Techniques for Low Jitter Clock Multiplication

by

Belal Moheedin Helal

Bachelor of Science in Electrical Engineering
King Abdulaziz University, June 1995

Master of Science in Electrical Engineering and Computer Science
Massachusetts Institute of Technology, February 2002

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

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

Author . . . .
Department of Electrical Engineering and Computer Science
February 15, 2008

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

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

Techniques for Low Jitter Clock Multiplication

by

Belal Moheedin Helal

Submitted to the Department of Electrical Engineering and Computer Science
on February 15, 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Phase realigning clock multipliers, such as Multiplying Delay-Locked Loops (MDLL), offer significantly reduced random jitter compared to typical Phase-Locked Loops (PLL). This is achieved by introducing the reference signal directly into their voltage controlled oscillators (VCO) to realign the phase to the clean reference. However, the typical cost of this benefit is a significant increase in deterministic jitter due to path mismatch in the detector as well as analog nonidealities in the tuning circuits.

This thesis proposes a mostly-digital tuning technique that drastically reduces deterministic jitter in phase realigning clock multipliers. The proposed technique eliminates path mismatch by using a single-path digital detection method that leverages a scrambling time-to-digital converter (TDC) and correlated double sampling to infer the tuning error from the difference in cycle periods of the output. By using a digital loop filter that consists of a digital accumulator, the tuning technique avoids the analog nonidealities of typical tuning paths. The scrambling TDC is not a contribution of this thesis.

A highly-digital MDLL prototype that uses the proposed tuning technique consists of two custom 0.13 μm ICs, an FPGA board, a discrete digital-to-analog converter (DAC) with effective 8 bits, and a simple RC filter. The measured performance (for a 1.6 GHz output and 50 MHz reference) demonstrated an overall jitter of 0.93 ps rms, and estimated random and deterministic jitter of 0.68 ps rms and 0.76 ps peak-to-peak, respectively. The proposed MDLL architecture is especially suitable for digital ICs, since its highly-digital architecture is mostly compatible with digital design flows, which eases its porting between technologies.

[1] Post-publication note: this PDF version of the thesis was updated after degree-fulfillment submission. Comments and noteworthy edits will be mentioned in the footnotes.

Additionally, a Pulse Injection-Locked Oscillator (PILO) structure is proposed for use in applications that require more stringent phase noise and jitter requirements that can only be provided by LC oscillators. A PILO prototype that uses the proposed tuning technique consists of a custom 0.13 μm IC, an FPGA board, a discrete DAC with effective 10 bits, and a simple RC filter. The measured performance (for a 3.2 GHz output and 50 MHz reference) demonstrated estimated random and deterministic jitter of 270 fs rms and 310 fs peak-to-peak, respectively [1].

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

[1] Post-publication note: later results demonstrated 130 fs rms and 200 fs peak-to-peak, respectively, as reported in [Helal, *et al.*, JSSC, May 2009].

To those who carry the torch of enlightenment through the ages . . .

“Verily, with every difficulty there is relief”, Quran, 94:6.

Acknowledgments

MIT has been a truly wonderful and life-changing environment for me; not only academically (as expected) but also spiritually and romantically (strangely enough). All the way to and through MIT, God placed countless individuals whose inspiration, help, and love paved the way. In the following few pages, I will try to acknowledge some of those who shaped my experience at MIT, or who helped me get there. Those folks have my heartfelt gratitude and sincere thanks, but above all others, I am most thankful to God whose signs of mercy and love have never ceased to envelop me.

Professor Mike Perrott, my advisor, has not only been a great mentor, but, with the astute personal advice he has offered over the years, can be considered a life coach. I learned so much from his technical expertise, thinking style, and communication skills, and I was equally inspired by his kindness, fairness and humility.

My thesis committee, Professor Franz Kaertner and Professor Vladimir Stojanovic, offered many comments and insights that deepened my understanding of my research and its context. Professor Gu-Yeon Wei of Harvard University, who co-advised me at the beginning of this research, always had thought-provoking questions to ask, and shared many of his insights and design tips. I worked for a few months as part of his research group, which gave me a solid start, and allowed me to interact with his fine group of students; among them Amber Tan and Ruwan Ratnayake.

Professor Harry Lee, the chairman of area III, lent me his ear and was a constant support whenever I needed his guidance, especially while navigating the old quals process. His assistant, Carolyn Collins, was very supportive and always cheered me on. She sent many prayers that undoubtedly helped me along the way.

Marilyn Pierce, the former EECS graduate administrator, never had anything but kind and encouraging words to offer. Marilyn's guidance and support were indispensable, especially through the few rough spots in the road to the PhD. Janet Fischer, the EECS graduate administrator during the last few months of my stay at MIT, was tremendously helpful in finishing the last mile of this long trip.

The former and current EECS education officers: Professors Eric Grimson and

George Vergese, who gave me invaluable opportunities to teach MIT undergraduate students through TAing 6.002 and 6.003. Lisa Bella, the education officer assistant, was always a delight to drop by her office, whether for a question, a favor (for which I was a repeat recipient) or just for a quick hello. Patsy Sampson, helped me reserve EECS conference rooms on many occasions, and would always offer encouraging words that helped calm my nerves during quals and doctoral committee meetings.

The lecturers and recitation instructors, who I was quite honored to be their colleague as a TA, Professors Jeff Lang, Alan Willsky, Gerry Sussman, Victor Zue, Denny Freeman, and Paul Gray. My fellow TAs with whom I had a lot of fun working and who helped make it quite a learning experience, including Taka Takahashi, Syed Alam, Siddhartan Govindasamy, Amy Englehart, James Sun, and Kevin Pipe. And my students who kept me on my toes and helped me become a better teacher.

Valerie DiNardo, our group administrator and dear friend, was helpful with all administrative issues (especially reimbursements). Debroah Hodges-Pabn, the MTL dynamo, who is my inspiration in getting things done with a hand of steel and heart of gold. Debb invited us grad students to MTL events and made us feel that it would not be the same without our attendance (needless to say, she got us at ”free food”).

My HSCS labmates, who taught me so much and enriched my tooling years with their company. Kerwin Johnson saved many days during our group tapeout and testing periods, and was instrumental in getting my last tapeout out the door. We shared many owl hours, technical and not-so-technical discussions, and trips to Costco. Kerwin also proofread large parts of this thesis. Chun-Ming Hsu practically taped out a test chip on my behalf, and spent many hours during my last tapeout helping me make the deadline. Matt Straayer developed the GRO that was a cornerstone block in the MDLL and PILO systems that constitute this thesis. Matt Park brought many smiles to our group that made the long tapeout hours much more enjoyable, and taught me a lot while using me as a sounding board. Min Park shared many discussions both technical and personal, and helped me with my last tapeout. Charlotte Lau was very helpful in getting me started when I joined the group and I benefited greatly from our interactions, both technical and personal. Scott Meninger, the group’s first PhD, was very generous with his knowledge and time, and I was probably greedy with both; especially whenever I needed a procrastination venue to sweeten the long nights at the lab. Ethan Crain picked up my ”occasional” slack TAing 6.976 while learning the ropes of newborn care, and we bonded over the shared experience of grad school parenting. Muyiwa Ogunnika and Shawn Kuo were quite pleasant company and their presentations during group meetings always taught me something.

Colleagues who shared our sunny office space of 38-246, including Oh Rayanakorn, Tania Khanna, Jose Bohorquez, Willie Sanchez, and Jack Holloway. Colleagues from other circuit groups, including Keng-Hoong Wee, Byungsub Kim, Ranko Sredojevic, Farinaz Edalat, Lunal Khuon, Fred Lee, Alex Kern, Denis Daly, and David Wentzloff.

Professor Kamal Youcef-Toumi, my Masters advisor, and my MRL labmates: Osamah and Khalid El-Rifai, Ahmed Elmouelhi, Eric Hoarau, Namik Yilmaz, Pablo Valdivia y Alvarado, Bernardo Aumond, and Pradya Prempraneerach.

Fellow 6.376ers with whom I spent many tooling hours working on biologically-inspired circuits, especially Syed Alam, Ming Tang, Ghinwa Choueiter, and our great TA and dear friend Micah O’Halloran (who received the GSC graduate teaching award for TAing this class; his first time at MIT).

Professor Jane Dunphy, my writing and communication professor and dear friend: I promise I will try to slow down ... my speech that is. Dean Blanche Staton whose support at several occasions made my life at MIT much easier. Anne Hunter, who organized the IAP Tai Chi sessions that introduced me to this wonderful art, and Dorri Li, the instructor who taught me how to enjoy the relaxing movements without obsessing about the correctness of the form. Zan Barry, who led the invigorating Yoga sessions I attended while writing my thesis.

The Living an Extraordinary Life crew, Lauren Zander, Gaby Jordan, and Aditi Thatte, and fellow LELers, Anthony, Samantha, Zahra, Qudsia, Catherine and Taro. This class was true to its name, and its lessons and inspirations will last me a lifetime.

The MBC gang: Babak Ayazifar, Kashif Khan, Mohammad Saeed, Ayman Shabra, Jalal Khan, Ihsan Djomehri, Ammar Al-Nahwi and Farhan Rana, whose vibrant discussions forced me to reevaluate my axioms and changed me in ways I never thought possible. MSA friends: Suheil, Numan, Basel, Aida, Humera, Tarik, Fadilah, Nazbeh, Sarah, Tania, Minhaj, Akram, Omar, Nadeem, Basier, Afreen and Hamid, Ali and Soraya, Tarek and Rokaya, and Husain and Zaynab, who were my family in Boston.

The many KAU professors whose strong foundation brought me to MIT, especially Professor Mahmoud Wanas, my Bachelors thesis advisor (an MIT alum himself). And Ghazi Al-Rawi, my KAU classmate, who helped keep the bar high. Dr. Mohamed Lazzouni, my role model for combining a brilliant technical career with inspiring spiritual explorations. Mohammed Al-Mudarris, my colleague at Al-Afandi, who helped me navigate through the grad school application process.

My father and mother in-law, who were my role models for raising a family while in grad school, but more importantly who entrusted me with their sweet daughter, Sarah, despite the fact that we were both students at MIT.

My parents, whom being their son was a great blessing. You were great examples of wisdom, patience, love, compassion, and spirituality. It is with the love of knowledge that you instilled in me that I am here today. I hope you will always be proud of me, and that I will be able to follow your example with my own kids.

My siblings Somayah, Usamah, Zilal and Ahmad, with whom, being their middle, I experienced the whole spectrum of siblinghood, including many laughs. You guys rock! We have been through a lot together, formed an amazing bond in the process, and I am truly proud to be your brother.

My children Yousef, Mariam and Hasan, who always brighten my day with their sweet smiles, soften my heart with unconditional love, and bring out the child in me with their innocence. You make me aspire to be the role model that you deserve.

And finally, to Sarah Ibrahim: my wife, the love of my life, and my partner for the last six years and for many years to come. You are the best thing that happened to me at MIT, my source of peace and tranquility, my cheerleader, and my Editor-in-Chief. This PhD would not have been if it were not for you; in fact you would be the first author on this thesis if that title was determined by the amount of effort put in. I can not wait until it is your turn to submit your thesis, and I hope I will be as much help to you as you were to me (although you set the bar pretty high).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As technology has advanced, on-chip clock multiplication has become a necessity for nearly all digital integrated circuits (ICs) in order to realize high speed clock signals from lower speed external sources such as crystal oscillators. The typical approach to achieve such clock multiplication is to employ a phase locked loop (PLL) circuit consisting of a phase detector, analog loop filter, frequency divider, and voltage-controlled oscillator (VCO). Unfortunately, the analog component of PLLs prevents their design from simple compatibility with a typical digital design flow.

Multiplying Delay-Locked Loops (MDLL) [1, 2] have been introduced recently as an alternative to PLLs for clock multiplication. Although similar in concept to [3], MDLLs offer much better jitter performance for high frequency clocks. For integrated applications, ring oscillators are preferred over LC oscillators due to their smaller areas. The disadvantage of ring oscillators, however, is their higher phase noise compared to LC oscillators with the same power consumption. MDLLs suppress the VCO phase noise more effectively than PLLs by periodically replacing the rotating edge with a clean reference edge. The effect of this operation is the suppression of the VCO phase noise at a bandwidth more than double that which is typically possible in PLLs. The cost of this advantage is that the reference must be clean, in addition to deterministic jitter that would occur if the reference edge does not replace the rotating edge of the ring oscillator at exactly the right instant. Perfect tuning of the VCO to reduce deterministic jitter is a challenging task due to path mismatch in the phase detector and other analog nonidealities in the tuning path. Minimization of deterministic jitter in MDLLs is an area of active research [4, 5] and is the primary objective of this thesis.

For LC oscillators, injection-locking is the analogous technique used for more effective suppression of VCO phase noise. Subharmonic injection-locking oscillators are used as clock multipliers [6]. However, continuous tuning of an injection-locked oscillator is challenging because the average frequency of an injection-locked oscillator is already at a multiple of the reference frequency. Continuous tuning is important to achieve a sufficient locking bandwidth to track thermal variations that would otherwise require a large injection power to achieve . Injection-Locked PLLs (ILPLL) take advantage of the relative phase shift between the locked oscillator and the reference when the oscillator is not perfectly tuned. However, the path mismatch between the injection and the PLL paths causes deterministic jitter similar to that discussed in MDLLs. The same problem is illustrated in realigned PLLs [2], which are similar to ILPLL except that ring oscillators are used. The secondary objective of this thesis is to devise an injection-locking structure that is amenable to continuous tuning with minimum deterministic jitter.

All mentioned objectives of this thesis have been achieved. A digital period correlation technique is proposed to detect tuning error without path mismatch. A highly-digital tuning technique that uses the proposed detection technique and eliminates analog nonidealities of the typical analog tuning paths is proposed. A MDLL prototype that incorporates the proposed tuning technique demonstrates sub-picosecond overall jitter. A Pulse Injection-Locked Oscillator (PILO) structure that is amenable to continuous tuning by the proposed technique is proposed and a prototype demonstrates estimated random and deterministic jitter of about 300 fs (rms, and peak-to-peak, respectively).

1.1 Area of focus

The focus of this thesis is on clock multiplication architectures that introduce the reference signal directly into the VCO to suppress its phase noise and generate high frequency clocks with low jitter, but which are prone to deterministic jitter. The first architecture is the Multiplying Delay-Locked Loop (MDLL), which periodically replaces the rotating edge of its ring oscillator with the clean reference edge. The second architecture is introduced in this thesis and is based on subharmonic injection locking, in which the reference signal is injected directly into the VCO.

Figure 1-1 shows a block diagram of the general architecture that is the focus of this thesis. The reference signal, *Ref* is introduced into the VCO, and is also used by the detector (explicitly or implicitly) to measure the tuning error, Δ . When not perfectly tuned, the output of the VCO, *Out*, exhibits inconsistency in the period of its cycles, resulting in deterministic jitter with a level of Δ (peak-to-peak). The tuning path detects the error, Δ , and integrates it with the loop filter to generate the VCO tuning voltage, V_{tune} .

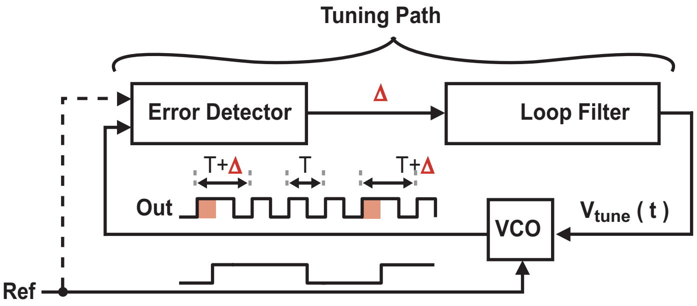


Figure 1-1: General Architecture of Focus.

Ideally, the detector and loop filter do not suffer from any mismatch or offset issues. But as we will discuss, those are in fact problems with classical architectures, which make them unable to sufficiently remove the tuning error, Δ . The primary objective of this thesis is to propose detection and tuning techniques that minimize Δ , and its deterministic jitter manifestation, to a level comparable or below random jitter. In addition, since digital ICs are the intended application for the MDLL architecture, a mostly digital tuning path is another objective of the thesis.

1.2 Thesis Organization

Before we present the detection and tuning techniques that minimize deterministic jitter, it is important to understand the context of the problem and its causes, after which the the proposed technique is presented. The thesis then presents a MDLL prototype as the first application that uses the proposed tuning technique. Then, a proposed Pulse Injection-Locked Oscillator (PILO) structure is presented, and its prototype is discussed, after which measurement results for the two prototypes are presented. Finally, the thesis ends with a summary of the thesis contributions and future research. Expanded details of the thesis organization are provided below.

