump output impedance included. The simulation assumes $r_{o-up} = 20k\Omega$, $r_{o-down} = 680k\Omega$, and $V_{nom} = 1.1V$. These values were determined from Hspice simulations of the PFD/DAC unit elements that will be discussed in Chapter 6. Note that the positive current sources have much worse output impedance than the negative current sources. This design issue will be discussed in section 5.4.

The dashed waveforms in the right plot of Figure 5-30 represent the outputs of the charge-pump before being processed by the filters that were added to model finite PFD/DAC settling. The zoom-in inset clearly shows that charge-pump current changes as the charge-pump output voltage changes. It is also clear from the phase noise plot that the addition of finite PFD/DAC output impedance has not degraded overall performance. The negative current changes by approximately $1\mu A$, while the positive current by approximately $30\mu A$. The simulation results show that the non-linearity introduced by PFD/DAC finite output impedance is not significant to the

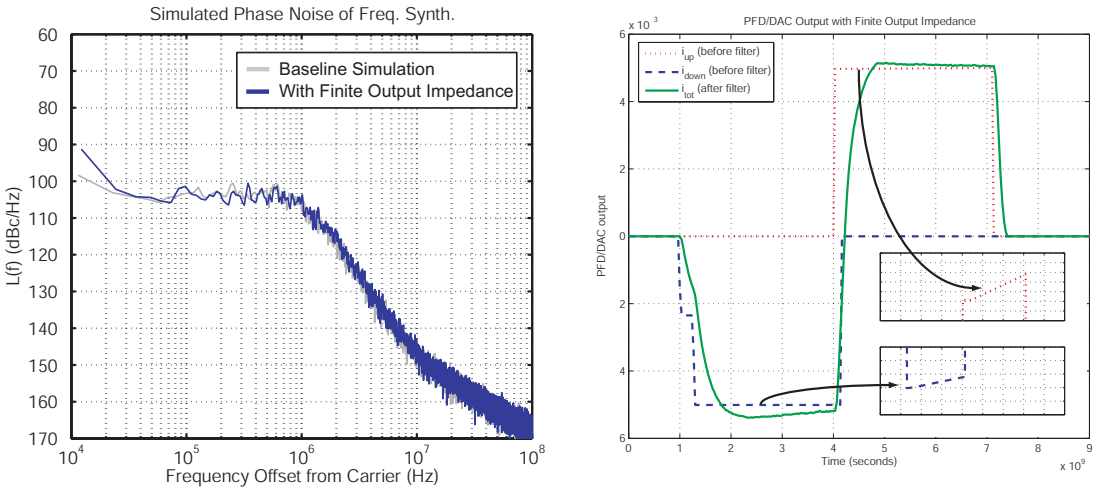


Figure 5-30 Simulation Including Finite Unit Element Output Impedance

design of the simulated synthesizer for the values of output impedance extracted from Hspice simulations.

In order to verify our intuition that, at some point, modulation of the PFD/DAC current by finite output impedance should affect performance, we run a simulation with the output impedances reduced by a factor of 100. Figure 5-31 shows the results of this simulation. The output spectrum from Figure 5-30 is included as a lightly shaded reference. The noise performance is altered in that the closed loop bandwidth is increased. Since the output impedance was made to be very bad, the average value of output current is very different from its nominal value. The loop gain is therefore higher, increasing closed loop bandwidth. A somewhat unexpected result is that the non-linearity introduced by finite output impedance has not caused any issues with fractional-spur cancellation, indicating that the chosen PFD/DAC architecture is robust in the face of finite PFD/DAC output impedance.

5.3.8 Impact of Unity Gain Buffer Non-linearity

The choice to use negative current to create the variable charge-box was made for reasons of speed and power. For a given voltage headroom and switch speed, NMOS devices are much better than PMOS devices due to their increased mobility. The prototype transmitter IC that will be presented in Chapter 6 utilizes a 7-bit PFD/DAC,

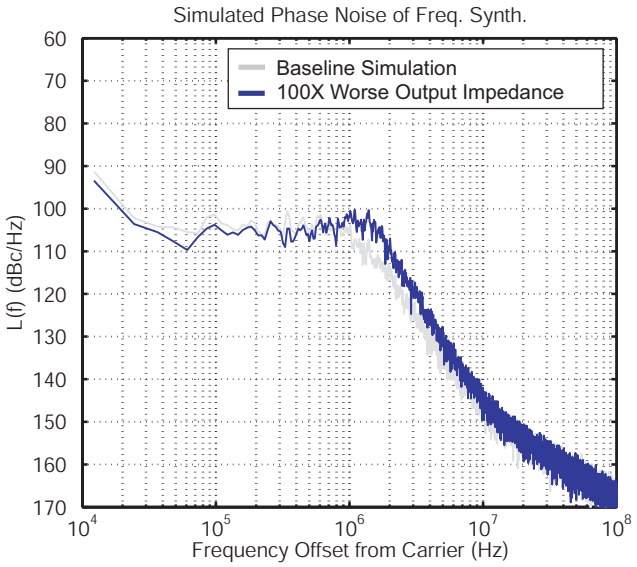


Figure 5-31 Simulation with 100X Worse-than-expected Charge-pump Output Impedance

consistent with our behavioral modeling simulations. A 7-bit PFD/DAC requires 128 unit elements, and therefore 128 switches to be controlled in order to generate the charge-box. In order to minimize the power needed to drive these switches, NMOS devices are chosen to create the variable charge-box.

If the active loop filter of Figure 5-28 is used, an extra inversion is introduced to the loop, since the op-amp acts as an inverting trans-impedance amplifier. This inversion has the undesired side-effect of causing the PLL negative feedback loop to become a positive feedback loop that will not be stable. Because the off-chip VCO used in the prototype synthesizer has a positive gain, we have to account for the added inversion. If an integrated VCO was used, we could simply design the VCO to have a negative gain to compensate for the added inversion introduced by the active loop filter.

There are two options to eliminate the instability caused by the additional inversion. First, the PFD/DAC can be made to control a charge-box created by PMOS current sources, effectively changing the polarity of the charge-pump gain. As has been discussed, this solution is not desirable for power and speed reasons. The second solution is to add a unity gain inverting amplifier between the op-amp output and

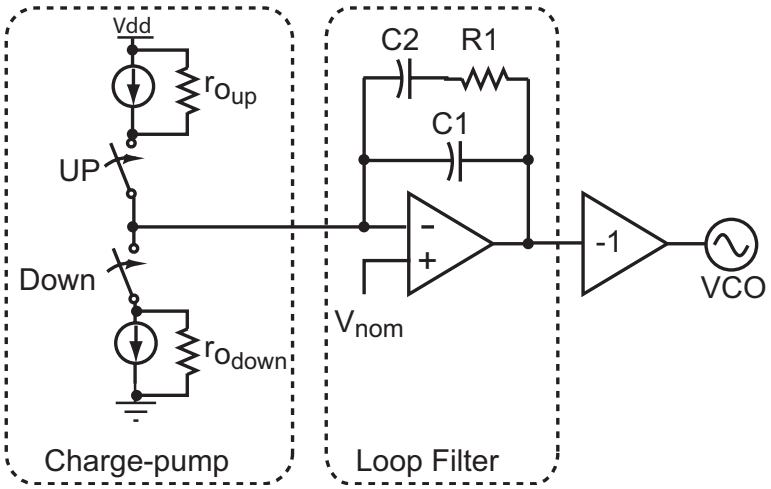


Figure 5-32 Active Loop Filter With Unity Gain Inverting Buffer

VCO input, as depicted in Figure 5-32.

The amplifier provides another inversion, restoring the loop feedback to a negative polarity. As with any added active circuitry, the amplifier will introduce its own noise source and non-idealities. As will be shown in Chapter 6, the unity gain inverting buffer amplifier noise is small enough to be ignored in behavioral simulation.

The amplifier will have some non-linearity associated with its transfer function from input to output, and it is possible that this non-linearity will affect synthesizer performance. Figure 5-33 presents the results of an Hspice simulation of the inverting buffer and a 4^{th} Order polynomial approximation used by CppSim to model the amplifier. The amplifier output has been normalized to 0 volts input and output at nominal operation to simplify its integration into the CppSim model.

For the behavioral model, the SPICE simulated gain curve is shifted so that $v_{in} = 0$ results in $v_{out} = 0$. The model output has been approximated by linear functions for $v_{in} < -0.6V$ and $v_{in} > -0.5V$. Note that the actual circuit does not operate with negative supplies. In order to achieve IC voltage levels from the curve shown in Figure 5-33, offset voltages would be added to the input and output voltages. These offsets were removed from the SPICE simulated gain curve to simplify its use in the behavioral model.

If the linear approximations are not made, there may be startup conditions where

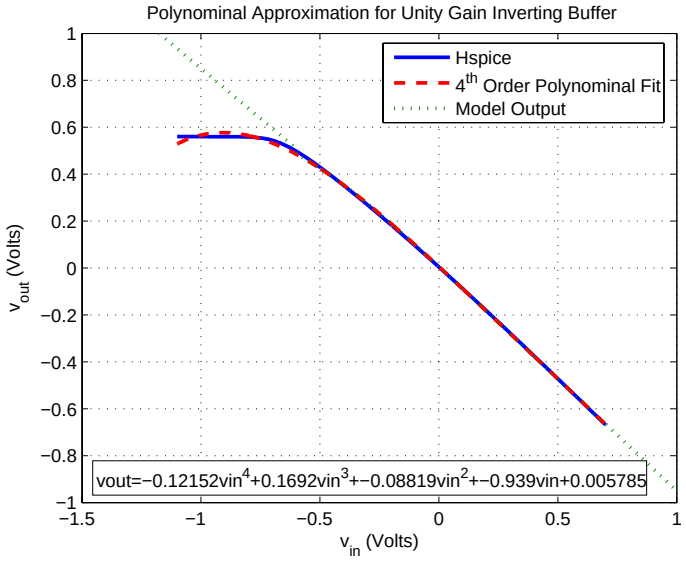


Figure 5-33 Inverting Buffer Amplifier Polynomial Approximation Model Response

the 4^{th} order polynomial approximation causes the buffer gain to change from negative polarity to positive polarity. This behavior is shown in Figure 5-33, where we see that, for $v_{in} < -0.8V$, the first derivative of the polynomial approximation changes sign, indicating that the inverter gain would change sign. Therefore we find that the linearized regions are good safeguards against undesirable instability introduced by the polynomial approximation.

Figure 5-34 shows a simulation result with the amplifier non-linearity included. The output spectrum is overlayed on top of the spectrum without amplifier non-linearity. They are *almost* identical. The slight difference is due to the fact that the amplifier gain is not exactly -1.0, so the closed loop bandwidth is slightly different than 1MHz. The slight decrease in gain can be offset by increasing gain in the loop filter, if desired.

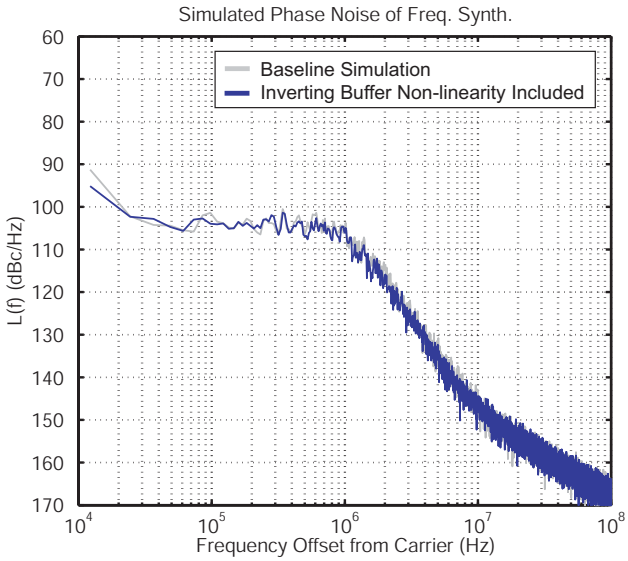


Figure 5-34 Output Phase Noise Profile with Inverting Amplifier Model Included

5.4 Choosing the PFD Architecture for Best Charge-Pump Linearity

The final design trade-off we will discuss in this chapter relates to the choice of PFD architecture. We will investigate four possible choices for the PFD typed used by the PFD/DAC, and discuss design trade-offs of each.

5.4.1 Classic Tri-state PFD

We begin with the simple tri-state PFD depicted in Figure 5-35. For simplicity only one of the two divider phases used by the PFD/DAC is shown in the figure. The tri-state PFD attempts to lock a PLL such that there is zero phase difference between the divider and reference. Mismatch between the charge-pump currents, i_{up} and i_{down} as well as fractional-N dithering of the divider edge location create non-linearities, as will be shown.

Also shown in Figure 5-35 is the phase noise response of the PFD/DAC synthesizer with this PFD configuration. All sources of possible error and additional noise that have been discussed thus far are included in the model. Overlayed in light gray is the

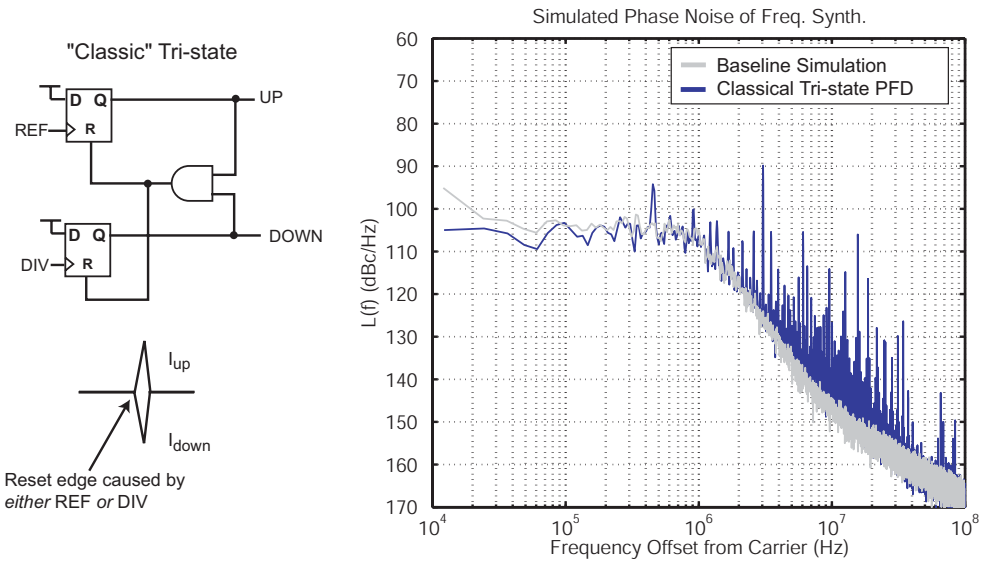


Figure 5-35 Classical Tri-State PFD Architecture

output response of Figure 5-34, which is used as a reference performance target. The classic tri-state PFD based PFD/DAC synthesizer suffers from large fractional spurs. The main issue centers around operation of the PFD in steady-state.

A fractional-N synthesizer using the tri-state PFD exhibits a steady-state behavior where the divider edge location changes over time, but occurs, *on average*, at the same time as the reference edge. This is because, for a fractional-N synthesizer, the tri-state PFD locks the PLL to zero *average* phase difference. The instantaneous divider edge locations can occur before or after the reference signal, and therefore the phase error moves on both sides of zero phase error in steady-state. Mismatch between positive and negative currents and mismatch sources affecting activation of the current pulses lead to nonlinearity in the overall PFD performance.

Figure 5-36 captures such behaviors, where we see that the two divider signals used in the PFD/DAC synthesizer align, on average, with the reference edge, but sometimes come before, and sometimes after the reference. Both positive and negative charge packets change over time, and the current pulses do not completely settle. All of these non-idealities contribute to the non-linear behavior of this PFD topology.

Fractional-N synthesizers are particularly vulnerable to non-linearities because they can lead to incomplete fractional-spur cancellation, spurious mixing products,

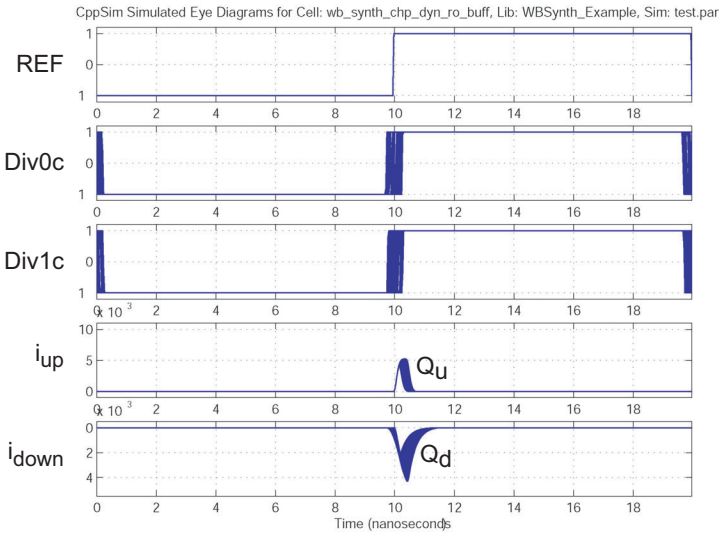


Figure 5-36 Eye Diagram of Key Signals in a PFD/DAC Employing the Classic Tri-state PFD

and noise folding, which are all highly undesirable effects.

One source of non-linearity in a classic tri-state PFD is depicted in Figure 5-37. For simplicity, only phase errors within a $\pm 2\pi$ window are shown. The steady-state phase offset conditions for four different tri-state configurations are marked in the figure. The classic tri-state based synthesizer locks to zero phase error. The three other PFD topologies that will be discussed are also represented in the figure. As will be discussed, the overlapping tri-state PFD synthesizer also locks to zero steady-state phase error, while both the offset tri-state PFD synthesizer and offset-and-overlapping tri-state PFD synthesizer architectures lock to a non-zero steady-state error. The steady-state value of phase error has linearity implications.

Returning to our discussion of the classic tri-state PFD, we see in Figure 5-37 that unequal positive and negative charge-pump outputs create a non-linearity in the system. A classical tri-state PFD based synthesizer locks to near zero phase error within the ability of the system to resolve small phase offsets and within the match between i_{up} and i_{down}. If the current magnitudes are nearly equal, then as the system locks and fractional-N dithering causes the divider edge to move, system phase error will move on both sides of zero such that the *average* phase error is zero.

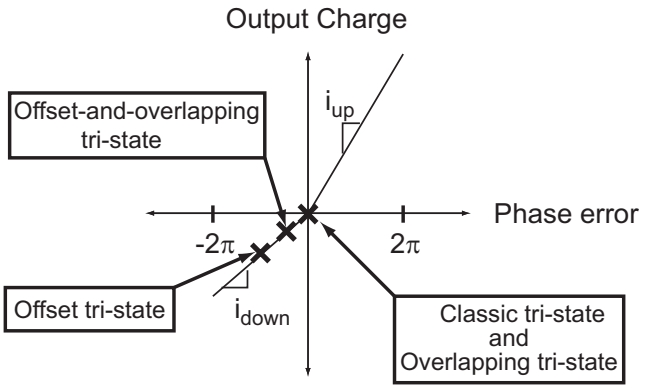


Figure 5-37 Tri-state PFD and Charge-pump Transfer Curve for $i_{up} \neq i_{down}$

As the phase error modulates around zero, it moves along both portions of the curve in Figure 5-37, and the synthesizer experiences a non-linearity as it moves between the different gain regions of the curve. Clearly, the transfer curve depicted in Figure 5-37 is not linear, and operating the PFD at zero degrees phase error is the worst place to operate when attempting to achieve high linearity.

The various sources of non-linearity in the classic tri-state PFD disturb the finely controlled charge-balance a PFD/DAC synthesizer creates. As Figure 5-35 shows, PFD/DAC synthesizer performance using the classic tri-state PFD is very poor. The non-linearity associated with this topology causes the output spectrum to be highly spurious. The classic tri-state is clearly a poor choice for the PFD/DAC synthesizer, and potentially a performance limiter for any fractional-N synthesizer due to its highly non-linear behavior.

5.4.2 Overlapping Tri-state PFD

An improvement to the simple tri-state PFD is the overlapping tri-state PFD, shown in Figure 5-38. This topology removes the small pulses associated with the classic tri-state configuration, but still operates around zero phase error in steady-state. Once again, the reference phase noise profile of Figure 5-34 is included as a light gray overlay reference for comparison. In the overlapping reset tri-state, both phase detectors are left on for some amount of time to avoid the issue of non-linearity due to the impact of finite rise and fall times on the ability of the PFD to resolve small

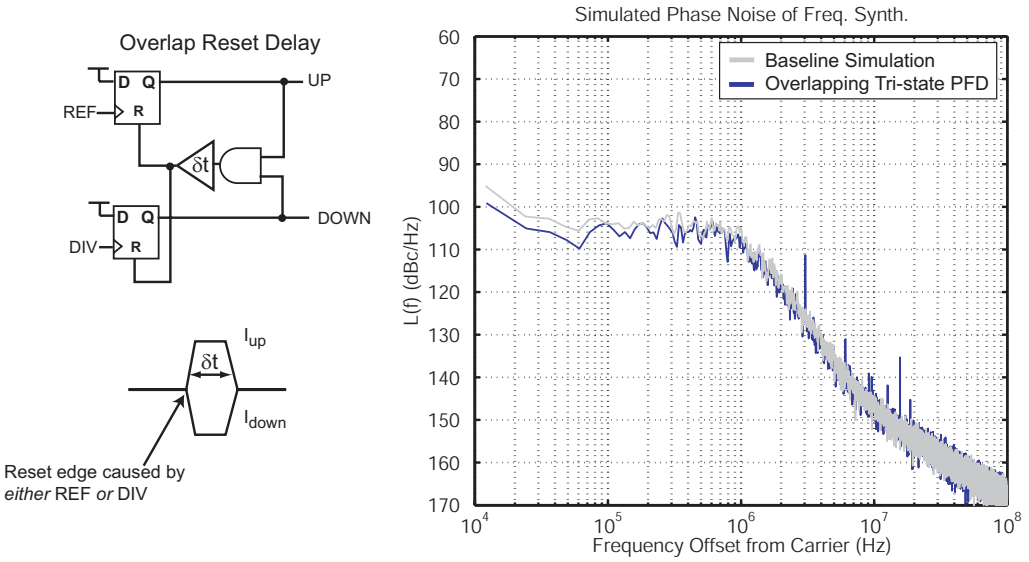


Figure 5-38 Tri-State PFD Architecture With Current Overlap

phase offsets. The on-time is determined by a delay cell that delays the reset pulse to both flip-flops. In the simulation, $\delta t = 3ns$.

Unfortunately, we observe spurs in the output spectrum in Figure 5-38. Although this architecture does offer more linear performance that the classic tri-state, it is not completely linear. Figure 5-39 shows the output eye diagrams for the key PFD signals. Both i_{up} and i_{down} are on long enough that they properly settle to their final values. However, we observe that both positive and negative charge packets change over time, indicating that there is some inter-modulation of charge over time. Creating a situation where the system locks to zero steady-state phase error means that any non-linearity experienced as the phase error moves back and forth through the zero error region in Figure 5-37 will be emphasized in the system.

One final point to note is that, while the currents in an overlapping tri-state PFD ideally cancel, their current *noise* will not. Current noise for the up and down current sources will be uncorrelated. Therefore, calculation of charge-pump noise should account for the presence of both noise sources at the output.

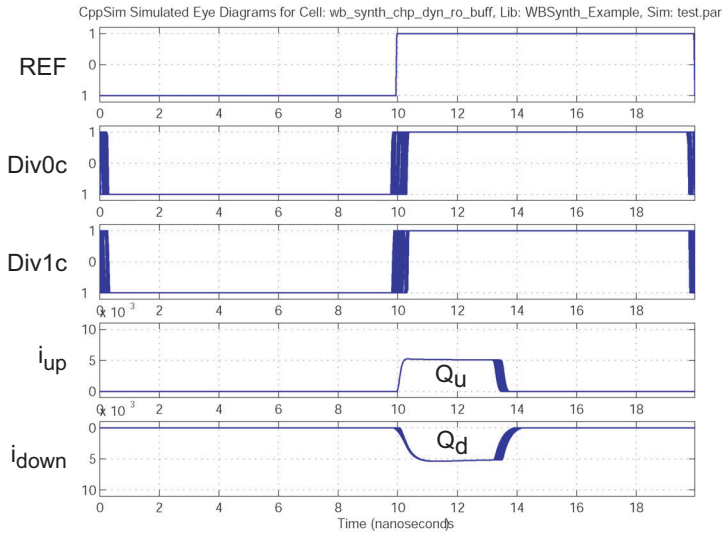


Figure 5-39 Eye Diagram of Key Signals in a PFD/DAC Employing the Overlapping Tri-state PFD

5.4.3 Offset Tri-state PFD

Figure 5-40 presents the offset tri-state PFD architecture. In this case, the delay is placed such that i_{up} always occurs after i_{down} is finished. Another way to understand its operation is to realize that the reference edge causes i_{down} to turn off also causes i_{up} to turn on. A benefit of this PFD topology is that the steady-state phase error is not centered around zero, and so the non-linearity associated with unequal i_{up} and i_{down} is avoided. We see that, in steady-state, the offset tri-state PFD based synthesizer operates only on one portion of the gain curve in Figure 5-37, which results in improved linearity compared to the classic and overlapping tri-state PFD based synthesizer topologies.

Because the reference edge is constant in time, the reset edge *always* occurs in the same place. Of course, a practical reference will have some phase noise. Reference phase noise (or jitter) is low-pass filtered by the PLL [1], and is typically assumed to contribute less to output phase noise than charge-pump noise, and so is not generally included in simulations. We assume a "clean" reference, meaning its noise does not contribute in any significant way to the output phase noise. We will see later that our prototype synthesizer, however, is ultimately limited by reference jitter induced

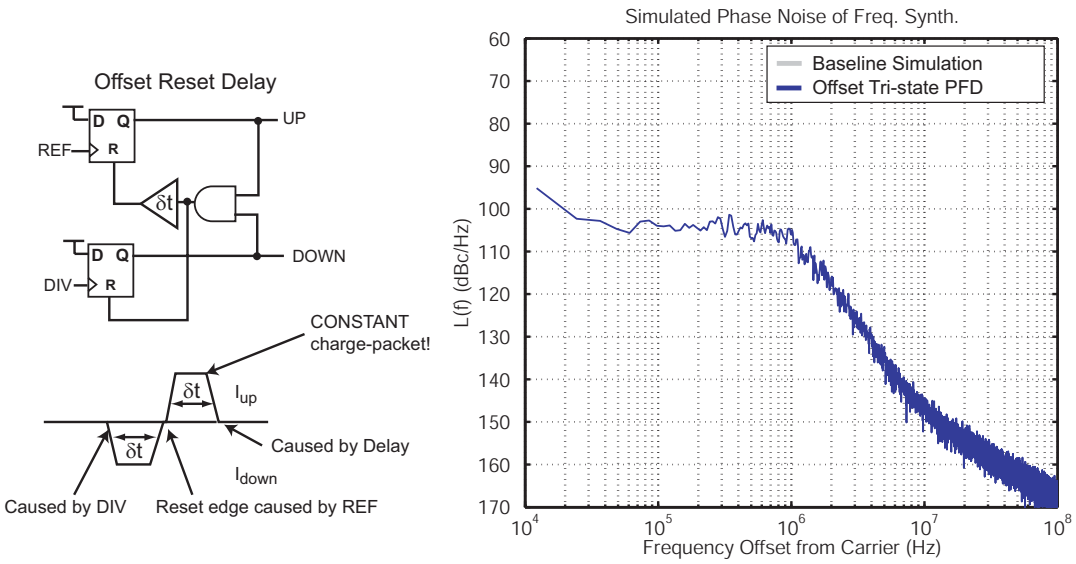


Figure 5-40 Tri-State PFD Architecture With Offset Currents

phase noise.

If the delay time, δt, is constant, then the amount of positive charge is constant every period. The negative charge can be made constant using the PFD/DAC technique, as described in Chapter 4. This is accomplished by offsetting the pulses in time, and operating under non-zero steady-state phase difference between the REF and DIV signals. The offset tri-state PFD therefore does not suffer from non-linearities associated with locking the PLL to zero steady-state phase error that the overlapping tri-state PFD does. We see in Figure 5-40 that the offset PFD based PFD/DAC synthesizer does not exhibit any residual fractional spurs or noise folding due to non-linearity. The offset PFD is, in fact, the default PFD architecture for simulations in this thesis.

Figure 5-41 shows the eye diagrams for charge transfer using the offset tri-state PFD topology. Unlike the classic tri-state and overlapping tri-state which lock to zero average phase error, the offset PFD locks to a static phase offset between REF and DIV. Positive charge contributed by the reference edge is now constant in time, and serves to act as a reference charge that the PFD/DAC balances. The result is that overall PFD linearity is improved. The output noise contribution of the tri-state PFD is determined according to equation 5.6, and will be the same as the overlapping

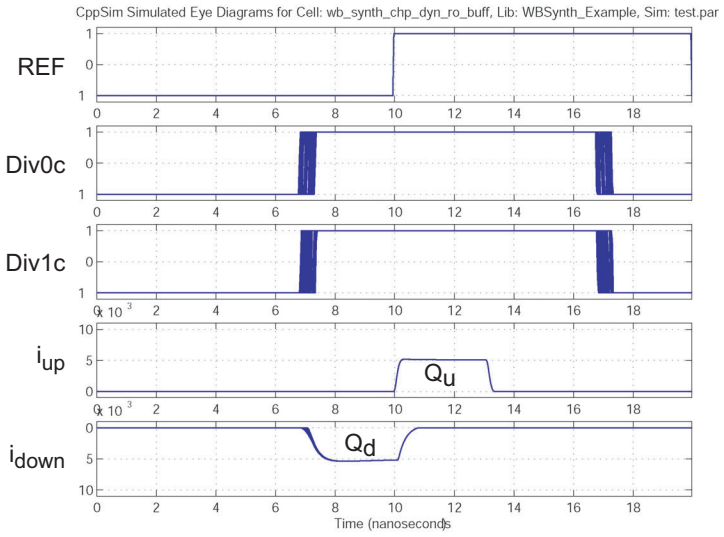


Figure 5-41 Eye Diagram of Key Signals in a PFD/DAC Employing the Offset Tri-state PFD

PFD for the same value of δt.

5.4.4 Overlapping and Offset PFD

The final tri-state architecture we present is depicted in Figure 5-42. It is a combination of the overlapping and offset tri-state architectures. It has the offset PFD advantages of having a constant positive charge packet for a reference and avoiding current magnitude mismatch non-linearity by locking the synthesizer to non-zero steady-state phase error.

Figure 5-42 shows that the overlapping and offset tri-state has the same performance as the offset PFD for the PFD/DAC synthesizer of our behavioral model. We have included the offset PFD phase noise profile as a light gray overlay in the plot. For completeness, we include eye diagrams of the key charge transfer signals in Figure 5-43. As with the offset tri-state topology, linearity is improved compared to the classic and overlapping tri-state approaches. Current noise is the same for the PFD of Figure 5-42 as for both the offset and overlapping PFD topologies, and is described by equation 5.6.

In the case where a S/H is used, the only benefit the overlapping and offset PFD

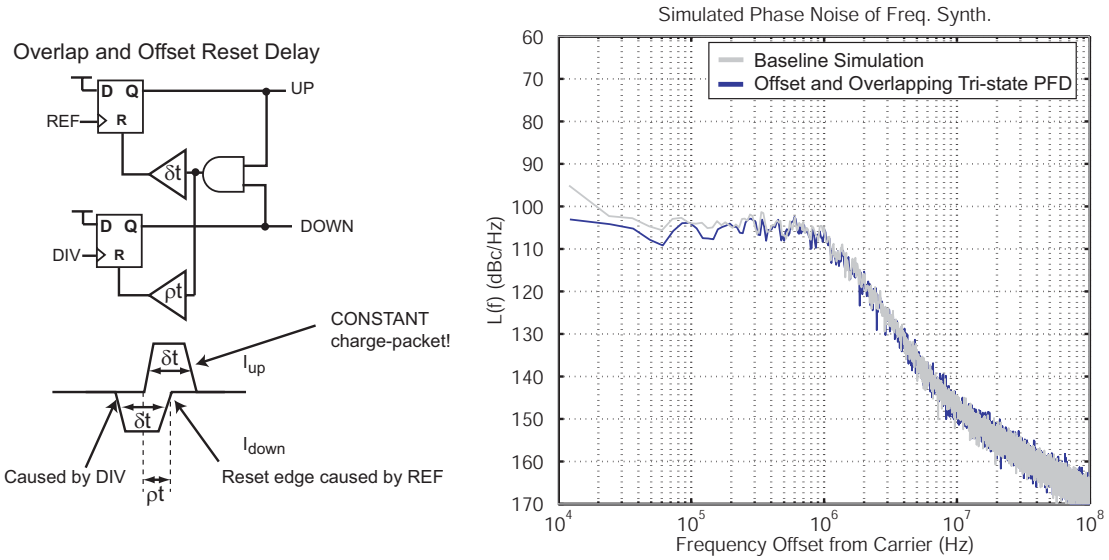


Figure 5-42 Tri-State PFD Architecture With Both Overlapping and Offset Currents

has over the offset PFD is a gain in timing margin. The timing margin in question is the time from which the PFD outputs are off to when the next PFD outputs begin. Overlapping the current pulses adds ρt seconds of additional margin here. This timing margin gain could become a potential benefit for systems with very high reference frequencies, where the reference period, and therefore the phase comparison period, is small.

5.5 GMSK Modulated Synthesizer Model

In this section we present simulation results for a GMSK (Gaussian Minimum Shift Keying) modulated PFD/DAC synthesizer [43]. To examine the worst case, we turn on all noise sources, enable all non-idealities, and assume a passive loop filter configuration. We use the offset PFD architecture with $\delta t = 3ns$.

5.5.1 Direct GMSK Modulation

The transmitter architecture is depicted in Figure 5-44. A 3.6GHz output frequency is used to generate 1.8GHz and 900MHz outputs. Random binary data is processed

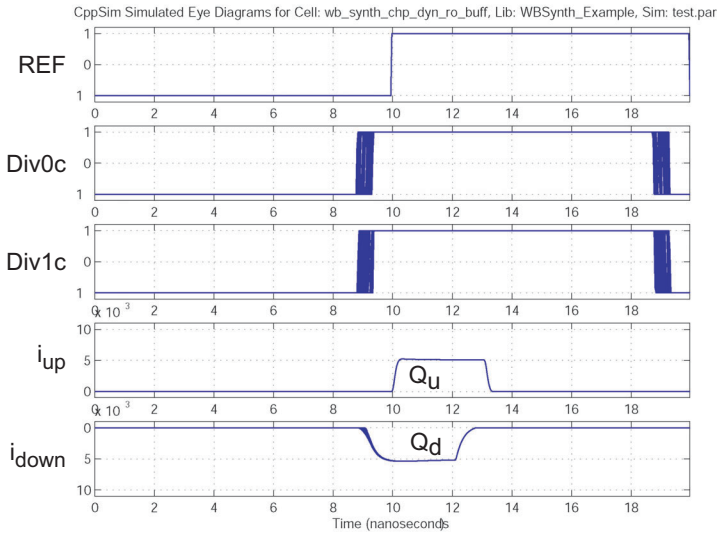


Figure 5-43 Eye Diagram of Key Signals in a PFD/DAC Employing the Offset and Overlapping Tri-state PFD

by a Gaussian filter and input to the synthesizer. 3.6GHz output is divided by either two or four depending on whether a 1.8GHz or 900MHz output band is desired.

Figure 5-45 presents simulation results for the transmitter modulated by GSM data rate 271kb/s data. The 900MHz spectrum meets the stringent GSM transmit spectral mask, and the eye diagrams for both bands are wide open.

Figure 5-46 shows that, as the data rate is increased to 500kb/s, the eyes remain wide open. Figure 5-47 reveals that, when the data rate is further increased to 1Mb/s, the eyes remain open, but inter-symbol interference (ISI) begins to close the eye. ISI is caused by the low-pass nature of the synthesizer dynamics filtering the data as it passes from the divider through to the VCO output.

In [7], a pre-emphasis technique is proposed which filters the data sequence with the inverse of the PLL transfer function. In this way, the data is pre-emphasized and overcomes the bandwidth limitation of the synthesizer bandwidth. A simple modification can be made to the digital Gaussian filter used to generate the synthesizer input. The drawback to this technique is that, if very aggressive filtering (2.5Mb/s GFSK data was transmitted through a 80kHz bandwidth synthesizer in [7]) is desired, the closed loop PLL transfer function must be known to within 10%. A technique

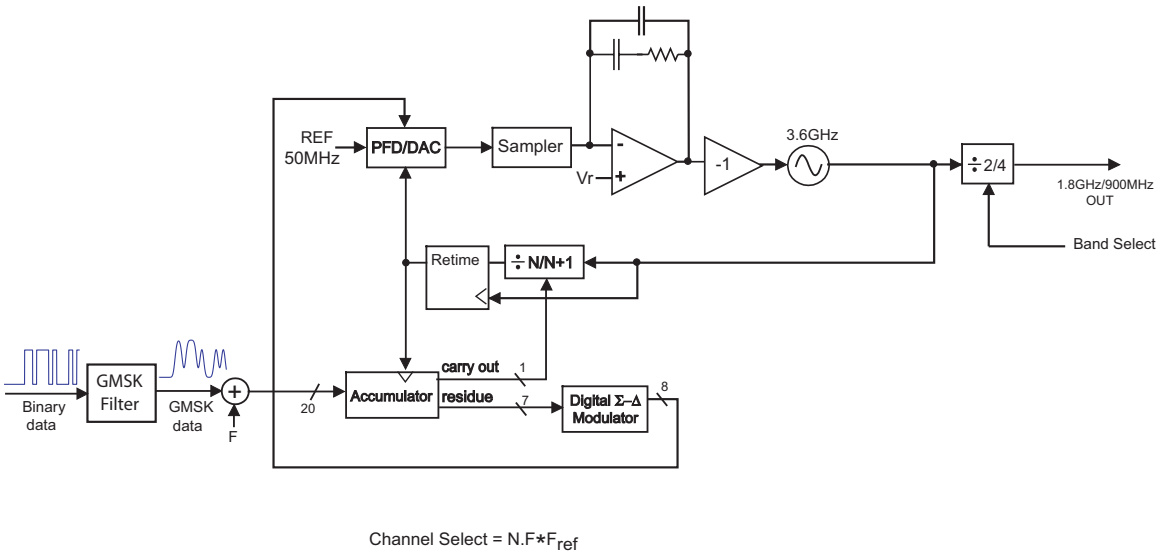


Figure 5-44 Dual Band GMSK Transmitter

in [34] is proposed to actively calibrate such a pre-emphasis filtering system. For the PFD/DAC system simulated in this chapter, the very high, 1MHz, closed loop bandwidth suggests that much less aggressive pre-filtering will be required to achieve multi-Mb/s data rates.

5.6 Summary

In this chapter, techniques to model and simulate high performance fractional-N synthesizers were proposed. In particular, results of Hspice simulations demonstrating non-ideal behavior of circuit blocks were integrated into the behavioral model to enhance its validity. By incrementally increasing the complexity of the model, a better intuition can be gained as to the actual bottlenecks in the overall system operation. A discussion of various tri-state PFD architectures was presented. Based on simulations of each tri-state topology, we have shown that best overall performance is achieved by employing an offset PFD architecture. Simulation of the synthesizer configured as a GMSK transmitter demonstrated the ability of the system to transmit data rates in excess of the GSM standard 271kb/s without significant degradation to the received data eye diagram. In Chapter 6 we will present circuits designed to implement the

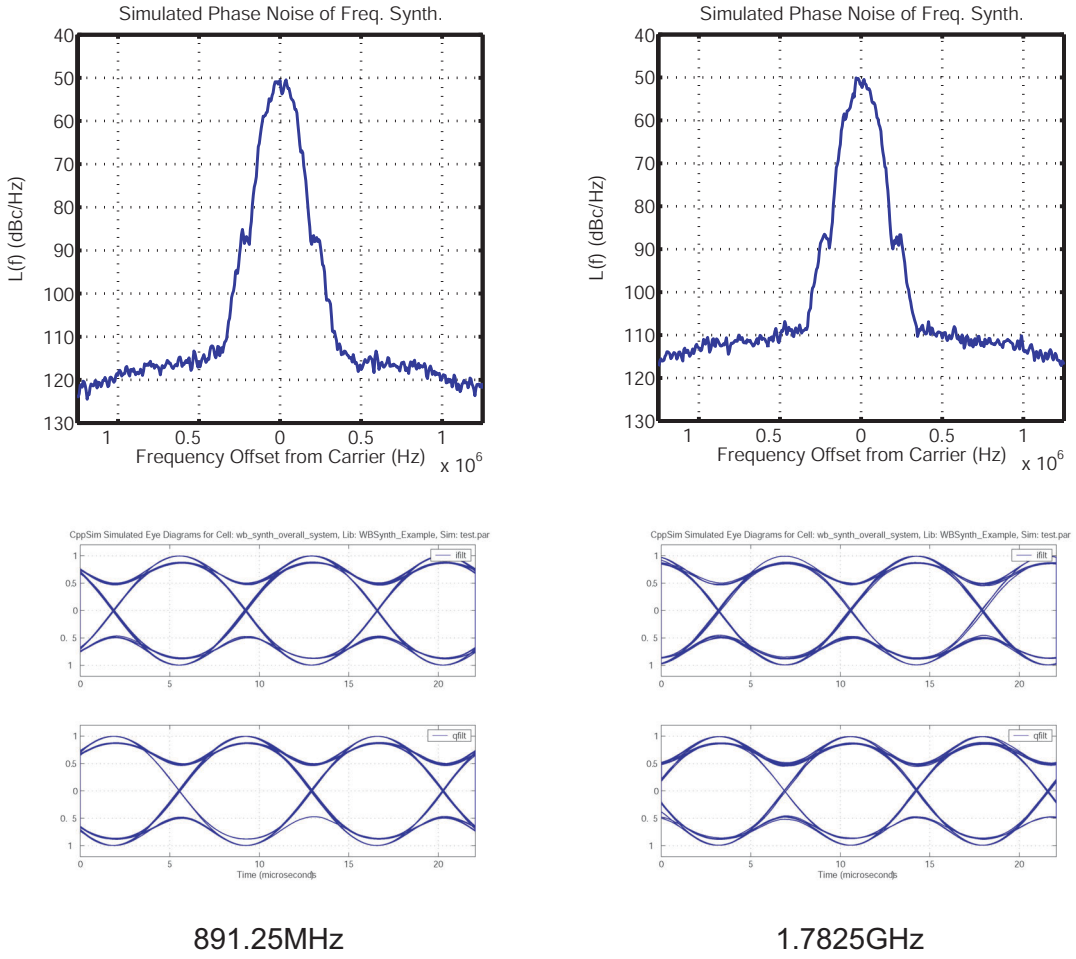
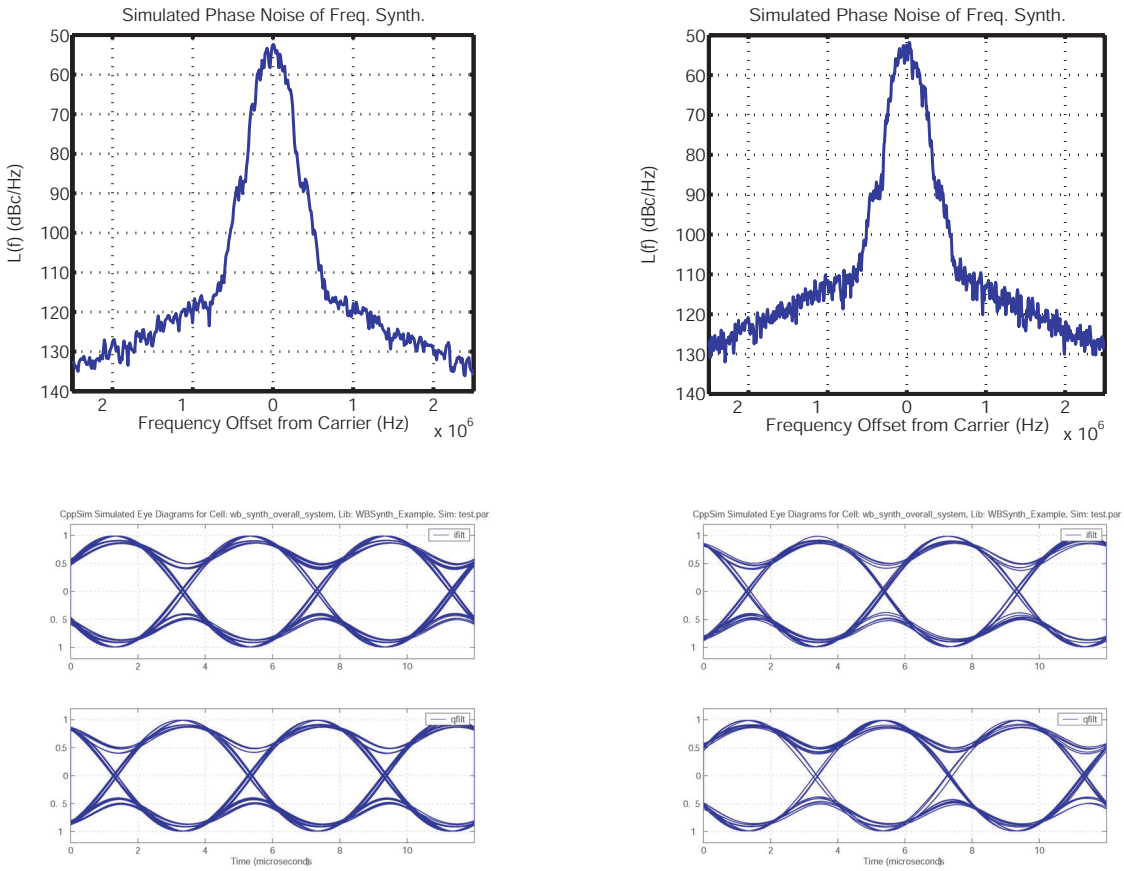


Figure 5-45 Simulated Output Spectra and Eyes for 271kb/s GMSK Modulated PFD/DAC Synthesizer

PFD/DAC synthesizer.



891.25MHz

1.7825GHz

Figure 5-46 Simulated Output Spectra and Eyes for 500kb/s GMSK Modulated PFD/DAC Synthesizer

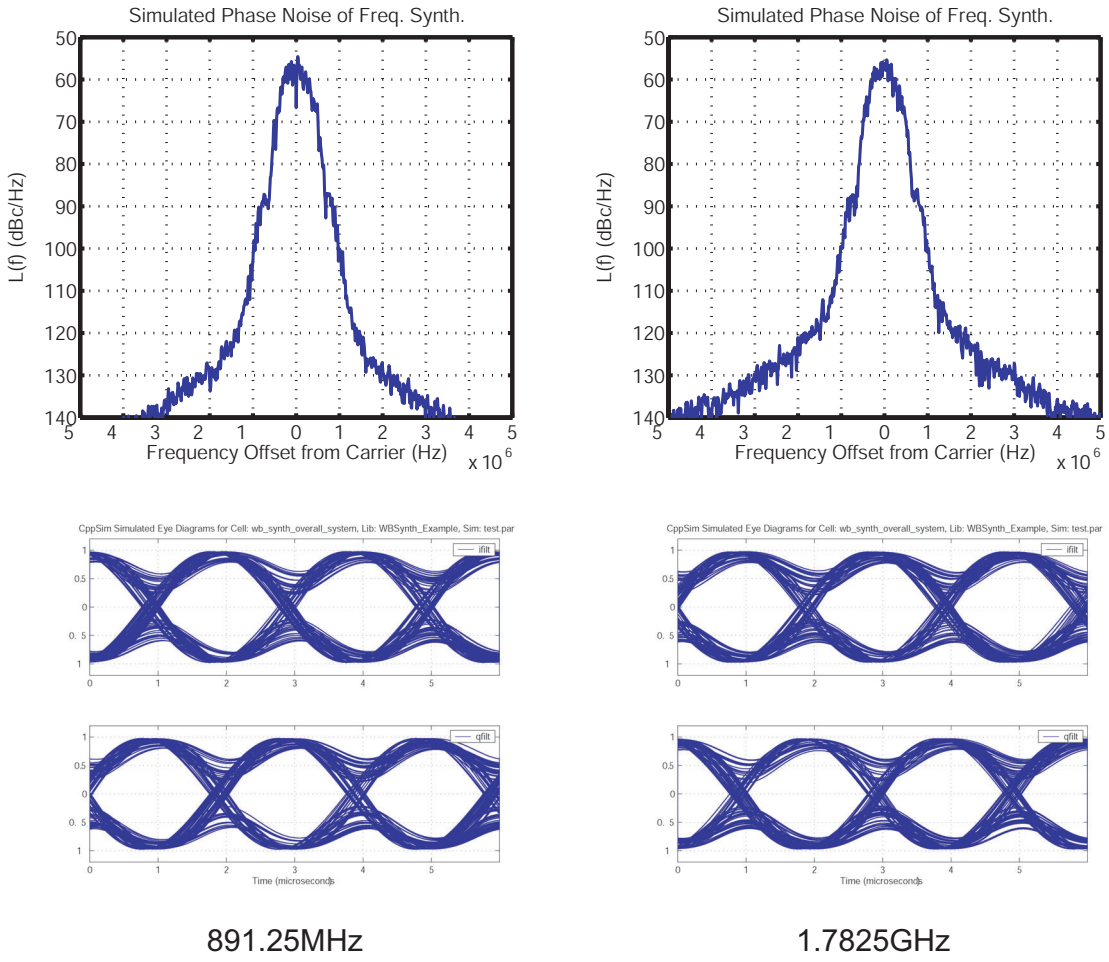


Figure 5-47 Simulated Output Spectra and Eyes for 1Mb/s GMSK Modulated PFD/DAC Synthesizer

Chapter 6

Circuit Design

The key to the PFD/DAC synthesizer architecture is the generation of a well defined charge-box. Since the charge-box is referenced to a single VCO period, this implies that the mismatch compensated PFD/DAC must generate fast edge rates to generate an accurate time window. In the prototype synthesizer fabricated for this thesis, the nominal VCO output frequency is 3.6GHz, so the charge-box is referenced to 278ps, a very small time interval.

In addition to the need for fast edge rates, noise performance is also a primary concern. As the behavioral simulations of Chapter 5 demonstrate, it is possible to eliminate the fractional-N quantization noise impact on the synthesizer output spectrum. Other noise sources in the system, such as charge-pump and mismatch noise, become dominant, and so it is desirable to reduce their magnitudes as much as possible. Therefore, circuit noise reduction techniques are also proposed.

We begin with discussion of the circuits used to construct the mismatch compensated PFD/DAC structure depicted in Figure 6-1. As has been discussed in Chapters 3 and 5, there are four main sources of systematic error that must be accounted for in the PFD/DAC to obtain maximum performance. These errors are magnitude mismatch between the DAC unit elements, timing mismatch between the phase paths used to construct the quantization noise cancellation window, initial re-timing error due to the asynchronous divider, and shape mismatch between the error and cancellation signals. To minimize the impact of these four error sources we propose

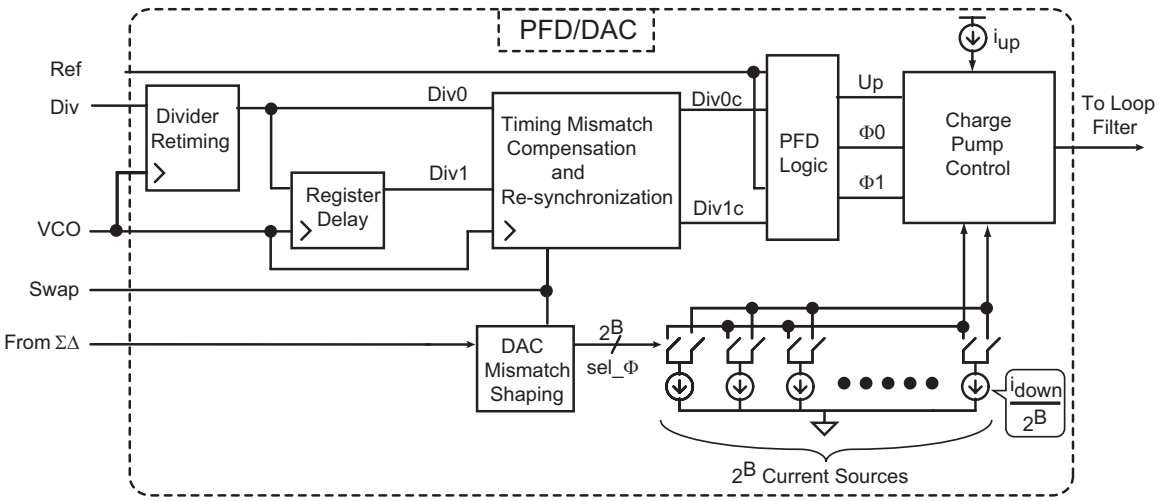


Figure 6-1 Mismatch Compensated PFD/DAC Circuit Block

four circuit blocks: a DAC mismatch shaping block, a timing compensation block, a divider retiming block, and a S/H block.

In addition to these four key blocks, we also aim to satisfy the low frequency noise specification of -110dBc/Hz, a very aggressive goal, which requires low noise design of the unit element current sources. The charge window being resolved in the system is only 278ps wide, so the PFD logic and DAC control must operate with fast edges to create an accurate charge-box. Circuit techniques for low-noise, high-speed performance will be presented.

Finally, we will provide details of the numerous support functions that need to be designed on any integrated circuit, such as bias, I/O, single-ended-to-differential and differential-to-single-ended conversion.

6.1 Divider and Divider Retimer

The proposed target application for the PFD/DAC synthesizer is a dual-band (1.8GHz/900MHz) GMSK transmitter, as discussed in Chapter 5. We will divide down a 3.6GHz output to create the two desired bands, and we will use a 50MHz reference frequency.

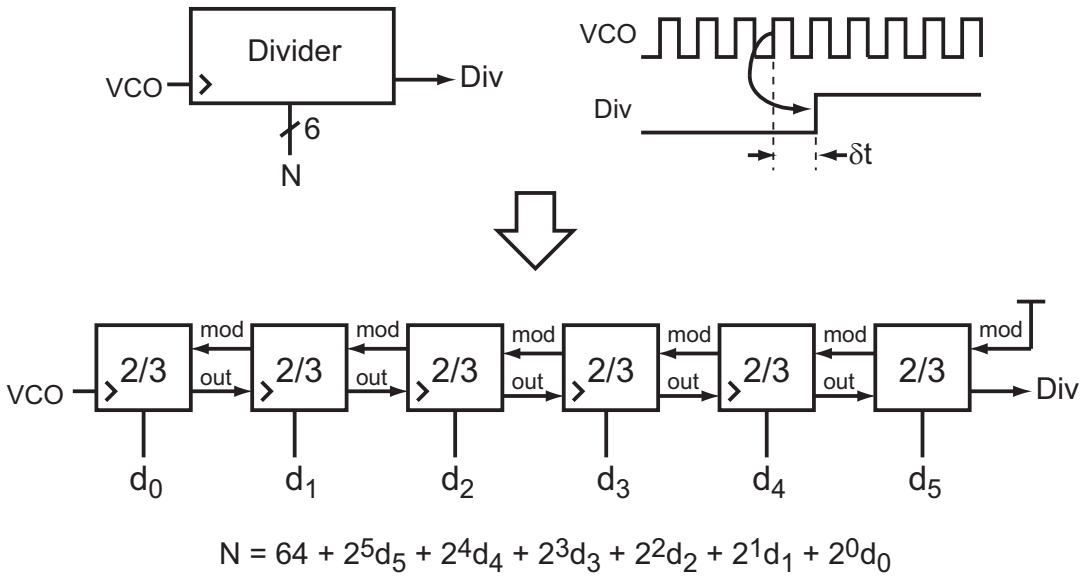


Figure 6-2 High-Speed, Asynchronous, Multi-Modulus Divider

6.1.1 High-speed, Multi-modulus Divider

In order to be able to cover the desired transmit range of 890MHz to 915MHz, we require the synthesizer output to span from 3.56GHz to 3.66GHz, corresponding to a divide range of N=71.2 to N=73.2. The divider architecture chosen is the high speed, asynchronous, multi-modulus architecture proposed in [27]. Consisting of divide-by 2/3 stages, the divider, depicted in Figure 6-2, operates by selectively "swallowing" pulses from preceding stages. For the six stage divider utilized in the prototype synthesizer, the divider is capable of producing divide values in the range of N=64 to N=127.

The asynchronous nature of the divider means that there is an unknown delay, δt, between the VCO rising edge and divider output rising edge. The divider output therefore must be re-synchronized to the VCO before the PFD/DAC can use it to generate the two required divider phases that make up the charge-box.

6.1.2 Retiming and the Issue of Meta-stability

Meta-stability is an issue that can arise when an asynchronous signal, such as the divider output, is resynchronized to the VCO by simply clocking it into a flip-flop. If

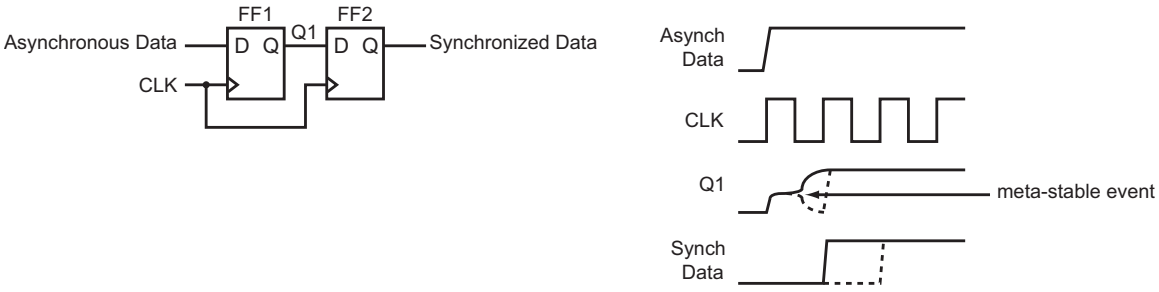


Figure 6-3 Synchronization in a Digital System

the setup and hold timing constraints of the flip-flop are not met, the flip-flop output may evaluate to an incorrect value, and may also take a long time to evaluate [44,45]. Digital data systems that synchronize asynchronous signals use a cascade of flip-flops, as depicted in Figure 6-3, to reduce the overall probability that a meta-stable event is *observed* by the system [44, 45].

In the example of Figure 6-3, flip-flop FF1 experiences a meta-stable condition because the asynchronous data signal transitions during its setup-and-hold window. FF1's output, Q1, may resolve to either a digital high or low, according to some probability distribution described in [44, 45]. In the figure, Q1 is shown to evaluate to a high, but the dashed line represents the possibility that it evaluates to a low. Using a second flip-flop, FF2, to evaluate FF1's output reduces the probability that the digital system downstream will see a meta-stable signal, because an entire clock period (minus a setup and hold time for FF2) is allowed for Q1 to settle to a valid value before it is clocked into FF2. Use of regenerative latch input stages in FF1 and FF2 help the decision process resolve faster [44, 45].

The problem with using the simple resynchronization solution shown in Figure 6-3 in a fractional-N synthesizer is that, while the circuit of Figure 6-3 can shield the system from observing a meta-stable output, it allows the possibility that the synchronizer will resolve to an incorrect value. If the synchronizer experiences a situation where it sometimes follows the dashed path in the figure, and sometimes follows the solid path, then the effective delay through the synchronizer dynamically changes by a VCO period. Since the asynchronous data in this instance is the divider output, a dynamically changing synchronizer delay is equivalent to adding a jitter

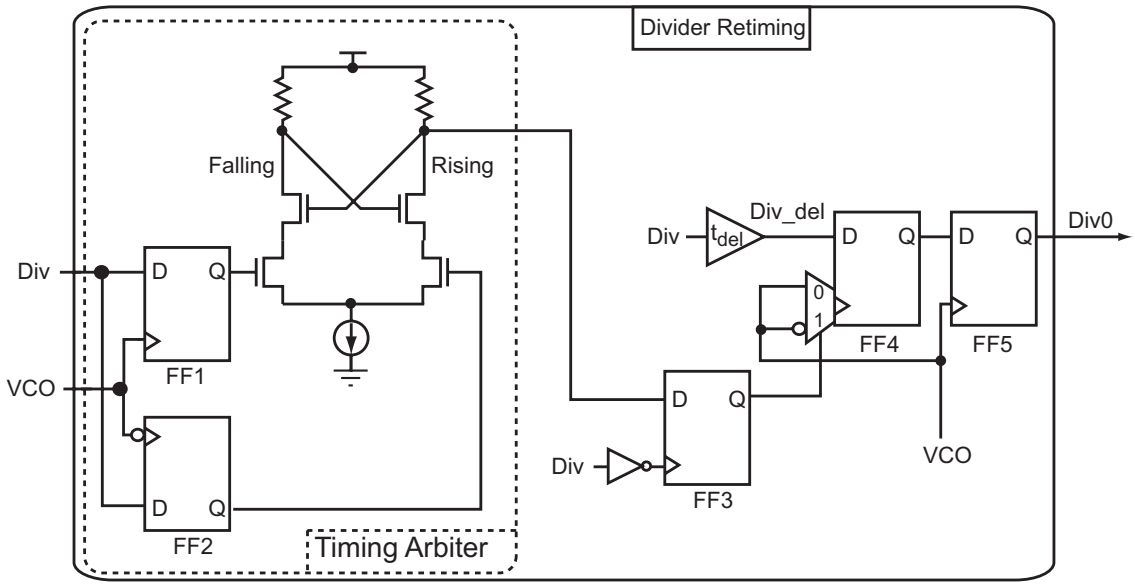


Figure 6-4 Divider Retiming Circuitry

noise source to the divider output, which increases noise in the system. For this reason we desire a synchronizer solution that not only reduces the probability of observing a meta-stable event, but one that consistently evaluates to the same output value.

6.1.3 Divider Retimer

Figure 6-4 presents the architecture proposed for the divider retiming circuit. Unlike prior approaches [19, 28] that use information propagating through the stages of the asynchronous divider, the circuit in Figure 6-4 directly determines if there is a meta-stable event and re-times accordingly.

The outputs of two flip flops, FF1 and FF2, are input to a high speed differential decision circuit that decides whether rising edge retiming or falling edge retiming evaluates first. The differential, cross-coupled load decision circuit is based on a dynamic design proposed in [46]. The arbiter output is sampled by a low-speed flip-flop, FF3, which is clocked on the divider falling edge. In order to add margin to the retiming, the divider is delayed by a time, t_{del}, equivalent to slightly more than a setup and hold time via a buffer delay stage, and then clocked into FF4 on the *opposite edge* output from the arbiter. FF5 is a final retiming flop that ensures that the PFD/DAC

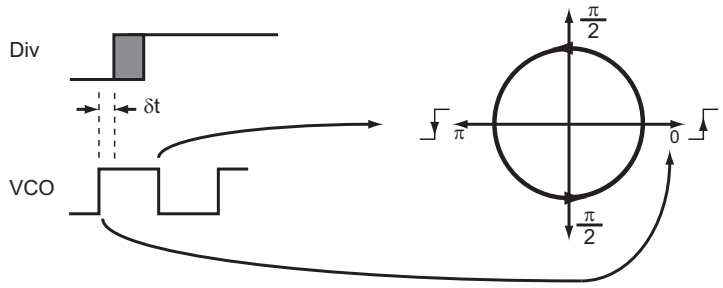


Figure 6-5 Mapping the VCO Edges to Phase-space

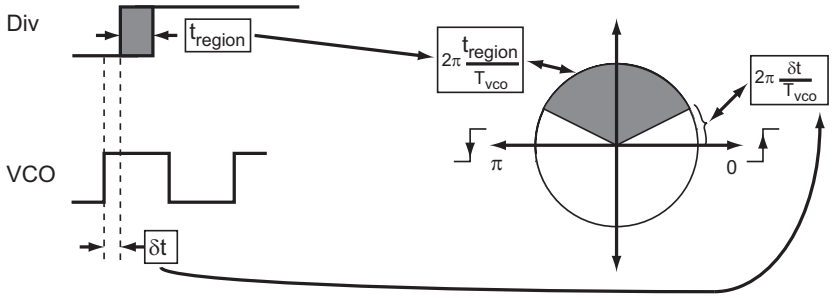


Figure 6-6 Mapping the Divider signal to Phase-space

receives a retimed edge that is always synchronized to the VCO rising edge. This final output, labeled Div0, is passed to the rest of the mismatch Compensated PFD/DAC structure depicted in Figure 6-1.

6.1.4 A Phase-space Methodology for Understanding Divider Retiming

We will use a phase-space representation to explain the divider retimer operation. A step-by-step method to map signals from the time domain to phase space begins with Figure 6-5. We define the VCO period as being the phase-space variable of interest. The VCO period therefore corresponds to 2π radians. The VCO rising edge is defined as being 0 radians, while the falling edge is mapped to π radians.