Chapter 2 discusses the MDLL architecture and its benefits, provides background information on previous approaches to MDLL tuning, and highlights their sensitivity to analog nonidealities. In addition, it discusses injection locking and the need for, and the challenge of, continuous tuning of an injection-locked oscillator.

Chapter 3 presents the proposed detection technique and overviews the requirements and important implementation details of its constituent blocks, including the Enable Logic and the time-to-digital converter (TDC), and the digital correlator. Afterwards, the chapter discusses the modeling of general architectures that use the proposed tuning technique to calculate their bandwidth, phase noise and jitter performance.

Chapter 4 discusses the MDLL prototype that incorporates the proposed tuning technique. The chapter starts with an overview of the prototype architecture and its test board. Next, it discusses details of key prototype blocks, including the MDLL core, gated ring oscillator (GRO) TDC, FPGA digital functions, digital-to-analog converter (DAC) and RC lowpass filter. Expected jitter performance is calculated using the noise model presented in Chapter 3 and parameters of various components of the prototype. The chapter ends with the discussion of some important implementation issues, including the effect of power supply noise.

Chapter 5 starts with the motivation for PILOs, and then presents its concept, linearized analysis of its operation, and two pulse injection methods. The chapter ends with the effect of oscillator tuning to motivate the use of the proposed tuning technique to minimize deterministic jitter

Chapter 6 presents an architectural overview of the PILO prototype and discusses its blocks that are functionally different from the MDLL prototype, including the injected VCO, and the injected pulse generator. The chapter then discusses blocks that are similar to the MDLL prototype in functionality, but not in implementation, including the Enable Logic, GRO TDC, and the correlator that includes an accumulate-and-dump block to decimate the correlator output. Expected jitter performance is calculated using the noise model presented in Chapter 3 and parameters of various components of the prototype. Finally, some implementation issues related to loop locking and noise coupling are also discussed.

Chapter 7 starts with discussing a method to estimate deterministic jitter, whose minimization is the yardstick by which the success of the proposed tuning technique is measured. The measured results of the MDLL prototype are presented first, including overall jitter and estimated random and deterministic jitter. The jitter performance is compared with previous MDLL architectures to illustrate the effectiveness of the proposed tuning technique. Finally, the chapter presents the measured results of the PILO prototype, including the estimated random and deterministic jitter.

Chapter 8 starts with a summary of thesis and then lists thesis contributions. Finally, the chapter and the thesis end with the discussion of suggestions for future research, including an optical PILO.

Chapter 2

Background

With the ever-increasing speed of microprocessors, it has become harder to generate the required clocks with a low-enough jitter. Jitter specifications affect the net speed of a microprocessor because jitter causes timing uncertainty. This uncertainty necessitates more conservative timing margins to satisfy the timing requirements of different blocks. A noisy clock is even more troublesome if it is used as a reference source for a frequency synthesizer. The resulting phase noise will reduce the signal-to-noise ratio of the received signal and might interfere with adjacent channels.

The classic method of generating high frequency clocks is a phase-locked loop (PLL). A PLL generates a high frequency clock from an off-chip low-jitter reference clock that can be generated easily and inexpensively, but at a relatively low frequency (100 MHz or lower). By comparing the reference clock with a frequency-divided version of the noisier high-frequency output of the voltage controlled oscillator (VCO), the PLL is able to generate the desired high frequency clock. The phase noise and jitter of the generated output will depend on various factors, including the phase noise of the VCO, the noise of the detector, and the loop bandwidth [7, 8]. However, there is a fundamental bandwidth tradeoff in PLLs that prevents the loop from optimally suppressing both the VCO and detector noise. This tradeoff is especially an issue for ring oscillator topologies that typically have larger phase noise than LC oscillators. This motivates the search for other clock multiplication architectures that do not exhibit such limitation, and that can offer better phase noise and jitter performance than PLLs of comparable power consumption and area.

This chapter presents an overview of two alternatives to PLLs for clock multiplication. The first is Multiplying Delay-Locked Loops (MDLL) and the second is subharmonic injection locking. Section 2.1 discusses the MDLL architecture, including its concept, benefits, classical architecture and the challenge of deterministic jitter, while Section 2.2 discusses subharmonic injection locking and the associated challenge of continuous tuning of an injection-locked oscillator.

2.1 Multiplying Delay-Locked Loop(MDLL)

Ring oscillators are typically used in clock multipliers of digital chips due to their smaller area, easier design, and larger frequency range compared to LC oscillators. However, ring oscillators typically exhibit much larger phase noise than their LC counterparts (for the same power consumption). This tradeoff provided an additional motivation to search for techniques to reduce phase noise in ring oscillators and MDLLs offered an attractive alternative.

2.1.1 Concept

As seen in Figure 2-1, the MDLL operates by replacing every N^{th} edge of a naturally running ring oscillator VCO with a reference frequency edge, where N corresponds to the frequency multiplication factor. This allows significant suppression of jitter caused by phase noise of the VCO [2]. Specifically, the edge multiplexing action of the MDLL rejects the VCO phase noise in a fashion similar to a type 1-order 1 PLL with a bandwidth of about one fourth the reference frequency. With the VCO phase noise suppressed at such a high bandwidth, the bandwidth of the tuning loop can now be lowered to better suppress the noise originating from the detector-referred sources. However, as shown in the figure, the price paid for such an advantage is that an incorrect setting of the V_{tune} voltage on the VCO (which tunes its corresponding frequency) leads to undesired ”deterministic jitter” due to corresponding periodic changes in the output period. Reduction of such deterministic jitter is the purpose of the tuning loop. The analog tuning approach used in classical MDLL architecture is presented in the next subsection.

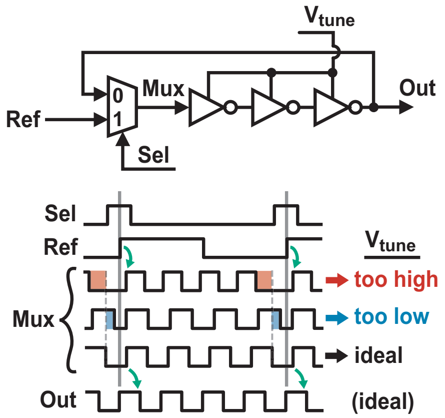


Figure 2-1: Conceptual MDLL clock multiplier and impact of tuning voltage on its associated signals.

2.1.2 Classical Architecture and the Challenge of Deterministic Jitter

Figure 2-2 shows the classical feedback approach used for adjustment of V_{tune} . The key idea of this approach is to use a phase detector to measure the difference in time, Δ , between two appropriate edges in the system, and then use a charge pump and loop filter to integrate the resulting error signal to form V_{tune} . Ideally, V_{tune} will then be adjusted by the feedback loop until Δ goes to zero, which would lead to zero deterministic jitter under steady-state conditions.

Practical circuit implementations for the traditional MDLL tuning approach are sensitive to nonidealities that cause the V_{tune} feedback loop to settle to a non-zero value of Δ , such that a substantial amount of deterministic jitter is introduced into the MDLL output [1, 2, 5, 4]. The major nonidealities are path mismatch in the multiplexer and phase detector, mismatch between the currents of the charge pump, and finite DC output impedance of the charge pump output. While several techniques have been recently proposed to reduce the impact of these nonidealities [5, 4], the relatively high analog design effort required by these approaches makes

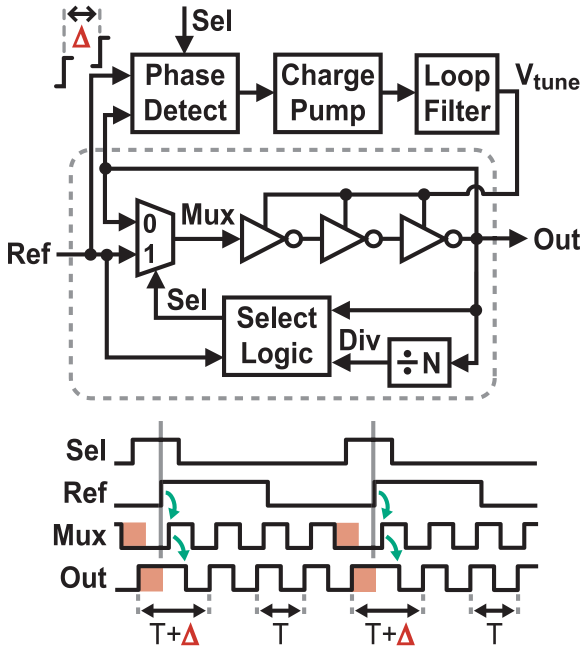


Figure 2-2: Classical approach to MDLL tuning.

them less amenable to inclusion within standard digital design flows. Therefore, it is attractive to develop an MDLL tuning architecture that is insensitive to such analog nonidealities and which requires minimal custom analog design effort for its implementation.

2.1.3 Literature Review

The MDLL concept of using the clean edge of the reference signal to purge out the accumulating jitter in a ring oscillator [9, 10] was first introduced by [3] in 1994. However, it was not until 2002 that an architecture suitable for high speed clock multiplication was introduced at ISSCC 2002 [11]. Realigned PLL (RPLL) is a similar concept that was introduced at the same conference [12]. Instead of multiplexing the reference edge, a RPLL injects the reference edge directly into a ring oscillator to realign its phase. The author also presented a model for the general class of clock multipliers that inject or multiplex a reference edge into an oscillator, and showed their benefit in suppressing the oscillator phase noise at a high bandwidth [12]. As the MDLL architecture started to gain wider acceptance, the issue of its deterministic jitter challenge started to be addressed. An architecture that can run as either PLL or MDLL was presented in [13] and showed measurements demonstrating that an MDLL achieves lower jitter only when the jitter of the reference source is relatively low. The same work used a secondary digital loop to minimize the effect of the charge pump current mismatches.

Two papers were presented at CICC 2006 [14, 15] which approached the reduction of deterministic jitter using different analog techniques. In [5], a low-bandwidth auxiliary loop was used to reduce deterministic jitter by comparing the N^{th} period with the average period of the MDLL output. In [4], each source of deterministic jitter was elaborately dealt with using various circuit techniques, including a sampling phase detector with chopper and an auto-zeroed operational transconductance amplifier.

2.2 Subharmonic Injection Locking

2.2.1 Overview of Injection locking

Injection locking of oscillators is a phenomenon that has been observed since the eighteenth century [16], but was not rigorously studied until the 1920s by Van Der Pol [17] and others. In the 1940's, Adler proved a relationship, known by his name, that relates the injected signal properties to the behavior of the injected oscillator [18]. The study of injection locking picked up steam in the 1970's, especially with the invention of lasers. During that decade Kurokawa showed a different approach to the injection relationship [19]. Recently, Razavi reviewed injection pulling and locking and provided a number of insights into the oscillator injection phenomenon [20]. Noise properties of injection-locked oscillators were investigated in a number of papers, for example [21].

In the last decade, there has been a significant increase in publications related to injection locking, mainly in the optics and microwave fields. In addition, there has been an increasing interest in injection-locking for applications in CMOS processes.

The CMOS application that saw the greatest number of publications and generated the most interest in the solid-state community is injection-locked frequency dividers (ILFD). The main motivation towards that interest is that the amount of power and area that an injection-locked frequency divider consumes is significantly less than conventional dividers running in the multi-Gigahertz region [22, 23]. Other applications of interest included quadrature oscillators [24], subharmonic injection locked oscillators [6], clock and data recovery circuits [25], and injection-locked PLLs [26, 27].

2.2.2 The Challenge of Tuning while Injection-Locked

In most cases of injection-locked oscillators, the oscillator is not directly tuned to the input signal. If the the tuning voltage is not kept constant, it is possibly either adjusted according to a replica VCO in a PLL [25], or initially tuned to bring the natural frequency of the oscillator to a value close to the injection frequency, before injecting the signal into the oscillator with enough power to insure locking [6]. The lack of tuning might pose a problem in a practical context, especially for sub-harmonic injection locked oscillators such as [6], where the oscillator is locked to a harmonic of the input signal whose power level might not be sufficient for a locking bandwidth that can track thermal variations.

Injection-Locked Phase-Locked Loop (ILPLL) is an architecture that can provide continuous tuning [27]. However, ILPLLs are prone to increased spurs due to the mismatch between the injection and the PLL paths, similar to realigned PLLs described in [2]. A major aim of this thesis is to address this issue by presenting a structure that can be continuously tuned without path mismatch issues.

Chapter 3

Proposed Detection and Tuning Techniques

This chapter introduces the proposed detection and tuning techniques that use a single-path detection, a scrambling time-to-digital converter (TDC), a digital correlator, and a highly-digital tuning path to drastically reduce deterministic jitter in MDLLs and similar phase-realigning clock multipliers. The single-path detection eliminates path mismatch, while the scrambling TDC enables the use of a digital correlator and a digital loop filter that eliminates offsets and mismatches found in analog approaches. The scrambling TDC used in the prototypes that demonstrated the proposed techniques was developed by Matt Straayer, who is a colleague in the same research group as the author.

Section 3.1 reviews the motivation for the proposed technique and possible applications where it can be used. Section 3.2 presents the proposed technique. Section 3.3 overviews the requirements for the Enable Logic. Section 3.4 discusses the TDC requirements and the disadvantages of classical TDCs, leading to an overview of the Gated Ring Oscillator (GRO) TDC, which was used in the prototypes. Section 3.5 reviews the various correlator implementations. Finally, Section 3.6 discusses the modeling of architectures that uses the proposed technique.

3.1 Motivation

The goal of the proposed detection technique is to solve the deterministic jitter problem discussed in Chapter 2, which is mainly caused by path mismatch and analog non-idealities. Figure 3-1 summarizes the two features desired in the proposed technique. First, path mismatch is eliminated by using only one signal to detect the tuning error. Second, analog non-idealities are eliminated by devising a detector that outputs the error in a digital format which can then drive a digital accumulator and loop filter.

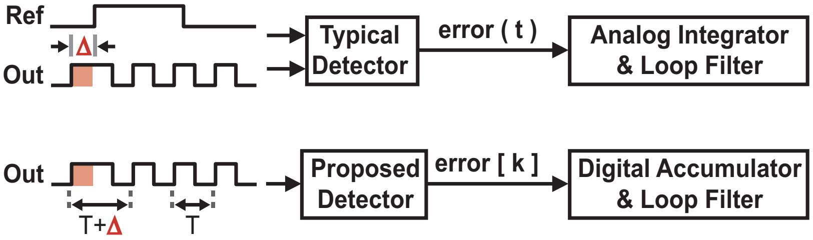


Figure 3-1: Error Detectors (a) Typical (b) Desired.

The desired detector shown in Figure 3-1 detects the type of error that is manifested as periodic inconsistency in the output period as illustrated in the *Out* waveform. This type of error is found in typical MDLLs, as discussed in Chapter 2, as well as PILOs (which will be presented in Chapter 5). For the proposed technique to be successful, the level of the tuning error, manifested as deterministic jitter, needs be reduced to about the level of random jitter, ideally in the sub-picosecond range.

3.2 Concept of the Proposed Techniques

The proposed period error detector, shown in simplified form in Figure 3-2, dramatically reduces the impact of path mismatch by avoiding the comparison of two different edge signals as in classical detectors. Instead, one signal is examined, *Enable*, whose pulse width alternates twice every reference cycle between the free running period of the oscillator, T, and the period of the error-affected cycle, T + Δ . By performing a relative comparison of each consecutive pulse period of the *Enable* signal, the value of Δ can be obtained in a manner such that the issue of mismatch is greatly mitigated since only one signal is being examined. This technique is referred to as correlated double-sampling when used in analog circuits, and is commonly applied in applications such as imagers and switched capacitor circuits to reduce DC-offset and 1/f noise [28, 29].

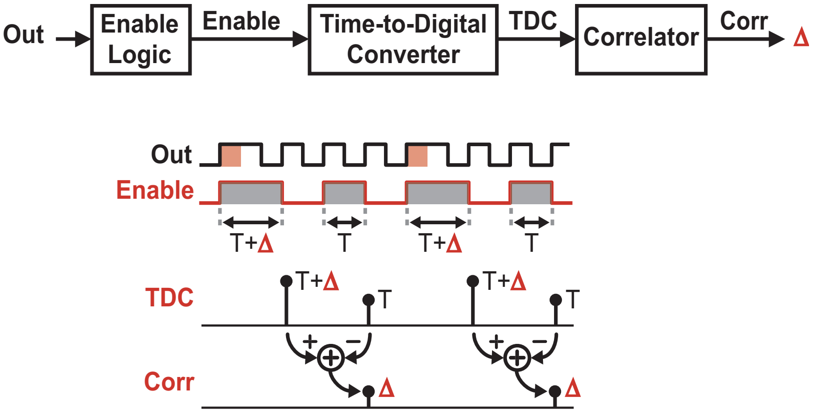


Figure 3-2: Proposed Error Detector.

While classical analog applications of the correlated double-sampling technique leverage switches, sampling capacitors, and operational amplifiers in their implementation, the goal for the proposed application examined here is to avoid such analog blocks and instead seek a highly digital implementation. As shown in Figure 3-2, this goal can be realized by leveraging a time-to-digital converter (TDC) structure to measure the periods in the *Enable* signal. The requirement of the TDC is discussed in more detail in the following section. As shown in Figure 3-2, this block outputs a digital signal, TDC, which is updated at the end of each *Enable* pulse and corresponds to a quantized measurement of the corresponding *Enable* pulse period (i.e., T or T + Δ). A digital correlator circuit simply subtracts consecutive pairs of the TDC samples to yield a stream of samples, *Corr*, which correspond to quantized estimates of Δ . By passing these Δ samples into a digital accumulator, as shown in Figure 3-3, the quantization error of the Δ samples is reduced by the averaging effect of the accumulation operation (assuming that the quantization error varies in an appropriately random fashion). Using a digital-to-analog converter (DAC) and a low pass filter (LPF), the accumulator output, D_{tune} , is converted to an analog voltage, V_{tune} , that adjusts the VCO until Δ reaches zero at steady-state.

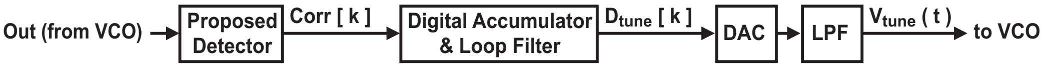


Figure 3-3: Proposed Tuning Path.

There are several advantages of the proposed tuning structure over previous approaches. First, the only analog blocks required by this tuning approach are a DAC and a simple lowpass filter, so that custom analog design effort for the proposed tuning approach is reduced compared to competing approaches given that an appropriate DAC structure is available to the designer. Second, the architecture is inherently insensitive to analog mismatch and offset due to the use of a correlated double sampling technique. Third, the digital accumulator structure has infinite DC gain, compared to the limited DC gain of analog integrators, so that the full range of V_{tune} can be achieved without any secondary impact on the steady-state value of Δ . In addition, the compact area of the digital accumulator allows easy integration of a low bandwidth tuning loop without concern for large capacitor area or degraded leakage characteristics. Finally, the highly digital architecture, which is insensitive to analog mismatch and offset, would greatly improve portability of the design between different CMOS technologies.

3.3 Enable Logic

For the single-path benefit of the proposed technique to hold, the Enable Logic must be able to select the desired output periods without adding any differential time offsets between the samples. Thus, it is important to devise an Enable Logic circuit that does not have different terminal states at different period samples. The details of two Enable Logic circuits that satisfy this requirement will be discussed in in Chapters 4 and 6.

3.4 Time-to-Digital Converter

The performance of the proposed technique and its desired residual tuning error will be limited by the time-to-digital converter (TDC) and its effective resolution. Depending on the TDC structure, the effective resolution can be smaller than the its raw resolution. The differentiating factor is whether the quantization noise of the TDC is sufficiently random, in which case the effective resolution can be improved by averaging [30].

3.4.1 Enhancing the TDC Effective Resolution

The improvement in the TDC resolution by averaging is proportional to the square root of the number of averaged samples, N. Assuming the TDC quantization noise is white, its effective resolution can be determined using the relationship shown below.

$$T_{tdc,eff} = T_{tdc,raw}\sqrt{\frac{1}{N}} \approx T_{tdc,raw}\sqrt{\frac{1}{OSR}} = T_{tdc,raw}\sqrt{\frac{2BW}{F_{tdc}}}. \tag{3.1}$$

, where $T_{tdc,eff}$ is the TDC effective resolution, $T_{tdc,raw}$ is the TDC raw resolution, OSR is the oversampling ratio (defined as the ratio of the sampling frequency to the Nyquist rate), F_{tdc} is the the sampling frequency, and BW is the loop bandwidth.

For example, a scrambling TDC (with a white noise profile) with a sampling rate of 100 MHz and a raw resolution of 50 ps, would have an effective resolution of about 0.71 ps for a 10 kHz bandwidth. This example shows the importance of choosing a TDC that has random quantization noise. TDCs with shaped quantization noise benefit more from averaging, though this is not the case when correlation is used, as will be explained in Subsections 3.4.3 and 3.5.1.

3.4.2 Classic TDC Structure

Figure 3-4 shows the classical delay chain TDC structure, which uses delay cells and synchronized registers to detect the time difference between two input edges [31]. The quantization error for this approach is set by the delay cell, which typically corresponds to an inverter delay. Since inverter delays are currently greater than 10 ps in modern CMOS processes [31], the classical structure falls over an order of magnitude short in meeting the desired sub-picosecond resolution that we seek. Unfortunately, the classical TDC always yields the same quantization error for a fixed time difference between the input edges, so averaging is not helpful for increasing the effective resolution. In practice, the time differences measured by the TDC will vary according to cycle-to-cycle jitter in the output, but the goal of achieving sub-picosecond jitter performance eliminates the possibility of this jitter being an adequate dithering source for achieving higher effective TDC resolution through averaging.

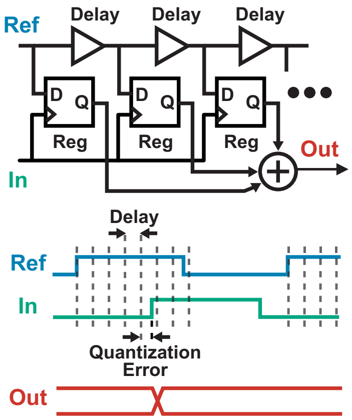


Figure 3-4: Classical time-to-digital structure and associated signals.

3.4.3 GRO TDC Structure

A TDC structure that has the desired property of scrambled quantization noise was used in the clock multiplying prototypes based on the proposed detection and tuning technique. The TDC is based on a Gated Ring Oscillator (GRO) structure [32, 33], and important details about its concept and implementation will be given in Chapters 4, 6. In addition to scrambling, the GRO TDC also first-order noise shapes its quantization noise. This property is very useful for other applications such as PLLs and DLLs, however, the proposed detector does not take advantage of the noise shaping due to the frequency translation effect of the correlator, which will be explained in subsection 3.5.1.

3.5 Digital Correlator

A number of approaches can be used to implement the digital correlator used in the proposed technique, either as a stand alone block or combined with other blocks. Before exploring some of those various implementations, a frequency-domain prospective of the correlation function will be discussed to clarify the correlation operation. The spectrum prospective will aid in appreciating the value of different implementations.

3.5.1 Correlation in the Frequency-Domain

As shown in Figure 3-2, the TDC output (ignoring noise), $TDC[n]$, is an alternating sequence of T and T+Δ samples. In the frequency domain, the TDC output has a DC component of T+Δ /2 and a high frequency component at $F_{tdc}/2$ with the value of Δ /2, as shown in the $TDC(f)$ plot of Figure 3-5 (ignoring scaling).

The useful information in the TDC output is the error, Δ , which resides at $F_{tdc}/2$ (uncorrupted by DC offset, in the form of the period T). The essential function of the correlator is to extract the Δ from high frequency and shift it to DC so it can properly drive the accumulator and lock the loop. The frequency translation is achieved by multiplying the TDC output, $TDC[n]$, by a sequence of alternating ones and minus ones, i.e. [..., -1, 1, -1, 1, ...], as shown in Figure 3-5.

Once the error information is shifted to DC (on average), what remains of the correlation operation is eliminating the T+Δ /2 component at $F_{tdc}/2$ (which originally was at DC) since, otherwise, it would remain as a spurious tone. Various correlator implementations have different methods of eliminating that high frequency component, some of which are discussed in the next subsections.

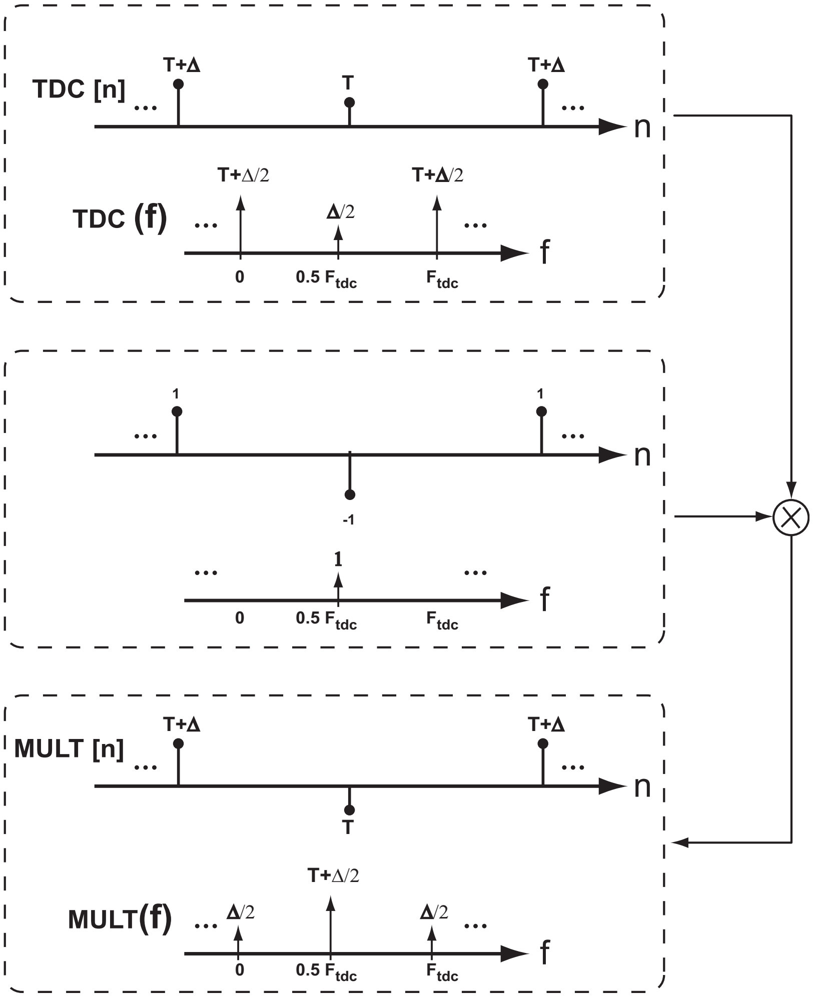


Figure 3-5: Multiplication step of the Correlator.

3.5.2 Using a Low-Pass Filter

The straightforward approach to filtering the tone at $F_{tdc}/2$ is to use a low-pass filter. A simple filter in the form of two-samples boxcar FIR would eliminate the tone, but the correlator output would be at the TDC sampling frequency, F_{tdc}. However, a two-sample accumulate-and-dump would achieve the same result, but with an output rate of $F_{tdc}/2$. The latter method is depicted in Figure 3-2. Higher downsampling ratios can be used to decimate the digital data to reduce the power consumption and complexity of subsequent digital blocks.

3.5.3 Using a High-Pass Filter

Instead of low-pass filtering after the multiplier, high-pass filtering before the multiplier would eliminate the low frequency component before it shifts to high frequency by the multiplying action. The simplest high-pass filter is a first difference block. Additionally, the first order difference can be combined with the multiplier by sequentially alternating the sign of the storage element in the first difference. In both cases the correlator output would be at the TDC sampling frequency, F_{tdc}.

3.5.4 Down-sampling after the Accumulator

In this approach no explicit filtering is used, but instead, the output of the accumulator is down sampled by a ratio of 2. The downsampling translates the high frequency content to DC again (by aliasing). However, since the downsampling is performed after the accumulator, the level of the originally high frequency content is significantly reduced and its effect is similar to a DC offset after an integrator, which is eliminated by the loop. This method provides a simple implementation that could relax the timing margins for the loop and avoid extra clock delays in the digital path.

3.6 Proposed Architecture Modeling

Figure 3-6 shows the general architecture that uses the purposed technique, e.g. MDLLs, with important specifications shown on top of each block. The specifications and the functions of the blocks will be discussed, after which the architecture modeling will be presented.

The heart of the architecture is the VCO whose phase is directly corrected by the reference either through edge multiplexing, as in MDLLs, or through injection locking, as in PILOs (discussed in Chapter 5). Note that there is no line from *Ref* to the detector because the proposed detector does not use the reference signal directly for detection. However, the detector error measurement is equivalent to measuring the time separation between the output edge and the reference edge. In addition,

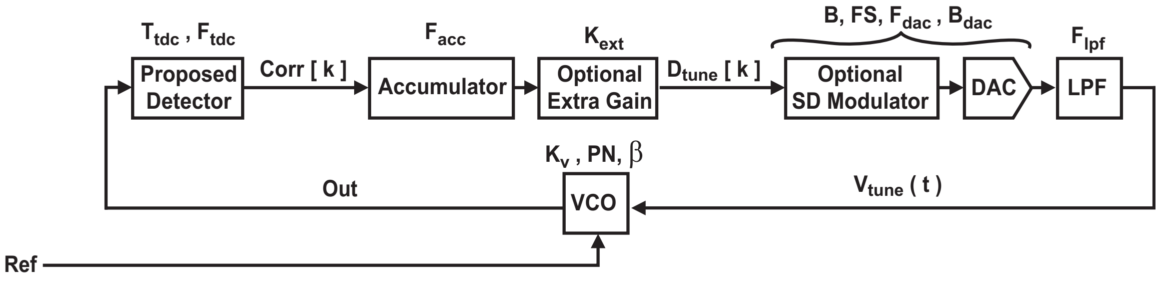


Figure 3-6: General Proposed Architecture.

the reference signal can be used in the detector to determine which sign is applied to each sample in the correlator. K_v is the frequency-voltage gain factor of the VCO, β is the realignment factor of the VCO, as defined in [2], which describes the strength of the phase correction (or realignment) due to introducing the reference signal in the VCO. β ranges between zero (for normal VCOs) to one (for ideal MDLLs), with values in between for injection-locked architectures. Finally, PN is the phase noise specification of the VCO.

As explained in section 3.2, the tuning path consists of the proposed detector, which extracts the tuning error, Δ , to be integrated by the accumulator to drive it to zero at steady state. The raw resolution of the TDC, T_{tdc}, along with its sampling rate, F_{tdc}, determine the noise contributed by the detector and its ability to remove the tuning error. The clocking rate of the accumulator determines its gain, and depends on the downsampling ratio of the correlator and whether an accumulate and dump is used. Simply, the value of F_{acc} is the rate of accumulation of the detected error. For example, if the correlator detects one Δ every two cycles of F_{tdc}, then $F_{acc} = F_{tdc}/2$. Note that no explicit loop filter is needed in the proposed architecture, since the inherent VCO pole at the origin is canceled by the direct introduction of the reference in the VCO (which has a high-pass filtering effect [2]). The optional extra gain block, with the gain of K_{ext}, adjusts the gain of the loop to achieve the desired loop bandwidth, and it can be implemented by a simple bit shifting operation, possibly through a multiplexer (for programmability).

The digital tuning signal, D_{tune} , is converted to an analog format using the

DAC. However, a Sigma-Delta modulator can be used before the DAC to lower its resolution requirement, or to lower the gain of the loop without additional digital quantization. B is the bit width of D_{tune} , and B_{dac} is the input bit width of the DAC, which is the same as B if no Sigma-Delta modulator is used. FS is the full scale output of the DAC, while F_{dac} is its clocking rate (which is the same as that of the Sigma-Delta modulator). Typically, F_{dac} is the same as F_{acc}, unless reduction of quantization noise is desired, in which case the Sigma-Delta modulator and DAC can be overclocked (which of course causes its power consumption to increase). Finally, the lowpass filter block attenuates the DAC quantization noise and glitch energy, and can be implemented as a simple RC filter. Note that the order of the filter needs to be at least the same as the order of the Sigma-Delta modulator to filter the quantization noise at higher frequencies. A higher order filter will suppress the noise further, at the expense of complexity and area.

In the next two subsections, two models for the general architecture will be discussed. The first model is phase-based and is used for noise modeling and estimating the jitter performance of the system. The second model is period-based and provides a simple method for loop gain calculation (to determine the loop bandwidth).