The second step is to map the asynchronous delay between the VCO rising edge and beginning of the region where the divider edge can occur, δt, as depicted in Figure

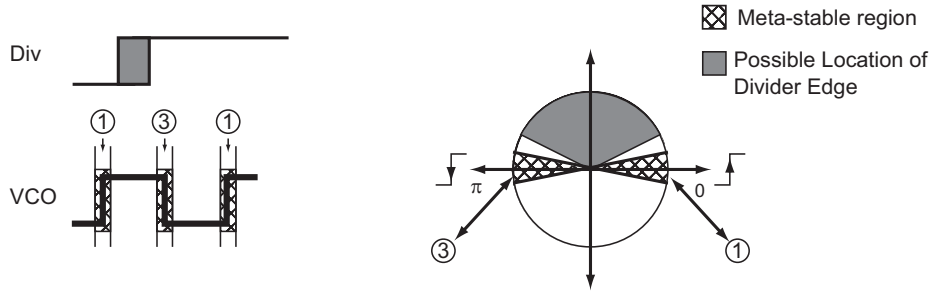


Figure 6-7 Mapping the Meta-stable Timing Regions to Phase-space

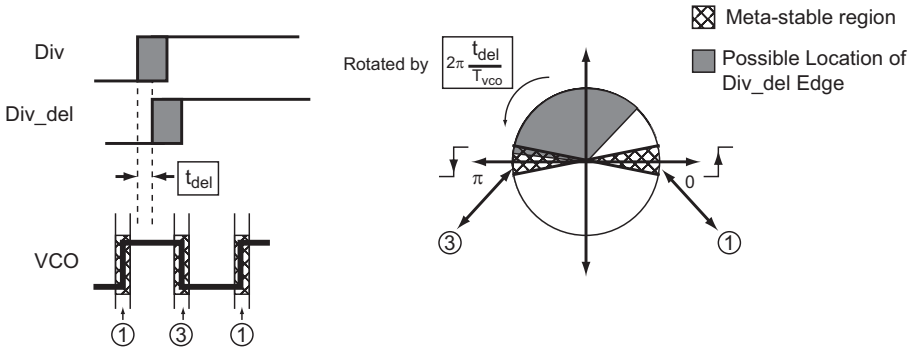


Figure 6-8 Mapping the Delayed Divider Signal to Phase-space

6-6. The asynchronous delay mapping takes place according to

$$\delta_t \quad seconds \rightarrow \text{maps to} \rightarrow 2\pi\frac{\delta t}{T_{vco}} \quad radians, \tag{6.1}$$

where, T_{vco} is the VCO period. We define a region where the divider edge may occur as t_{region}, and represent it by a shaded region in the time domain that maps to a shaded region in phase space that spans

$$t_{region} \quad seconds \rightarrow \text{maps to} \rightarrow 2\pi\frac{t_{region}}{T_{vco}} \quad radians. \tag{6.2}$$

The third step in creating a phase-space representation of the divider retimer is to map the meta-stable retiming regions. These are the banded regions 1 and 3 around the VCO rising and falling edges in Figure 6-7. If the divider edge location falls in region 1, arbiter flip-flop FF1, which clocks the divider on the VCO rising edge, experiences meta-stability. Likewise, if the divider edge falls in region 3, arbiter

flip-flop FF2, which clocks the divider on the VCO falling edge, experiences meta-stability. The retimer output is not taken from either FF1 or FF2, however, which leads us to the final step in phase-space mapping for the divider retimer.

The final step in the mapping process is depicted in Figure 6-8. Because the retimer ultimately produces an output that synchronizes a delayed divider signal, *Div_del* in Figure 6-4, we account for the delay, t_{del}, by rotating the divider edge signal according to

$$t_{del} \quad seconds \rightarrow \text{maps to} \rightarrow 2\pi\frac{t_{del}}{T_{vco}} \quad radians. \tag{6.3}$$

In the example used in the phase-space mapping derivation, the phase rotation depicted in Figure 6-8 has created a situation where, if the delayed divider is retimed on the VCO falling edge, meta-stability will result. This is represented by the fact that the delayed divider signal shaded region overlaps with the falling edge meta-stable region.

6.1.5 Divider Retimer Operation In Phase-space

Having arrived at a phase-space explanation for meta-stability in the divider retiming application, we proceed with operation of the actual retimer circuit proposed for use in the prototype synthesizer IC.

Figure 6-9 presents the phase-space explanation of the divider retimer circuit operation. The possible location of the divider edge is divided into four regions with respect to the VCO edge. There are two regions where meta-stability can result, Regions 1 and 3, and two regions where meta-stability is avoided, Regions 2 and 4. The actual location of the divider edge within a region is unknown, and can vary continuously as the edge moves around due to temperature variations, changes in the divide value, or jitter, and so the actual location within a given region is represented by a gray shading. The left side of the plot is the phase-space representation of the arbiter circuit that synchronizes the divider output, and the right side of the plot is the phase-space representation of the output synchronizer FF4 that operates on the

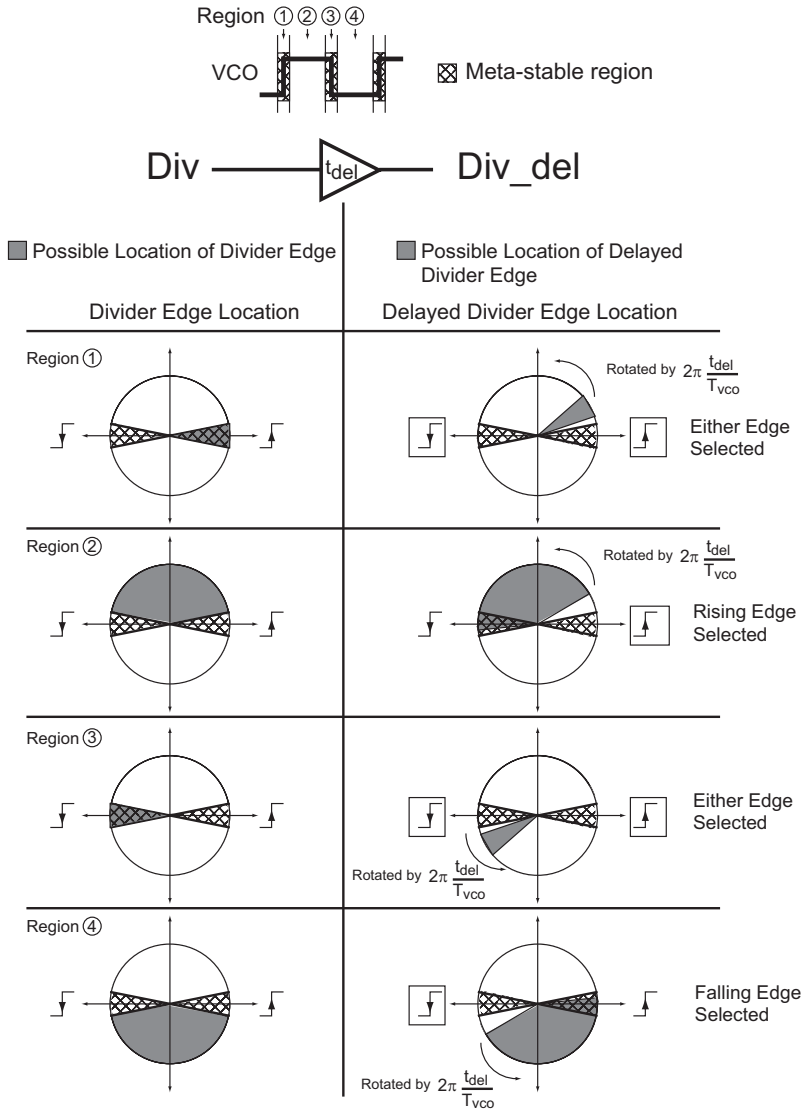


Figure 6-9 Phase Space Explanation of the Divider Retimer Circuit

delayed divider signal.

Here we examine the behavior of the arbiter circuit, corresponding the left side of Figure 6-9.

- Region 1: The divider edge falls such that arbiter flip-flop FF1 is meta-stable. It is possible that either FF1 or FF2 will produce a valid output first, depending on where within the meta-stable region the divider edge occurs.

- Region 2: The divider edge falls such that both arbiter flip-flops produce valid outputs and are not meta-stable. Because the divider edge occurs *after* the

VCO rising edge and *before* the falling edge, falling edge triggered arbiter FF2 will always produce a valid output first.

- Region 3: The divider edge falls such that arbiter flip-flop FF2 is meta-stable. It is possible that either FF1 or FF2 will produce a valid output first, depending on where within the meta-stable region the divider edge occurs.

- Region 4: The divider edge falls such that both arbiter flip-flops produce valid outputs and are not meta-stable. Because the divider edge occurs *after* the VCO falling edge and *before* the rising edge, rising edge triggered arbiter FF2 will always produce a valid output first.

Having established the behavior of the arbiter circuit, we now explore the operation of the output synchronizer, FF4, as the arbiter operates in each region. This corresponds to the right side of Figure 6-9. In region 1, the arbiter may select to clock the delayed divider signal on either rising edge or falling edge. Either way the decision is a good one, because the delayed divider signal present at the retiming flip-flop FF4 has been delayed by enough time so that its the edge location, as represented by the shaded area, does not fall within a meta-stable region. Simulations of the retiming circuit suggest that during operation in region 1, the falling edge is always valid first. However, the delay through the meta-stable flip-flop depends on many factors, and so there is a finite (but small) probability that the arbiter could, under some circumstances, determine that the rising edge was valid first. For this reason, we propose that, for maximum stability, the retiming circuit not be run continuously. This is to avoid the rare case that the divider edge occurs in region 1 at such a time that it sometimes causes the arbiter to choose the rising edge and sometimes the falling edge, which would appear as a change in the effective divide value, and therefore as an additional noise source. The preceding arguments apply for region 3.

In region 2, the divider edge occurs such that the falling edge triggered flip-flop FF2 will always produce a valid output first, and therefore trigger the decision circuit. Delaying the divider edge in this region can introduce meta-stability. This is evidenced by the fact that, in Figure 6-9, the divider signal in region 2 does not overlap with a

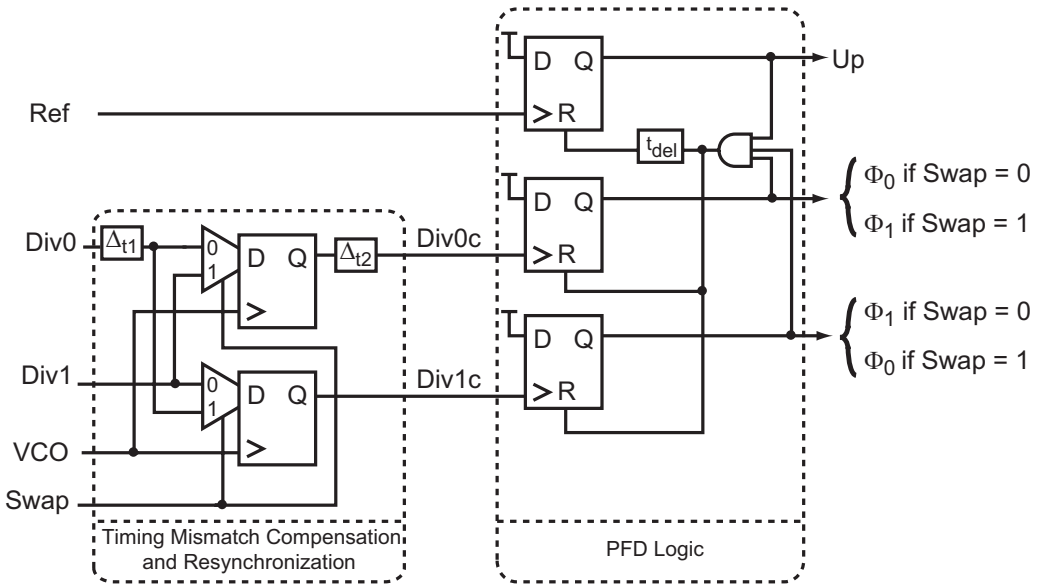


Figure 6-10 Timing Compensation and Resynchronization Block and PFD Logic

meta-stable region, but the delayed divider region does. However, since the retimer ultimately re-clocks output FF4 on the *opposite* edge from that which the arbiter chooses as evaluating first, meta-stability is avoided. The same arguments apply to region 4.

In short, the proposed retiming circuit delays the divider phase input to retiming flip-flop FF4 to solve the problem of being in the meta-stable region of either retiming edge (Regions 1 and 3). The choice to re-clock on the opposite edge is to account for those times when delaying the divider phase would otherwise hurt performance by introducing meta-stability (Regions 2 and 4).

As has already been mentioned, to avoid the rare situations where the arbiter makes time varying decisions due to operating in its meta-stable regions, we recommend operating the retimer in a burst mode. Tracking of environmental changes can be done by periodically operating the retimer.

6.2 PFD Logic and Timing Compensation

Once the divider is properly re-timed, it is used to create the two divider phases that are processed by the phase detector logic. Mismatch between the two divider phases

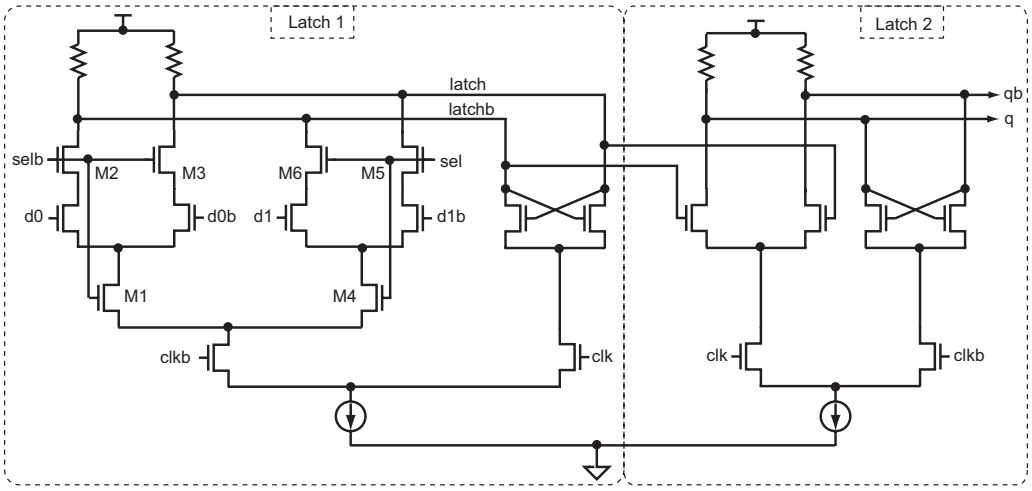


Figure 6-11 High Speed Differential Flip-Flop with Muxed Input Stage and State Mismatch Compensation

is averaged out via the timing compensation and resynchronization circuitry shown in Figure 6-10, which was discussed in Chapter 4.

Differential, source-coupled logic (SCL, also known as current mode logic (CML)) is used to achieve high speed operation [47, 48]. The flip-flops used by the timing compensation block have muxes embedded in the first latch stage to save power and area and increase speed. One of these flip-flops is shown in Figure 6-11. Typically, only transistors M1 and M4 are used in the first latch stage if an input multiplexer function is desired. However, without transistors M2, M3, M5, M6, there will be a change in loading at the internal nodes *latch* and *latchb* as the de-selected phase input signals switch. Transistor pairs M2,M3 and M5,M6 isolate the first stage latch nodes from the de-selected input pair, and eliminate this input state dependent mismatch, reducing residual timing error Δ_{t2}.

The phase swapping process is controlled by a 23 bit LFSR acting as a random number generator with an average duty cycle of 0.5 [49]. The phase swapping process results in a white noise PSD contributing to output phase noise, as demonstrated in the behavioral simulation results presented in Chapter 5.

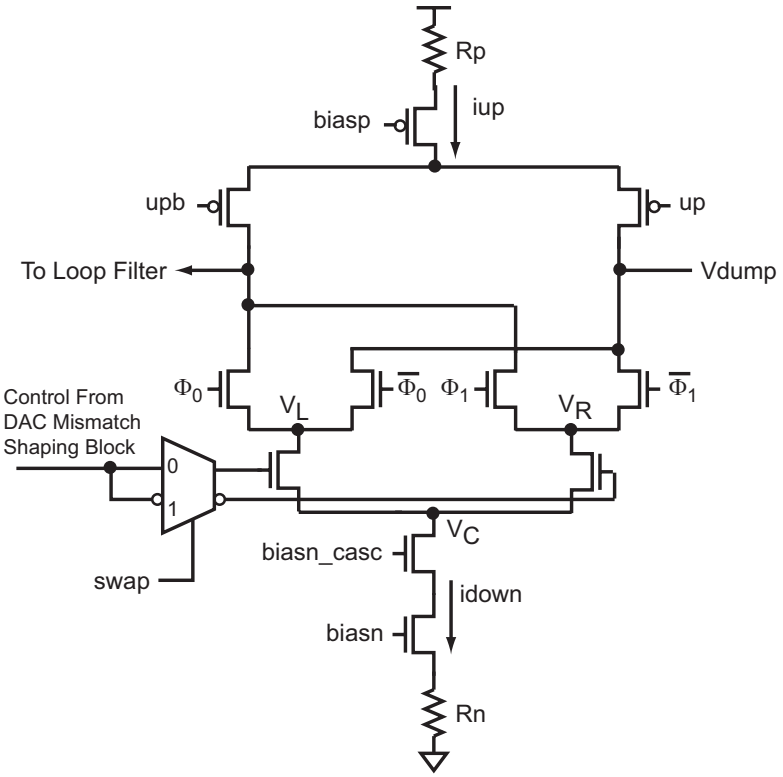


Figure 6-12 PFD/DAC Unit Element

6.3 PFD/DAC Unit Element Current Source

The proposed unit element used to construct the PFD/DAC is depicted in Figure 6-12. For high speed operation, a differential structure is employed. As the divider phases are swapped by the timing mismatch compensation and resynchronization block in Figure 6-10, the phase selection control of the unit element must be swapped as well. The multiplexor accomplishes this phase swapping function. The same swap signal that controls phase swapping in the PFD determines which phase, Φ_0 or Φ_1, switches the unit element current.

Switching transients can disturb the value of output current presented to the loop filter. Node V_C in the unit element circuit of Figure 6-12 is particularly sensitive to switch transients, since any voltage deviations at V_C will change the unit element output current according to its output impedance [50]. To minimize voltage transients at node V_C, two techniques are used. First, the swap signal is switched on the falling edge of the divider. Phase comparisons in the PFD are done on the divider rising

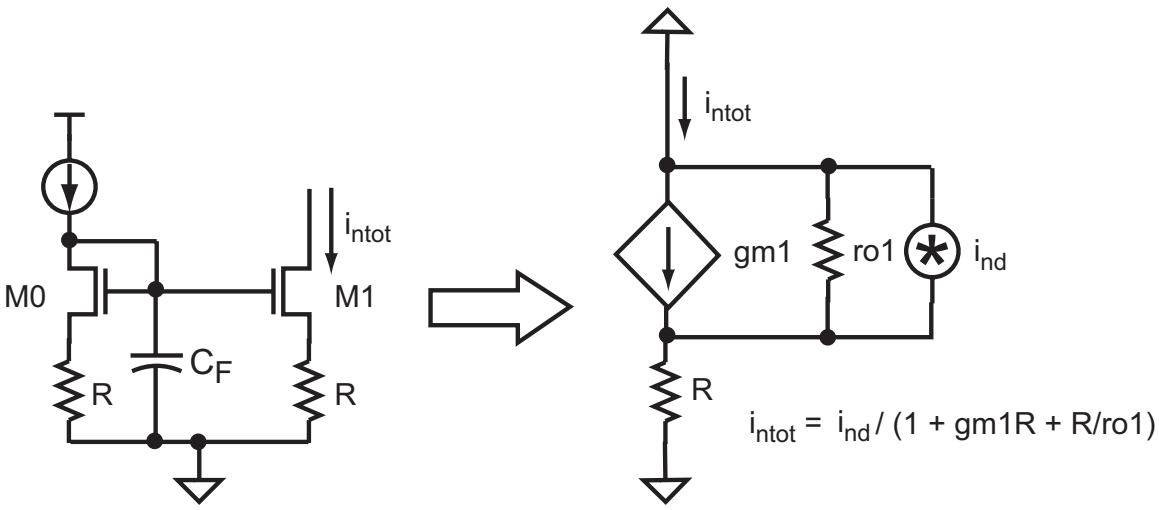


Figure 6-13 Resistive Degeneration for Reduced Current Source Noise

edge, and therefore Φ_0 and Φ_1 are also generated by the divider rising edge. Switching the swap signal on the divider falling edge gives half a divider period (10ns) for any voltage transients on all of the differential pair common sources nodes V_L, V_R,and V_C to settle before an output is required. The second technique is to apply a constant voltage, V_{dump} at the unused output node. If the active loop filter of Figure 5-32 is used, and $V_{dump} = V_{nom}$, the conditions at the drains of all differential pairs is nominally the same, and switching transients during operation of the Φ_0 and Φ_1 differential pairs is minimized.

Resistive degeneration is employed to reduce the magnitude of current noise produced by the unit element current source [51]. Figure 6-13 presents a small signal model for a current source with resistive degeneration. The resistor creates a feedback loop whereby, as the source voltage of M1 increases due to noise, the gate-to-source voltage decreases, reducing the current through M1, and thereby lowering the drain current noise. The amount of noise reduction can be calculated from the small signal model equivalent of the M1 and R current source circuit. The output noise power of the circuit in Figure 6-13 is

$$\overline{i_{ntot}^2} = \left| \frac{1}{1 + gm_1 R + \frac{R}{r_{o1}}} \right|^2 \cdot \overline{{i_{nd}}^2} \quad A^2/Hz \quad , \tag{6.4}$$

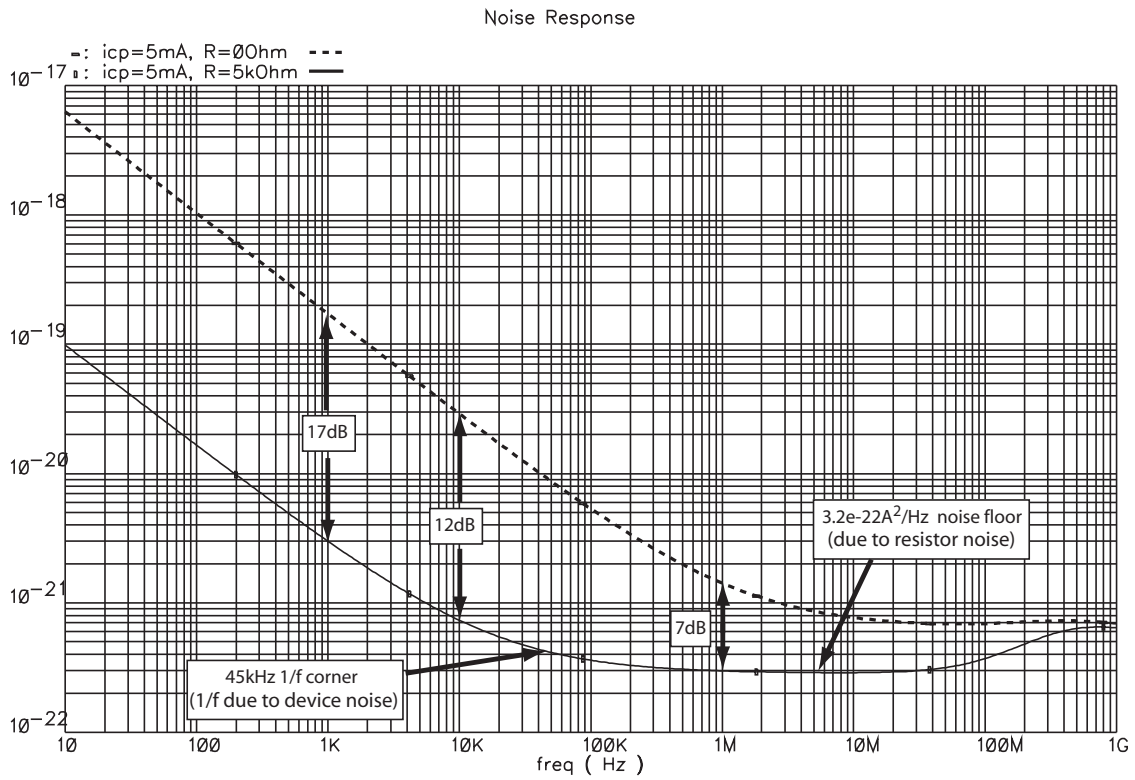


Figure 6-14 Hspice Simulation of Charge-pump Noise With and Without Resistor Degeneration

where $\overline{i_{nd}^2}$ is the current source device's drain current noise PSD, r_o is the device output impedance, and we have ignored, for now, the noise contributed by the degeneration resistor. For reasonable values of gm, R, and ro, the circuit can achieve 20-40dB of noise attenuation. Hspice simulations result in a final noise profile for the unit element current sources of $3.2e-22\ A^2/Hz$ thermal noise with a 1/f corner frequency of 45kHz.

The tradeoff associated with using resistor degeneration is that voltage headroom is sacrificed due to the voltage drop across the resistor. Additionally, the resistor adds some current noise of its own that will appear unattenuated at the unit element output. In the prototype IC, $R = 5e3$, and $i_{element} = 50\mu A$, which leads to a 250mV sacrifice in headroom. The output current noise of the $5k\Omega$ resistor is

$$\overline{i_{nR}^2} = \frac{4kT}{R} = 3.2e-24\ A^2/Hz \quad . \tag{6.5}$$

A simulation of the resistively degenerated NMOS current sources demonstrates $17dB$ of output referred current noise attenuation, as depicted in Figure 6-14. In the simulation, all 128 unit elements were active and the output charge-pump current was set to 5mA. We see that low frequency noise is attenuated by 17dB while the ultimate low frequency noise floor of $3.2e{-}22\ A^2/Hz$ is set by the total noise contributed by the degeneration resistors. In other words, the device noise has been attenuated enough that resistor noise becomes the thermal noise limit. The effective 1/f corner of output frequency noise has been reduced from 1.2MHz to 45kHz. A last point to note is that, above 100MHz, the degeneration resistor begins to be shorted out by parasitic capacitance, and output noise rises until it follows the profile of the non-degenerated circuit. Since the PLL bandwidth is set to 1MHz and this high frequency noise is filtered by the PLL, this behavior is not a concern. The simulated final output current noise for a 7-bit PFD/DAC was

$$\overline{i_{ntot}^2} = 3.2e-22\ A^2/Hz \quad . \tag{6.6}$$

We can use equation 5.6 to find the output current noise and convert to dBc/Hz, resulting in:

$$S_{\Phi_{out}|\overline{i_{ntot}^2}}(f) = 10\log\left(\overline{i_n^2}\left(\frac{2\pi N_{nom}}{I_{cp}}\right)^2 |G(f)|^2 \cdot D\right) \quad \text{dBc/Hz}, \tag{6.7}$$

In the equation, N_{nom} is the nominal divide value, I_{cp} the full-scale charge-pump current, and D the duty cycle of the charge-pump. Using the nominal parameter values, we find that output referred charge-pump noise power spectral density is simulated as -123dBc/Hz, indicating that the added noise due to the degeneration resistor is negligible. The 1/f corner frequency in unit element noise simulations is approximately 45kHz using the degeneration scheme, down from approximately 1.2MHz before degeneration was added. To offset the lost headroom due to degeneration, the unit elements are biased to be wide-swing current sources [52].

The final point to note about the unit element current sources is that only the

NMOS current source is cascoded. As has been discussed and simulated in Chapter 5, the offset tri-state PFD architecture offers enhanced linearity by creating a constant reference charge packet via the UP current that is then canceled by the PFD/DAC controlled DOWN current sources. Since the positive charge is always the same value and simply used as a reference, the output impedance of the positive current sources is not a critical performance parameter. Because an offset PFD architecture is chosen, the positive and negative current sources are on at different times, and so their output impedances are isolated from one another. The tolerance of the proposed mismatch compensated PFD/DAC synthesizer to low PMOS current source output impedance was demonstrated in the behavioral simulations of Chapter 5, where a 34X difference between positive current source output impedance and negative current source output impedance was shown to not adversely affect output phase noise performance.

Voltage headroom is sacrificed by the addition of resistor degeneration. The behavioral simulation results of Chapter 5 show that the offset PFD performs well in the face of poor positive current source output impedance. We use this information to make a decision to not cascode the positive current sources. The negative current sources are cascoded since their performance is critical to achieving a high quality noise cancellation in the charge-box. The decision to not cascode the positive current sources, and to "spend" the headroom on cascoding the negative current sources, which are more critical to performance, demonstrates the power of iterating between SPICE level and behavioral level simulations.

6.4 Loop Filter

The loop filter is configured as an active lead-lag filter, as depicted in Figure 5-32. The op-amp used by the filter is presented in Figure 6-15, and is a modified version of a basic two-stage architecture [52, 53]. Op-amp noise adds directly to VCO input referred noise, an issue that will be addressed in section 6.6.

The input stage tail current source, consisting of devices M12 and M10, is biased to act as a wide-swing current source. The current mirror load on the first stage,

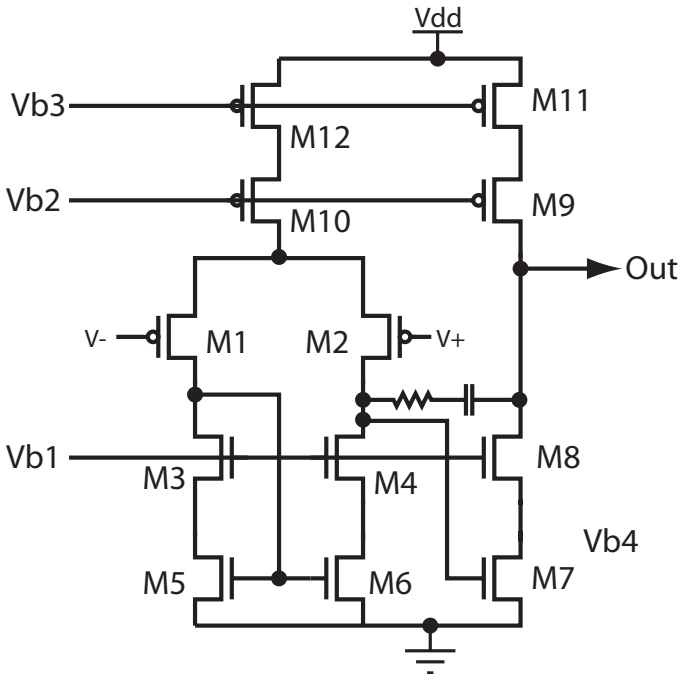


Figure 6-15 Loop Filter Op-amp

consisting of devices M3, M4, M5, and M6, is also configured in a wide-swing topology. The current mirror load is cascoded for several reasons. Primarily, it allows the output stage to be cascoded and simultaneously achieve a wide-swing and low systematic offset [52]. Second, the cascoding provides an extra 6dB of DC gain in the first stage when compared to an un-cascoded load. The first stage gain is

$$A_{v1} = gm_1 R_{tot1} \quad , \tag{6.8}$$

where R_{tot1} is the parallel output resistance seen at the gate of M7. With no cascoding (M3 and M4 removed), R_{tot1} is found as

$$R_{tot1} = r_{o2}||r_{o6} \approx \frac{r_o}{2} \quad , \tag{6.9}$$

where it has been assumed that all device output resistances are approximately equal. With cascode devices M3 and M4 included,

$$R_{tot1} = r_{o2}||gm_1 r_{o6} r_{o4} \approx r_o \quad . \tag{6.10}$$

Specification	Value
DC Gain	$\geq 77dB$
Unity Gain Bandwidth	$40MHz - 70MHz$
Phase Margin	$\geq 78degrees$
Power	$5mW$ (@1.8V)
Output Referred Noise	$1e-17V^2/Hz$ @ 20MHz

Table 6.1 Op-amp Simulation Results

By increasing the effective output resistance of the first stage by a factor of two, the first stage gain is increased by the same amount.

For maximum output swing, the output stage consists of the cascoded amplifier created by M7 and M8, loaded by the wide-swing current source M9 and M11. The second stage gain is calculated as

$$A_{v2} = gm_7 R_{tot2} = gm_7(r_{o8}gm_7 r_{o8}||r_{o9}gm_{11}r_{o9}) \approx \frac{(gmr_o)^2}{2} . \quad (6.11)$$

Overall gain of the op-amp is the product of first and second stage gains:

$$A_v \approx \frac{(gmr_o)^3}{2} . \quad (6.12)$$

The architecture of Figure 6-15 therefore offers high gain.

The op-amp is compensated in the classical way, using a dominant pole contributed by the compensation capacitor and a zero added by the compensation resistor to offset a right-half plane zero that results from introducing the compensation capacitor. A detailed discussion of op-amp compensation is presented in [52]. Table 6.1 presents results of Hspice simulations performed on the op-amp.