3.6.1 Noise Model

Figure 3-7 shows the noise model for the general proposed architecture of Figure 3-6. This model is based on [2] (specifically, the derivation of the $H_{rl}(j\omega)$ and $Hup(j\omega)$), with the addition of the blocks and noise sources that are relevant to the proposed technique.

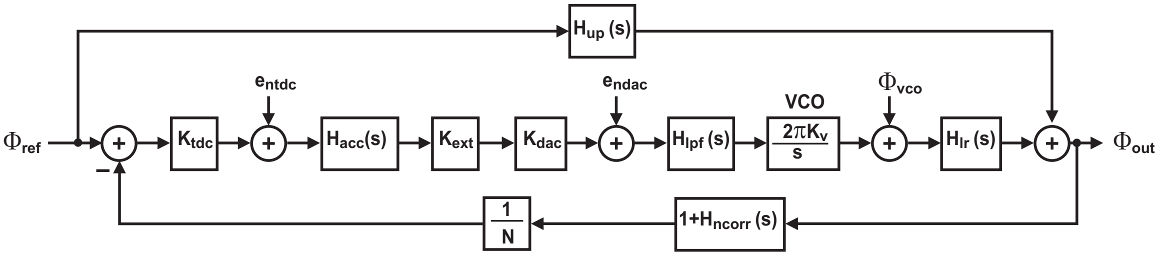


Figure 3-7: Noise Model.

The formulas for H_{rl} and H_{up} [2] are rewritten below in terms of G_{ref}, which is the unity DC-gain LPF that represents the effect of the introducing the reference directly to the VCO. This representation is similar to that used in [8] as a base function to parameterize PLL dynamics. Note that if the architecture is run open-loop, i.e. with tuning disabled, and for β equals to one, the H_{rl} and H_{up} resemble the transfer functions of the VCO and reference noise in that the first is a unity-gain high pass filter and the second is a low pass filter with a gain of N, respectively, similar to the modeling in [8].

$$\begin{aligned}
H_{rl}(j\omega) &= 1 - G_{ref}. && (3.2)\\
H_{up}(j\omega) &= NG_{ref}. && (3.3)\\
G_{ref}(j\omega) &= \frac{\beta}{1+(\beta-1)e^{-j\omega T_{ref}}}\, e^{-j\omega T_{ref}/2}\, \frac{\sin(\omega T_{ref}/2)}{\omega T_{ref}/2}. && (3.4)
\end{aligned}$$

$H_{ncorr}(j\omega)$ is the transfer function that corresponds to the noise added by the correlation operation, which involves the substraction of two periods, instead of comparing the output and reference edges as in classical detectors. As shown in section 3.6.4, the additional noise by the correlation operation can be neglected compared to the noise contribution of the implicit comparison of the output and reference signals, and, hence, the value of $H_{ncorr}(j\omega)$ can be set to zero.

The transfer functions of the rest of the blocks of Figure 3-7 are listed below.

$$\begin{aligned}
K_{tdc} &= \frac{T_{ref}}{2\pi T_{tdc}} && (3.5)\\
H_{acc}(j\omega) &= \frac{1}{1-e^{-j\omega T_{acc}}} \approx \frac{F_{acc}}{j\omega} && (3.6)\\
K_{dac} &= \frac{FS}{2^B} && (3.7)\\
H_{lpf}(j\omega) &= \frac{1}{1+\frac{j\omega}{2\pi f_{lpf}}}. && (3.8)
\end{aligned}$$

Using the same parameterized modeling approach of [8], the open-loop transfer function of the architecture, $A(j\omega)$, and the unity DC-gain LPF base function, $G(j\omega)$, are given below.

$$A(j\omega) = K_{tdc}H_{acc}(j\omega)K_{ext}K_{dac}H_{lpf}(j\omega)\frac{2\pi K_v}{j\omega}H_{rl}(j\omega)/N \tag{3.9}$$

$$G(j\omega) = \frac{A}{1+A}. \tag{3.10}$$

The explicit noise sources in Figure 3-6 are the TDC quantization noise, e_{ntdc}, the DAC quantization noise, e_{ndac}, and the VCO phase noise Φ_{vco}. The transfer functions from major noise sources to the output, using the derived base function, $G(j\omega)$, are listed below. [1]

$$TF_{tdc}(j\omega) = \frac{\Phi_{out}}{e_{ntdc}} = N\frac{G(j\omega)}{K_{tdc}} \tag{3.11}$$

$$TF_{dac}(j\omega) = \frac{\Phi_{out}}{e_{ndac}} = H_{lpf}(j\omega)\frac{2\pi K_v}{j\omega}(1 - G_{ref}(j\omega))(1 - G(j\omega)) \tag{3.12}$$

$$TF_{vco}(j\omega) = \frac{\Phi_{out}}{\Phi_{vco}} = H_{rl}(j\omega)(1 - G(j\omega)) = (1 - G_{ref}(j\omega))(1 - G(j\omega)). \tag{3.13}$$

The typical noise power density for the major noise sources are shown below.

$$S_{tdc,white}(f) = \frac{1}{12F_{tdc}} \tag{3.14}$$

$$S_{tdc,1^{st}ordershaped}(f) = \frac{1}{12F_{tdc}}(2\cos(\pi f/F_{tdc}))^2 \approx \frac{1}{6F_{ref}} \tag{3.15}$$

$$S_{dac,nosd}(f) = \frac{FS^2\ 2^{-2B_{eff}}}{12F_{dac}} \tag{3.16}$$

$$S_{dac,sd}(f) = \frac{FS^2\ 2^{-2B_{eff}}}{12F_{dac}}(2\sin(2\pi f/F_{dac}))^2 \tag{3.17}$$

$$S_{\Phi_{vco}}(f) = \frac{F_{noff}^2 10^{\frac{noff}{10}}}{f^2}(1 + \frac{F_{f3corner}}{f}). \tag{3.18}$$

[1] Post-publication note: the reference noise transfer function was omitted originally. It was reported in [Helal, *et al.*, JSSC, May 2009], and repeated here for convenience: $TF_{ref}(j\omega) = \frac{\Phi_{out}}{\Phi_{ref}} = N\frac{A(j\omega)+G_{ref}(j\omega)}{1+A(j\omega)} \approx NG_{ref}(j\omega)$.

The noise from the TDC is assumed white for a scrambling structure with no noise shaping, so the power density has a simple quantizer format [34]. On the other hand, if the TDC has noise shaping (for example the first-order noise shaping in the GRO TDC that is used in the prototypes[32]), then the noise power density will have a factor of $(2\sin(\pi f/F_{tdc}))^2$ [34]. However, due to the frequency translation of the correlator, the noise shaped peak will be at DC instead of $F_{tdc}/2$, and thus the effective noise shaping factor will be $(2\cos(\pi f/F_{tdc}))^2$. In addition, since the TDC noise will affect the output phase noise according to the loop bandwidth, then the TDC noise can be approximated as a white noise quantizer scaled by 4. The formula shown for the phase noise of the free running VCO, assumes a $1/f^2$ roll off with power density of *noff* dBc/Hz at an offset of F_{noff} Hz, and a $1/f^3$ corner frequency of $F_{f3corner}$ Hz, which is the typical phase noise distribution (more detailed treatment can be found in [7, 8]).

Finally, the output phase noise will combine the various noise sources, scaled by their respective transfer functions as shown below.

$$\begin{aligned} S_{\Phi_{out}}(f) &= S_{\Phi_{ntdc}}(f) + S_{\Phi_{ndac}}(f) + S_{\Phi_{nvco}}(f) \qquad (3.19) \\ &= S_{tdc}(f)|TF_{tdc}(f)|^2 + S_{dac}(f)|TF_{dac}(f)|^2 \\ &\quad + S_{\Phi_{vco}}(f)|TF_{vco}(f)|^2. \qquad (3.20) \end{aligned}$$

Note that the total output timing jitter, or the individual jitter contributions, can be calculated by integrating the corresponding phase noise from the lowest frequency of interest, F_{low}, according to Equation 3.21 below which shows the variance of timing jitter [7]. The factor 2 is included in the equation because the calculated phase noises in the previous equations describe the phase noise on only one side of the carrier, so the other symmetric side has to be included.

$$\sigma^2\ (sec^2) \approx \frac{2}{(2\pi F_{out})^2} \int_{F_{low}}^{\infty} S_{\Phi}(f)\ df \qquad (3.21)$$

3.6.2 Simplified Noise Model

Figure 3-8 shows a simplified noise model that graphically illustrates how the noise sources affect the output phase noise through their respective transfer functions. The noise contribution from the reference is neglected, due to the assumption that it is supplied by a clean source. The numbers on the roll off lines are in dB/decade, F_o is the loop bandwidth, and F_{eff} is the effective bandwidth of G_{ref}.

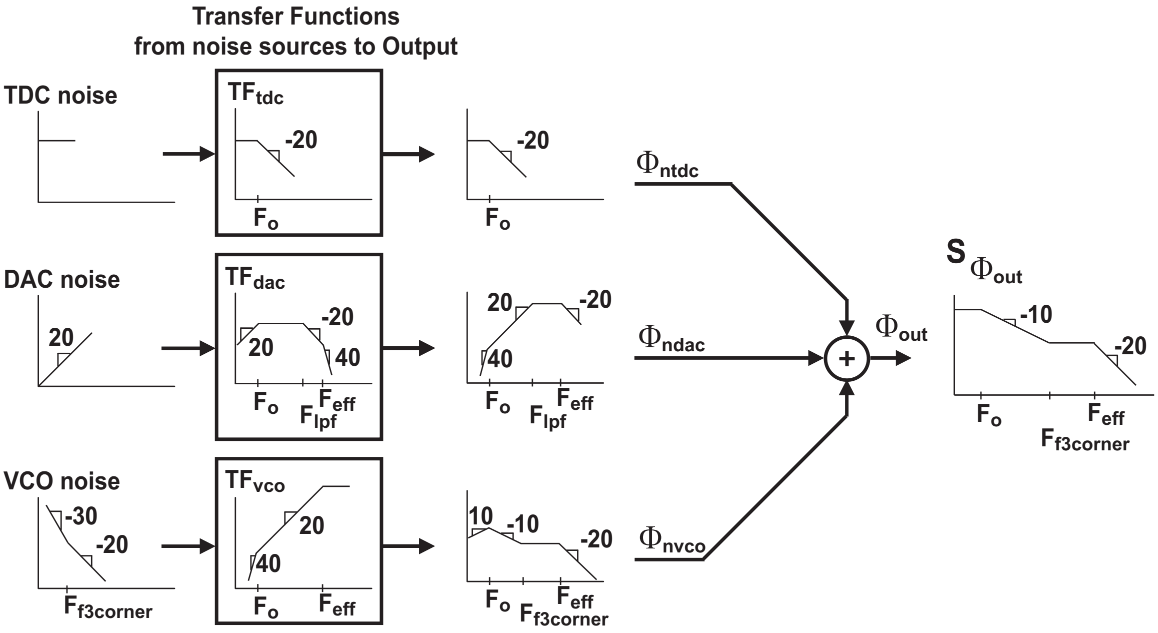


Figure 3-8: Simplified Noise Model.

An important design insight to be gleaned from the figure is that the optimum noise performance can be achieved by choosing a loop bandwidth at approximately the intersection of the scaled contributions from the TDC and the VCO's -10 dB/decade roll off (which originally was a -30 dB/decade roll off before the the direct introduction of the reference in the VCO). In addition, the DAC noise contribution can be minimized by lowering the cutoff frequency of the LPF, F_{lpf}, by increasing the filter order, by increasing the DAC number of effective bits, or by increasing the clocking frequency.

3.6.3 Estimate of the Minimum Achievable Deterministic Jitter

A plausible definition of the minimum achievable deterministic jitter is the tuning error value at which the signal to noise ratio (SNR) at the detector input within the loop bandwidth is unity.

If we assume that the tuning error, Δ , is static at steady state, then its power is simply Δ^2. The noise at the detector input is dominated within the loop bandwidth by the detector noise. The power of the detection noise is approximately the power density of the TDC quantization noise multiplied by the loop bandwidth. Thus, when using a GRO TDC (which is noise-shaped), the deterministic jitter at unity SNR has the following minimum value:

$$\Delta_{min} \approx T_{tdc}\sqrt{\frac{BW_{loop}}{3F_{ref}}}. \tag{3.22}$$

However, if the detection SNR is defined over the whole spectrum, then the minimum achievable deterministic jitter (peak-to-peak) will simply by the total random jitter of the output (in rms), which is calculated using Equations 3.19 and 3.21.

3.6.4 Noise due to Correlation

A number of noise sources caused by the correlation operation can be added to the noise model. The noises added by the Enable Logic, the buffer that drives it, and the buffer that drives the TDC are approximately white, and are filtered by the loop, which makes their contributions insignificant compared to the noise of the TDC and VCO. This can be demonstrated by considering the phase noise spectrum of a buffer, which can be estimated by Equation 3.23 below [35]. The buffer has a timing jitter variance of $\overline{\Delta t_d}^2$, an input signal with a frequency of F_o, and its noise is assumed to have a white distribution.

$$\text{Buffer phase noise floor} \cong (2\pi)^2 \cdot F_o \cdot \overline{\Delta t_d}^2 \tag{3.23}$$

For example, for a 1.6 GHz signal going through a buffer with 0.2 ps jitter rms (which can be estimated from a SpectreRF simulation), the resulting phase noise PSD is -146 dBc, which is more than four order of magnitude less than the expected low frequency phase noise of the output (typically -100 dBc or worse).

The period subtraction operation of the correlator can be analyzed by separating the measured periods of the *Enable* signal into an offset component, which contains the free running period of the VCO, T, and an error component, which contains the tuning error, Δ , as shown in Figure 3-9. This is similar to jitter decomposition in [36]. The term jitter will be used in the following discussion instead of noise, since time is what is being measured by the detector. Assuming linearity, the jitter carried by the *Enable* signal equals the sum (in terms of variance) of the offset jitter and the error jitter . The jitter bands superimposed on the shown waveforms are denoted by hatched green for the reference jitter and blue for the output jitter.

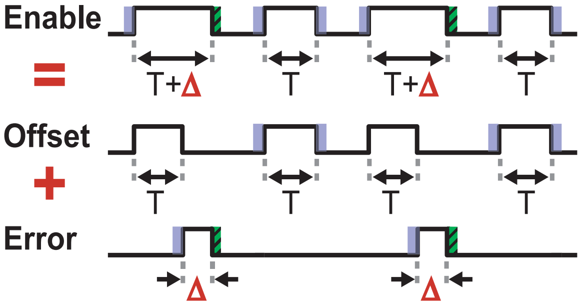


Figure 3-9: Correlator Period Subtraction and its Noise Contribution.

The correlation subtracts consecutive periods, so the offset jitter becomes period jitter, while the error jitter corresponds to the jitter of the output compared to the reference (which approximately equals to the absolute jitter, assuming a clean reference). Since the period jitter is calculated from the phase noise by scaling it by $\sin^2(\pi f F_{out})$ [7], and with such scaling factor significantly less than one within the loop bandwidth, the period jitter can be neglected compared to the error jitter. Therefore, H_{ncorr} in Figure 3-7 can be safely set to zero.

3.6.5 Period-Based Model for Bandwidth Calculation

Figure 3-10 shows the period-based model for the general architecture of Figure 3-6, which provides a simple method to calculate the tuning loop bandwidth.

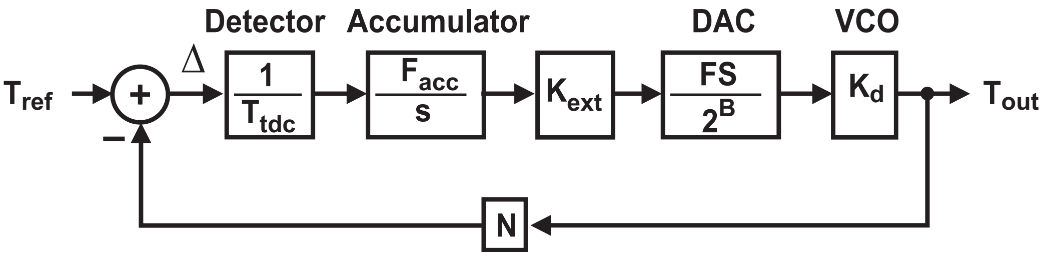


Figure 3-10: Period-Based Model.

The output node of the model is the output period of the VCO which, if not ideally-tuned, will accumulate a total time error of Δ over the period of a reference cycle. During the reference period, N output cycles are completed, and thus the error is the difference between T_{ref} and NT_{out}. The detector scales the timing error by the reciprocal of the TDC raw resolution, $\frac{1}{T_{tdc}}$, while the accumulator approximates an integrator with a gain of F_{acc}. The extra gain block provides a gain of K_{ext}, which equals to 2^M if implemented by bit shifting, where M is the number of bits that the digital word is shifted by. The DAC, including the optional Sigma-Delta modulator , is represented by the gain block with the value of $\frac{FS}{2^B}$, where FS is the full scale of the DAC, and B is the number of input bits of the DAC (or the input bits of the Sigma-Delta modulator , if it is used). Finally, the change in the VCO period due to change in its tuning voltage is determined by the VCO period gain factor, K_d, which can be calculated from the frequency gain factor, K_v, using Equation 3.24 below. The minus sign in the expression is ignored for bandwidth calculation, but is included here for completeness.

$$K_d = -\frac{K_v}{F_{out}^2} \tag{3.24}$$

Since the simplified loop has only a pole at the origin, the loop bandwidth can be calculated directly from the loop gain, as shown in Equation 3.25.

$$\text{Tuning Loop Bandwidth (Hz)} \approx \frac{1}{2\pi}\frac{1}{T_{tdc}}F_{acc}K_{ext}\frac{FS}{2^B}K_dN \quad (3.25)$$

$$= \frac{1}{2\pi}\frac{1}{T_{tdc}}F_{acc}K_{ext}\frac{FS}{2^B}\frac{K_v}{F_{out}F_{ref}}. \quad (3.26)$$

Finally, note that the bandwidth predicted by the period-based model differs from what is predicted by the phase-based model discussed in Section 3.6.1. The ratio of the period-based to the phase-based bandwidths depends on β, and has maximum of two when $\beta = 1$ [2].

[2] Post-publication note: this paragraph was updated. An empirical relationship can be found to match the bandwidths predicted by the period-based and phase-based models. However, further study of this issue is warranted.

Chapter 4

MDLL Prototype

This chapter presents an MDLL prototype that incorporates the proposed detection and tuning technique discussed in Chapter 3. In addition to testing the proposed technique, the goal of the MDLL prototype is to demonstrate a highly-digital architecture that is amenable to inclusion in digital design flows and to porting between CMOS technologies, which makes it suitable for digital ICs. To show that the proposed architecture does not sacrifice performance for a highly-digital implementation, but in fact outperforms previous approaches, the target jitter performance is set to be in the sub-picosecond range in terms of both random and deterministic jitter.

This chapter starts with an overview of the prototype architecture and its blocks in Section 4.1. Section 4.2 discuses the test board and Section 4.3 presents the MDLL core blocks, including the Multiplexed Ring Oscillator, divider, and Select Logic. Section 4.4 discusses the Enable Logic circuit and Section 4.5 overviews important specifications of the GRO TDC. Section 4.6 discusses some of the issues related to the digital blocks that were implemented on the FPGA (including the correlator, accumulator and Sigma-Delta modulator). Section 4.7 overviews the digital-to-analog converter and the lowpass filter used in the prototype. Section 4.1 and 4.9 present results of the noise and behavioral models, respectively. Finally, Section 4.10 discusses some additional implementation issues.

4.1 Overall Architecture

Figure 4-1 shows a simplified form of the overall implementation of the MDLL prototype, which nominally multiplies a 50 MHz reference frequency to a 1.6 GHz output frequency. The prototype consists of two custom integrated circuits that implement the GRO and MDLL core logic, an FPGA board that implements the digital operations, a commercially available 16-bit DAC, and a first order RC filter with a pole at 3 MHz.

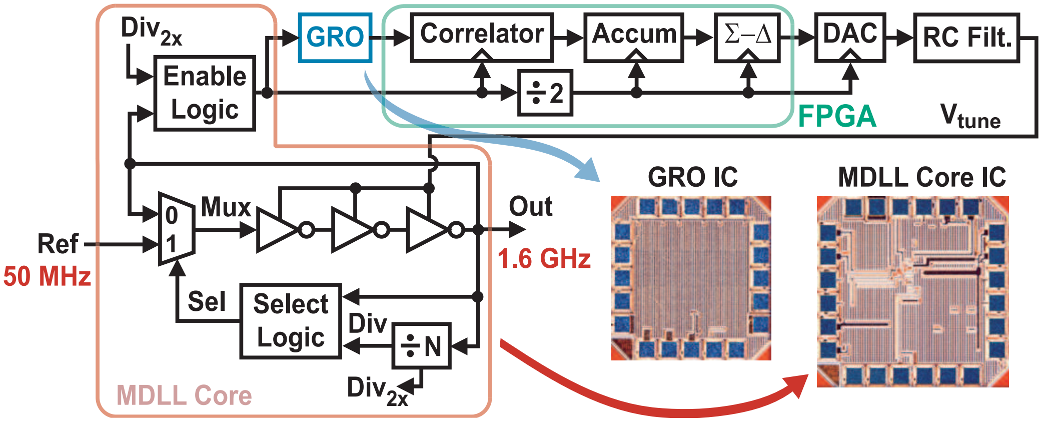


Figure 4-1: Implementation of the Proposed MDLL Architecture.

The major blocks of the MDLL core IC are a multiplexed ring oscillator, divider, select logic, and the edge generator (the main block of the Enable Logic). The GRO IC contains the GRO TDC which detects the period of the selected output cycles, in addition to a flip-flop that converts the edges generated by the Edge Generator to the *Enable* signal.

The FPGA provides the digital functions of the correlator, accumulator, and Sigma-Delta modulator, in addition to control of the DAC and the MDLL setup bits. The DAC's effective number of bits can be changed in the FPGA by using only the desired number of top MSBs, and connecting the rest of the lower LSBs to zero (to disable them). A first-order RC filter is used to filter the DAC noise before feeding the fine tuning port of the MDLL oscillator.

4.2 Prototype Test Board

Prototype components were mounted on a 4-layer FR4 gold-plated test board. The custom GRO and MDLL core chips were wire bonded directly on board in order to minimize bondwire inductance. The FPGA board (Opal Kelly, XE3010, with Xilinx Spartan-3) was socket-mounted to the test board using surface mount connectors (SAMTEC, BTE-040-01-F-D-A), and was programmed from a PC through a USB connection (using MATLAB).

Two voltage levels are required on the board: 3.3 V (for the FPGA and the DAC) and 1.2 V (for FPGA and the custom chips). The Main power was supplied to the board from a power supply at 3.3 V, while the 1.2 V was supplied to various chips from an on-board power regulator through jumpers. Optionally, these jumpers can be used to connect the supply lines directly to external power sources, in order to minimize cross-coupling of supply noise.

As shown in Figure 4-2, the MDLL IC drives the GRO IC with two signals, *en* and *dis*, whose edge time separation captures the cycle periods of the output and is converted to Enable pulses in the GRO IC. The digital output of the GRO IC is read by the FPGA board, In addition to the 10-bit data output, the GRO also provides a clock signal (for proper data registering) and a data-validity signal called outbad, which flags invalid data words.

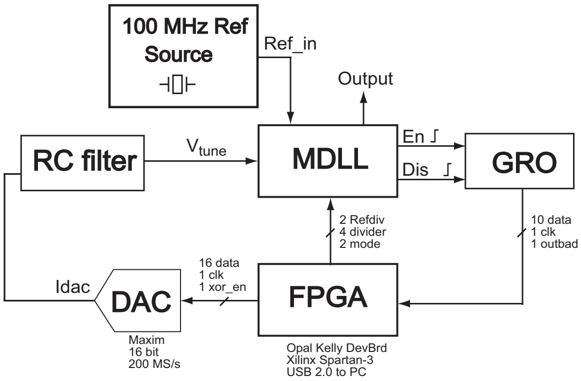


Figure 4-2: Overall MDLL Test Board.

The FPGA board drives the 16 bit DAC, which generates a current output that is converted to voltage by a load resistor. Adding a capacitor in parallel to the load resistor results in a first-order low pass filter, which attenuates the noise of the DAC and feeds the fine tuning port of the MDLL core, TuneF. The MDLL core has a coarse tuning port, TuneC, which is manually tuned using a potentiometer (not shown in Figure 4-2).

The reference signal is generated by an off-chip, low noise, 100-MHz oven-controlled crystal Oscillator (OCXO) reference source (Wenzel, 501-04516 Rev D) that is internally divided on the MDLL core IC to a range of possible reference frequencies including 12.5, 25, 50 and 100 MHz. Testing the architecture using different reference frequencies provides additional measurement points to corroborate the noise model. In addition, it allows a measured comparison of performance trade-off with the change of the reference frequency. The reference and output signals are coupled in and out of the test board using SMA connectors.

The next sections further describe the major prototype blocks.

4.3 MDLL Core Blocks

Figure 4-3 shows the die photo of the MDLL core IC, which was fabricated in a 0.13 μm CMOS process (IBM mixed-mode process, 8RF-DM). The die active size is 150 μm $\times$ 250 μm out of a total die size of 1.2 mm $\times$ 1.2 mm. Most of the rest of the chip area was filled by decoupling capacitors.

Figure 4-4 shows the MDLL core functional blocks. The MDLL core includes the following blocks: the multiplexed ring oscillator, divider, select logic, edge generator, and reference divider. Except for the multiplexed ring oscillator, all logic circuits used standard cells, except registers that were used in the dividing and retiming stages, which had a TSPC topology, but nonetheless were not custom designed for this prototype.

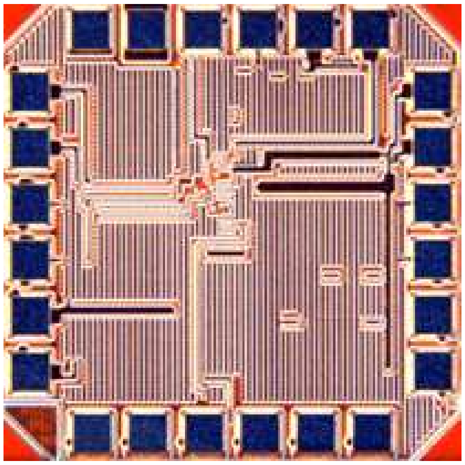


Figure 4-3: MDLL Core Die Photo.

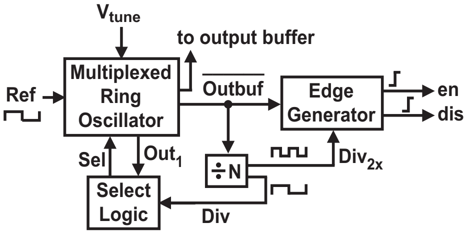


Figure 4-4: MDLL Core Functional Blocks.

4.3.1 Multiplexed Ring Oscillator

Figure 4-5 shows the multiplexed ring oscillator and its constituent delay cells. The delay cells are similar to [37], except that only a single-ended NMOS bias is used for frequency tuning in order to improve phase noise, increase speed and reduce complexity. Separate coarse and fine tuning ports, TuneC and TuneF, which are implemented by different size NMOS devices as shown in the figure, are used to achieve a wide frequency range and a relatively low K_v value for the MDLL tuning feedback loop (which feeds only the fine tune port). The narrowed tuning range offered by the fine tune port both reduces the impact of noise from the MDLL tuning loop and also helps prevents sub-harmonic locking during the tuning process. In the prototype, the voltage of the coarse port is tuned by hand to achieve the appropriate frequency range for the fine tune port.

An important design consideration of the multiplexed ring oscillator is to match

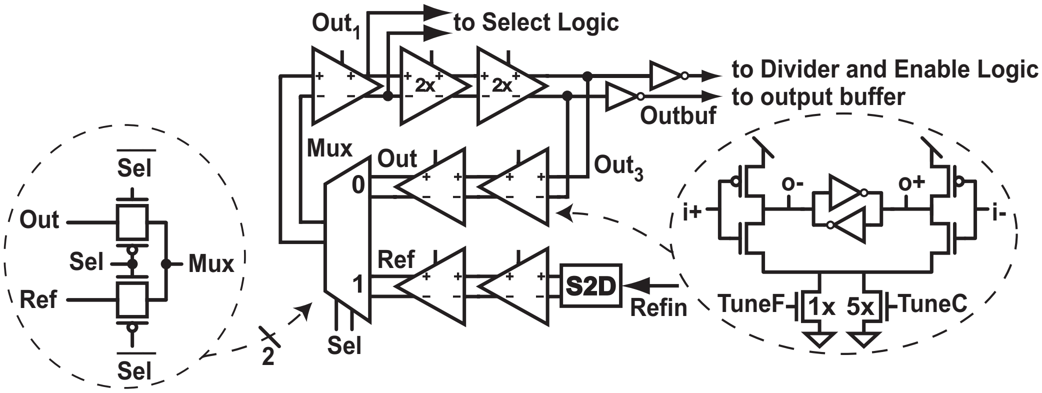


Figure 4-5: Multiplexed Ring Oscillator.

the slope of the edges of the two inputs feeding into the multiplexer, which correspond to the output of the last delay cell and the reference input, in order to minimize deterministic jitter [4]. In addition, care must be taken to avoid influence of the *Sel* signal on the edges running through the multiplexer since such influence would also lead to increased deterministic jitter [1]. To deal with the first issue, the reference input signal is buffered using two delay cells that are identical in design and tuning to the oscillator delay stages, as shown in Figure 4-5. Each of these delay stages are placed in close proximity to each other in the chip layout in order to achieve good matching between them. As for the second issue, the impact of *Sel* is sought to be minimized by striving for fast edges going through the multiplexer [1] so that there is a smaller time window for *Sel* to influence them. To this end, the number of delay cells is chosen to be as large as possible while still supporting the desired frequency range of the oscillator, which leads to less delay per stage and, therefore, faster edges. It should be noted that increasing the number of delay stages does not have a significant impact on phase noise [38]. In this 0.13 μm CMOS design, the choice of five delay stages allows oscillation frequencies high enough to achieve our 1.6 GHz target with a comfortable design margin. Note that the second and third delay stages were doubled in size in order to drive external blocks. Furthermore, edges going in and out of the multiplexer are kept sharp by eliminating external loading on its input and output.

Differential load balancing in the oscillator delay stages produces a more symmetric waveform, and thus, lowers 1/f noise [38]. To achieve that goal, care was taken to provide matching loads for all delay cells. To that effect, the output of the first delay cell after the multiplexer, Out_1, drives two identical gates in the select logic, and Out_3 drives two identical inverters, with one output feeding the output buffer, $Outbuf$, and another, $\overline{Outbuf}$, that drives the divider and the edge generator.

Proper design of the multiplexer is required to avoid mismatch between the *Out* and *Ref* edges while they pass through it. This issue is especially problematic for architectures that detect the error by comparing the edges of the two multiplexer inputs since the measurement circuitry will not be able to detect any error due to path mismatch in the multiplexer which occurs after the observation nodes. Fortunately, this issue is significantly mitigated in the proposed architecture since the single-path detection method that is employed will detect the error regardless of its source. Nevertheless, care was taken to match the two paths of the multiplexer, and to minimize its propagation delay so that the impact of any remaining mismatch would be reduced.

4.3.2 Divider

The divider was designed to have the required division ratios for a number of reference frequencies and output frequencies. The divider consists of two divide-by-2 stages (providing a divide-by-4 ratio), followed by three bypass-able divide-by-2 stages (providing division ratios of 1,2,4,8), and two dual-modulus divider stages (together providing division ratios from 4 to 7). This configuration provides division ratio from 16 to 224, which allows the testing of the MDLL at possible output frequencies of 0.8, 1, 1.2, 1.4, 1.6 and 2 GHz from reference frequencies of 12.5, 25, 50 and 100 MHz (internally-divided from the 100 MHz reference source).

Retiming stages were used at various points in the chain to eliminate metastability due to variation in clock-to-output delays. Additional retiming stages are used at the output of the divider to synchronize its output with the enable edges, as described in Section 4.4. However, it is worth mentioning that a more programable and power efficient divider design uses a modular-divider topology [39], with the output taken from the mod-out port. In such configuration the divider output is inherently retimed by the input signal, and thus eliminates the need for explicit retiming stages to avoid metastability, as discussed in Section 6.3.

Reference Divider

The reference divider uses three divide-by-2 stages whose outputs are multiplexed and retimed by the reference input (to remove asynchronous divider jitter). A second 2-to-1 multiplexer allows the selection of the reference input without division.

4.3.3 Select Logic

The select logic circuit and its timing diagram are illustrated in Figure 4-6. The main goal of this block is to generate a select signal, *Sel*, with sharp edges that are sufficiently separated in time from the falling *Out*and the rising *Ref* edges, such that the *Sel* signal edges fall in approximately the middle of the ring oscillator transitions. This is important in order to minimize the influence of multiplexing on the edges passing through the multiplexer around the time of its switching, and reduce the effect of multiplexing nonidealities, e.g. charge injection and the feedthrough from the *Sel* signal,.