6.5 Unity Gain Inverting Buffer

The inverting buffer used after the active loop filter is depicted in Figure 6-16. Note that this buffer is required only for support of the external VCO and could be eliminated with an on-chip VCO implementation. The topology chosen is a source degen-

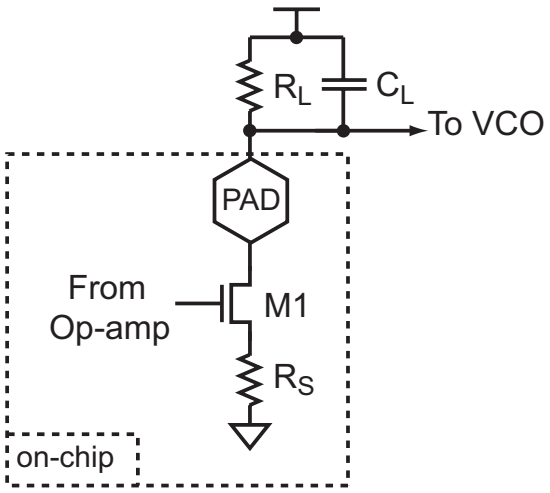


Figure 6-16 Unity Gain Inverting Amplifier

erated common source amplifier. The output gain is described by

$$A_v = -\frac{gmR_L}{1 + gmR_S + \frac{R_S}{r_o}} \approx -\frac{R_L}{R_S} \quad . \tag{6.13}$$

The amplifier bandwidth is determined by the RC time constant at the output node. Both R_L and C_L are external and can therefore be varied to accommodate a variety of gains and bandwidths.

The chosen topology has the benefit of creating a gain that relies on resistor ratioing rather than absolute values. Additionally, any noise contributed by M1 is degenerated by R_S in the same manner as the degeneration circuit employed by the PFD/DAC unit element current sources. A final, practical benefit of this circuit is that it easily accomplishes a DC shift in voltage because R_L and the amplifier supply are external. This is beneficial because the prototype synthesizer utilizes an off-chip VCO. Depending on the vendor and center frequency chosen, discrete VCOs can have nominal input voltages higher than the 1.8V nominal supply voltage allowed by the 0.18um CMOS process used to design the synthesizer. To aid in flexibility in test, M1 was chosen to be a 3.3V, thick oxide device. This way, up to 3.3V can be tolerated at the output node, and a larger selection of discrete VCOs can be used for test.

6.6 Op-amp and Buffer Noise Considerations

As with any active circuitry, noise contributed by the two amplifiers is a concern. To gain an understanding of how small the total noise contributed by the op-amp and buffer amplifier should be, we re-examine the noise model for the op-amp presented in Figure 5-11, where the output VCO noise is described by

$$S_{\Phi o|_{vco}} = \overline{v_{nvco}^2} \cdot \left(\frac{2\pi K_v}{2\pi f}\right)^2 \quad . \tag{6.14}$$

In Chapter 5, we proposed that VCO noise can be input referred and treated as a white noise source by solving for $\overline{v_{nvco}^2}$. From this proposition, we can make two very interesting insights, and gain a better understanding for the analysis of active filter noise.

First, VCO output noise is equivalent to an integrated input referred white noise in the band of interest, and VCO output referred noise is high-pass filtered by the PLL dynamics. Op-amp and buffer amplifier noise will add to VCO input referred noise, and therefore is treated in the same way as VCO input referred noise. Namely, it is integrated by the VCO and high-pass filtered. This is good news, because it means that low frequency (1/f) noise performance of these amplifiers is not critical because the noise is suppressed by the loop.

Second, since the noise performance of the two amplifiers is only critical at high frequencies above the loop bandwidth, we should compare noise performance of the amplifiers to the VCO input referred noise at a high offset frequency, such as 20MHz. For the ZComm VCO used in the prototype system, the input referred VCO noise at 20MHz offset is 3.6e-18V^2/Hz. How good is this noise performance? It is equivalent to the thermal noise PSD of a 225Ω resistor, which is very good indeed!

The unfiltered op-amp noise of $1e{-}17V^2/Hz$ @$20MHz$ is larger than the intrinsic VCO performance and is therefore unacceptable. However, by using the $R_L C_L$ filter at the output of the unity gain inverting amplifier to our advantage, we can reduce this noise. For the prototype system, we choose C_L such that the pole appears at 2.5MHz. The unity gain inverting buffer pole was included in the behavioral model

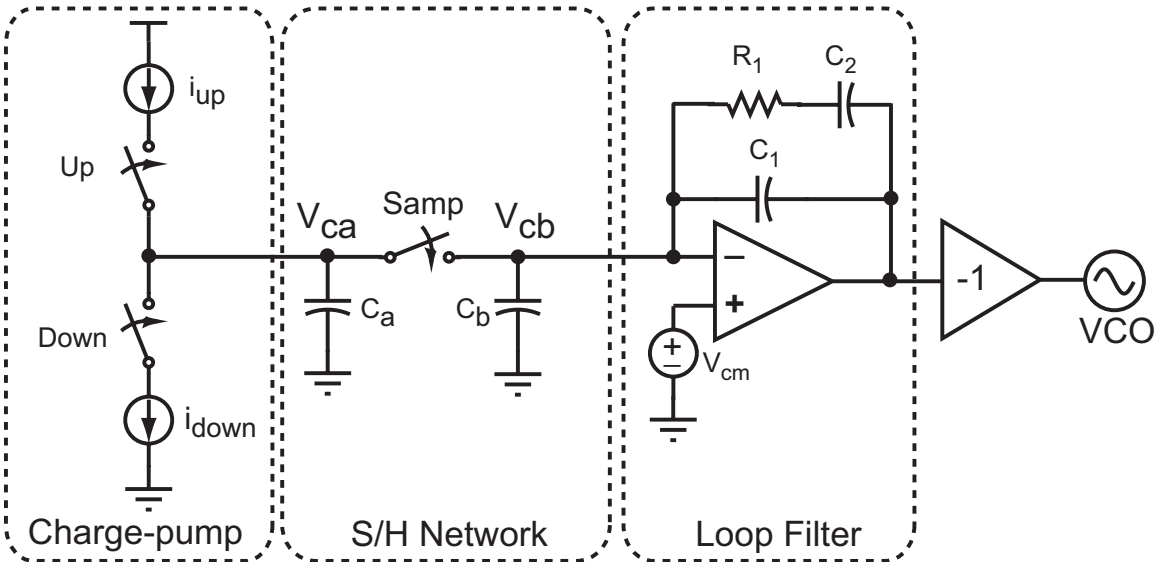


Figure 6-17 Proposed Sample-and-hold Loop Filter

simulations presented in Chapter 5, and reduces the op-amp noise at the output by 18dB at 20MHz to $1.25e-18V^2/Hz$. Total noise is degraded by only 1.3dB from intrinsic VCO performance. If desired, the pole can be lowered, or an additional pole added to increase noise attenuation.

The unity gain inverting amplifier has three primary sources of noise. The two resistors have values $R_L = R_S = 100\Omega$. Noise contributed by these resistors will appear directly at the output node. In noise voltage, the noise power contributed by each 100Ω resistor is $\overline{v_{nR}^2} = 1.6e-18V^2/Hz$. The noise contributed by M1 is degenerated by R_S and is not significant. The raw noise performance of the inverting amplifier is comparable to the intrinsic VCO input referred noise if left unfiltered. However, the filter at the buffer output reduces the buffer output noise by 18dB at 20MHz to $4e-19V^2/Hz$, effectively removing it from consideration.

6.7 Sample and Hold Circuitry

To eliminate the shape mismatch non-ideality that occurs with the PFD/DAC approach, we propose introducing a sample-and-hold (S/H) into the loop filter. The proposed S/H loop filter is presented in Figure 6-17 [54,55]. During the time that the

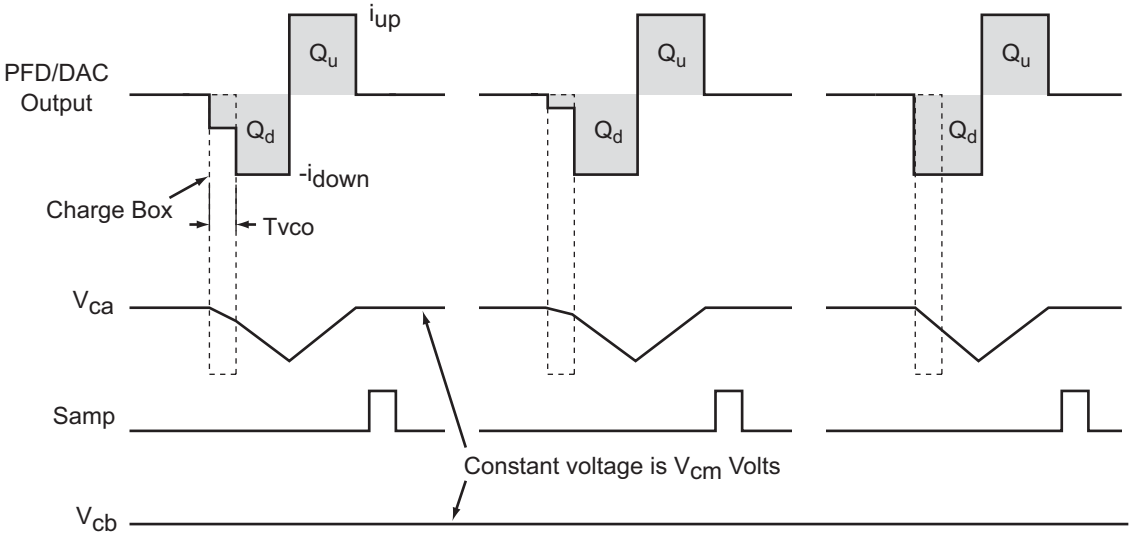


Figure 6-18 Employing a Sample-and-hold to Eliminate Shape Mismatch

PFD/DAC currents are on, the sample switch is open and the current sources charge C_a. When the PFD/DAC completes its operation, $Samp$ goes high, and the op-amp summing junction is connected to C_a.

The S/H shields the VCO from the voltage excursions made by the loop filter due to PFD/DAC operation, as depicted in Figure 6-18. The PFD/DAC ensures that the net charge transferred to capacitor C_a in any given period is zero in steady-state, ignoring noise. By sampling C_a *after* the PFD/DAC completes its operation, no charge is transferred to the loop filter in steady-state. The VCO sees no disturbance on its control voltage, and *all* spurs can theoretically be eliminated! We speculate that switch kT/C noise should be first order shaped due to charge conservation in the S/H architecture and can be ignored in noise calculations.

Since the op-amp positive terminal is set to V_{cm} $Volts$, the minus terminal is also nominally at V_{cm} (plus or minus any input offset in the op-amp), and so the nominal voltage at the charge-pump output is also V_{cm} $Volts$. Note, however, that the voltage at the charge-pump output (node V_{ca}) does make an excursion below V_{cm} during normal operation. The output impedance of the charge-pump can therefore be a possible concern since i_{up} and i_{down} will both vary from their nominal values according to their output impedance. Capacitor C_a is chosen to be large enough to

constrain the voltage swing at the charge-pump output to a reasonable amount of variation.

In the prototype synthesizer, $C_a = 50pF$. For 6.6mA charge-pump current, the peak voltage swing is 374mV below V_{cm}. This effect was behaviorally simulated to verify that synthesizer performance would not be adversely affected. Detailed behavioral simulation results presented in Chapter 5 demonstrated that the PFD/DAC synthesizer is fairly insensitive to charge-pump output impedance or the voltage swing at the charge-pump output.

Capacitor C_b acts as an immediate charge-transfer reservoir during transient events. For example, if the synthesizer output frequency is stepped, there will be a transient period as the loop is settling during which there is a net charge transfer to the loop filter. If the initial phase error due to the frequency step is large, it is possible that the error charge will be too large for the op-amp to supply during the sample window. Capacitor C_b is equal in value to C_a and acts to absorb half of the error charge at the switch instant. If the op-amp cannot completely cancel the *total* error charge on both capacitors when the sample switch is closed, some of the error charge will be left on capacitor C_a when the sample switch opens. By contrast, capacitor C_b has the entire reference period to be discharged by the op-amp.

The aforementioned incomplete charge cancellation scenario will be very rare in a practical implementation of the S/H loop filter, and can be completely avoided by ensuring that the op-amp can supply the worst case expected error charge caused by a frequency step, so it is not considered critical when introducing the S/H function to the loop filter.

6.7.1 Charge Injection and Compensation

Charge injection occurs in any real switched circuit. Unintended error charge due to charge injection mechanisms causes the loop filter output to move, and therefore represents an undesirable noise source different than kT/C noise. The nature of charge injection in the S/H loop filter is to introduce reference spurs, since the sampling event occurs once every reference period. Reference spurs will be attenuated by the low-

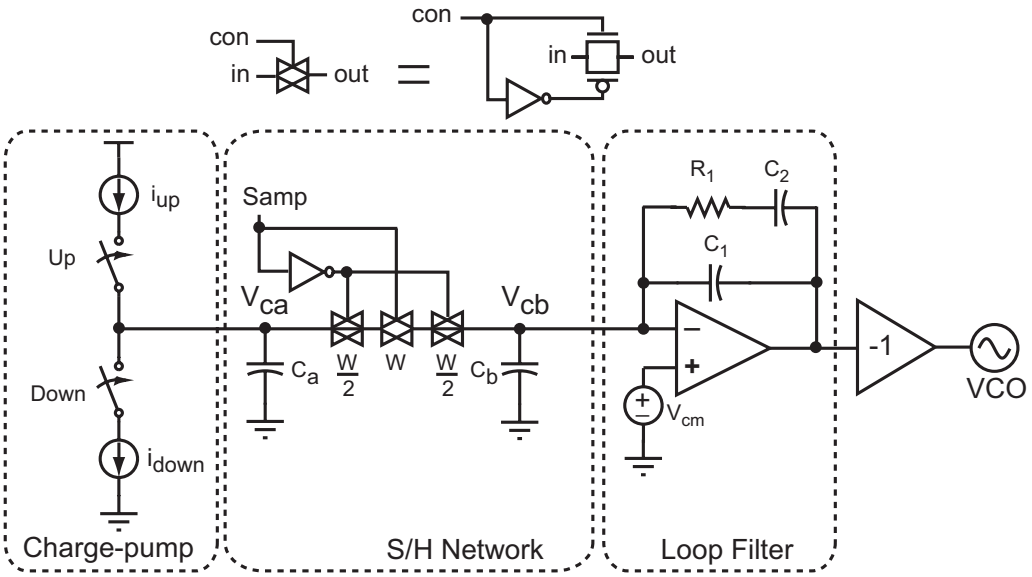


Figure 6-19 Proposed Sample-and-hold Loop Filter

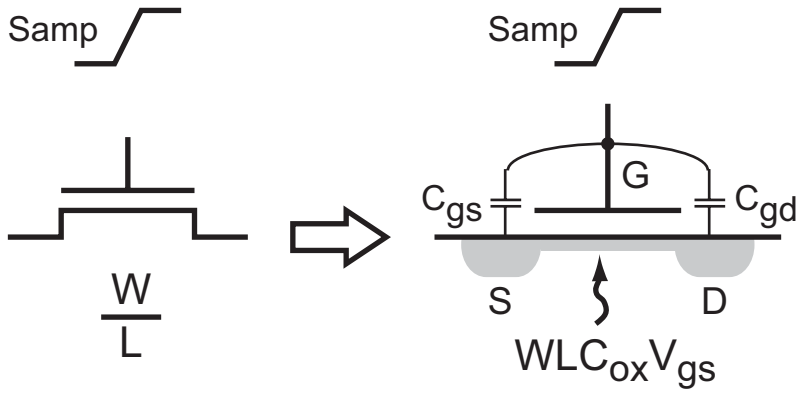


Figure 6-20 Charge Injection Mechanisms in the S/H Switches

pass nature of the PLL dynamics, but any added spurious energy is undesirable, so charge injection should be minimized.

Figure 6-19 presents the implementation of the S/H loop filter used in the prototype synthesizer with the sample switch network included. A transmission gate is used for the sample sample switch. As Figure 6-20 shows, the gate-to-source and gate-to-drain overlap capacitances of the two MOSFETs will couple in error charge when the control signal (*Samp*) switches. Additionally, since the sample switch is a pair of MOSFETs operating in the linear region, the channel charge WLC_{ox}, must be supplied when the switch turns on, and be absorbed when the switch turns off.

The overlap injection is process dependent to a large extent. C_{gs} and C_{gd} are proportional to transistor width, W, times an overlap capacitance C_{ol} that is specified in units of Farads/m and is determined by photolithographic effects. Transistor width is sized so that switch on-resistance is low. Generally speaking, the desire for low on resistance in the switch translates to a wide device, and therefore larger than desired overlap capacitance, and correspondingly high overlap charge injection.

Channel charge is determined by gate area, gate oxide capacitance, and gate-to-source voltage V_{gs}. We see that achievement of low charge injection due to the sample transistors requires small W, coinciding with the desire for small overlap charge injection, but conflicting with the need for low on-resistance in the switch.

Since the magnitude of channel charge depends on V_{gs}, it is highly desirable to keep V_{gs} constant for all switching events so that a signal-dependent charge injection non-linearity is avoided [52]. Constant V_{gs} switching is accomplished by the proposed S/H loop filter because the output node is maintained at V_{cm} by the op-amp, and the input node is also at V_{cm} because the charge-pump output node returns to V_{cm} as depicted in Figure 6-18.

Two techniques are used by the proposed sample switch to offset charge injection. First, complementary devices are used in the sample switch so that the overlap charge will cancel to the degree that C_{ol} is the same for NMOS and PMOS devices and the switch events are coincident for NMOS and PMOS switches. *Samp* occurs at the reference frequency, so incomplete cancellation of the overlap charge will result in a spur at the reference frequency that, fortunately, will be filtered by the PLL dynamics. Second, half-sized, out of phase transmission gates are introduced on both sides of the sample switch to supply and absorb the sample switch channel charge. These transmission gates have their inputs and outputs shorted, as depicted in Figure 6-19. Because the NMOS and PMOS transistors that comprise these transmission gates have a width of W/2 as compared to the sample gate which has width W, the compensation devices will each supply or absorb half of the channel charge of the switched device [52].

Hspice simulations of the proposed sample switch network suggest that spurious

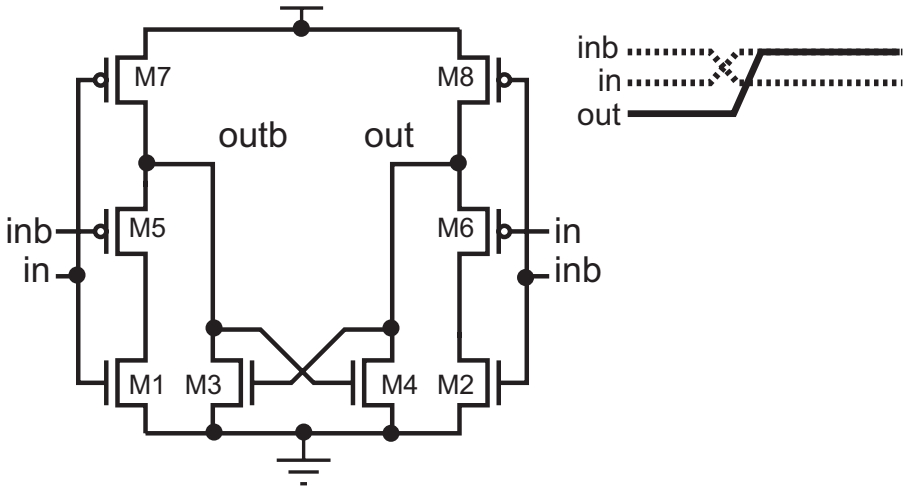


Figure 6-21 Differential to Single-ended Converter

performance due to charge injection is on the order of $-90dBc$ before filtering by the PLL dynamics, a very acceptable level. This spur occurs at 50MHz, and so is attenuated by the closed loop PLL response.

6.7.2 Differential-to-single-ended Converter

As already described, the PFD is built with high-speed, differential, source coupled logic. However, the sample switches must be driven with full-swing logic so that both NMOS and PMOS transistors in the transmission gate are fully turned on and off. Therefore, a differential to single ended converter is required.

Figure 6-21 depicts our proposed differential to single ended converter. It has the advantages over traditional topologies [56] that no static power is dissipated, and coincident full-swing output signals are generated. This is important for the overlap charge cancellation techniques already described, where an extra inverter delay between *out* and *outb* would cause a phase difference between the overlap charge packets delivered through the NMOS and PMOS devices in the transmission gate. Coincident switching means that the overlap charge has a better opportunity to cancel.

Operation of the converter is as follows. Signals *in* and *inb* are differential signals provided from the PFD that swing from 1.8V to 0.8V. If we assume *in* is high (1.8V) and *inb* is low (0.8V), then, in the left side of the circuit, transistor M7 is completely

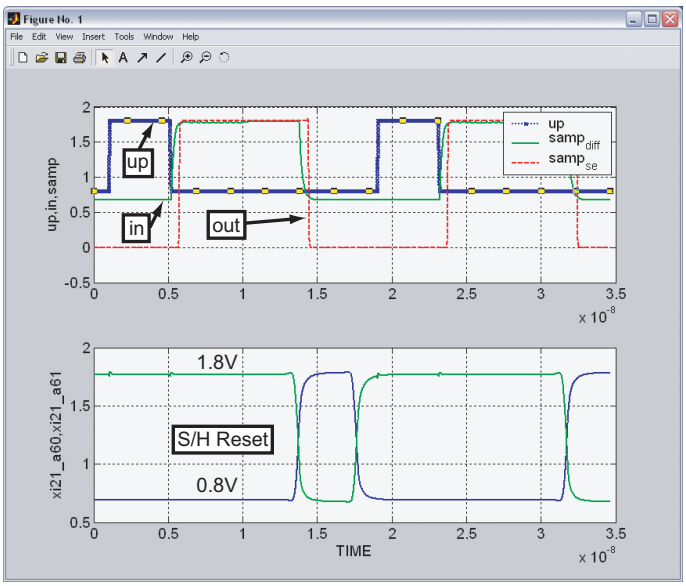


Figure 6-22 Hspice Simulation of Differential to Single Ended Converter

off, M5 is partially on, M1 is completely on, and node *outb* is pulled down. On the right side, M8 is partially on, M6 is fully off, and M2 is partially on. Because of the isolation provided between node *out* and M2's drain, *out* is pulled high. The cross-coupled latch structure M3 and M4 helps the negative going edge to transition all the way to 0V, since the two transistor stack M5 and M1 will not be able to accomplish this on its own. M3 and M4 are therefore made somewhat weak in comparison to the other devices, so that they are easily overcome when the inputs change value. The circuit operates better with larger input voltage swings since the isolation devices M5 and M6 are, therefore, able to turn on more during a pull-down event.

Figure 6-22 shows an Hspice simulation that demonstrates the differential to single ended converter in action. The sample operation initiates when the UP pulse falling edge occurs. A one-shot is then fired for a time that is programmable through a configuration register. The reset signal is shown to indicate when the sample pulse should terminate. The differential input and single ended output signals are shown. Hspice simulations indicate that the proposed differential to single ended converter offers robust operation over process corners and temperature variations.

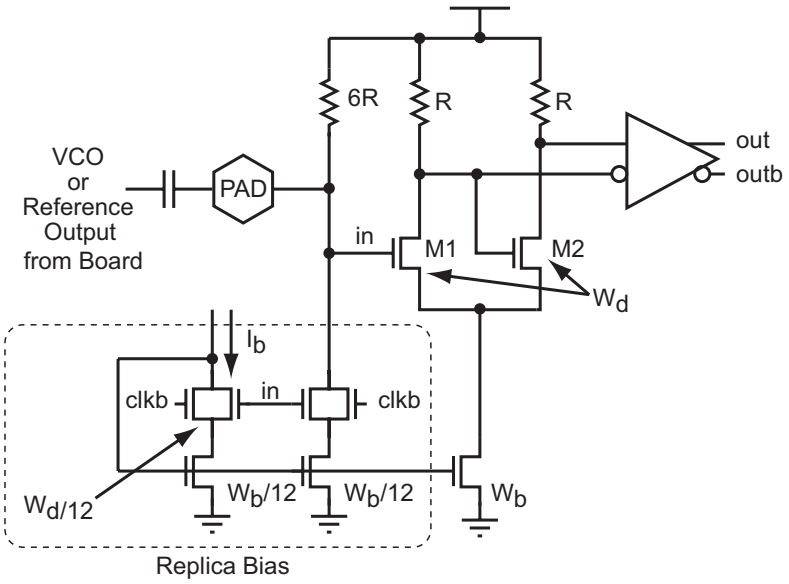


Figure 6-23 Input Buffer and Single-ended to Differential Converter used for VCO and Reference Inputs

6.8 High Speed I/O Design

Thus far, we have proposed circuits for high speed and low noise operation of the PFD/DAC and loop filter. Now we propose circuitry used to get the critical signals on and off the chip.

6.8.1 VCO and Reference Input Buffer

The off-chip VCO and reference frequency source used to implement the prototype synthesizer both have a single-ended output. Because the high-speed PFD logic is differential in nature, a single-ended to differential conversion is required. Figure 6-23 presents the circuit we propose to do the conversion. Based on a simple differential amplifier, feedback is used to perform the conversion. To see what is meant by feedback, we observe that M2's gate is tied to M1's output. This is equivalent to taking the negative output of a differential amplifier and tying it back to its inverting input. Since this leads to a delay between the differential amplifier's two inputs, there will be some duty cycle distortion introduced by this configuration. To restore uniformity to the on-chip duty output cycle, an additional differential amplifier is

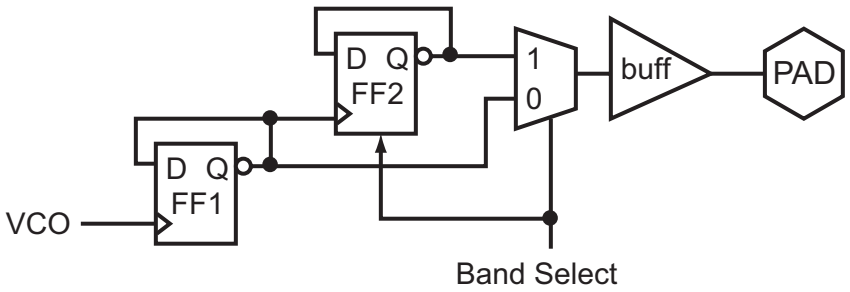


Figure 6-24 Band Select Divider

used to buffer the first stage outputs.

The off-chip signal is capacitively coupled on chip, and a replica bias circuit is used to set the DC bias point of the input node *in* to be halfway between the output high level and output low level. This is an optimal bias point for operation, since it is equivalent to the input stage being self-biased. To save power, the replica bias stage uses devices one-twelfth the width of the input devices M1 and M2.

As we will see in chapter 7, this circuit plays an important role in defining overall synthesizer noise performance. Namely, when the proposed circuit is stimulated by the 3.6GHz discrete VCO it performs very well. However, when the 100MHz reference is input to a scaled down version of the proposed circuit, low frequency noise suffers. In Chapter 7 we will show that this is the result of slow input edge transitions. Both the VCO input and reference input are sinewaves, whose risetimes are proportional to frequency. Because the reference buffer version of Figure 6-23 is scaled down to save power it is more susceptible to jitter sources. The combination of higher noise sensitivity and slower reference signal edge rates limit the overall low frequency performance of the synthesizer. In Chapter 7 we will also demonstrate techniques to improve the reference buffer jitter problem.

6.8.2 Output Band Select Divider

Once the off-chip VCO output is brought on chip, it is sent several places in parallel. First, it goes to the divider to be used by the synthesizer. Second, it is used by the PFD/DAC to re-time the divider and generate the charge-box. Finally, it is sent

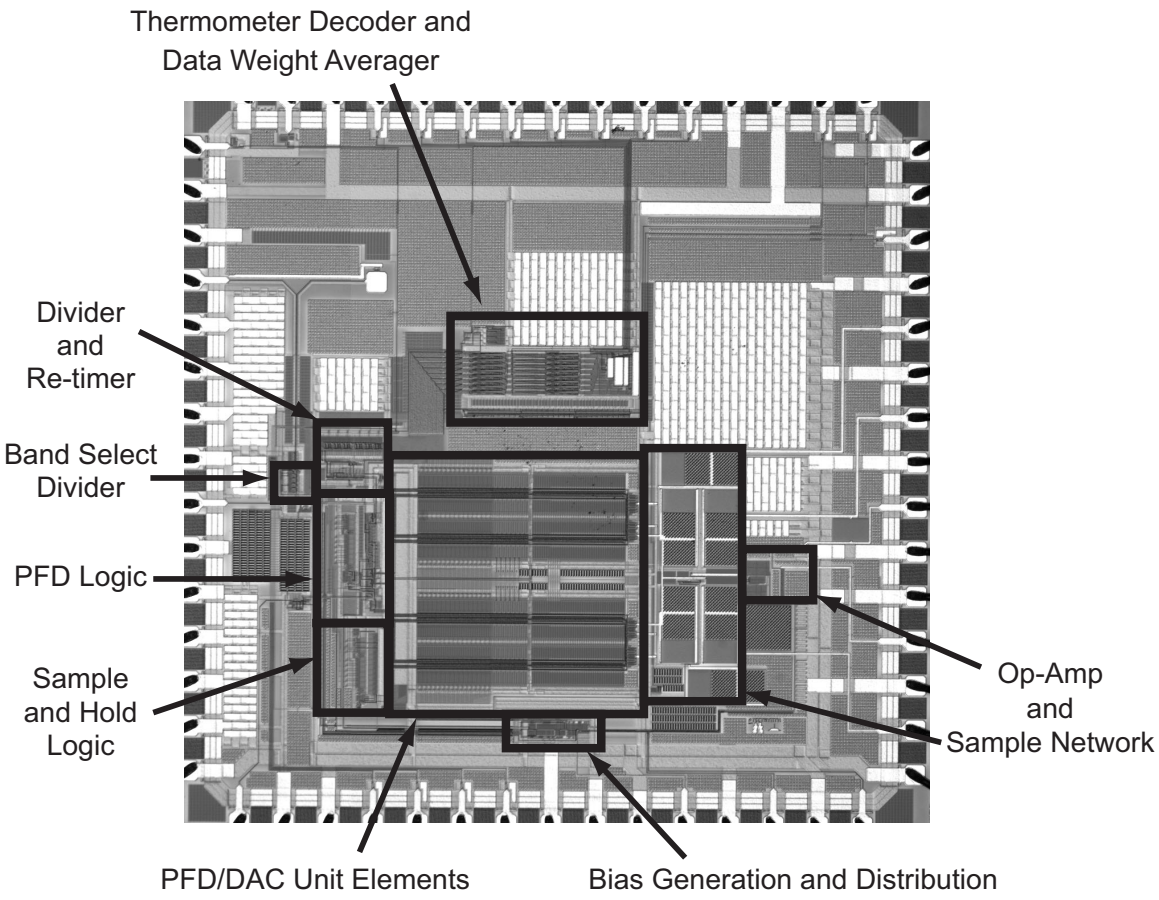


Figure 6-25 Mismatch Compensated PFD/DAC Synthesizer Chip Microphotograph

to the band select divider and then its divided down version is sent off chip. The band select divider is used to divide the 3.6GHz VCO down to either 1.8GHz or 900MHz, depending on the desired output band. All of the logic used by the band select divider is high-speed differential logic, but single-ended logic is shown in Figure 6-24 for clarity.

To save power, FF2 is turned off by the band select signal when a 1.8GHz output is chosen. The output buffer is a differential amplifier with 50Ω on-chip termination resistors.

6.9 Prototype PFD/DAC Synthesizer IC

Figure 6-25 is a chip microphotograph of the fabricated mismatch compensated PFD/DAC synthesizer IC. Implemented in National Semiconductor's $0.18\mu m$ CMOS process, the chip measured 2.7mmX2.7mm with a 1.8mmX1.5mm active circuit area. The key circuits described in this chapter are highlighted in the figure.

6.10 Summary

In this chapter we have proposed several new circuit techniques for high speed, low noise operation. Circuits key to PFD/DAC performance have been proposed. We have shown that introduction of active circuitry into the loop filter has a performance penalty in terms of added noise. Op-amp and buffer noise considerations center around classical noise concerns, namely thermal noise impacting synthesizer noise performance at high offset frequencies. S/H charge injection noise, on the other hand, impacts noise performance by introducing spurious energy at the reference frequency. In the first case, filtering is used to gain additional suppression and achieve the desired performance levels. In the second case, an active cancellation scheme is used to reduce the magnitude of the charge injection induced spurs in addition to filtering considerations. In the next chapter we will present measured results of the prototype mismatch compensated PFD/DAC synthesizer IC shown in Figure 6-25, which is built using the circuits described in this chapter.

Chapter 7

Measured Results

In this chapter we present measurements taken using a prototype PFD/DAC synthesizer fabricated in a 0.18um CMOS process. In addition to presenting the raw performance of the synthesizer, comparisons will be made between measured performance and predicted performance based on the behavioral simulation techniques proposed in Chapter 5.

Measurement data will lead us to examine two sources of phase noise that were not included in initial calculations. Using the programmability incorporated into the synthesizer IC, we de-couple these additional noise sources and propose board level circuits to reduce their impact on overall performance.

We will also demonstrate the power of the analytical model presented in Chapter 3 by extracting values for parameters such as reference jitter and PFD/DAC internal timing mismatch based on correspondence between the analytical model and measured results. This analytical back extraction is a technique that has not yet appeared in the literature.