In normal operation, the select logic is enabled (by pulling the signal *mode* high), and the select signal is generated as follows: the last falling edge of Out_3 (before multiplexing *Ref*) causes the divider output, *Div*, to rise and trigger a D flip-flop with reset (DFFR), which allows the NAND gate and the subsequent inverter to generate a rising *Sel* edge after a rising Out_1 edge. The falling Out_1 edge, which occurs after the *Ref* edge is passed, causes *Sel* to fall and resets the DFFR (when mode is high) to make it ready for the next select cycle. The signal Out_1 is specifically chosen to drive the select logic in order to guarantee that the multiplexer switches at a time that is approximately in the middle of the Out transition, or specifically, after the edge has fully passed the multiplexer. This design alleviates the challenge of optimal positioning of the *Sel* signal as posed in [4], in addition to providing large

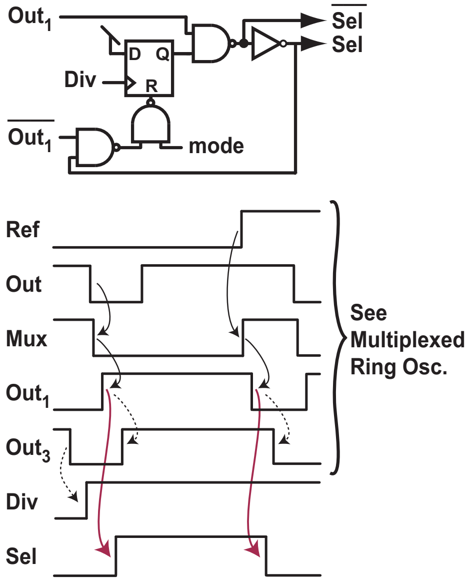


Figure 4-6: Select Logic and associated signals.

timing margins that enhance its robustness. However, for a much higher output frequency or for slower processes, it might be necessary to reevaluate the retiming budget and the choice of the oscillator branch that drive the divider. In that case, it must be made sure that the multiplexer is completely switched before the rise of the last *Out* edge at the highest target frequency of the oscillator.

Note that all NAND gates used in the select logic circuitry are identical so that the load is symmetric on the Out_1 branch of the ring oscillator. In addition, the design is almost entirely based on standard cells for ease of design and portability. The only cell that is not strictly a standard-cell is the DFFR, which has a TSPC topology, as previously mentioned. And finally, setting the signal *mode* to low disables the reference multiplexing for testing purposes or to run the architecture as a PLL, if desired. If such functionally is not needed, the NAND gate connected to the reset input of the DFFR can be replaced by an inverter.

4.4 Enable Logic

The Edge generator circuit and its timing diagram are illustrated in Figure 4-7. This is the first stage in generating the Enable signal that drives the GRO TDC, and it generates two signals, *en* and *dis*, whose relative delay captures the period of two MDLL output cycles every reference cycle, namely T and T+Δ .

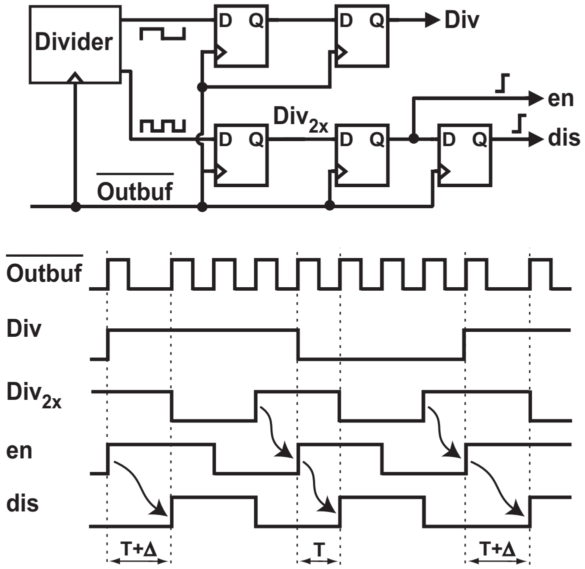


Figure 4-7: Edge Generator and associated signals.

As shown in Figure 4-7, the edge generator has two inputs. The first input, $\overline{outbuf}$, is an inverted version of Out_3 (the output of the third delay cell in the ring oscillator), and it carries the period information. The second input, Div_{2x}, is the divider output that runs at twice the reference frequency, and it selects the proper period to sample, however, its exact edge location does not affect the measurement. Div_{2x} is retimed twice by $\overline{outbuf}$ to generate two signals, *en* and *dis* which have the desired property of being separated in delay by the corresponding MDLL output period. These two signals go off-chip to the GRO TDC to form the Enable signal. A retiming stage is used in the Div_{2x} path to increase immunity to metastability for the retiming DFF that generates the *en* signal, as shown in Figure 4-7. The divider output, *Div*, is sent through two retiming stages. One stage is used in order to relax the timing margins for the select logic, while the second is needed to delay the rising edge of *Div* for one output cycle in order to synchronize it with the rising edge of *en*, such that the period that includes the error, Δ , is captured. Note that any offset or non-ideality that affects the generation of the *en* and *dis* signals is consistent between samples and, hence, will be canceled by the subtraction operation in the digital correlator. Furthermore, the design is simple and robust, and the DFF cells do not need to be custom-designed, which makes it amenable to porting between technologies.

4.5 Scrambling TDC (GRO)

The TDC used in the prototype was developed by Matt Straayer, who is a colleague in the same research group as the author. Since the design of the TDC is out of the scope of this thesis, only important details about the TDC will be given here. However, the reader is referred to [32, 33] for further details.

The TDC is based on a Gated Ring Oscillator (GRO) architecture [32], in which a ring oscillator is allowed to run only during the measured time, and is turned off otherwise. As shown in Figure 4-8, the number of oscillator transitions during the ON time is counted and it corresponds to the measured time. Assuming that the period of the oscillator is not correlated to the measured time, the consecutive detected samples will exhibit different error values, which in essence scrambles the quantization noise. The scrambling property allows the improvement of the TDC effective resolution according to the amount of averaging the loop performs, as explained in Section 3.4.

In contrast to the VCO-based TDC in [30] which has a continuously-running oscillator, the gating of the oscillator of the GRO TDC between measurements drastically reduces the power consumption of the ring oscillator (especially for small duty cycles, which is the case in the MDLL prototype). In addition, the gating of the ring oscillator provides a first order shaping of the TDC quantization noise. Such a property can be of great benefit to applications such as PLLs and DLLs, however it is not taken advantage of here due to the frequency translation property of the correlation technique [33] as explained in the following section.

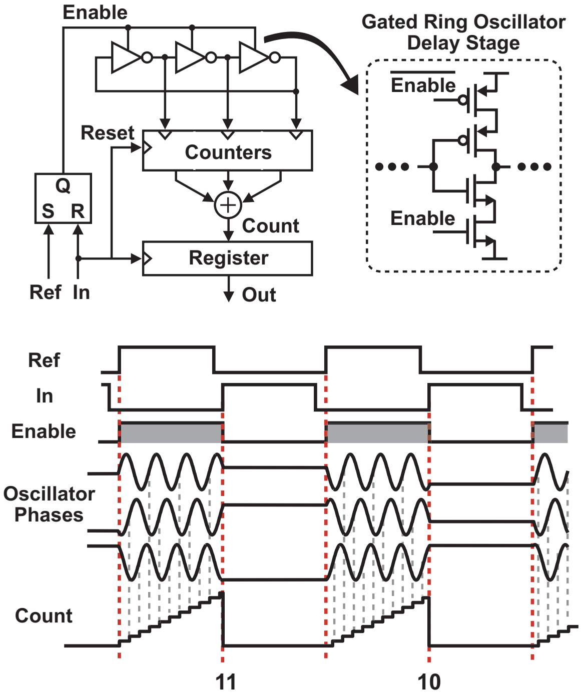


Figure 4-8: GRO TDC structure and associated signals.

4.5.1 GRO Noise-Shaping Nullification due to Correlation

The issue at stake, which is illustrated in Figure 3-5, is that the correlated double-sampling technique used in the MDLL tuning loop requires a relative comparison of alternate samples of the GRO output, which effectively leads to multiplication of the GRO samples by the alternating sequence {..., 1, -1, 1, -1,...}. This multiplication operation, in turn, causes mixing of the higher frequency GRO quantization noise down to lower frequencies, thereby removing the noise shaped characteristics in the original GRO signal. Fortunately, by using a relatively low bandwidth of the MDLL tuning loop (and the high degree of averaging that it provides), the scrambling action of the GRO quantization noise is sufficient without noise shaping to achieve the effective sub-picosecond resolution that we desire.

4.5.2 GRO Dead Zone

An important GRO issue worth mentioning is that the specific GRO topology used in the MDLL implementation exhibits dead zones in its transfer characteristic. This issue was originally discovered during the characterization procedure of the MDLL. During that procedure, the multiplexed ring oscillator was set to a free running mode, while the GRO detected the period of the output cycle. By averaging the GRO digital output over millions of samples, a very precise estimate of the period was achieved, and a smooth detection transfer curve was expected. However, the transfer curve showed flat areas around integer multiples of the raw GRO delay. Fortunately, the dead zones have a predictable and limited region of influence in the measurement space, so simple hand tuning of the GRO (by proper adjustment of its supply voltage) is sufficient to alleviate this issue when testing the MDLL prototype. Further explanation of the dead zone issue can be found in [33], although it is worth noting that the second GRO generation, used in the PILO prototype in Chapter 6, had a negligible dead zone issue.

4.5.3 Implementation Details

The GRO IC was fabricated in a 0.13 μm CMOS process (IBM mixed-mode process, 8RF-DM), and Figure 4-1 shows its die photo. The active size of the GRO chip is 120 μm × 172 μm out of a total die size of 1 mm × 1 mm. The raw resolution at a supply of 1.2 V was about 45 ps.

4.6 Digital Blocks (FPGA)

The FPGA performed the digital operations that included the correlation, accumulation, gain, and first-order $\Sigma\Delta$ modulation. There are a number of ways these operations can be implemented on an FPGA or when integrated on-chip. Figure 4-1 shows the digital blocks in a simplified way, where there is a separate block for each function, and all blocks after the correlator are clocked by a divided-by-2 version of the *Enable* signal. Such a simplified block diagram conveys the essence of the architecture, however, it is not robust and it does not offer the most efficient implementation. Four issues require elaboration: the sign ambiguity of the error(which motivates synchronizing the correlator with the reference), the clock rate of the Sigma-Delta modulator and DAC (which affect the quantization and glitch noise), the implementation of the correlator and accumulator(which affects the power and area efficiency of the implementation), and the adjustment of the loop gain.

4.6.1 Correlator Reference Synchronization

An omitted detail in Figure 4-1 is an explicit path to synchronize the sign of the correlator with the reference. Without such synchronization, the sign of the correlator output, Δ , might be non-inverted (when T is subtracted from T+Δ , which is the case shown in Figure 4-9) or inverted (when T+Δ is subtracted from T). The actual sign will depend on the arbitrary initial state of the divide-by-2 divider that provides the sign input for the correlator. By synchronizing the correlator with the reference, ambiguity about the sign of the detected error would be removed and the sign required for the negative feedback loop can be guaranteed.

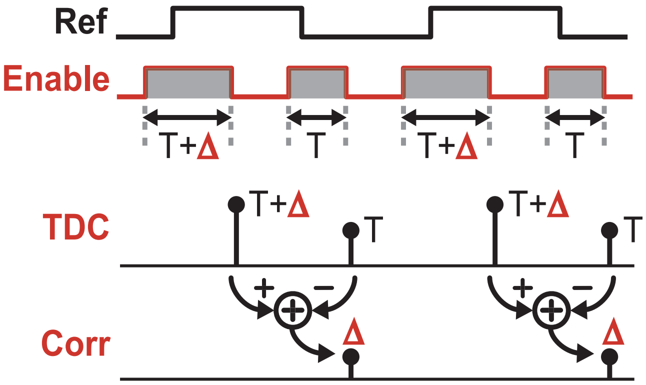


Figure 4-9: Correlation and Reference Synchronization.

While designing the circuit that synchronizes the correlator with the reference, attention should be given to the delay through the GRO(for example, due to pipelining), and the reference duty cycle . For a general system, it is preferable to use the reference transition, rather than its level, to avoid the effect of its duty cycle. In all cases, the addition of optional sign flipping is useful for testing, but it can not be used for a self-starting system (unless a built-in test checks for the correct sign and flips it if needed). In the case of the prototype, no reference synchronization was used, and the sign was flipped manually when needed.

4.6.2 Efficiency of Digital Blocks

Instead of implementing the digital path with a separate block for each function, as shown in Figure 4-1, the correlator and the accumulator can be combined into one block. In such a case the GRO output is connected to the accumulator input with a sign that alternates between samples (for example using an XOR gate), and a down sampler at the accumulator output reduces that data path rate, as explained in Section 3.5

Such block combining can reduce the digital path complexity, and possibly the power consumption, however, this will depend, among other things, on the actual implementation and the bit width of both the correlator and accumulator. The combined correlator/accumulator was used in the MDLL implementation on the FPGA since it minimizes timing constraints.

4.6.3 DAC Clock Rate

The DAC needs to be clocked at the reference frequency for the simplest architecture. However, by running the DAC at the GRO clock rate (i.e., double the reference clock), the DAC quantization noise power density would be halved and the glitch energy would be better attenuated by the RC filter (by 6 dB since it is a first order filter). This is especially useful when a Sigma-Delta modulator is used, as the added quantization noise can significantly affect the MDLL jitter performance. However, the benefit of the increased clock rate has to be balanced with the increased power and complexity that would result from such an increase in the sampling frequency and reduction in timing margins.

4.6.4 Digital Adjustment of the Loop Gain

The gain of the loop can be adjusted digitally to achieve the desired the loop bandwidth. To increase the loop gain, bit shifting can be used, in which the wiring between two blocks in the digital path can be multiplexed to allow connecting each output bit level to a higher input level bit. For example, for a gain increase by 2^k, each bit in one of the blocks would be connected to the input of the next block at k bits higher. Reducing the loop gain can be achieved using the same method, but with the opposite wiring direction. However, that would cause the loss of the lowest LSBs, which can increase quantization noise or cause limit cycles. Instead a Sigma-Delta modulator can be used to modulate the lower LSBs into one bit, and in this case, the additional quantization noise would be shaped and easier to reduce by the lowpass filter.

4.7 Digital-to-Analog Conversion

The DAC used in the MDLL prototype is a Maxim MAX5885 16 bit, 200 MS/s current-steering DAC with a 48-QFN package that runs on a 3.3V supply. A 16 bits DAC offers the flexibility to test the MDLL architecture at different bandwidths without the use of a sigma-delta modulator. In addition, a 16 bit DAC has generally much better overall specifications than an 8 bit DAC, and thus would not cause a bottleneck in the prototype performance. For example, a 16 bit DAC generally has much less current noise than an 8 bit DAC (which is expected, since its LSB is 256 times smaller), and thus the current noise can be ignored in the output phase noise calculation. Since the FPGA maximum running speed is about 200 MHz, this DAC can enable the system to run at the highest reference frequency of 100 MHz to demonstrate the best performance. The DAC has a differential current output which is useful when linearity is an issue, however, a single ended configuration was used for simplicity, since the DAC is driving a VCO that has a slightly non-linear, but monotonic, Kv. The DAC has a full scale current output of 20 mA, which can be lowered by adjusting the value of a reference resistor (which is nominally set at 2 kΩ). The DAC has a maximum output voltage of 1.1 V, so a 55 Ω resistor was used as a parallel load, which is connected to the fine tuning port of the MDLL chip, TuneF.

4.7.1 RC Filter

A simple RC filter is used to attenuate the DAC glitches and quantization noise (especially when using a Sigma-Delta modulator). Since the DAC output is driving a 55 Ω resistor, a capacitor is placed in parallel to form the needed pole. The exact value of the pole is not very critical when no ΣΔ modulation is used, since the white quantization noise of a Nyquist DAC is not dominant in the output phase noise. In that case, the pole value can be selected to be high enough compared to the locking bandwidth, in order to avoid affecting the loop stability, and such that it can be reasonably easy to implement on-chip. In addition to the aforementioned condition, for the ΣΔ modulation case, the pole value needs to be low enough to attenuate as much as possible of the ΣΔ noise-shaped quantization noise. A pole at approximately 3 MHz was chosen, for which a parallel capacitance of 1 nF was used. Another 0.1 nF capacitor was placed close to the tuning pin of the MDLL chip, to achieve a lower capacitor ESR and attenuate noise picked up on the PCB trace.