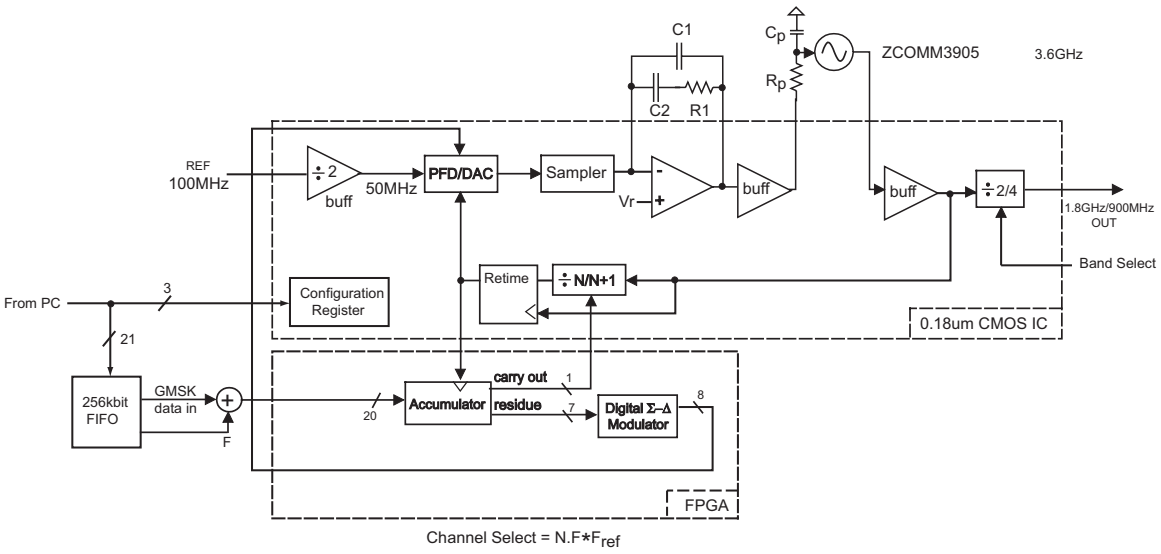


Figure 7-1 Prototype PFD/DAC Synthesizer

7.1 Prototype Mismatch Compensated PFD/DAC Synthesizer System

Figure 7-1 depicts the prototype synthesizer. For flexibility in testing, the loop filter, VCO, and digital $\Sigma\Delta$ modulators are kept off chip. The $\Sigma\Delta$ modulators are implemented in an FPGA, which can be configured to make the overall system act as an integer-N synthesizer, a classical $\Sigma\Delta$ synthesizer, or a PFD/DAC synthesizer. A 256kbit FIFO is employed to buffer the GMSK data generated on a PC before it is received by the FPGA when the system is configured to act as a dual band GMSK transmitter.

7.1.1 System Programmability

A configuration register on the synthesizer IC is used to select operational parameters. The PFD reset delay that determines the on-time of the current pulses produced in the PFD/DAC is programmable, as is the S/H signal pulse-width. The S/H function can be enabled or disabled, as can phase swapping in the PFD/DAC. The PFD/DAC function can be enabled or disabled. The PFD logic can be configured in either an offset or overlapping tri-state topology (as discussed in Chapter 5). Bias currents

for the PFD logic and multi-modulus divider are programmable through a bias DAC that is controlled by the configuration register. Other programmable functions include enabling the band select divider to output 900MHz, 1.8GHz, or disabling it, selecting whether to use the divider output or a buffered version of the reference signal as a clock source for the internal circuitry and external FPGA, and configuration of the divider retimer circuit to work in continuous mode, sampled mode, or to be overridden and have a retiming edge forced.

As will be shown, the considerable amount of flexibility added to the prototype synthesizer has considerable benefits when examining measured performance. By selectively turning circuits on or off, and re-configuring the PFD structure, we are able to understand all of the noise sources affecting overall system performance. Table 7.1 lists the relevant design parameters used for the prototype synthesizer based on Hspice simulations and calculations. The charge-pump current was modified from the simulated value of 5mA to 6.6mA for improved performance. As will be discussed in section 7.3.5, on-chip coupling mechanisms lead to larger than desired fractional spurs. Having a larger value of charge-pump current reduces the impact of spurious charge injected into the control loop.

All phase noise and step response plots are measured using an Agilent Technologies E5052A Signal Source Analyzer. Modulated spectra and spurious plots are measured with an Agilent Technologies 8595E Spectrum Analyzer. Demodulated GMSK eye diagrams are measured using a Hewlett Packard 89440A Vector Signal Analyzer.

The un-modulated synthesizer is the best measurement starting point, because constant valued inputs are a worst case scenario for low ordered $\Sigma\Delta$ modulators such as those controlling the PFD/DAC. In order to see the impact of intrinsic noise sources on overall performance, we initially configure the synthesizer as an integer-N type. Integer-N synthesizers exhibit no fractional-N quantization noise, so we are able to evaluate synthesizer performance in the absence of quantization noise. We then add quantization noise by configuring the test system as a PFD/DAC synthesizer and explore the various techniques utilized by the PFD/DAC synthesizer to improve quantization noise performance.

Design Parameter	Value
Nominal Output Frequency	$3.6GHz$ with $1.8GHz$ and $900MHz$ available
Reference Frequency	$50MHz$
Closed Loop Bandwidth	$1MHz$
Loop Filter Pole	$2.81MHz$
Loop Filter Zero	$111kHz$
Added Filter Pole	$2.5MHz$
$R1$	222Ω
$C1$	$265pF$
$C2$	$6.42nF$
R_p	180Ω
C_p	$350pF$
PFD/DAC Full-scale Current	$6.6mA$

Table 7.1 Prototype Synthesizer Design Parameters

7.2 Baseline Measured Performance: The Integer-N Synthesizer

Figure 7-2 presents measured results for the the prototype system configured as an integer-N synthesizer with the phase detector configured first as an offset tri-state PFD, and then as an overlapping tri-state PFD. We see that the two spectra are different, with the overlapping tri-state PFD based synthesizer exhibiting *less* noise than the offset PFD synthesizer. In Chapter 5 it was demonstrated that the offset PFD exhibits more linear behavior than the overlapping PFD, so it is desirable to use the offset structure when operating as a fractional-N synthesizer. We therefore must determine what is different between the implemented system and our analytical and behavioral models.

In order to compare the measured results to the analytical model proposed in Chapter 3, we utilize the PLL Design Assistant (PDA) tool and Matlab. Figure 7-3 presents calculated performance using the analytical model along with the measured results depicted in Figure 7-2. Measured low frequency noise performance is much worse than expected from calculation using simulation parameters and the analytical model.

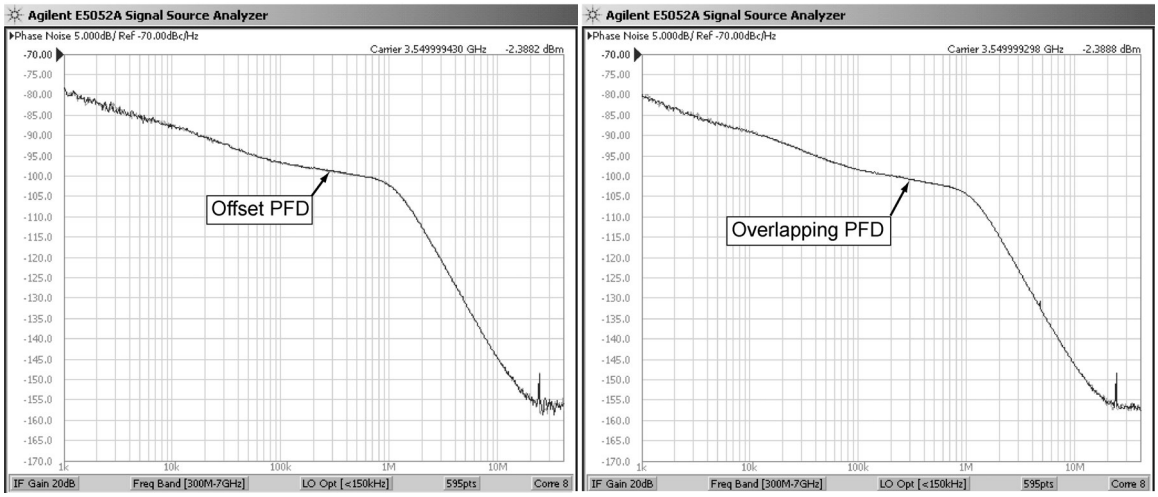


Figure 7-2 Measured Phase Noise Plot for Integer-N Synthesizer

The fact that the overlapping and offset PFD topologies exhibit different noise performance for the integer-N synthesizer gives a clue as to possible reasons for the poor low frequency noise exhibited by the synthesizer in our initial measurement. Calculated performance using the PDA and analytical model was initially based on the assumption that charge-pump thermal noise dominates the phase noise performance at low frequencies. In reality, the system is suffering from two additional noise sources that were not included in the initial calculations.

Figure 7-4 provides an explanation for the increased low frequency noise. In the figure, charge-pump thermal noise has been ignored, and two sources of jitter are examined. Div0c and Div1c are the two divider signals output from the timing mismatch compensation and resynchronization block inside the PFD/DAC, which was described in Chapter 6.

The first source of noise shown in Figure 7-4 is due to reference jitter. Jitter on the reference input will cause significant noise for both topologies. For the offset PFD, the reference jitter causes a charge error $Q_{\text{n\_off1}}$, while for the overlapping PFD it generates $Q_{\text{n\_ov1}}$. This jitter induced error charge disturbs charge balance in the system.

The second noise source is particular to the offset PFD. If the delay cell used to generate the reset pulse produces jitter, the reset edge will jitter and error charges

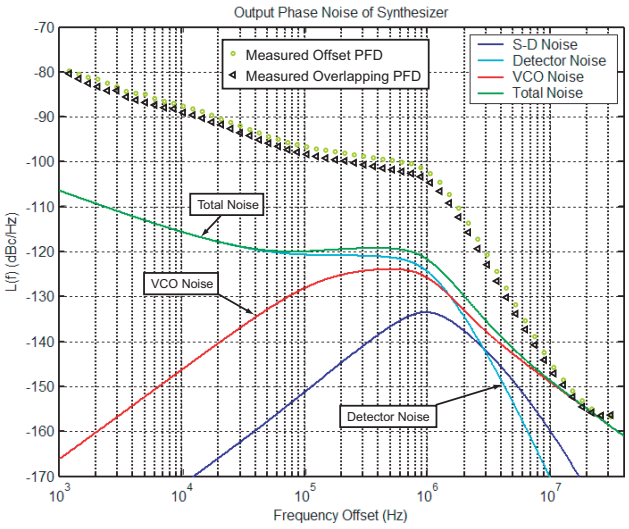


Figure 7-3 Measured Vs. Calculated Performance

$Q_{\text{n\_off2}}$ and $Q_{\text{n\_ov2}}$ are created. For the offset PFD, reset jitter error changes the value of the positive charge, Q_u, upsetting steady-state charge balance. For the overlapping PFD, the positive error charge cancels the negative error charge produced by reset jitter if the positive and negative current magnitudes are equal. The overlapping PFD based synthesizer suffers from only one of the two jitter induced noise sources, and therefore exhibits better low frequency noise performance, as observed in the measured results.

From the previous arguments and measurements we will soon show, we attribute the unexpected low frequency noise seen in Figure 7-3 to reference jitter and reset jitter in the PFD. The overlapping PFD performance is slightly better than the offset performance because PFD reset jitter induced noise minimally impacts it. The overlapping PFD based synthesizer is therefore limited by reference jitter. Therefore, future designs should focus on reducing reference buffer jitter to obtain better low frequency noise performance.

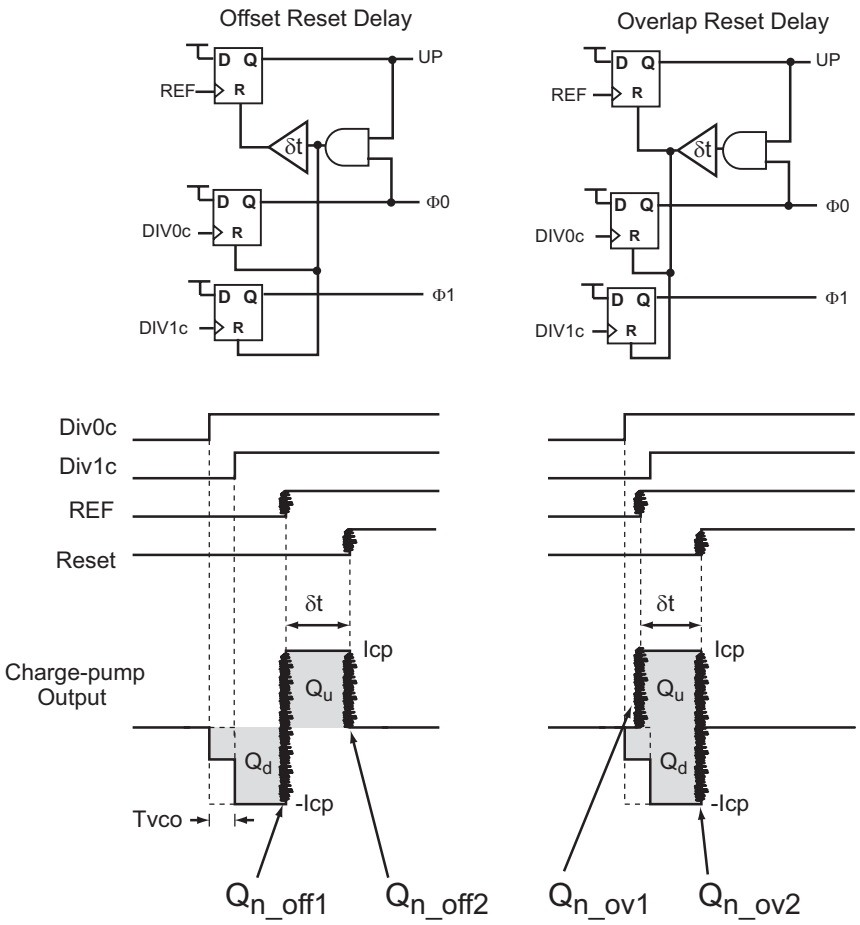


Figure 7-4 PFD Jitter Induced Noise Sources

7.2.1 Reference Buffer Jitter Induced Noise

To verify the proposed explanation and improve performance, we start by reducing reference jitter in the prototype system. Figure 7-5 presents both the problem and a method of improving the reference jitter problem in the prototype. The low noise crystal reference used to act as the reference input to the synthesizer produces a 100MHz sinewave output, which is then processed by the reference input circuitry and divided down to 50MHz for use by the synthesizer.

As depicted in the figure, a sinewave input to the chip will be very susceptible to noise. Any voltage noise, ΔV, injected to the buffer, will cause a timing error, $\Delta t1$. The most sensitive point for noise injection is at the sinewave zero crossings [57–60]. By contrast, the squarewave signal has a much higher zero crossing slope, and

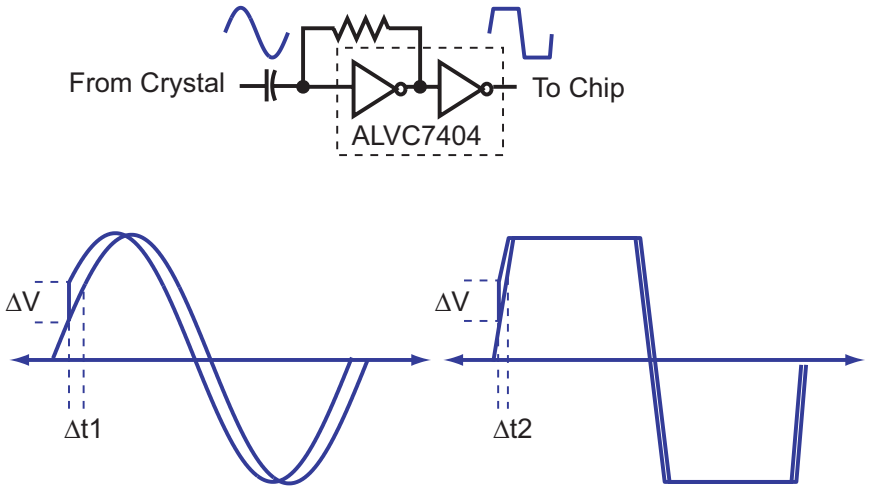


Figure 7-5 Reference Buffer Jitter and Proposed Solution

therefore produces a much smaller time-step, $\Delta t2$, in response to the same voltage disturbance, ΔV.

It is the function of the reference input buffer circuitry to square up the crystal sinewave for use by the rest of the chip. However, because a 100MHz crystal has a very slow slope relative to the rest of the signals processed by the system, it is possible that the reference input buffer does not achieve a sufficient degree of "squareness" in response to an input sinewave. Noise may also be added by the first stage of the input buffer, which processes the crystal output.

To improve reference jitter performance, the circuit in Figure 7-5 is added to the test board. Two high-speed logic inverters are used to improve the edge rates seen by the synthesizer IC. The first inverter is self-biased with a resistor to configure it as a high-speed amplifier. The crystal output is AC coupled into the buffer, and a second inverter is used to square up the output of the first inverter. A $10\mu F$ decoupling capacitor is used across the inverter chip on the board to ensure that it has a clean supply. The inverters add two stages of gain between the crystal oscillator output and on-chip reference buffer input, thereby increasing the edge rates of the waveform and lowering susceptibility to on-chip noise.

Figure 7-6 compares the offset and overlapping PFD based integer-N synthesizers using the proposed buffer circuit. The offset tri-state PFD based synthesizer phase

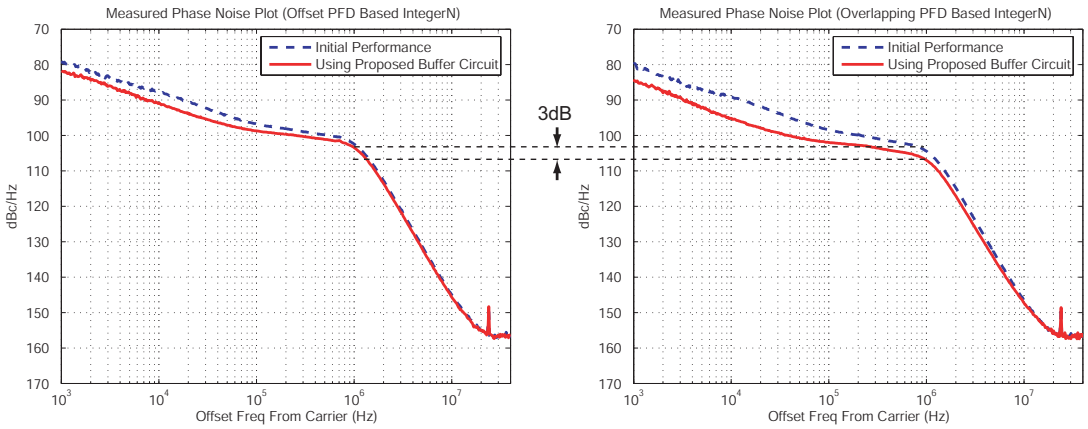


Figure 7-6 Measured Phase Noise Plot for Integer-N Synthesizer

noise improves by approximately 3 dB at low frequencies, while the overlapping tri-state PFD based synthesizer improves by 5dB. The difference in improvement is attributed to the fact that the offset PFD based synthesizer is also influenced by PFD reset jitter induced noise, while the overlapping PFD based synthesizer is not. The absolute level of noise in the overlapping PFD based synthesizer is 3dB lower than in the offset PFD based synthesizer, indicating that the performance of the offset PFD based synthesizer can be further improved.

7.2.2 PFD Reset Jitter Induced Noise

Figure 7-7 presents a technique to reduce the reset jitter induced phase noise in the prototype offset PFD synthesizer. A better way to lower reference jitter in future designs is through improved circuit design of the input buffer. For the prototype system, we see that by lowering the magnitude of i_{up}, the jitter induced error charge is reduced. An additional benefit is that reference jitter induced error charge is also reduced because it is created by charge contributions of both i_{up} and i_{down}.

We are able to decrease the magnitude of i_{up} in the prototype synthesizer through the use of a board level trim current. Figure 7-8 presents a simplified schematic of the bias circuit used to generate i_{up} and i_{down} in the PFD/DAC. Bias current is delivered to M1 from a current source on the test board. M1, M2, and M3 are all sized such that $i_{down} = i_{biasn} = i_{biasp}$. M2 and M4 establish a bias voltage at M5's gate such

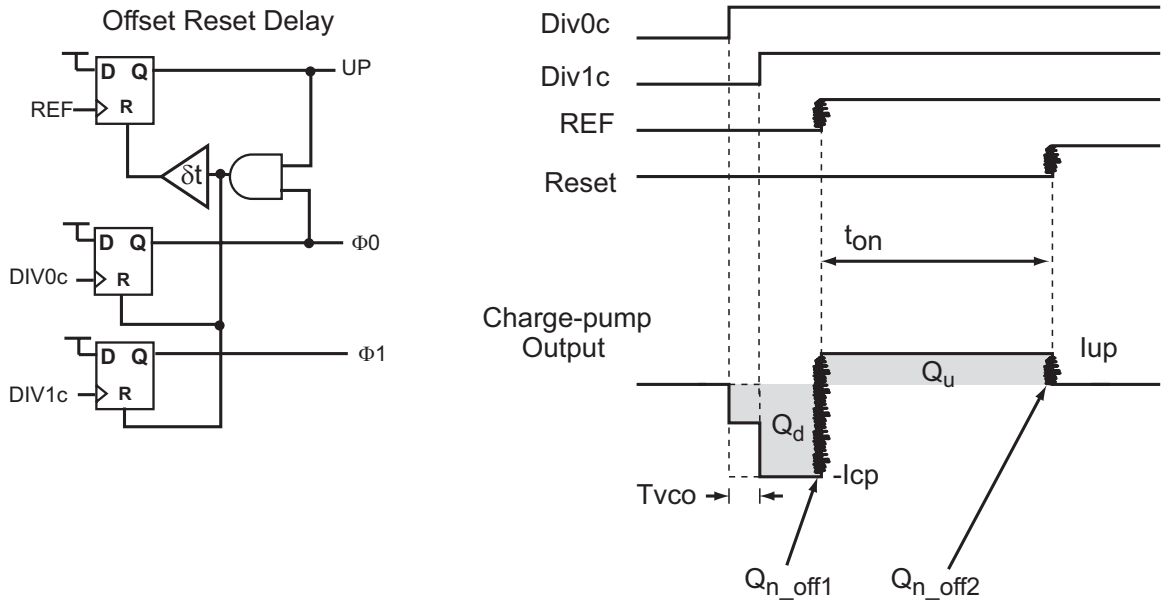


Figure 7-7 Reducing Reset Jitter Induced Phase Noise in the Offset Tri-state PFD

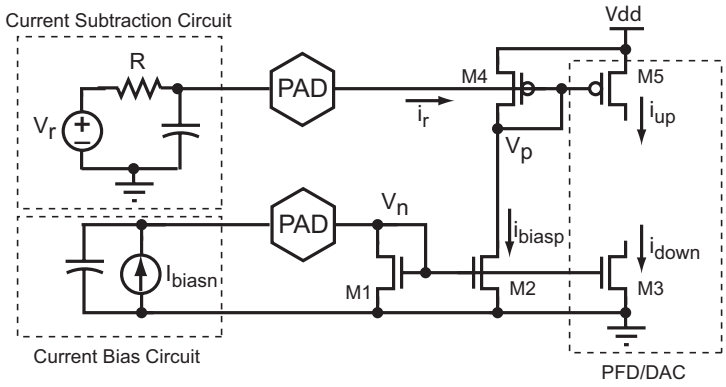


Figure 7-8 Lowering i_{up} to Reduce Reset Jitter Induced Phase Noise

that, nominally, $i_{up} = i_{down}$. The bias nodes (V_p and V_n) are bypassed with off-chip capacitors acting as noise filters.

Because the bias node for the PMOS devices is brought out to a pad for bypass, it can also be used to add or subtract current from i_{biasp} and therefore change i_{up}. The voltage source and a resistor are used to generate i_r, which either adds to or subtracts from i_{biasp}, depending on the voltage dropped across the resistor. By varying V_r, i_{up} can be made different from i_{down}.

Figure 7-9 shows the measured results achieved by reducing i_{up} from its nominal value of 6.6mA by increasing i_r. As expected, as the magnitude of i_{up} is lowered, the

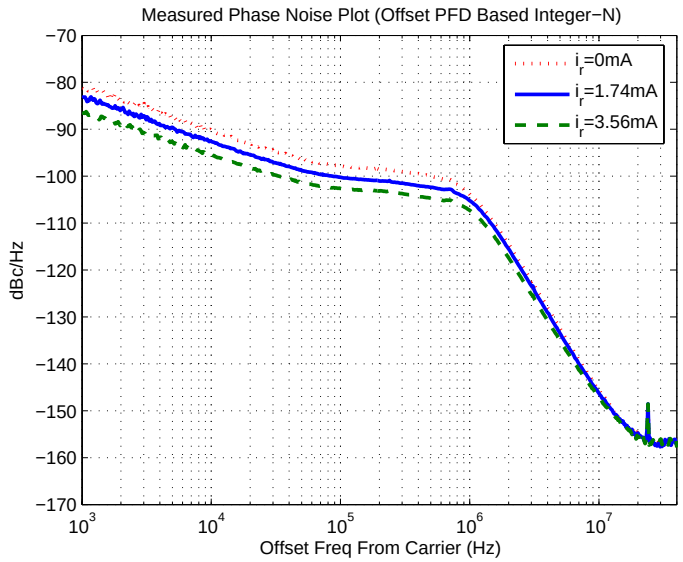


Figure 7-9 Offset PFD Synthesizer Measured Phase Noise With Current Subtraction

low frequency noise improves because both reference jitter induced phase noise and reset jitter induced phase noise decreases.

The question as to which of the two noise sources ultimately limits performance of the offset PFD based integer-N synthesizer at low frequency is answered in Figure 7-10. The value of trim current used by the synthesizer, i_r in the figure, is $3.56mA$, corresponding to $i_{up} = 3.04mA$. The reset pulse-width is varied by changing δt in Figure 7-7. On-chip, this corresponds to re-configuring a programmable delay cell that is comprised of a series of delays that can be selected between a fast path and a slow path. Slow path cells will have more jitter than fast path cells since their rising and falling slopes are slower.

Figure 7-10 clearly shows that, as the reset delay is made shorter, and therefore has a lower value of rms jitter, phase noise between 100kHz and 1MHz decreases, but phase noise below 100kHz does not appreciably change. This suggests that the reference jitter is dominant for both curves below 100kHz, while the reset jitter noise is dominant for the dashed curve above 100kHz. Performance for the integer-N synthesizer is therefore ultimately limited by reference jitter in the prototype system. Therefore, future implementations should be designed with particular attention paid to this issue.

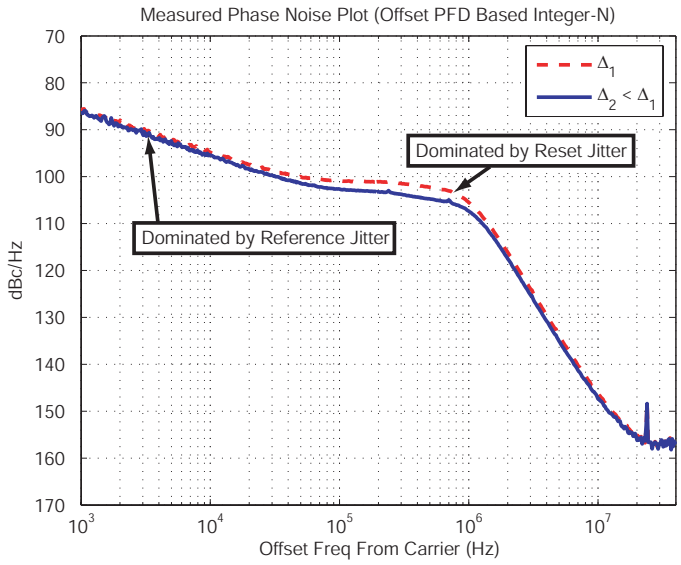


Figure 7-10 Measured Phase Noise While Varying the Reset Pulse-width

7.2.3 Reference Jitter Extraction Using the Analytical Model

Using the analytical model of Chapter 3, it is possible to extract the value of reference jitter that would result in the measured performance. Reference jitter induced phase noise at the synthesizer output is derived from the analytical model as

$$S_{\Phi_{o|REF}}(f) = \overline{\Delta t_{jitt}^2} \cdot \frac{1}{T} \cdot \left(\frac{2\pi}{T}\right)^2 \cdot (NT)^2 \cdot |G(f)|^2 \cdot \frac{i_{up}}{i_{down}} \quad , \tag{7.1}$$

where $\overline{\Delta t_{jitt}^2}$ is the variance of the jitter distribution, and we have assumed a white spectral density profile for the jitter. The factor $\frac{i_{up}}{i_{down}}$ is included to reflect the difference between the currents in the PFD/DAC. Because i_{up} is lowered, so is the noise.

The PLL Design Assistant allows us to input reference noise (referred to the output) and compare it to measured results. Initially, we assumed that charge-pump noise was dominating detector noise at low frequencies. We have since determined that reference jitter is dominating low frequency noise, so we change the PDA parameters to reflect this new information. Using the PDA, Table 7.1, and equation 7.1, we can back extract the rms value of reference jitter seen by the synthesizer! This is a very valuable and powerful analysis technique, since it allows us to determine the value of an important system parameter that is not directly measurable.

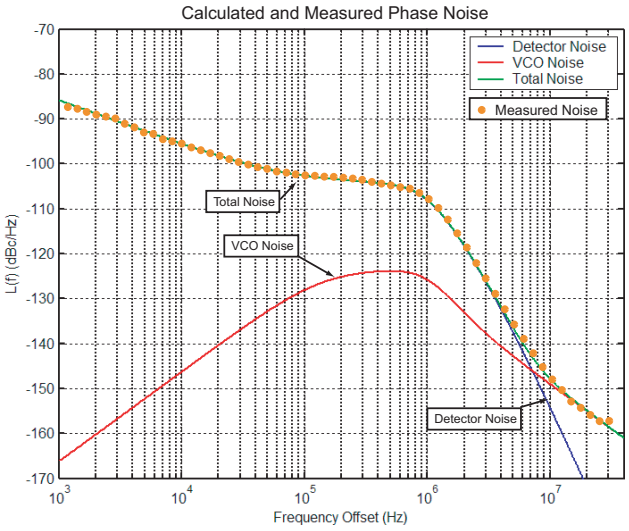
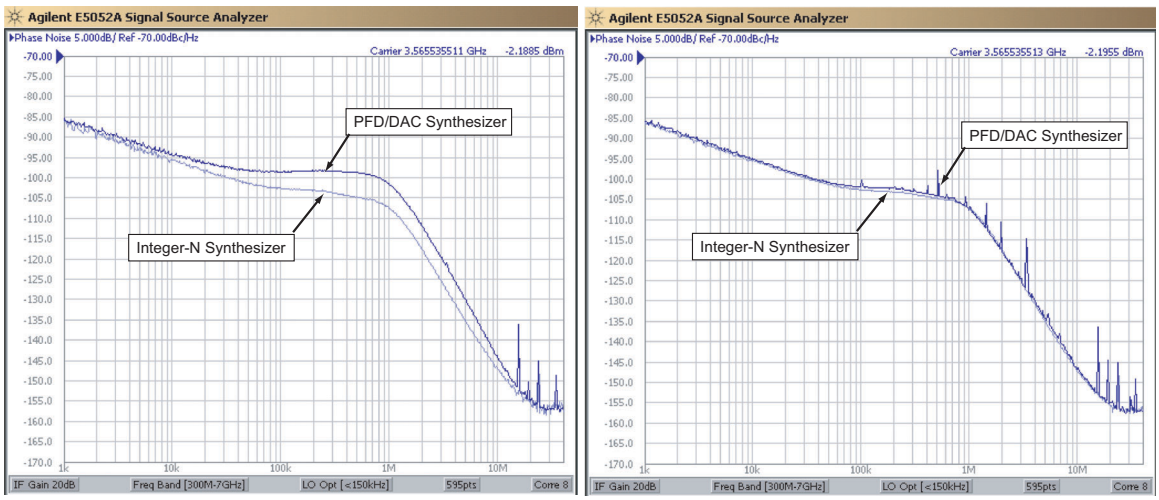


Figure 7-11 Analytical Model and Measured Performance for Extraction of Reference Jitter

Figure 7-11 presents the result of analytical calculations using the PDA and measured results for the integer-N synthesizer. In order to obtain the plot in Figure 7-11, the detector noise parameter was set to $-107dBc/Hz$, with a 1/f corner frequency of 130kHz and a -10dB/decade 1/f noise slope. There is excellent agreement, as the measured data is within a dB or two over the entire frequency span.