4.8 Calculated Jitter Performance

Using the noise modeling presented in Section 3.6, the phase noise contributions of major noise sources were calculated using MATLAB, as shown in Figure 4-10. These calculations shed a light on the performance that can be expected, and provide a tool for optimizing various parameters.

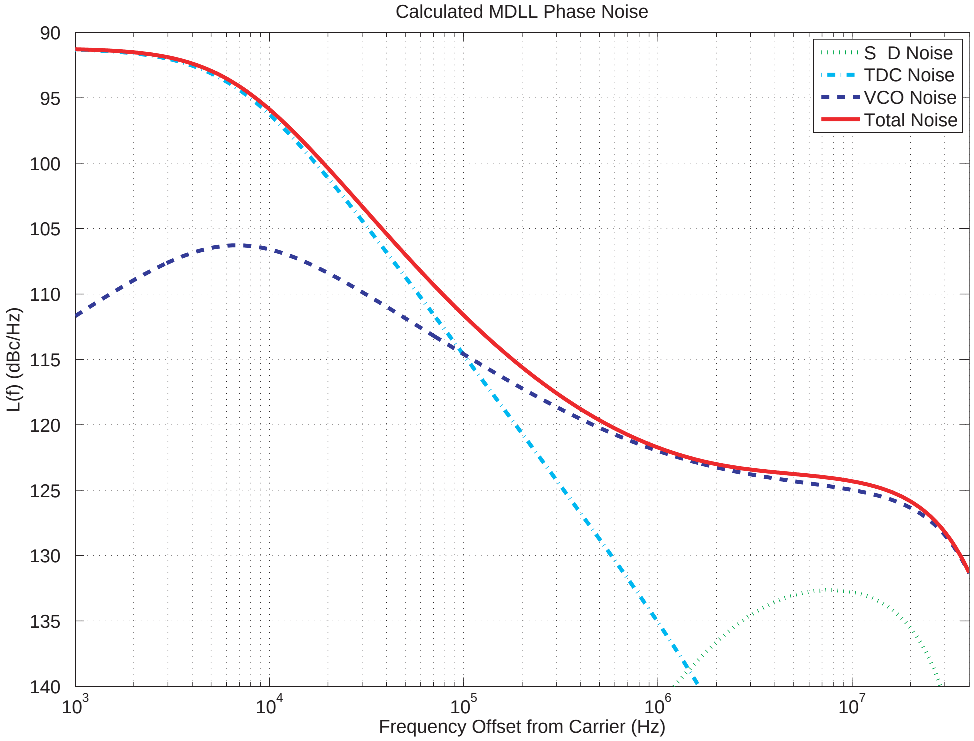


Figure 4-10: Calculated Phase Noise of the MDLL Prototype

The random jitter contributions due to major noise sources were calculated from the corresponding phase noises using Equation 3.21, and are listed below in Table 4.1. Note that the VCO is the dominating source of jitter, and that the Sigma-Delta modulator does not add significant jitter[1].

Table 4.1: MDLL Calculated Jitter

	SDM-DAC	TDC	VCO	Total
Calcuated Jitter (fs rms)	142	378	467	618

Listed below are the various specifications used for calculating the phase noise, which were measured or estimated from the prototype and the circuit simulations. The units for all frequencies are in MHz, and the reference and output frequencies are 50 MHz and 1.6 GHz, respectively.

[1] Post-publication note: calculated phase noise and jitter were updated.

Table 4.2: Parameters used for MDLL jitter calculations

T_{tdc}	F_{tdc}	F_{acc}	K_{ext}	B	FS	F_{dac}	B_{dac}	F_{lpf}	K_v	$noff$	F_{noff}	$F_{f3corner}$
45 ps	100	50	4	16	1.0	100	8	3	100 MHz/V	-127 dBc/Hz	20	1

4.9 Behavioral Simulation

The MDLL prototype was simulated using CppSim, which is a behaviorial simulator based on C++ [40] and is available freely at `http://www.cppsim.com` .

By using behavioral simulations instead of circuit simulations, new ideas can be tested and modified at a much faster pace, and time-prohibitive circuits simulation of the complete system is avoided. Simulation files, MATLAB scripts, excel worksheet, and a tutorial on simulating MDLLs using CppSim is available at the link above.

Figure 4-11 depicts the phase noise of the simulated prototype with the calculated phase noise superimposed on the same figure, showing reasonable matching that corroborates the noise model [2]. The calculated total random jitter is 618 fs(rms), while the simulated total random jitter is 631 fs(rms). The loop bandwidth was set at about 10 kHz.

The number of samples that are needed to capture the phase noise with a small enough bin size, especially for low loop bandwidths, requires a long simulation time. In the case of the shown figure, it took about 20 hours for 2 billion samples. Fortunately, a much faster method is available to measure the jitter of a CppSim simulated system. Figure 4-12 shows the jitter plot of the same MDLL system. The simulation took only about two minutes for 2 million samples, which was sufficient for the system to sufficiently settle. As shown in the figure, the overall jitter is 759 fs(rms). This includes deterministic jitter in addition to the random jitter that is calculated using the noise model and listed in Table 4.1.

[2] Post-publication note: compared to the noise model, simulation results show slightly lower phase noise due to TDC noise (below the tuning loop bandwidth) and VCO 1/f noise, which warrants further study

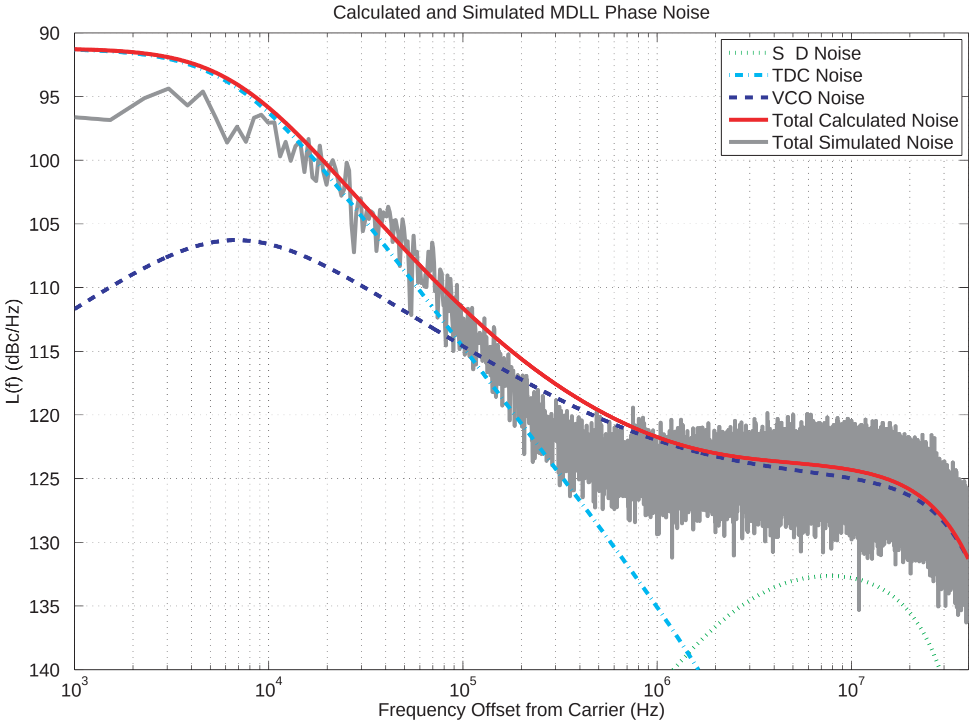


Figure 4-11: Simulated and Calculated Phase Noise of the MDLL Prototype.

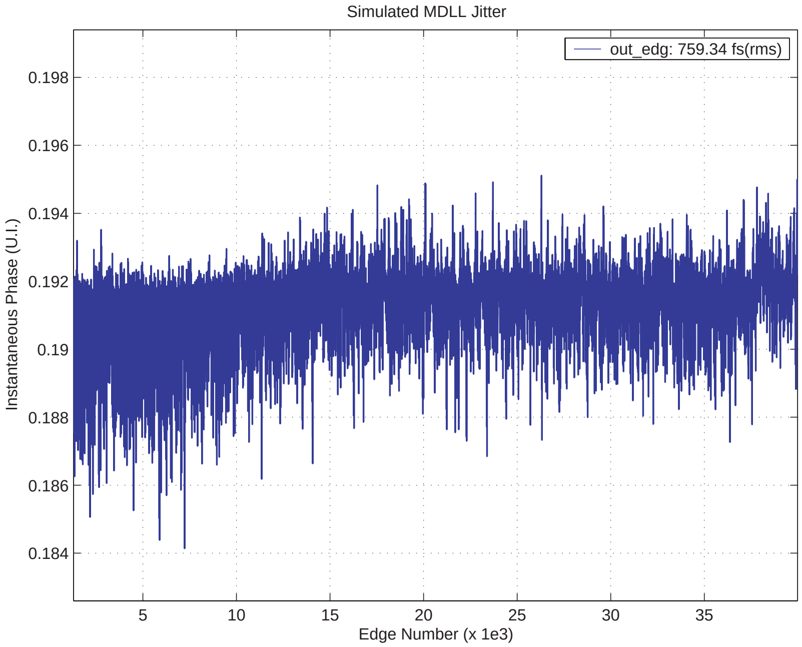


Figure 4-12: Simulated Jitter Plot of the MDLL Prototype.

4.10 Implementation Issues

4.10.1 Power Supply Noise

The MDLL architecture is primarily intended for digital applications, in which case an important issue to be considered is the sensitivity to power supply noise, especially if the same supply is shared among various blocks or if noisy digital blocks are in close vicinity. In such case, it is essential to adequately decouple, and possibly regulate, the supplies of the VCO, DAC, and any blocks that the reference signal passes through. However, the GRO and the Enable Logic blocks are less critical since the correlation operation and the low bandwidth of the loop suppress their supply noise more effectively, though proper decoupling and regulation is certainly advised for these blocks as well.

Power supply and substrate noise affect different blocks in the system in different ways. Each block has a transfer function from the supply line to its output, which can be derived or measured [36]. To qualitatively investigate the effect of supply noise on the system, a rough estimate of the supply noise spectrum can be formed using published measured supply noise spectrums, for example [41]. Using the system model, the effect of the supply noise on the output phase noise and jitter, can then be calculated [36]. By comparing the effect of supply noise on various blocks, decisions can be made regarding the benefit of using supply regulation. To aid with those decisions, a wealth of information is available regarding the effect of, measurement, simulation, and reduction of power supply noise.

In the case of the MDLL prototype, the three major noise sources (TDC, DAC and VCO) are also the blocks which are most affected by supply noise. We will briefly overview the effect of the supply noise through the TDC, since it has a unique aspect compared the other blocks. As discussed in Subsection 3.5.1, the correlator shifts the spectrum of the TDC such that spectrum content at F_{ref} is shifted to DC. The transfer function from the correlator to the MDLL output is a lowpass filter with a transition frequency equal to the loop bandwidth. This means that the detector (i.e. Enable Logic, TDC and correlator) is most sensitive to frequencies around F_{ref} which will be translated to low frequency and then lowpass filtered. Effectively, the detector is sensitive to supply noise in a bandpass filter fashion, with a peak around F_{ref}.

4.10.2 Low Yield due to On-Chip Decoupling Capacitors

Metal-Insulator-Metal (MIM) capacitors were used for on-chip decoupling of the supply lines in addition to thick-oxide MOS capacitors. A significant number of test dies had the problem of excessive current consumption that often made them unusable. After analyzing different factors, MIM capacitors ended up being the main suspects of causing this problem, since their very thin insulator is vulnerable to defects, especially due to ESD events.

After this issue was realized, dies were tested before wirebonding by using a probing station to measure their I-V curves and discarding dies with linear curves (which signify a shorting defect that acts as a linear resistor). Two dies were selected using this approach (for an MDLL prototype and the PILO early test chip), and in both cases no defect issues were observed. As an example of the usefulness of using such method, out of 11 dies tested in one case, only 2 had the expected diode-like IV curves for all susceptible pads.

Chapter 5

Proposed Pulse Injection-Locked Oscillator

This chapter presents a Pulse Injection-Locked Oscillator (PILO) that can be used as a clock multiplier to generate a high-frequency low-jitter output from a clean reference source. The advantage of a PILO over typical injection locking structures is that its operation can be intuitively understood by linearized analysis. More importantly, a PILO can be continuously tuned by leveraging the proposed tuning technique of Chapter 3, which allows significantly reduced deterministic jitter and reference spurs, compared to typical subharmonic injection locked oscillators or injection-locked PLLs (ILPLL).

The chapter starts with Section 5.1, which expands on the motivation for PILOs. Section 5.2 presents a simplified model of an LC oscillator circuit, on which Section 5.3 builds to discuss the concept of a normalized phasor diagram that relates the voltage, current and phase of the oscillator. Section 5.4 presents a linearized analysis of the oscillator phase shift due to an injected pulse, and Section 5.5 compares two methods of pulse injection. Finally, Section 5.6 discusses the effect of tuning, or lack thereof, on PILOs.

5.1 Motivation

Since a narrow pulse can be approximated as an impulse, the effect of pulse injection on the oscillator's LC-tank can be linearized for the short period of the injection, which makes PILOs operation more intuitive to understand. This is in contrast to continuous-wave injection, where the injecting signal continuously affects the oscillator, leading to the need for non-linear analysis of its effect [42, 43, 44]. Additionally, the linearized PILO analysis enables the modeling of its phase noise using the model in [2].

Similar to typical subharmonic injection-locked oscillators, PILOs can be used as clock multipliers that reject the oscillator phase noise at a relatively high bandwidth. The difference is that subharmonic injection-locked oscillators are either not continuously tuned [6], or are tuned using an injection-locked PLL (ILPLL) architecture [26]. In the first case, the oscillator is initially tuned to bring its natural frequency to a value close to a multiple of the injection frequency before injecting the signal into the oscillator with enough power to ensure locking. The need to achieve an adequate injection power level such that injection-locking is maintained across thermal variations will potentially undermine the ability to achieve a low power implementation. In the second case, ILPLLs are prone to mismatch between the injection and the PLL paths, similar to realigned PLLs as described in [2], which causes increased reference spurs and deterministic jitter.

5.2 Simplified LC Oscillator Circuit

Before deriving the relationship between pulse injection and shift in the tank phase, we will first suggest a simplification of the oscillator circuit. At the beginning of the oscillation the negative conductance of the active device, G_m, pumps energy into the oscillating tank. This increases the oscillation amplitude until the average energy pumped into the oscillator equals the average energy dissipated by the effective resistance of the tank, R. At equilibrium, G_m and R cancel each other since the energy in the tank is, on average, constant. Figure 5-1 shows an ideal oscillator, where G_m cancels the tank resistance, and thus both are removed. By removing the non-linearity of the active device from the circuit, it becomes simpler to analyze.

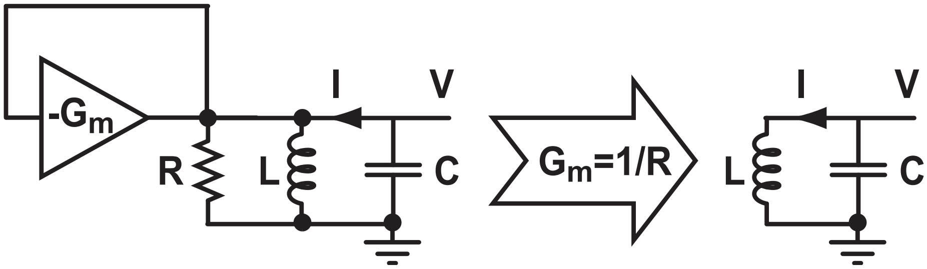


Figure 5-1: Ideal oscillator with canceled G$_m$ and R.

5.3 Normalized Phasor Diagram

To analyze the effect of pulse injection on an ideal oscillator at equilibrium, a normalized phasor diagram is proposed to intuitively depict the relationship between the oscillator phase, the voltage drop over the tank capacitor, and the current through the tank inductor. A normalized phasor diagram is similar to a typical phasor diagram but removes dependency on I and V levels (i.e makes them unitless) in order to depict both the current and voltage of the tank on a unit circle.