As an approximation to calculate reference jitter, we ignore the 1/f portion of the curve and use equation 7.1. This approximation is reasonable, since the 1/f portion of the curve is only dominant up to 100kHz, and so will contain a relatively small portion of the total noise contributing to reference jitter.

The result is a calculated reference jitter $\Delta t_{jitt} = 3.04ps$, indicating very good, but not excellent, rms jitter performance is obtained by the reference in the prototype system. In order to achieve a noise level below that of the calculated charge-pump noise level of $-123dBc/Hz$, the reference buffer jitter would have to be below $1ps$ rms for the same configuration. It is important to note that the value of reference jitter being extracted is referred to the *input* of the PFD circuitry. This includes all jitter added by any circuitry between the reference crystal output and the PFD input, which, in the prototype system includes the the board level reference buffer circuit



Phase Swapping Enabled

Phase Swapping Disabled

Figure 7-12 Measured Phase Noise: PFD/DAC Synthesizer Vs. Integer-N Synthesizer

proposed in Figure 7-5, the reference input buffer, and the PFD input stage.

7.3 Un-Modulated PFD/DAC Synthesizer Measured Performance

Now that the intrinsic noise performance of the synthesizer has been established, we configure the system to be a PFD/DAC fractional-N synthesizer and compare it with the integer-N synthesizer results to see how far from intrinsic noise performance the PFD/DAC synthesizer operates. We then compare the PFD/DAC synthesizer with a state-of-the-art $\Sigma\Delta$ synthesizer, with an emphasis on the impact of quantization noise on overall phase noise performance. By programming the synthesizer IC configuration register, we examine the impact of the various noise reduction and management techniques already proposed.

7.3.1 PFD/DAC Synthesizer Vs. Integer-N Synthesizer

Figure 7-12 presents measured results comparing the 7-bit PFD/DAC synthesizer with a divider value N=71.3107 with the integer-N synthesizer configured with a divide value N=71.

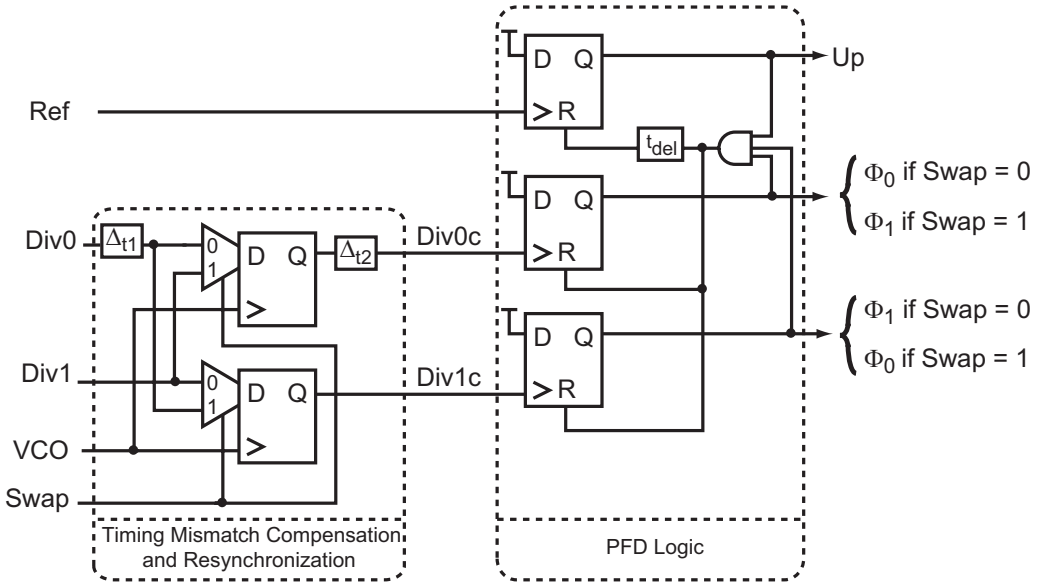


Figure 7-13 PFD/DAC Logic and Timing Block

The left plot indicates that the PFD/DAC synthesizer has worse low frequency noise performance than the integer-N synthesizer. In order to demonstrate the source of this additional noise, the phase swapping technique used to eliminate a timing mismatch in the PFD/DAC structure (described in Chapter 3) is disabled.

As the right plot demonstrates, when phase swapping is disabled, we see that the two synthesizers exhibit the same noise performance, but the gain mismatch in the PFD/DAC leads to incomplete cancellation of fractional spurs, which appear in the spectrum. Therefore, we see that timing mismatch limits noise performance of the prototype mismatch compensated PFD/DAC synthesizer.

7.3.2 PFD/DAC Timing Mismatch Extraction Using the Analytical Model

As with reference jitter induced phase noise, we can use the analytical model to determine the magnitude of the timing mismatch in the PFD/DAC. For convenience, we repeat the phase swapping circuit in Figure 7-13 to demonstrate the source of the low frequency noise. The two divider phases used by the PFD/DAC to create the charge-box are re-timed by the timing mismatch compensation block and dynamically

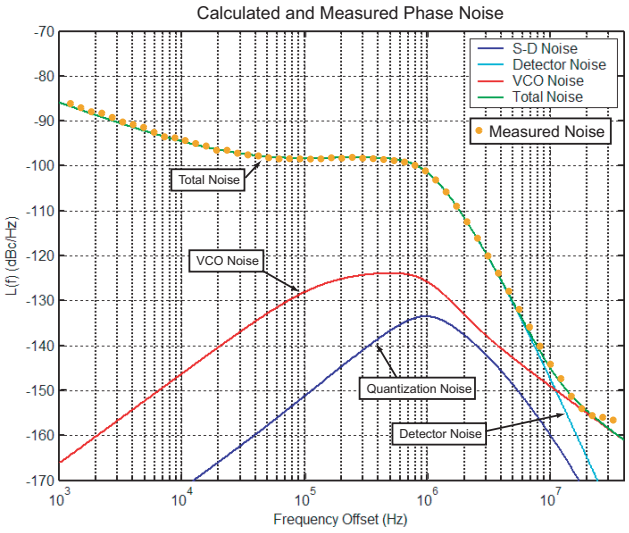


Figure 7-14 Extraction of Timing Mismatch in the PFD/DAC Using the Analytical Model

swapped so that, on average, both Div0 and Div1 signals will see the same average delay, Δ_{t2}, which is referred to the flip-flop outputs. The swapping process is controlled by a 23-bit LFSR, and the resulting phase noise has a white profile described by equation 4.10, which is repeated here for convenience:

$$S_{\Phi_{out}|\Delta_t} = \frac{1}{T}\frac{\overline{\Delta_{t2}^2}}{12}(2\pi N_{nom})^2 \cdot |G(f)|^2 \quad . \tag{7.2}$$

Figure 7-14 compares the PFD/DAC measurement with phase swapping enabled to the analytical model with a -100dBc/Hz detector noise floor. Because the phase swapping noise raises the flat portion of the low frequency noise and does not affect 1/f noise, the 1/f corner frequency input to the PLL Design Assistant used to generate Figure 7-14 was changed to 20kHz to keep the 1/f portion of the curve the same as in Figure 7-11.

Using the parameter values from Table 7.1 and equation 7.2, Δ_{t2} is determined to be 10.9ps. Once again we are able to back extract an internal parameter by employing the analytical model. This information provides useful feedback for layout methodology. Hspice simulations on extracted layout predicted that Δ_{t2} would be approximately 6ps, so it is probable that the parasitic extraction control file and

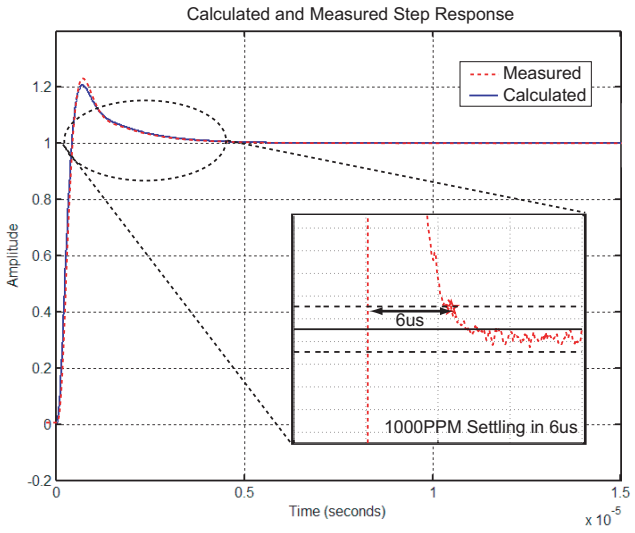


Figure 7-15 PFD/DAC Synthesize Measured Vs. Calculated Step Response and Settling Time

settings are not capturing all of the parasitic capacitances that contribute to Δ_{t2}. Device mismatch is also a concern, as well as statistical variation in the mismatch profile.

It may be possible to change the simple LFSR swap signal control to a scheme that would result in shaped timing mismatch noise. Such a technique would remove Δ_{t2} from consideration as the key parameter limiting the ultimate performance of the PFD/DAC synthesizer. This topic is left as the subject of future work.

7.3.3 Dynamic Response

Before moving on to comparisons between the PFD/DAC synthesizer and $\Sigma\Delta$ synthesizer noise performance, we examine the dynamic performance of the synthesizer. As has already been shown, there is excellent agreement between measured and calculated performance using the analytical model implemented by the PLL Design Assistant tool. In addition to validating frequency domain behavior via phase noise performance, we can compare calculated and measured time domain behavior via the step response.

Figure 7-15 shows the nominal calculated response as well as the measured re-

sponse of the PFD/DAC synthesizer to a 10MHz frequency step in the divide value. One point of note is that, for such a large frequency step, classical frequency synthesizers would typically exhibit non-linear cycle slipping. The PFD/DAC synthesizer does not cycle slip because of its very high closed loop bandwidth, but rather exhibits small signal linear settling behavior.

Comparison between the step responses reveals a slight discrepancy between calculated and measured performance. Experiments with the PDA reveal that the difference can be attributed to the prototype system open loop gain being 8% higher than nominal, an acceptable error given component tolerances. We know from phase noise measurements that this small gain error results in minimal change to the shape of the phase noise profile.

The inset in the figure shows the settling response of the PFD/DAC synthesizer. Settling to within 1000ppm is achieved in $6\mu s$, and is dominated by the long tail transient associated with the pole-zero doublet present in the loop filter [35]. Settling behavior was limited to a 1000PPM measurement by the equipment used to perform measurements. The PFD/DAC synthesizer settling time is much faster than classical fractional-N synthesizers, which typically have settling times on the order of 10's to 100's of μs.

7.3.4 PFD/DAC Synthesizer Vs. $\Sigma\Delta$ Synthesizer

The primary goal of the PFD/DAC synthesizer is to reduce the impact of quantization noise on synthesizer phase noise performance. Thus far, we have not determined how much quantization noise suppression is achieved by the PFD/DAC synthesizer in comparison to state-of-the-art $\Sigma\Delta$ synthesis.

Figure 7-16 presents measured results for the PFD/DAC synthesizer with phase swapping enabled compared to a 2^{nd} order $\Sigma\Delta$ synthesizer. Because the $\Sigma\Delta$ synthesizer does not employ phase swapping, its low frequency noise is 2dB lower than the PFD/DAC synthesizer. However, at intermediate offset frequencies, the PFD/DAC synthesizer demonstrates 29dB of quantization noise suppression! This is much more noise suppression than achieved by prior work that employs active noise cancellation

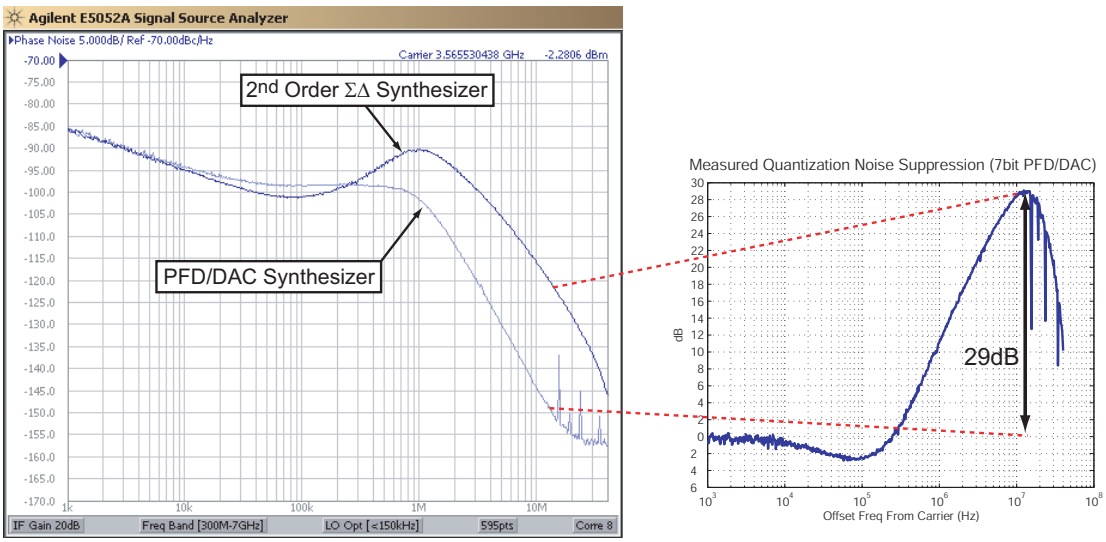


Figure 7-16 Measured Performance: PFD/DAC Synthesizer Vs. ΣΔ Synthesizer

techniques [19, 20], which achieve 16dB and 15dB suppression, respectively.

It is likely that the PFD/DAC synthesizer attenuates the quantization noise by more than 29dB, but the filtered white noise produced by the phase swapping process masks the noise suppression provided by the PFD/DAC. To reinforce this observation, Figure 7-17 presents measurements performed with the phase swapping function disabled. The noise suppression with phase swapping disabled is 31dB, an intuitive result since the phase swapping process increases low frequency noise by 2dB. Of course, phase swapping is an important technique that is necessary if the synthesizer is used in a mixer based transmitter since spurs are highly undesirable.

7.3.5 Impact of Sample-and-Hold Loop Filter and Spurious Performance

In this section we compare synthesizer performance with the S/H circuitry enabled and disabled. Ideally, the S/H eliminates all spurs in the system by removing the shape mismatch present in the PFD/DAC (see Chapter 3) and the reference spur. As discussed in Chapter 6, practical S/H circuitry will suffer from charge injection related spurs at the reference frequency. Hspice simulations suggest that the reference

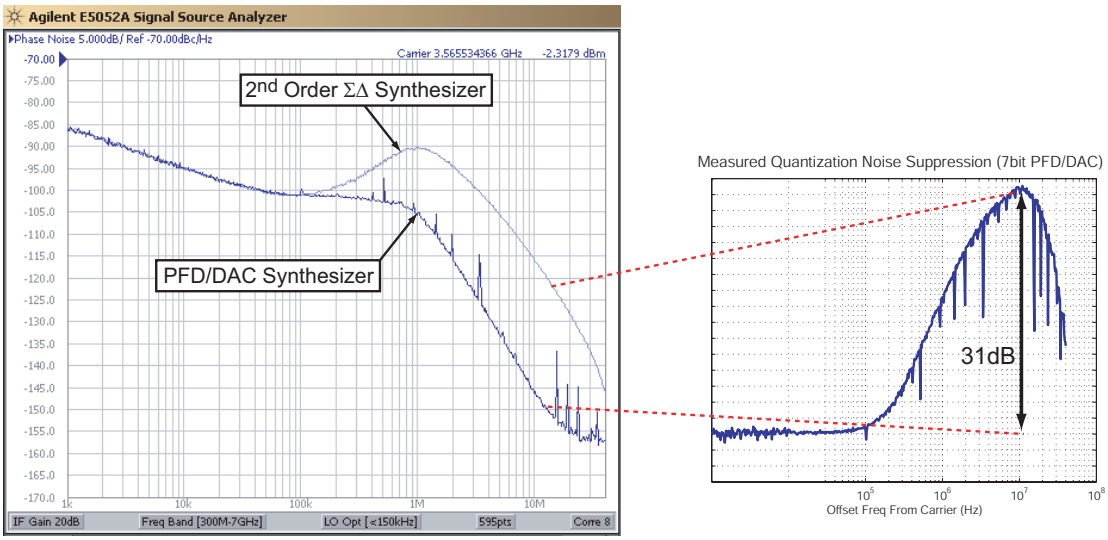


Figure 7-17 Measured Performance: PFD/DAC Synthesizer Vs. ΣΔ Synthesizer With Phase Swapping Disabled

spur in the PFD/DAC synthesizer will be below -90dBc, a very good performance level.

We have already noted that spurs are present in the PFD/DAC synthesizer output spectra. This is because there is coupling on-chip, and possibly coupling on-board, as well as coupling through the package bondwires that adversely impacts spurious performance. The PFD/DAC synthesizer IC was laid out with the intent of utilizing a high resistance non-epitaxial layer substrate. However, at the time of fabrication, this material was not available, and low resistance substrate material was used to fabricate the IC. It is therefore also possible that noise couples through the substrate from digital circuitry switching at the fractional spur frequency to sensitive analog nodes. It is also possible that on-chip supply lines were insufficiently bypassed and that spurious noise therefore couples through them. Because spurs are present in the system, they must be characterized for worst case performance. A solution to the spurious problem that can be implemented in the future is to use spread spectrum clocking techniques. By randomizing the clock signal to the digital circuitry, periodic noise caused by the periodic sequences processed by the digital ΣΔ modulator and accompanying circuits can be reduced.

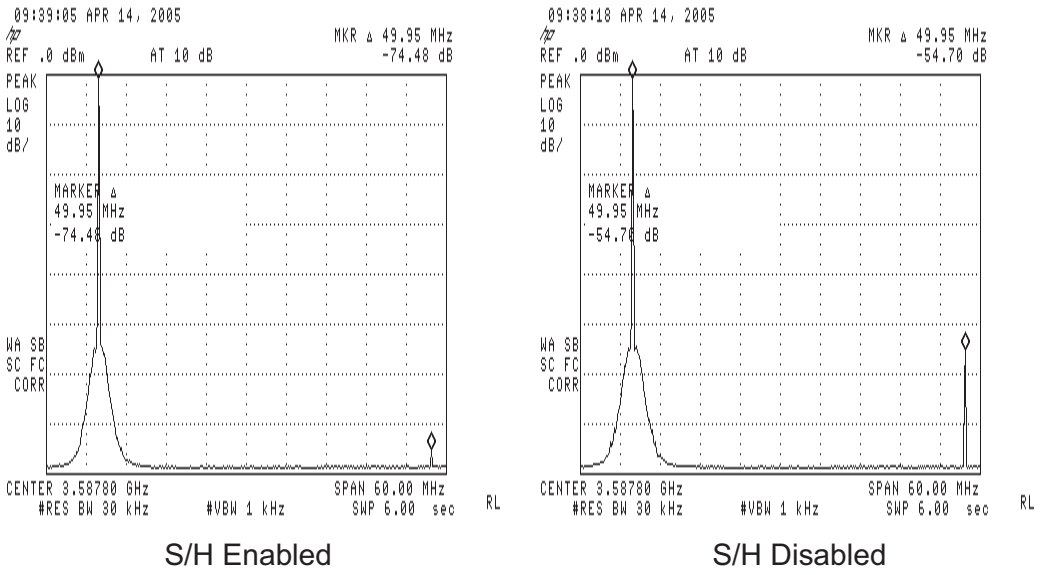


Figure 7-18 S/H Loop Filter Attenuation of the Reference Spur

Figure 7-18 demonstrates the ability of the S/H loop filter to attenuate the reference spur. The left plot presents the synthesizer with the S/H enabled, and the measured reference spur magnitude is $-74dBc$. When the S/H is disabled, the reference spur increases to $-55dBc$, indicating that the reference spur is attenuated by $19dB$ by the S/H, a significant improvement.

Ideally, the reference spur would be completely eliminated. As has already been discussed, charge injection in the S/H network will prevent complete removal of the reference spur. Additionally, the same on-chip coupling mechanisms that limit fractional spur cancellation (namely, supply, substrate, and bondwire coupling) also will provide coupling paths for reference spur energy. In future implementations of the PFD/DAC synthesizer a high resistivity substrate should be used, and the VCO and loop filter implemented on chip to reduce or eliminate bondwire coupling mechanisms.

Fractional spur performance is measured by programming the synthesizer with a variety of fractional values and observing the resulting spurs. Because the synthesizer is constructed with a 20-bit input, there are 2^{20} possible input combinations, which is an unreasonably large number of output frequencies to be able to check every possible output channel. In a real system, not every channel will be used, and so we can check a subset of outputs based on some nominal channel spacing.

We can further reduce the search space by eliminating combinations that result in integer divide values, since the synthesizer will then be operating as an integer-N synthesizer. Finally, we understand that the worst case fractional spurs will be those that appear in-band (meaning, inside the 1MHz closed loop synthesizer bandwidth), because they will not be attenuated by the synthesizer dynamics. The PFD/DAC synthesizer spurious performance was measured by varying the divide value over a range of fractional combinations that result in in-band or near in-band spurs. The worst case measured spur is $-45dBc$, indicating that the on-chip coupling is severely limiting performance. While this level of spurious performance is worse than desired, it is comparable to prior work that utilizes active quantization noise cancellation [19, 20].

Figure 7-19 presents measured spur results for the PFD/DAC synthesizer 3.6GHz output over a range of 125 channels spaced at 800kHz increments between 3.56GHz and 3.66GHz. The measured channel spacing, when divided down by the band select divider, results in the 200kHz channel spacing used by 900MHz band GSM transmitters. In addition, five channels that result in large, low frequency fractional spurs were measured and included in the plot to ensure that in-band spurious performance is adequately measured. We note that, as expected, the worst spurs appear in-band, an area not often reported in the synthesizer literature. Once the 1MHz loop bandwidth is exceeded, the filtering action of the PLL dynamics clearly causes the spur profile to roll off according to the filter shape. There is some degree of overlap in the measured spurs since harmonically related fractional divide values can produce spurs at similar frequencies.

7.3.6 Comparison to Prior Work

The PFD/DAC synthesizer achieves excellent noise performance while achieving a very high (1MHz) closed loop bandwidth. In this section, we compare measurements of the proposed architecture against prior work that employs active cancellation of the quantization noise [19, 20].

As table 7.2 shows, the PFD/DAC synthesizer exhibits higher bandwidth, more

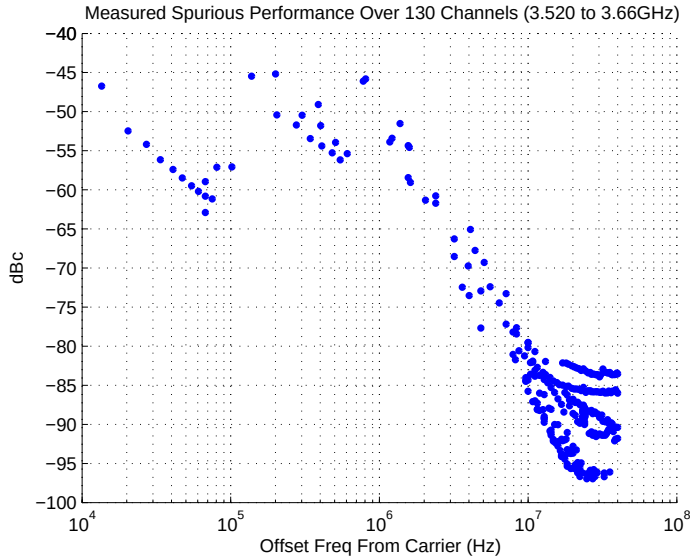


Figure 7-19 Measured Spurious Performance

noise suppression, simpler modulator control, and comparable spurious performance to prior work. The increased power cost is somewhat deceiving. The main ways to reduce power are to use a more advanced technology that achieves higher speed operation at lower power, and to lower the number of bits in the PFD/DAC. In the $0.18\mu m$ process used to fabricate the prototype synthesizer, the high speed differential logic used by the PFD operated slightly beyond the nominal process knee where speed and power do not tradeoff linearly.

Power is spent in the synthesizer mainly in the high speed logic required to create the one VCO period wide charge-box. In particular, the output logic that drives the unit element switch pairs must be capable of driving, in the case of the prototype 7-bit PFD/DAC synthesizer, 128 unit elements with fast enough edges to create a well defined 278ps wide charge-box. As measurements have shown, other noise sources in the system, particularly the timing mismatch in the PFD/DAC, limits the measurable noise suppression level from the calculated value $6.02 \cdot \mathrm{B}$ dB, where B is the number of bits in the PFD/DAC, to something less. For the prototype system, we are able to measure 29dB of suppression for a 7-bit PFD/DAC.

To reduce power and obtain comparable overall noise performance, a 5-bit PFD/DAC synthesizer could be employed. Offering 30dB of potential quantization noise suppres-

	[19]	[20]	This Work
Div Control	2^{nd} Order $\Sigma\Delta$	3^{rd} Order $\Sigma\Delta$	1^{st} Order $\Sigma\Delta$
DAC Control	3^{rd} Order $\Sigma\Delta$	2^{nd} Order $\Sigma\Delta$	1^{st} Order $\Sigma\Delta$
Bandwidth	$460kHz$	$700kHz$	$1MHz$
Output Frequency	$2.4GHz$	$2.1GHz$	$3.6GHz$
Phase Noise @ 10MHz (normalized to 2.1GHz output)	$-133dBc/Hz$	$-135dBc/Hz$	$-151dBc/Hz$
Largest In-band Spur	-45dBc	-55dBc	-45dBc
Noise Suppression	**16dB**	**15dB**	**29dB**
Core Power	61mW	28mW	110mW

Table 7.2 Comparison of Synthesizers Employing Active Quantization Noise Cancellation (All are implemented in 0.18um CMOS)

sion, a 5-bit PFD/DAC synthesizer would require dramatically lower power because only 32 elements would be driven by the PFD logic. Figure 7-20 shows the affect a 5-bit PFD/DAC would have on the prototype synthesizer's performance. As the plot shows, both the low frequency and high frequency (20MHz) noise performance are the same for the 5-bit and 7-bit PFD/DAC implementations. There is a slight performance degradation over the frequency range from 4MHz to 10MHz, where the 5-bit implementation raises the noise level by 2dB. If it is desired to meet the same performance target, a 6-bit PFD/DAC could be used, or an additional pole placed at 10MHz would attenuate the slight amount of added noise using the 5-bit PFD/DAC.

While the power reduction obtained by reducing PFD/DAC size would not be exactly linear because of overhead costs, a safe estimate would be $40-50\%$ core power savings for the 4X reduction in the DAC size. An additional benefit would be that the PFD/DAC timing mismatch that has been demonstrated to limit noise performance would be reduced, because the parasitic capacitance that determines its magnitude would be lowered.

Further power reduction could be obtained by designing the system for a 1.8GHz output rather than 3.6GHz output. This design change would have a large impact in a 0.18um CMOS process, because single-ended dynamic logic could be used to implement all of the PFD/DAC logic functions. Single-ended dynamic logic (such as the True Single Phase Clocked (TSPC) flip-flop [61]) offer dramatic power savings

Specification	Value
Technology	$0.18\mu m$ CMOS (National Semiconductor)
Date Rate	Up to 1Mb/s GMSK Measured (900MHz/1.8GHz Bands)
Reference Frequency	50MHz
Bandwidth	$1MHz$
Phase Noise @ 100kHz (3.6GHz output)	$-98dBc/Hz$
Phase Noise @ 20MHz (3.6GHz output)	$-155dBc/Hz$
Charge-pump Current	6.6mA
PFD/DAC Resolution	7-bit
Largest In-band Spur (3.6GHz output)	-45dBc
Quantization Noise Suppression	29dB
Core Power	$110mW(1.8V)$
Digital Power	5.4mW(1.5V)
I/O Buffer Power	37mW(1.8V)
Total Area	2.7mmX2.7mm
Active Area	1.8mmX1.5mm

Table 7.3 Summary of PFD/DAC Synthesizer/Transmitter Performance

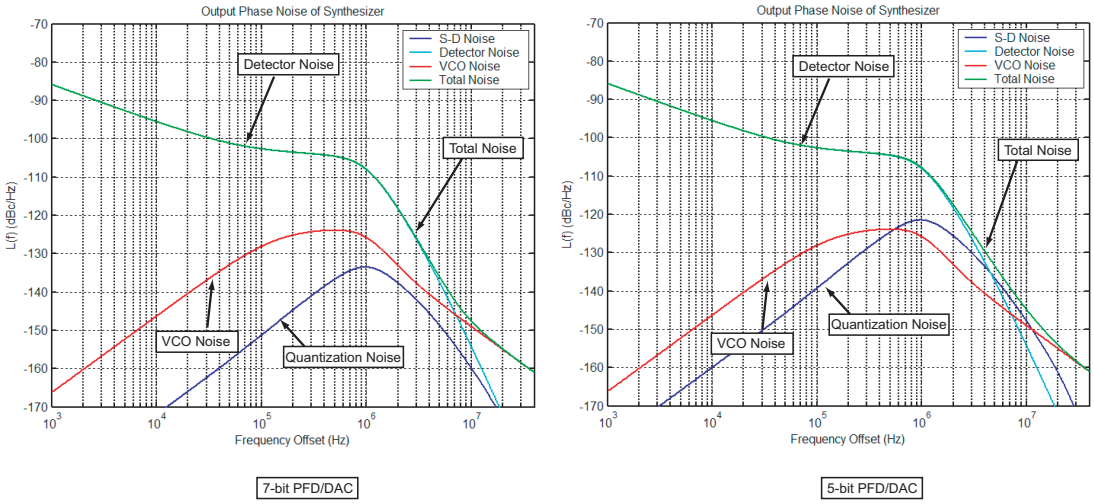


Figure 7-20 Comparison of Calculated Noise Performance for a 7-bit Vs. 5-bit PFD/DAC

compared to differential source-coupled logic designs. Lowering the output frequency would double the impact of quantization noise (because the quantization step-size, the VCO period, has doubled), which could be compensated by doubling the PFD/DAC resolution.

In summary, the PFD/DAC measurement results indicate that the proposed technique achieves excellent noise suppression compared to prior work. By targeting both less aggressive quantization noise suppression levels and an output frequency more in-line with the process technology used, power can be lowered significantly. For example, using a 4-bit PFD/DAC still achieved 24dB noise suppression, a significant amount when compared to prior art [19,20], but would only require 16 DAC unit elements. Lowering PFD/DAC operating frequency would allow the use of single-ended dynamic PFD logic, further lowering power.

Digital power would also be lowered for a lower resolution PFD/DAC, since the digital power is dominated by the thermometer decoder and data weight averager circuit used to dynamically match the charge-pump unit elements. Data weight averager circuit complexity, and therefore power, grows as 2^B for a B-bit PFD/DAC. A 4X reduction in PFD/DAC size therefore corresponds to a 4X reduction in digital power. Buffer power is somewhat over-exaggerated because, to speed design time, the

I/O buffers were not power optimized. In a product-driven environment, the VCO and digital ΣΔ modulators would be implemented on-chip and buffers would not be required. Additionally, no buffers would be used to explicitly drive measurement signals off-chip.

Table 7.3 summarizes the performance measurements of the PFD/DAC synthesizer, and represents a superset of the specifications listed in table 7.1.

7.4 Modulated Synthesizer Measured Performance

Having characterized the PFD/DAC synthesizer un-modulated performance, we now examine measured results for the synthesizer modulated by a GMSK data stream.

A GMSK filtered random data sequence was generated on a PC and downloaded into a pair of 256k-bit FIFO memories. This data is input to the FPGA containing the ΣΔ modulators used to control the PFD/DAC. Use of the FIFOs and FPGA allows flexibility in programming both the ΣΔ modulators as well as the GMSK data sequence.

Figure 7-21 presents measured results for the PFD/DAC synthesizer modulated by 271kb/s GMSK data for both 900MHz and 1.8GHz output bands. We see that, in both cases, the measured spectra conform to the expected GMSK output profile, and demodulated eye diagrams measured on the Hewlett Packard Vector Signal Analyzer are wide open. One way to characterize the quality of opening in the eye diagram is to measure the rms phase error, which corresponds to the difference, in phase, between an ideal reference waveform and the measured de-modulated waveform. Some cellular standards, such as GSM, have stringent specifications requiring rms phase errors for 270kb/s data to be below 5 degrees rms. The eye diagrams in Figure 7-21 have measured rms phase error magnitudes of 2.8 degrees and 2.7 degrees for the 900MHz and 1.8GHz bands, respectively, indicating the excellent performance of the PFD/DAC based synthesizer.

Higher data rates with good performance are achievable using the PFD/DAC synthesizer because of the very high bandwidth achieved. Figure 7-22 presents measured

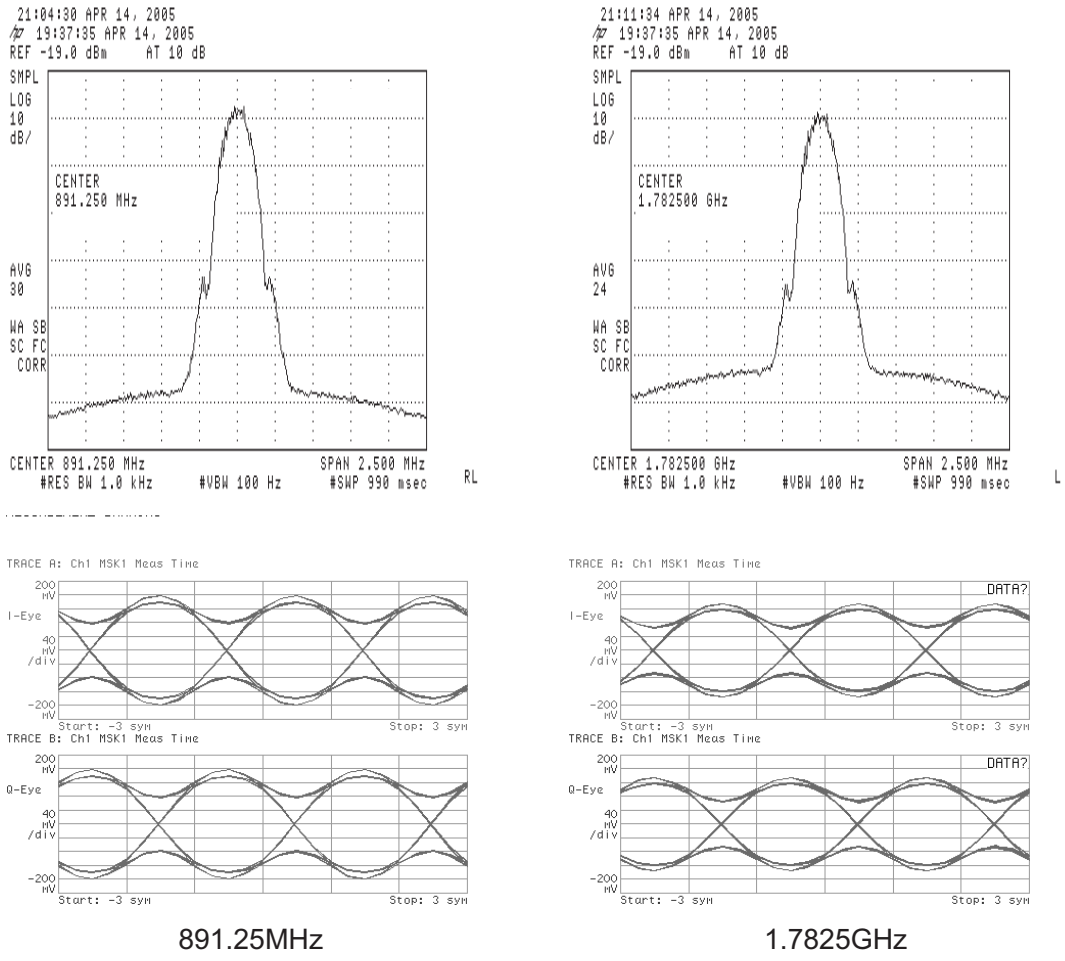


Figure 7-21 PFD/DAC Synthesizer Measured 271kb/s GMSK Spectra and Eye Diagrams

eye diagrams for the transmitter modulated by 500kb/s data. We observe that the eyes are still wide open, but not quite as open as for the 271k/s modulated synthesizer. This is because there is some small amount of inter-symbol interference (ISI) that is starting to close the eye. ISI can be caused by many different effects. Here, it is due to the filtering action the PLL dynamics perform on the GMSK data as it passes through the synthesizer. Rms phase error is 6.2 degrees and 6.5 degrees for the 900MHz and 1.8GHz bands, respectively.

As the data rate is increased further, increased levels of ISI are observed. Figure 7-23 shows measured results for a 757kb/s data rate, and Figure 7-24 measured results for a 1Mb/s data rate. Measured rms phase errors are 11 degrees (900MHz band)

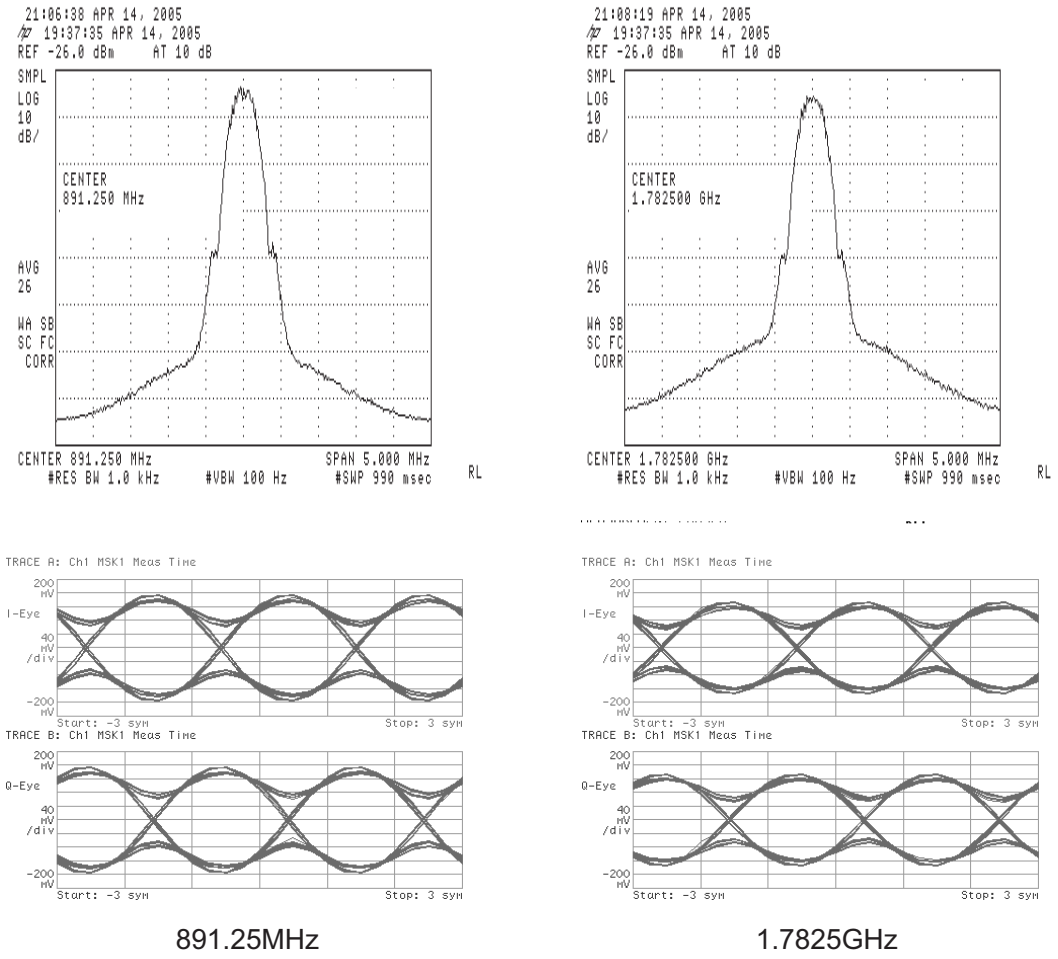


Figure 7-22 PFD/DAC Synthesizer Measured 500kb/s GMSK Spectra and Eye Diagrams

and 10.7 degrees (1.8GHz band) for the 757kb/s data, and 13 degrees (900MHz band) and 16 degrees (1.8GHz band) for the 1Mb/s data. The figures show that the data at 757kb/s exhibits eyes that are still wide open, while the 1Mb/s eyes are still open, but beginning to close due to increased ISI. This performance could be improved by applying a small degree of pre-emphasis filtering to the data to counteract the synthesizer dynamics. A pre-emphasis technique is proposed in [7], which makes a modification to the digital Gaussian filter used to generate the synthesizer input.

It would be possible to further increase the data rate by two means. The first is simply to increase the synthesizer bandwidth. This change would result in increased impact of low frequency noise, and, therefore, higher levels of output phase noise. A

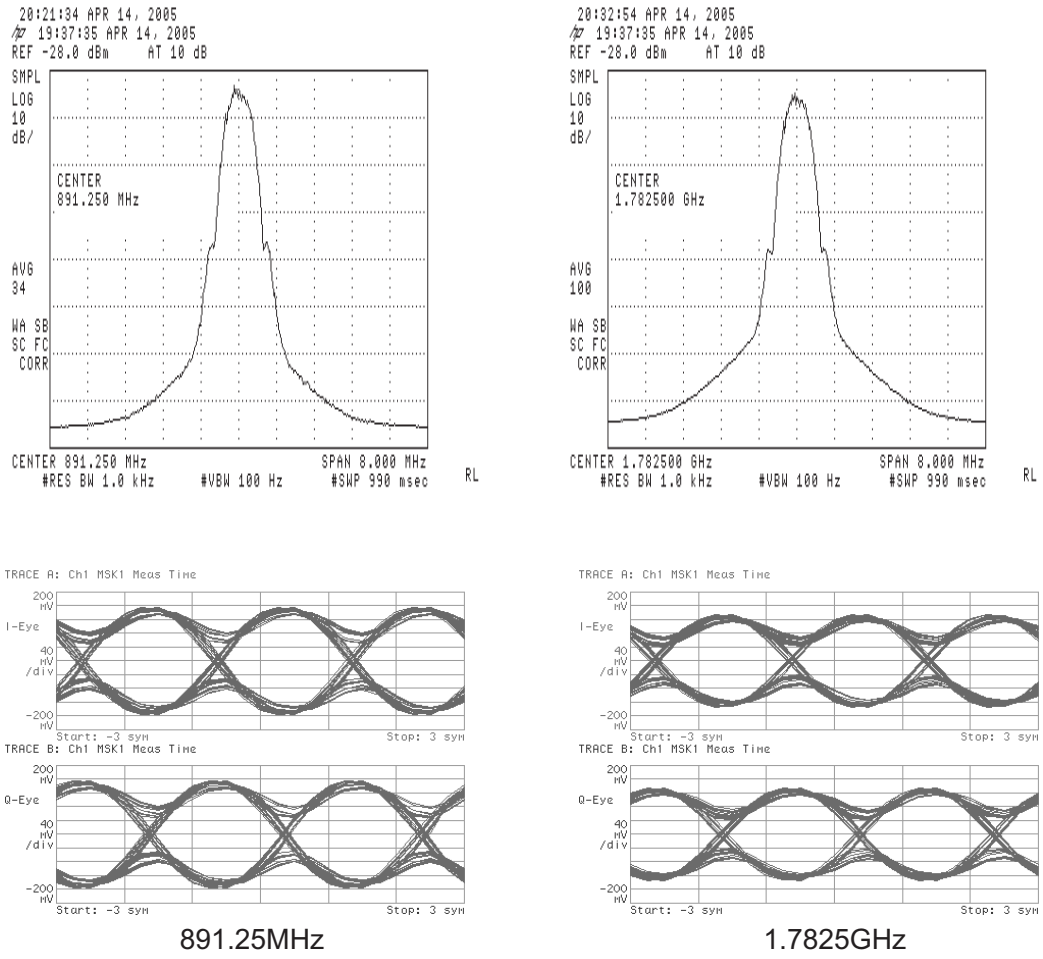


Figure 7-23 PFD/DAC Synthesizer Measured 757kb/s GMSK Spectra and Eye Diagrams

second approach would be to apply pre-emphasis filtering to the data in the manner proposed in [37]. The advantage of pre-emphasis is that it is a purely digital technique, and therefore adds little overhead cost. In [37], 2.5Mb/s GFSK data rates were achieved by pre-emphasizing data that passed through an 80kHz bandwidth $\Sigma\Delta$ synthesizer. The limitation of this technique is that the open-loop PLL gain must be known to properly pre-filter. A method to automatically calibrate the loop-gain using pre-emphasis is described in [34]. For the PFD/DAC synthesizer, much less aggressive pre-emphasis would be required due to its already high bandwidth compared to prior art, creating the attractive possibility of achieving data rates in excess of 1Mb/s with little digital overhead.

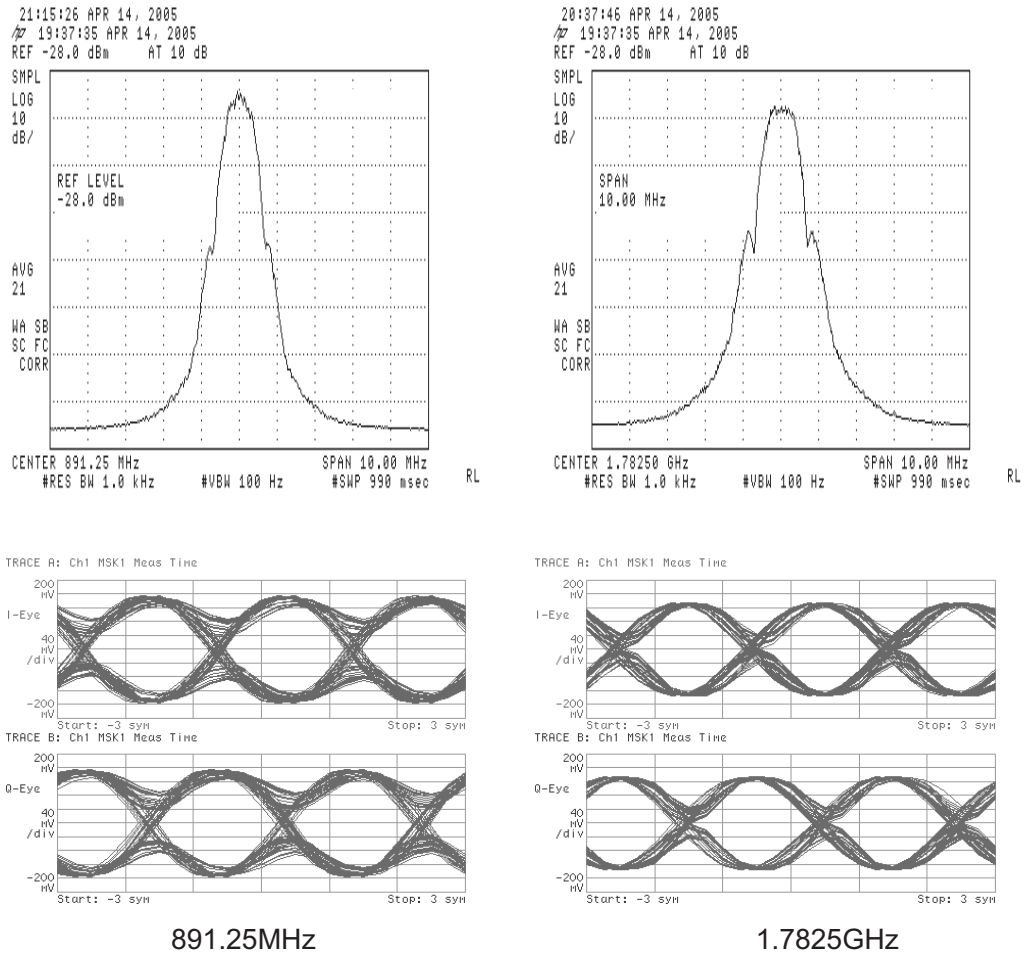


Figure 7-24 PFD/DAC Synthesizer Measured 1Mb/s GMSK Spectra and Eye Diagrams

7.5 Summary

In this chapter we have presented measured results for a prototype system implementing the proposed mismatch compensated PFD/DAC synthesizer architecture. Non-ideal circuit effects lead to reference and PFD reset jitter, two sources of phase noise that were not included in the initial modeling. Because of the large degree of programmability implemented in the synthesizer IC and test board, reference jitter induced phase noise and reset jitter induced phase noise were both identified as contributing to higher-than-expected noise performance. Circuit adjustments were implemented on the board level and subsequently proved through experiment that reference and reset jitter were the cause of the added noise.

Very close correlation between the analytical model proposed in [1] and measured results has been demonstrated, validating the analysis presented in Chapter 3 and simulations presented in Chapter 5. Using the analytical model, we are able to back extract key system level parameters that are not directly measurable. This is a powerful analysis technique that will be useful for a variety of PLL systems.

Finally, we have presented measured results for the system configured as a direct modulated GMSK transmitter, and proposed techniques to reduce the power requirements of the PFD/DAC architecture and to increase the achievable data.

Chapter 8

Conclusions and Future Work

This thesis has proposed techniques aimed at extending the bandwidth of fractional-N synthesizers. It has been shown that the current bottleneck to improved performance centers around managing the quantization noise introduced by the fractional-N dithering process. To this end, we have made several contributions and proposed several solutions.

8.1 Mismatch Compensated PFD/DAC Synthesizer

In Chapter 4, we proposed a mismatch compensated PFD/DAC synthesizer architecture that is able to dramatically reduce the impact of quantization induced phase noise. By creating a self-alignment between the cancellation charge-box and VCO period, the PFD/DAC achieves an intrinsic gain match between the quantization error and cancellation signal [21]. However, mismatch internal to the PFD/DAC structure will alter the cancellation charge-box from the ideal case and limit performance if left unchecked. We have proposed PFD/DAC mismatch compensation techniques that leverage dynamic element matching approaches described in the $\Sigma\Delta$ DAC literature. The timing mismatch and DAC unit element mismatch compensation blocks transform what would otherwise be a gain mismatch that would result in incomplete noise cancellation, into broadband noise sources that are filtered by the PLL. The non noise shaped timing mismatch was shown to be the limiting factor in overall syn-

thesizer low frequency noise performance for the prototype synthesizer, and should be the focus of future work. One possible solution would be to determine a means by which to noise shape the timing mismatch. The 29dB quantization noise suppression demonstrated by the prototype synthesizer allows the PFD/DAC synthesizer to simultaneously achieve very good noise performance and high bandwidth.

8.2 Analytical Modeling Contributions

In Chapter 3 we presented a new framework for analyzing fractional-N synthesizers by modeling them in an analogous manner to $\Sigma\Delta$ digital-to-analog converters, based on the analytical noise model proposed in [1]. The insight gained through this observation leads to analysis techniques that result in the mismatch compensated PFD/DAC architecture, which not only achieves an inherent gain match between the quantization noise signal and cancellation signal, but employs dynamic element matching techniques leveraged from the $\Sigma\Delta$ DAC literature to achieve high levels of quantization noise suppression.

Expressions derived from the model have been demonstrated to not only accurately predict simulated performance of the synthesizer, but to also allow key parameters that are not directly measurable to be back extracted by fitting measured data to the model. This powerful characterization technique is extremely useful for identifying key areas of concern for future iterations of synthesizer architectures.

8.3 Behavioral Modeling Contributions

In Chapter 5 we proposed behavioral modeling techniques that both validate the analytical model, as well as provide a good means by which to evaluate architectural performance tradeoffs. The behavioral model is used to evaluate the impact of circuit non-idealities on synthesizer performance. In particular, it was shown through behavioral simulation that the offset reset tri-state PFD topology, which has been frequently misunderstood in the literature, offers improved linearity over its overlap-

ping reset pulse counterpart, while offering similar noise performance if a S/H loop filter is employed and reference and reset jitter are low.

Perhaps more importantly, the methodology followed in the design of the synthesizer presented in Chapter 5 follows a logical progression that explores non-idealities in an evolving fashion. As more circuits are designed, SPICE simulation results are used to improve the model. Iteration between the behavioral model and SPICE level simulations provides a design methodology that is capable of capturing the impact of sometimes subtle non-idealities introduced by circuit imperfections without requiring prohibitively long simulation time.

Finally, as we see from the measured performance, the behavioral simulation results are only as accurate as the model allows them to be. In the case of the prototype PFD/DAC synthesizer, we discovered two noise sources (reference jitter and reset jitter) that were not included in the initial analytical model, but should be in future designs. It is important to note that the insight and intuition that a designer gains by going through the behavioral modeling process helps identify and eliminate new sources of noise very quickly. The model can later be updated to include these newly encountered non-ideal effects, and more emphasis given to the key circuits that determine their magnitude.

8.4 Circuit Contributions

Circuit techniques for high speed and low noise operation were proposed in Chapter 6. The proposed divider retiming circuit, when operated in a burst mode, offers the possibility of eliminating the meta-stability issue associated with re-synchronizing asynchronous signals. The retiming circuit can be activated during header times in burst mode systems, or at some pre-determined time to account for environmental drift. The proposed technique is general, applying to any circuit that operates at the same frequency, or at some sub-frequency, of a clock.

Another key circuit block used by the PFD/DAC is the unit current source element. The proposed unit element current source utilizes a dynamic element matching

technique to mitigate mismatch, and leverages the results of behavioral simulation to spend headroom not on the PMOS positive current sources, but rather on the critical, NMOS, negative current sources that create the noise cancellation charge-box key to PFD/DAC operation.

The high-speed logic used by the PFD/DAC uses several techniques to achieve high speed, low noise operation. Phase swapping transforms a gain mismatch that would otherwise result in large fractional spurs into a broadband noise source that is filtered by the synthesizer dynamics. A muxing function embedded into the differential flip-flops uses a proposed state mismatch insensitive architecture to isolate the flip flop input stage from a dynamically changing internal impedance that would otherwise add to timing mismatch.

8.5 Future Work

The mismatch compensated PFD/DAC technique reduces quantization noise to the point where intrinsic noise sources become the key concern in synthesizer design. This represents a paradigm shift by eliminating the noise-bandwidth tradeoff that exists in state-of-the-art $\Sigma\Delta$ synthesizer design. There are a few key considerations to be explored by future work.

8.5.1 Quantization Noise

The prototype synthesizer exhibits larger residual fractional spurs than desired. This behavior is attributed to coupling through the low resistivity substrate, on-chip supply lines, and bondwires. These spurs can be reduced through high levels of integration (to remove bondwire coupling of critical signals), use of non-epi substrates and improved circuit isolation strategy (to reduce substrate coupling), and increased on-chip de-coupling capacitance or by introducing linear regulators to isolate on-chip analog supplies(to reduce supply noise). An additional method that can be applied to reduce coupling-induced spurs is to introduce spread spectrum clocking [62] to randomize the edge transitions of the digital circuitry that processes information

containing fractional spur energy content.

The second conclusion drawn from measured prototype synthesizer results is that timing mismatch noise should be a key area of focus in future designs. In the prototype system, broadband timing mismatch noise dominates synthesizer phase noise performance, and results in higher low frequency noise levels than exhibited by the reference integer-N configuration. A general purpose synthesizer will have to process constant divide values, so phase swapping noise must be lowered.

One very easy fix to the timing mismatch problem is to use less unit elements in the system so that the parasitic capacitance contributing to its magnitude is reduced. Another solution is to target a lower output frequency so that single-ended logic and smaller devices can be used in the logic that creates the mismatch, again with the goal of creating less sources of parasitic capacitance. An added side benefit of implementing either of these changes will be a large decrease in overall system power, since driving the unit elements represents the major power sink in the system.

While the two solutions proposed above for timing mismatch are viable in a practical system, a more elegant approach would be to find a way to noise shape the timing mismatch. In the present system, an LFSR random number generator is used to control the phase swapping process, resulting in a white noise profile. As has been shown with the unit element mismatch, if mismatch noise can be shaped, it will not dominate the phase noise profile at low frequencies. It may be possible to shape the phase swapped timing mismatch noise as well. This is a key area for future research because, if it is possible, then the mismatch compensated PFD/DAC synthesizer will truly be limited by purely intrinsic noise sources, and quantization noise effectively eliminated from design consideration!

8.5.2 Intrinsic Noise

Once quantization noise is eliminated, intrinsic noise becomes the area of focus for future research. For very high performance systems, low frequency noise will be dominated by charge-pump noise, reference jitter, and reset jitter. Reference jitter should be attacked by technology advances used to create the reference source and

more careful reference buffer design. Reset jitter can be improved through more careful design and simulation of the reset circuitry employed by the PFD logic.

Circuit topologies utilizing noise suppression techniques, such as the resistive degeneration technique employed by the PFD/DAC unit element, offer lower charge-pump noise at the cost of voltage headroom. More power can be spent in generating the reset delay, but any tradeoff that results in increased power is undesirable. Because the charge-pump and delay represent analog signals, their performance does not scale well with technology. In fact, decreasing gmr_o products, increasing 1/f noise, and increasing transistor leakage currents adversely affect analog circuit performance.

The PFD/DAC synthesizer employs discrete time signal processing techniques, such as dynamic element matching in the charge-pump and PFD and employing a S/H loop filter to reduce spurs. Purely digital techniques scale very well with technology, so investigation of digital architectures for frequency synthesizers is a good avenue for future research. Some work has already begun in this area [63], and it offers a wide open space for future innovation.

Appendix A

Chip Pinout and Bonding Diagram

Table A.1 lists the pinout of the prototype PFD/DAC synthesizer IC. Figure A-1 depicts the bonding diagram used to package the die. The package has a metal casing, so numerous down-bonds are used for GND connections between the IC and the package.

Pin Number	Name	Function
1	TM_OUT	Test Mode Output
2	VDIGHI	3.3V Digital Supply
3	VCO_DIV	900MHz/1.8GHz Output
4,5,9	VDDVCO	1.8V VCO Buffer Supply
6,8	GND	
7	VCO_IN	VCO Input
10,11,14	VDDREF	1.8V Ref Buffer Supply
12,26	NC	No Connection
13	REF_IN	Reference Input
15,20,22,25,29	PWRP	1.8V Core Analog Supply
16	Div0	Divider Control LSB
17	Div1	Divider Control
18	Div2	Divider Control
19	Div3	Divider Control MSB
21	Bias Core	450uA (nom) Core Bias Input
23	Biasn	NMOS Charge-pump Bias
24	Biasp	PMOS Charge-pump Bias
27	Vcm	Op-amp V_+ Terminal
28	Amp_buff	Inverting Buffer Output
30	Amp_out	Op-amp Output
31	Amp_vm	Op-amp V_- Terminal
33	Vdump	Charge-pump Dump Voltage
34	SER_CLK	Serial Configuration Register Clock
35,38,47	DVDD	(1.5V) Digital Core Supply
36	SER_DAT	Serial Configuration Register Data
37	SER_LOAD	Serial Configuration Register Load
39	DAC0	PFD/DAC Control LSB
40	DAC1	PFD/DAC Control
41	DAC2	PFD/DAC Control
42	DAC3	PFD/DAC Control
43	DAC4	PFD/DAC Control
44	DAC5	PFD/DAC Control
45	DAC6	PFD/DAC Control
46	DAC7	PFD/DAC Control MSB
48	$\Sigma\Delta$ Clock	FPGA $\Sigma\Delta$ Clock Output to FPGA

Table A.1 Chip Pinout

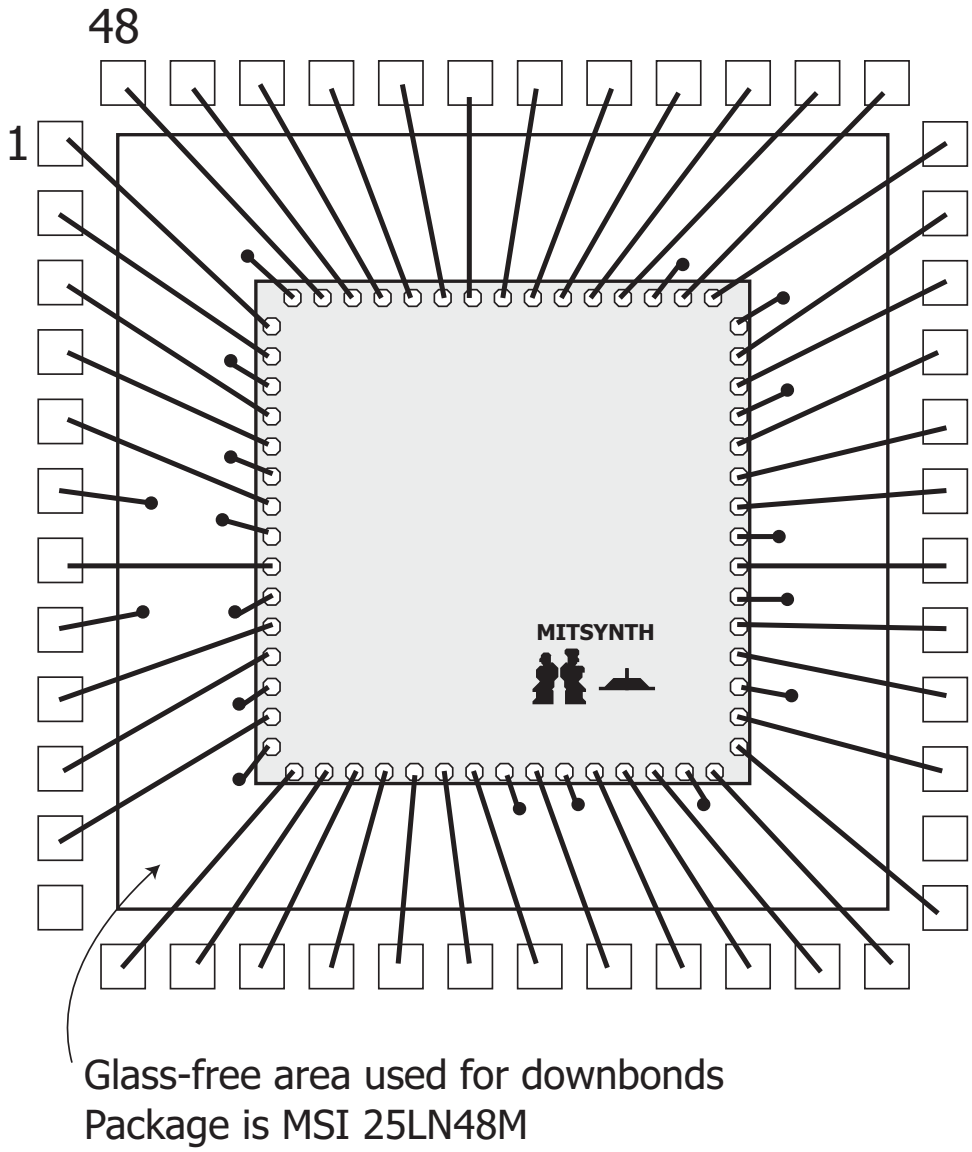


Figure A-1 Chip Bonding Diagram

Appendix B

Synthesizer Configuration Register

The prototype synthesizer IC uses a three wire, serial interface to program its configuration register. A simplified schematic of the serial interface and configuration register is depicted in Figure B-1.

B.1 Register Organization

The programmable functions are broken up into several categories. To create better understanding of the individual categories, different tables are used for each. Simplified schematics that explain various control bits are included in this appendix as well, so that a user of the prototype synthesizer IC can understand what operation each bit in the configuration register performs. The default values listed in the tables reflect a nominal *suggested* value for each bit, not a hard-coded default. In other words, the configuration register must be programmed at startup, and the default values listed in the tables are a recommended starting point.

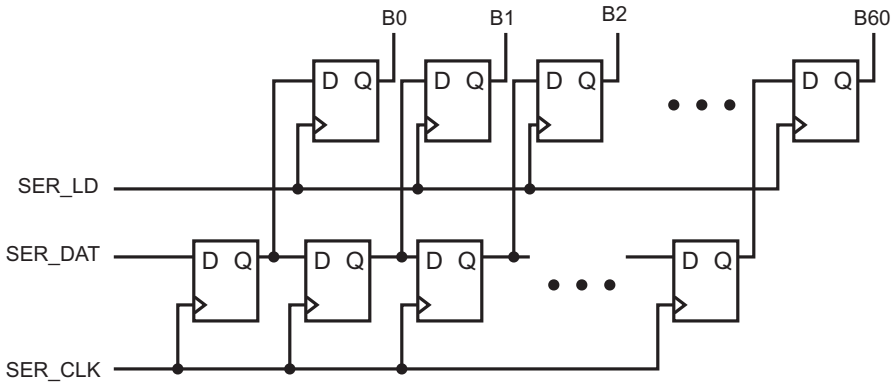


Figure B-1 Configuration Register

Bit	Name	Function	Default Value
1	RES_INT	Loop Filter Reset Switch	L
2-8	Not Used		L
9	SEL_ΣΔ _CLK	Selects FPGA Clock	L, (0=REF, 1=DIV)
10	EN7bit	7-bit PFD/DAC Enable	H
13	EN_DAC	PFD/DAC Enable and DWA Reset	Pulse L→H at Startup
14	DACRES	PFD/DAC Resolution	H (H=7-bit, L=6-bit)
15	EN_SWAP	Phase Swapping Enable	H (L=disable)
16	SH_DIS	S/H Disable	L (H=disable)
45	EN_BAND_SEL	Band Select Divider Enable	L
46	BAND_SEL	Band Selection Control	L (L=900MHz, H=1.8GHz)
47	REF_SEL	Reference Buffer Selection	H (L = Pass Reference input straight through H = Divide Reference input by 2)
48	Not Used		L
49	TM_SEL	Test Mode Select	L (L = Output S/H Samp signal H = Output PFD Swap Signal)
50	TM_EN	Test Mode Enable	L

Table B.1 Configuration Register General Function Control Bits

B.2 General Configuration

Table B.2 describes the functions of the general synthesizer architectural configuration bits controlled by the shift register.

To clarify some the functions provided by control bits in this portion of the configuration register, the user is referred to Figure B-2. The integrator reset switch is pulsed at startup to reset the loop filter components across the op-amp. During normal operation this switch is turned off. It has been sized small enough such that its parasitic capacitances are insignificant relative to the rest of the capacitances present at the op-amp output node and summing junction.

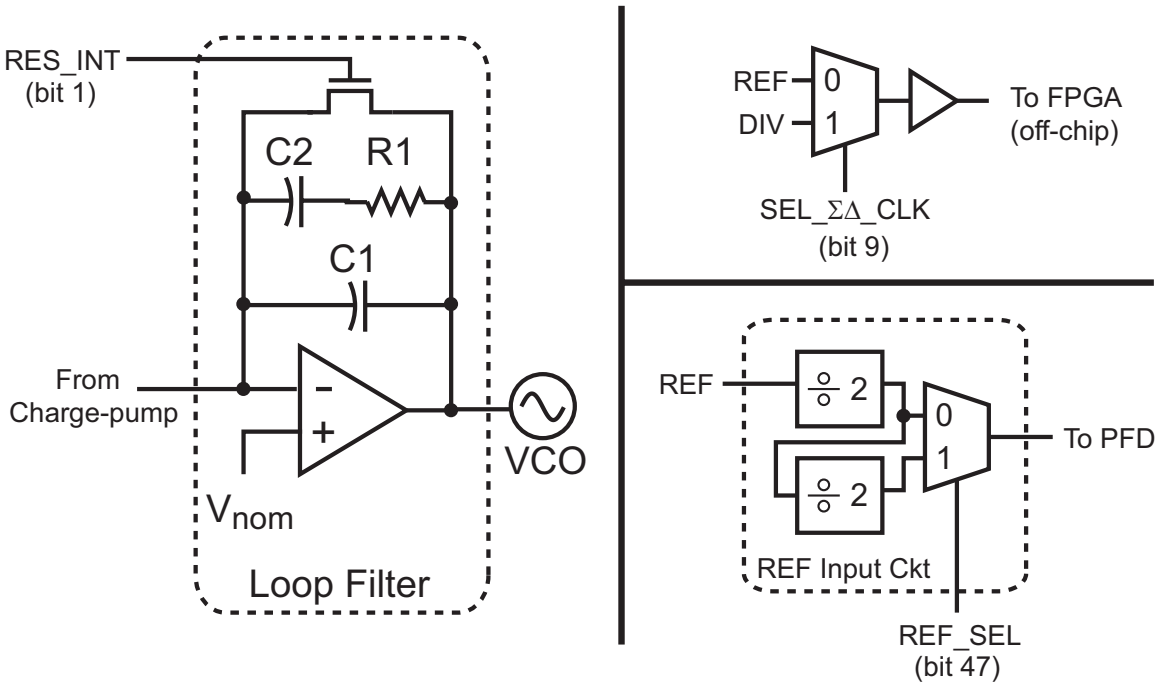


Figure B-2 Functions Controlled by the general configuration bits of the configuration register

Bit	Name	Function	Default Value
17	CASC_3	Charge-pump Cascode Length	H
18	CASC_2	Charge-pump Cascode Length	H
19	CASC_1	Charge-pump Cascode Length	H
20	CASC_4	Charge-pump Cascode Length	H
21	DIV_B2	Divider Bias MSB (150uA)	L
22	DIV_B1	Divider Bias (75uA)	H
23	DIV_B0	Divider Bias LSB (37.5uA)	H
24	PFD_B3	PFD Logic Bias MSB (1.2mA)	H
25	PFD_B2	PFD Logic Bias (600uA)	H
26	PFD_B1	PFD Logic Bias (300uA)	L
27	PFD_B0	PFD Logic Bias LSB (150uA)	L

Table B.2 Configuration Register Bias Control Bits

B.3 Bias Configuration

Table B.3 describes the bias functions controlled by the configuration register. The nominal bias configuration assumes that the synthesizer IC (Pin 24, as depicted in Table A-1), receives a 450uA bias current from the test board.

Figure B-3 presents the three bias functions controlled by the configuration register. The PFD and divider bias values are controlled by programmable DACs, whose LSB values are determined by the aforementioned 450uA bias input to the synthesizer IC pin 24. Any change from the nominal 450uA value will scale the LSB current of both PFD and divider DACs proportionally.

The unit element NMOS cascode bias transistor effective length was made programmable so as to account for unexpected headroom issues. Figure B-3 shows a simplified schematic of the scheme used to control the cacode bias device length.

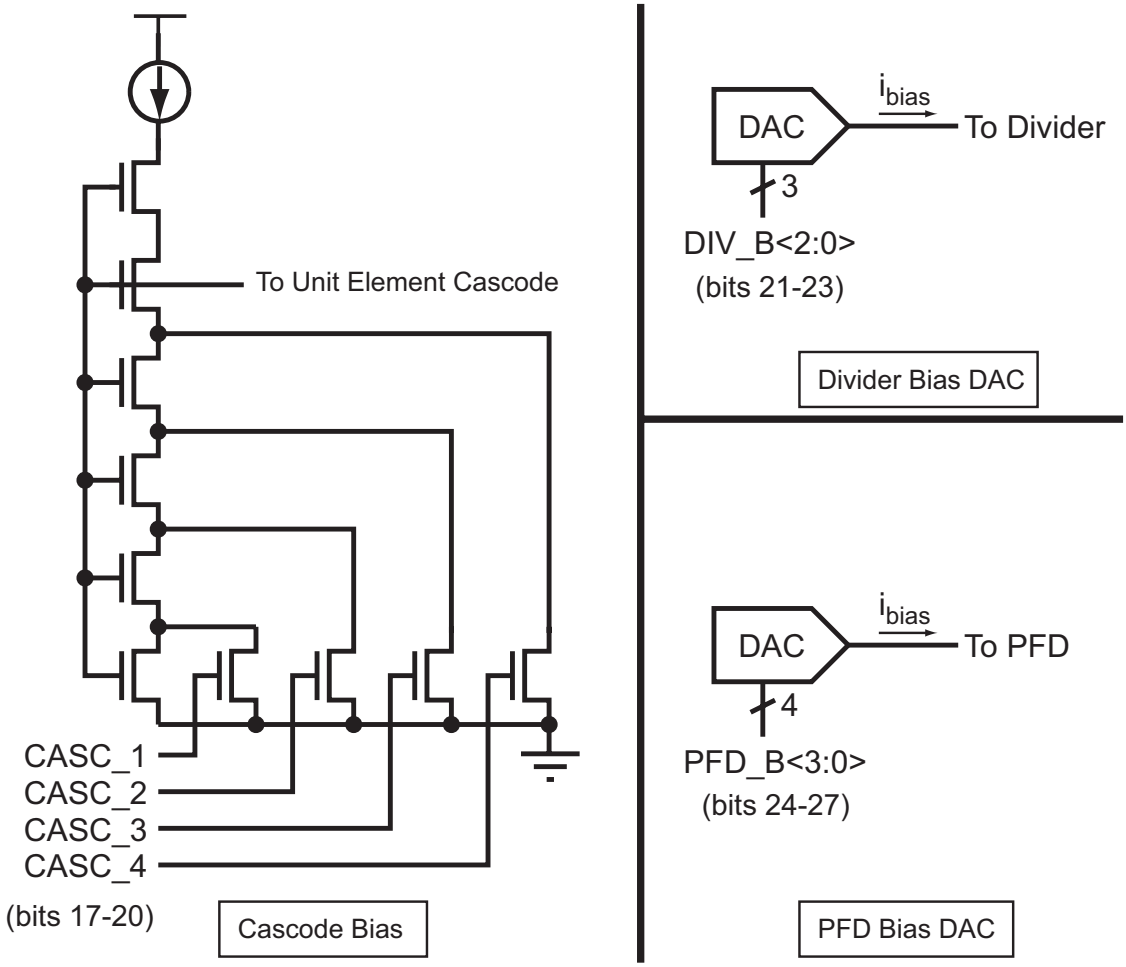


Figure B-3 Bias Circuitry

Bit	Name	Function	Default Value
11	Div5	Sets MSB of 6-bit Divider	L
12	Div4	Sets 2^{nd} MSB of 6-bit Divider	L
28	DR_SEL	Selects the Div Retimer Mode	L (L = Use Divider Retimer Output, H = Use Shift Register Output(Bit 29)
29	DIV_PHASE	Manual Divider Retiming Edge Selection. Used in conjunction with bit 28 to manually choose to retime the divider on a particular edge. This bit is only valid if DP_SEL is H.	L (L = Retime on VCO Rising Edge, H = Retime on VCO Falling Edge
30	SAMP_PHASE	Samples the divider retimer output so that it can be run periodically.	H (L = Use the last internally generated edge value from the divider retimer circuitry. H= Use the output of the divider retimer continuously. H→L produces a sampling action.)
40	RT_BIASEN	Enables the Divider Retimer Bias	

Table B.3 Configuration Register Divider Retimer Control Bits

B.4 Divider Retimer Configuration

Figure B-4 presents the divider architecture employed by the prototype transceiver. The overall divide range, N, is determined by the control bits $Div < 5 : 0 >$. Bits $Div < 5 : 4 >$ are hard-coded through the configuration register, while bits$Div < 3 : 0 >$ are controlled by the FPGA, and therefore can change during normal operation.

The divider retimer configuration control bits determine the overall state of the divider retimer circuit, a simplified schematic of which is presented in Figure B-5. DP_SEL determines whether the retimer output or a manual control bit (DIV_PHASE) is used to determine whether to synchronize the divider on the rising or falling VCO edge. SAMP_PHASE can be used to sample the retimer output, and RT_BIASEN

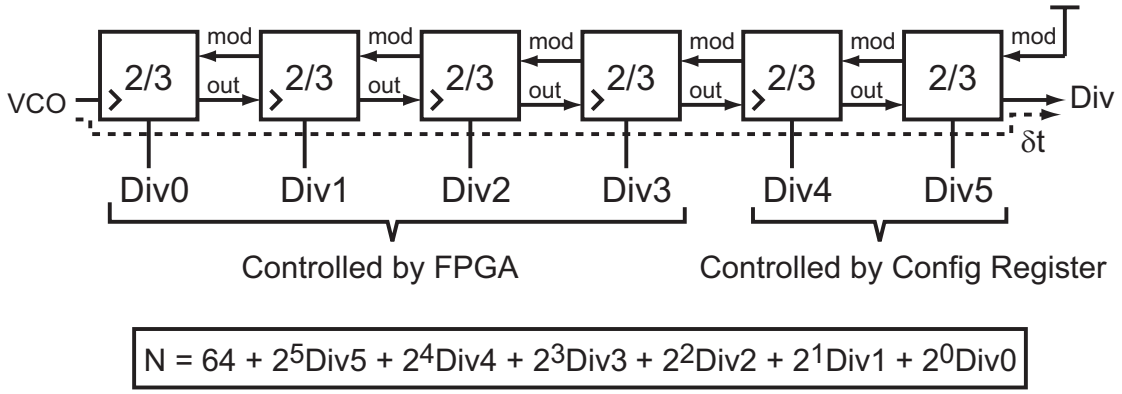


Figure B-4 High-speed, Multi-modulus Divider

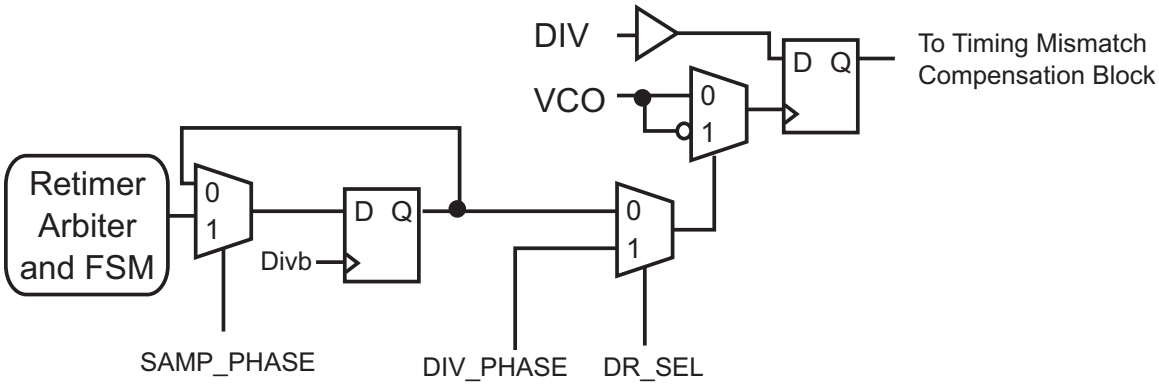


Figure B-5 Simplified Divider Retimer Schematic

enables the retimer circuitry. If RT_BIASEN is low, the retimer circuitry is disabled and draws no power.

Bit	Name	Function	Default Value
41	SAMPDEL_7	Delay Bit for S/H On-time	H
42	SAMPDEL_8	Delay Bit for S/H On-time	H
51	SAMPDEL_6	Delay Bit for S/H On-time	H
52	SAMPDEL_5	Delay Bit for S/H On-time	H
53	SAMPDEL_4	Delay Bit for S/H On-time	H
54	SAMPDEL_3	Delay Bit for S/H On-time	H
55	SAMPDEL_2	Delay Bit for S/H On-time	H
56	SAMPDEL_1	Delay Bit for S/H On-time	H
57	SAMPDEL_0	Delay Bit for S/H On-time	H
58	SAMP_SEL1	S/H Delay Mux Select	L
59	SAMP_SEL0	S/H Delay Mux Select	L

Table B.4 Configuration Register S/H Control Bits

B.5 S/H Configuration

Figure B-6 depicts a simplified version of the circuit used to generate the sample pulse used by the S/H loop filter. A fixed delay and programmable delay are used to determine the Samp signal pulse-width. The non-fixed delay is programmable between two binarily weighted delay lines. Each delay line is comprised of delays which can be programmed to be in either "fast" or "slow" mode. A large combination of possible delays is possible using the programmability of the delay line. The overall delay pulse-width can be measured by selecting the appropriate test mode

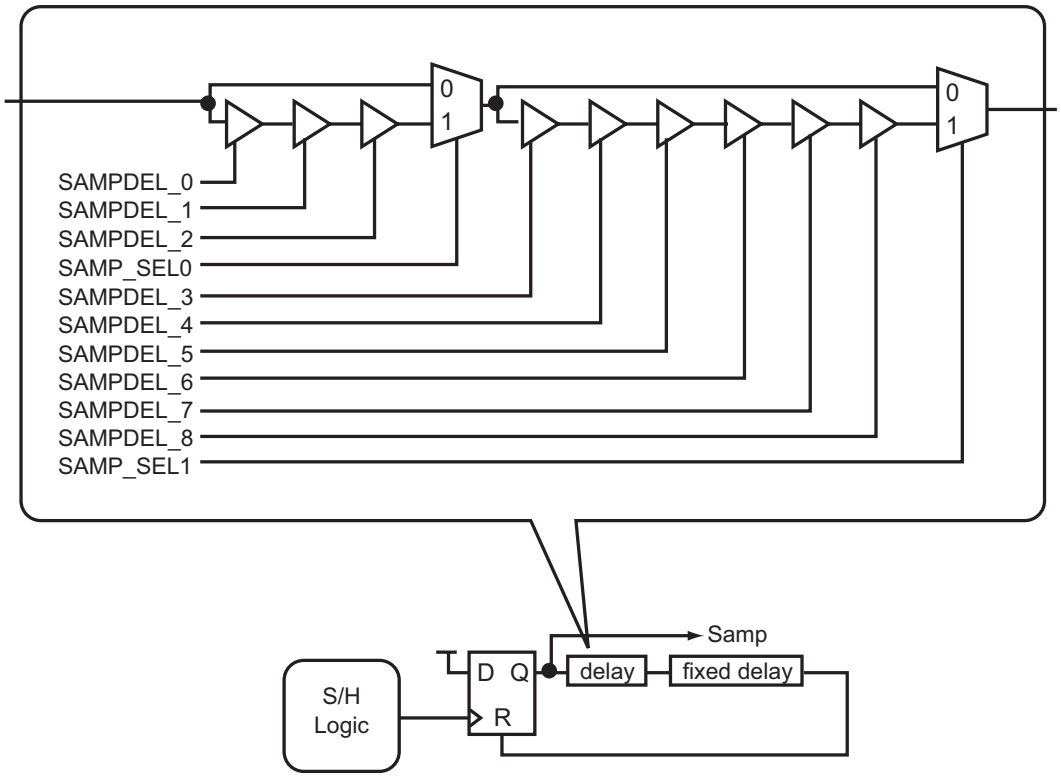


Figure B-6 Simplified S/H Pulse Generation Schematic

Bit	Name	Function	Default Value
31	PFDDEL_8	Delay Bit for PFD Reset	H
32	PFDDEL_8	Delay Bit for PFD Reset	H
33	PFDDEL_8	Delay Bit for PFD Reset	H
34	PFDDEL_8	Delay Bit for PFD Reset	H
35	PFDDEL_8	Delay Bit for PFD Reset	H
36	PFDDEL_8	Delay Bit for PFD Reset	H
37	PFDDEL_8	Delay Bit for PFD Reset	H
38	PFDDEL_8	Delay Bit for PFD Reset	H
39	PFDDEL_8	Delay Bit for PFD Reset	H
43	PFD_SEL0	PFD Delay Mux Select	L
44	PFD_SEL1	PFD Delay Mux Select	L
60	SEL_RES	PFD Reset Mode	L (L=Offset, H=Overlapping)

Table B.5 Configuration Register PFD Control Bits

B.6 PFD Configuration

The PFD reset pulse delay is set in much the same way as the S/H Samp pulse. The difference is that the entire delay for the PFD reset pulse is programmable, as depicted in Figure B-7.

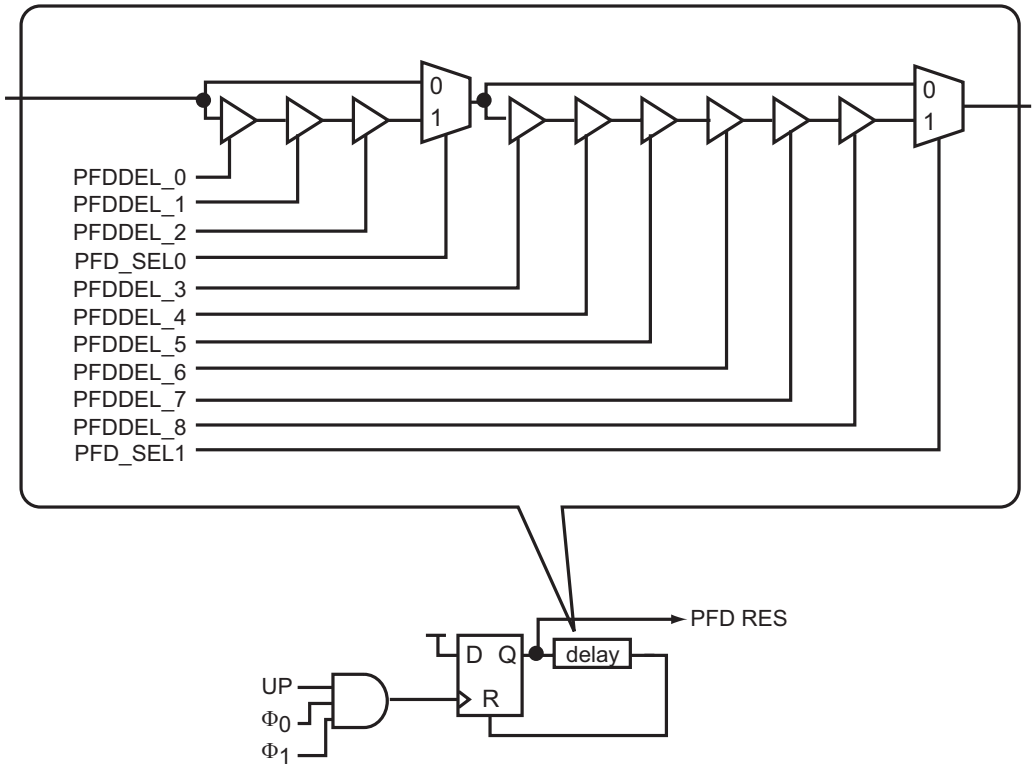


Figure B-7 Simplified PFD Logic Reset Pulse Generation

Bibliography

[1] M. Perrott, M. Trott, and C. Sodini, "A Modeling Approach for $\Sigma\Delta$ Fractional-N Frequency Synthesizers Allowing Straightforward Noise Analysis," *IEEE Journal of Solid-State Circuits*, vol. 37, pp. 1028–1038, Aug. 2002.

[2] B. Razavi-editor, *Monolithic Phase-Locked Loops and Clock Recovery Circuits*. IEEE Press, 1996.

[3] U. L. Rohde, *Digital PLL Frequency Synthesizers, Theory and Design*. Prentice-Hall, 1983.

[4] T. Riley, M. Copeland, and T. Kwasniewski, "Delta-Sigma Modulation in Fractional-N Frequency Synthesis," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 553–559, May 1993.

[5] B. Miller and R. Conley, "A Multiple Modulator Fractional Divider," *IEEE Transactions on Instrumentation and Measurement*, vol. 40, pp. 578–583, June 1991.

[6] I. Galton, "Delta-Sigma Data Conversion in Wireless Transceivers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, Jan. 2002.

[7] M. Perrott, T. Tewksbury, and C. Sodini, "A 27mW Fractional-N Synthesizer Using Digital Compensation for 2.5Mb/s GFSK Modulation," *IEEE Journal of Solid-State Circuits*, vol. 44, pp. 2048–2059, Dec. 1997.

[8] B. DeMuer and M. Steyaert, "A CMOS Monolithic $\Delta\Sigma$-Controlled Fractional-N Frequency Synthesizer for DCS-1800," *IEEE Journal of Solid-State Circuits*, vol. 37, pp. 835–844, July 2002.

[9] W. Rhee, B. Song, and A. Ali, "A 1.1GHz CMOS Fractional-N Frequency Synthesizer with 3-b Third Order $\Delta\Sigma$ Modulator," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 1453–1460, Oct. 2000.

[10] E. Hegazi and A. Abidi, "A 17mW Transmitter and Frequency Synthesizer for 900MHz GSM Fully Integrated in 0.35um CMOS," *2002 Symposium on VLSI Circuits Digest of Technical Papers*, pp. 234–237, June 2002.

[11] W. Rhee, B. Bisanti, and A. Ali, "An 18-mW 2.5GHz/900-MHz BiCMOS Dual Frequency Synthesizer With <10Hz RF Carrier Resolution," *IEEE Journal of Solid-State Circuits*, vol. 37, pp. 515–520, Apr. 2002.

[12] R. Magoon, et. al., "A Sing-Chip Quad-Band (850/900/1800/1900MHz) Direct Conversion GSM/GPRS RF Transceiver with Integrated VCOs and Fractional-N Synthesizer," *IEEE Journal of Solid-State Circuits*, vol. 37, pp. 1710–1720, Dec. 2002.

[13] N. Filiol, T. Riley, C. Plett, and M. Copeland, "An Agile ISM Band Frequency Synthesizer with Built-In GMSK Data Modulation," *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 998–1008, July 1998.

[14] H. Lee, et. al., "A Fully Integrated Fractional-N Frequency Synthesizer with a Wideband VCO using an AFC Technique and a 3-bit 4th Order $\Sigma-\Delta$ Modulator for GSM/GPRS/WCDMA Applications," *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 1164–1169, July 2004.

[15] C. Park, O. Kim, and B. Kim, "A 1.8GHz Self-Calibrated Phase-Locked Loop with Precise I/Q Matching," *IEEE Journal of Solid State Circuits*, vol. 36, pp. 777–783, May 2001.

[16] K. Lee, et. al., "A Single Chip 2.4GHz Direct-Conversion CMOS Receiver for Wireless Local Loop Using Multiphase Reduced Frequency Conversion Technique ," *IEEE Journal of Solid State Circuits*, vol. 36, pp. 800–809, May 2001.

[17] W. Rhee and A. Ali, "An On-Chip Compensation Technique in Fractional-N Frequency Synthesis," *Proceedings of the 1999 IEEE ISCAS*, vol. 3, pp. 363–366, July 1999.

[18] T. Riley and J. Kostamovaara, "A Hybrid $\Delta\Sigma$ Fractional-N Frequency Synthesizer," *IEEE Transactions on Circuits and Systems II*, vol. 50, pp. 176–180, Apr. 2003.

[19] S. Pamarti, L. Jansson, and I. Galton, "A Wideband 2.4GHz Delta-Sigma Fractional-N PLL With 1Mb/s In-Loop Modulation," *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 49–62, Jan. 2004.

[20] E. Temporiti, et. al, "A 700kHz Bandwidth $\Sigma\Delta$ Fractional Synthesizer With Spurs Compensation and Linearization Technique for WCDMA Applications," *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 1446–1454, Sept. 2004.

[21] Y. Dufour, *Method and Apparatus for Performing Fractional Division Charge Compensation in a Frequency Synthesizer.* US Patent No. 6,130,561, 2000.

[22] Y. Fan, "Modeling and Simulation of $\Sigma\Delta$ Frequency Synthesizers," *Proc. of the 2001 IEEE Symposium on Industrial Electronics*, pp. 684–689, June 2001.

[23] M. Perrott, "Fast and Accurate Behavioral Simulation of Fractional-N Frequency Synthesizers and Other PLL/DLL Circuits," *Proceedings of the IEEE 39th Annual Design Automation Conference*, pp. 498–503, 2002.

[24] X. Mao, H. Yang, and H. Wang, "Behavioral Modeling and Simulation of Jitter and Phase Noise in Fractional-N PLL Frequency Synthesizer," *Proc. of the 2004 IEEE Behavioral Modeling and Simulation Conference*, pp. 25–30, Oct. 2004.

[25] S. Meninger and M. Perrott, "A Fractional-N Frequency Synthesizer Architecture Utilizing a Mismatch Compensated PFD/DAC Structure for Reduced Quantization-Induced Phase Noise," *IEEE TCAS-II*, vol. 50, pp. 839–849, Nov. 2003.

[26] W. Egan, *Frequency Synthesis By Phase lock*. Wiley Press, 2000.

[27] C. Vaucher, et.al, "A Family of Low-power Truly Modular Programmable Dividers in Standard 0.35um CMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 1035–1045, July 2000.

[28] H. Chien, et. al, "A 4GHZ Fractional-N Synthesizer for IEEE 802.11a," *20004 Symposium on VLSI Circuits Digest of Technical Papers*, pp. 46–49, 2004.

[29] B. Razavi, "Challenges in portable RF Transceiver Design," *IEEE Circuits and Devices Magazine*, vol. 12, pp. 12–25, Sept. 1996.

[30] T. Lee and S. Wong, "CMOS RF Integrated Circuits at 5GHz and Beyond," *Proc. of the IEEE*, vol. 7, pp. 32–41, Sept. 2000.

[31] B. Razavi, "Design of CMOS Transceivers for Cellular Telephony," *IEEE Communications Magazine*, vol. 41, pp. 144–149, Aug. 2003.

[32] A. Abidi, "RF CMOS Comes of Age," *IEEE Journal of Solid State Circuits*, vol. 51, pp. 549–561, Dec. 2004.

[33] B. Razavi, *RF Microelectronics*. Prentice Hall, 1998.

[34] D. McMahill and C. Sodini, "A 2.5Mb/s GFSK 5.0Mb/s 4-FSK Automatically Calibrated $\Sigma\Delta$ Frequency Synthesizer," *IEEE Journal of Solid-State Circuits*, vol. 37, pp. 18–26, Jan. 2002.

[35] T. Lee, *Design of Radio Frequency Integrated Circuits*. Cambridge University Press, 1997.

[36] S. Norswothy, R. Schreier, and G. Temes, *Delta-Sigma Data Converters: Theory, Design, and Simulation*. IEEE Press, 1997.

[37] M. Perrott, "Techniques for High Data Rate Modulation and Low Power Operation for Fractional-N Frequency Synthesizers," *PhD Thesis, Massachusetts Institute of Technology*, 1997.

[38] C. Lau and M. Perrott, "Fractional-N Frequency Synthesizer Design at the Transfer Function Level Using a Direct Closed Loop Realization Algorithm," *Proceedings of the IEEE Design Automation Conference*, pp. 526–531, June 2003.

[39] M. Perrott, *PLL Design Using the PLL Design Assitant Program.* http://www-mtl.mit.edu/researchgroups/perrottgroup/tools.html.

[40] R. Baird and T. Fiez, "Linearity Enhancement of Multibit $\Delta\Sigma$ A/D and D/A Converters Using Data Weighted Averaging," *IEEE TCAS II*, vol. 42, pp. 753–762, Dec. 1995.

[41] R. Radke, A. Eshraghi, and T. Fiez, "A 14-bit Current Mode $\Sigma - Delta$ DAC Based Upon Rotated Data Weighted Averaging," *IEEE Journal of Solid State Circuits*, vol. 35, pp. 1074–1084, Aug. 2000.

[42] M. Perrott, *CppSim Reference Manual.* http://www-mtl.mit.edu/researchgroups/perrottgroup/tools.html.

[43] K. Murota and K. Hirade, "GMSK Modulation for Digital Mobile Radio Telephony," *IEEE Transactions on Communications*, vol. 29, pp. 1044–1050, July 1981.

[44] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design.* Addison-Wesley, 2000.

[45] W. Dally and J. Poulton, *Digital Systems Engineering.* Cambridge University Press, 1998.

[46] V. Gutnik and A. Chandrakasan, "On-chip Picosecond Time Measurement," *VLSI Circuits Digest of Technical Papers*, pp. 52–53, June 2000.

[47] S. Kiaei, S. Chee, and D. Allstot, "CMOS Source Coupled Logic for Mixed-mode VLSI," *Proc. of 1990 IEEE International Symposium on Circuits and Systems*, pp. 1608–1611, May 1990.

[48] B. Razavi, *Design of Integrated Circuits for Optical Communications*. McGraw-Hill, 2003.

[49] R. Mutagi, "Pseudo noise sequences for engineers," *Electronics & Communication Engineering Journal*, pp. 79–87, Apr. 1996.

[50] A. Van den Bosch, M. Borremans, M. Steyaert, and W. Sansen, "A 10-bit 1-GSample/s Nyquist Current-Steering CMOS D/A Converter," *IEEE Journal of Solid State Circuits*, vol. 36, pp. 315–324, Mar. 2001.

[51] T. Riley, N. M. Filiol, D. Qinghong, and J. Kostamovaara, "Techniques for In-band Noise Reduction in $\Delta\Sigma$ Synthesizers," *IEEE Transactions on Circuits and Systems II*, vol. 50, pp. 794–803, Nov. 2003.

[52] D. Johns and K. Martin, *Analog Integrated Circuit Design*. Wiley Press, 1997.

[53] P. Gray, P. Hurst, S. Lewis, and R. Meyer, *Analysis and Design of Analog Integrated Circuits*. Wiley Press, 2001.

[54] M. Cassia, P. Shah, and E. Brun, "Analytical Model and Behavioral Simulation Approach for a $\Sigma\Delta$ Fractional-N Synthesizer Employing a Sample-hold Element," *IEEE TCAS-II*, vol. 50, pp. 850–859, Nov. 2003.

[55] S. Lee, et. al., "A Quad-Band GSM-GPRS Transmitter With Digital Auto-Calibration," *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 2200–2214, Dec. 2004.

[56] J. Zhou and H. Chen, "A 1GHz 1.8V Monolithic CMOS PLL with Improved Locking," *Proc. of the 44th IEEE Midwest Symposium on Circuits and Systems*, pp. 458–461, Aug. 2001.

[57] A. Hajimiri and T. Lee, "A General Theory of Phase Noise in Electrical Oscillators," *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 179–194, Feb. 1998.

[58] A. Hajimiri and T. Lee, "Oscillator Phase Noise: A Tutorial," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 326–336, Mar. 2000.

[59] B. Razavi, "A Study of Phase Noise in CMOS Oscillators," *IEEE Journal of Solid-State Circuits*, vol. 31, pp. 331–343, Mar. 1996.

[60] J. McNeil, "Jitter in Ring Oscillators," *IEEE Journal of Solid-State Circuits*, vol. 32, pp. 870–879, June 1997.

[61] J. Yuan and C. Svensson, "High Speed CMOS Circuit Technique," *IEEE Journal of Solid-State Circuits*, vol. 24, pp. 62–70, Feb. 1989.

[62] H. Skinner and K. Slattery, "Why Spread Spectrum Clocking of Computing Devices is Not Cheating," *2001 IEEE International Symposium on Electromagnetic Compatibility*, pp. 537–540, Aug. 2001.

[63] R. Staszewski, C. Hung, D. Leipold, and P. Balsara, "A First Multi-Gigahertz Digitally Controlled Oscillator for Wireless Applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, pp. 2154–2164, Nov. 2003.