Figure 5-2 shows a normalized phasor diagram, where the horizontal axis corresponds to the tank voltage normalized to its maximum, V_{max}, and the vertical axis corresponds to the tank current normalized to its maximum, I_{max}. This phasor representation assumes that the output of the oscillator is in the form of $V_{max}\ cos(\omega_0 t)$, where ω_0 is the natural oscillation frequency of the tank (in radians) and its value is equal to $\frac{1}{\sqrt{LC}}$. The phasor vector rotates counter clock-wise with time, and the angle it makes with the positive voltage axis represents the oscillator phase, Φ. At equilibrium, when G_m cancels R, the length of the vector is unity and its tip traces a unit circle. Now it is easy to visualize a rotating phasor vector on the unit circle,and to relate its angle to the voltage and current of the tank.

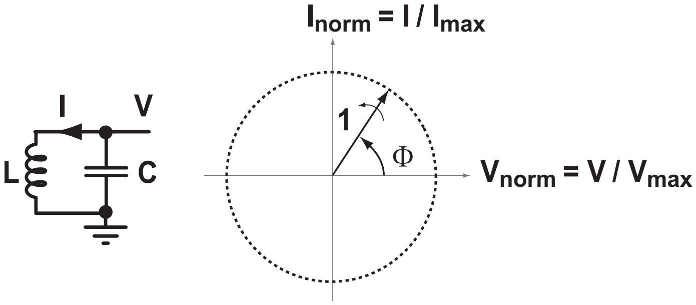


Figure 5-2: Normalized phasor diagram of an ideal LC oscillator.

5.4 Linearized Analysis

To estimate the phase shift of the oscillator phase due to a pulse injection, consider a current impulse that has an area (i.e. charge) of δQ and that is injected into an ideal LC tank, as shown in Figure 5-3. The capacitor will absorb all of the charge, and the voltage will jump by $\delta V = \frac{\delta Q}{C}$.

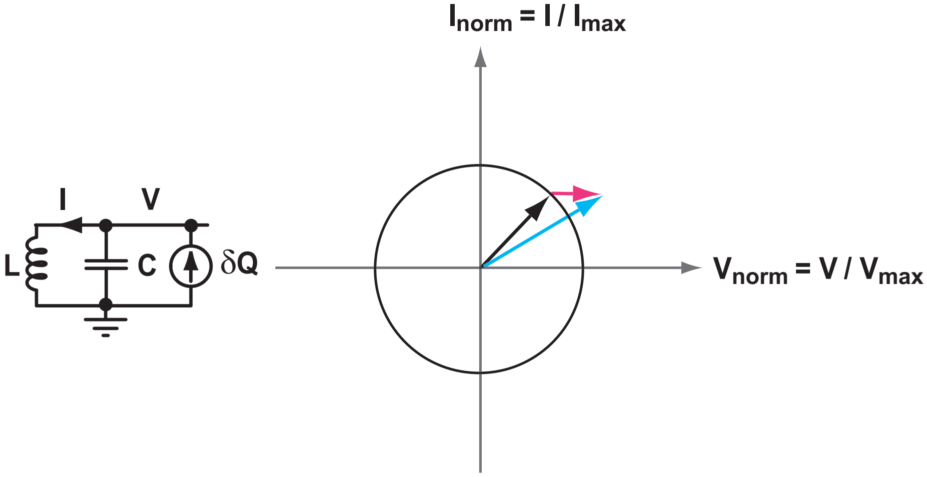


Figure 5-3: Normalized phasor diagram showing a voltage step at the instance of pulse injection.

To take a closer look, Figure 5-4 shows the phasor diagram with three vectors, corresponding to the phase vector just before injection with a unity length and a phase of Φ, the normalized voltage jump due to injection, β, and the resulting phase vector after injection, with a length of A_β and a phase shift due to injection of θ. Notice that θ in the figure has a negative sign because it has a clockwise direction.

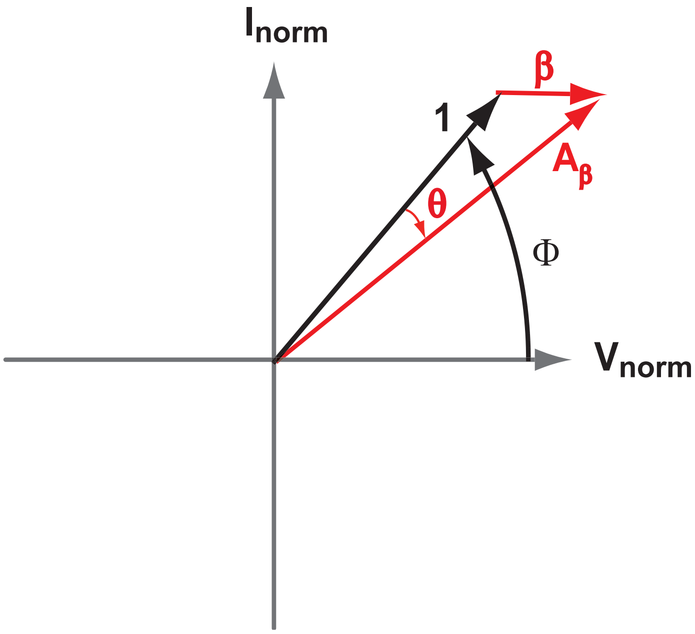


Figure 5-4: Oscillator voltage step due to pulse injection (phase shift derivation).

The phase shift due to injection, θ can be found using trigonometry, and can be derived using the following equations:

$$A_\beta = \sqrt{(\cos\Phi + \beta)^2 + (\sin\Phi)^2} = \sqrt{1 + \beta^2 + 2\beta\cos\Phi} \tag{5.1}$$

$$\sin\theta = \frac{\sin(\Phi)\beta}{A_\beta}. \tag{5.2}$$

For small θ, $\sin\theta \approx \theta$. This is a valid approximation, since by definition the injected pulses are relatively small, and will cause small phase shifts. Note that the result of the approximation of Equation 5.2 was derived through a different and lengthier approach in [45]. Similarly, $\sin\Phi \approx \Phi$. This is a valid approximation for a tuned and injection-locked oscillator, which would have a small phase difference compared to the reference, as would be the case if the proposed technique is used. Finally, for small β, $A_\beta \approx 1$. Using these approximations and by enforcing the counter clock-wise convention for positive angles, Equation 5.2 can be simplified to:

$$\theta \approx -\beta\Phi. \tag{5.3}$$

Equation 5.3 shows a linear relationship between the phase of an oscillator and the phase shift resulting from pulse injection. Through negative feedback, it can be deduced from Equation 5.3 that if a train of pulses is injected into an oscillator, with a close enough frequency, it will lock the oscillator to its phase, causing it to have its peak voltage at the injection time. At steady state, the phase of the oscillator relative to the injecting pulses will be such that the resulting periodic phase shifts balance the offset in frequency. Therefore, for a perfectly tuned oscillator that has no frequency offset, the oscillator phase will eventually be locked such that the injecting pulses occur during the peak voltage of the oscillator output, where the injecting pulses cause no phase shift. Practically, though, the finite pulse width will cause even a perfectly-tuned oscillator to have some residual phase offset to compensate for the slight change in frequency as the oscillator returns back to its equilibrium energy level.

Using the relationship of Equation 5.3 and the value of β, the phase noise of the PILO can be found using the model in [2] as discussed in Section 3.6.1. The value for β in that model is the same as the value of β in Equation 5.3 and is equal to $\frac{\delta V}{V_{max}}$, which can be found using circuits simulation.

5.5 Pulse Injection by Shorting

For the linearized analysis in Section 5.4, it was assumed that a current pulse of a constant area is injected. However, another method can be used to inject current in an LC tank, in which a switch shorts the capacitor of the tank for the period of the injected pulse, as shown in Figure 5-5. Shorting the capacitor will cause it to discharge by passing current to ground(i.e., between its two plates), and its voltage will be reduced by a value proportional to the lost charge.

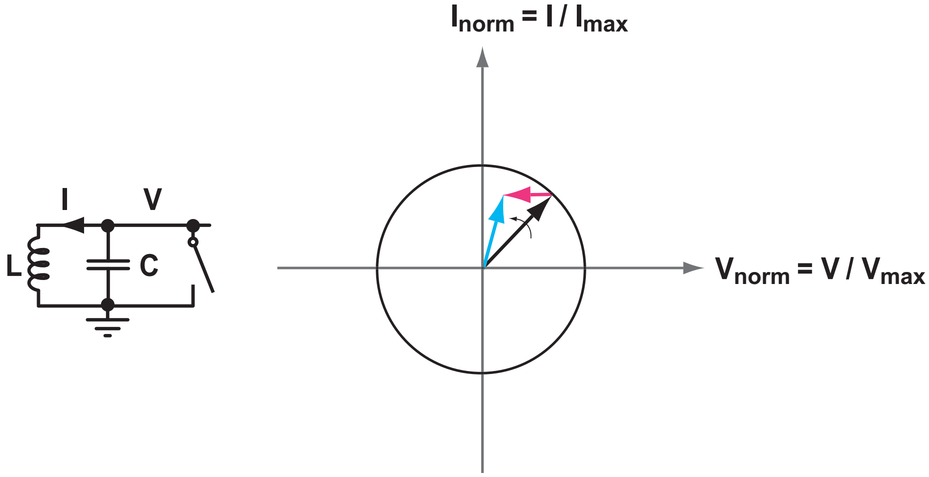


Figure 5-5: Pulse injection using a shorting switch.

This method has a similar linearized analysis to the current pulse method, but it also offers an additional advantage. The practical switch has a finite conductivity, so the shorting current will be proportional to the voltage of the tank at the injection instant. This means that if the switch is shorted when the tank's voltage is zero, no current will pass and the energy of the tank does not change. In contrast, the current injection method will always pump energy into the tank, which it would have to dissipate over some finite time, leading to a residual phase shift, even for a perfectly-tuned oscillator.

A more important advantage of the shorting method is that it can be readily applied to differential oscillators, which are advantageous due to their better immunity to supply and substrate noise compared to single-ended oscillators [45]. In contrast, a current pulse can be injected into one of the sides of the differential oscillator, or converted to a differential pulse and injected into both sides. In the first case, the resulting injection asymmetry would distort the output and it might take longer time to recover from the one-sided injection. In the case of differential current injection, the asymmetry will be smaller, but since active devices of opposite polarity would be used for injection into opposing sides, the symmetry of the injected differential pulse would be limited.

On the other hand, the shorting switch method has a minor disadvantage because the PILO can lock any of its zero crossing points to the reference, which requires additional precautions to remove or avoid such ambiguity[1].

Finally, due to its overwhelming advantages, the shorting switch method was used for the prototype discussed in Chapter 6.

5.6 Impact of Frequency Tuning

Figure 5-6 shows the normalized phasor diagrams for the shorting switch injection. If the shorting pulse occurs when V_{osc} equals zero, it will have no effect (assuming the pulse is very narrow). On the other hand, if the shorting occurs before or after the zero crossing, the oscillator phase will be pushed towards the zero crossing point such that its phase is either advanced or delayed. When the oscillator is injection-locked, this periodic phase shifting will cause the average frequency to match the desired frequency even if the natural frequency of the oscillator is slightly different than the injection source frequency. However, since the injection is occurring at a sub-multiple of the oscillator frequency, the presence of such periodic phase shifting due to frequency offset will lead to a different cycle time for the oscillator during the injection cycle as compared to the free-running cycle time, and thereby induce deterministic jitter and frequency spurs at the output of the PILO.

The tuning technique proposed in Chapter 3 will be used to detect the difference in the cycle periods and tune out the frequency offset of the PILO. A prototype that implements the PILO and the tuning technique is presented next, in Chapter 6, and its measured results are discussed in Chapter 7.

[1] Post-publication note: for example, phase locking the VCO to the reference, using a simple PLL, before enabling the PILO, such that the injected pulse occurs close to the desired zero-crossing phase .

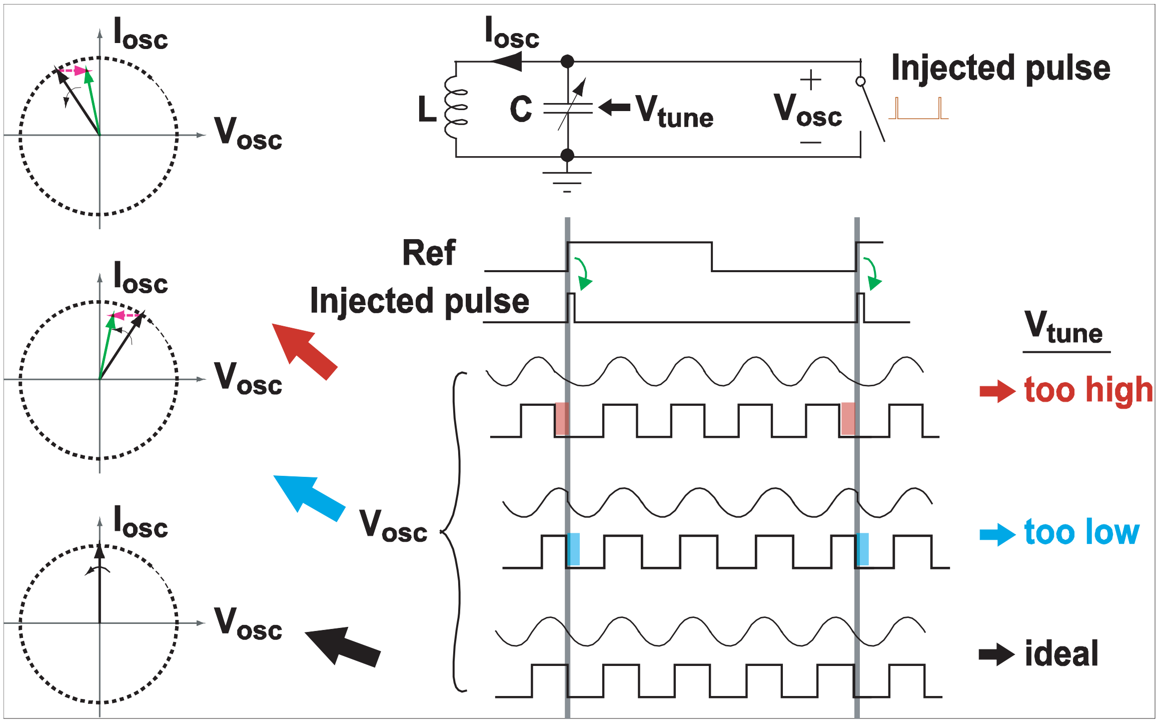


Figure 5-6: Impact of frequency tuning on PILO output.

Chapter 6

PILO Prototype

This chapter presents a PILO prototype that incorporates the proposed detection and tuning technique discussed in Chapter 3. A suitable application for PILOs is where the required jitter performance of a clock multiplier exceeds that of a PLL (using an LC-VCO) with an optimum bandwidth(in which detection and VCO noise contributions are balanced). In that case, and assuming the availability of a clean reference, a PILO architecture can prove to be a good alternative to PLLs since it can reject the two major sources of noise with two decoupled bandwidths instead of one optimum bandwidth that represents the best compromise. The PILO can reject the VCO noise at a relatively high bandwidth (set by the reference frequency and the injection effectiveness) and at the same time reject detection noise at a low bandwidth (set by the tuning loop bandwidth). The target jitter performance of the PILO prototype is set to be less than three hundred femtoseconds in terms of both random and deterministic jitter, given the available VCO, GRO and reference source.

The PILO prototype is similar to the MDLL prototype discussed in Chapter 4, in that they both use the proposed detection and loop locking technique, and have a very similar test board structure that includes an FPGA board, a discrete DAC and an RC filter. Therefore, PILO details that are identical to the already-discussed MDLL prototype will not be repeated, and the focus will be on the unique aspects of the PILO prototype.

After an overview of the architecture of the PILO prototype and its blocks in

Section 6.1, Section 6.2 presents the injected VCO and the pulse generator, and Section 6.3 presents the Enable Logic. Section 6.4 briefly discusses the specifications of the GRO TDC used in the PILO prototype, while Section 6.5 presents the correlator implementation that includes an accumulate-and-dump block to decimate the correlator output. Section 6.6 calculates the expected jitter performance using the model discussed in Section 3.6 and the specifications of various components of the prototype. Finally, Section 6.7 discusses some of the implementation issues.

6.1 Overall Architecture

Figure 6-1 shows a block diagram of the proposed prototype of the continuously tuned PILO structure, which provides details of the feedback circuit used to tune the varactor of the LC oscillator such that the average error value becomes zero. The prototype system consists of a custom 0.13 μm CMOS IC, an FPGA board, and an off-chip DAC with a passive RC filter at its output. The IC contains a pulse generater, injected VCO, Enable Logic (which generates the *Enable* signal that samples the cycle periods), the GRO TDC (which measures the sampled periods), the correlator (which extracts the tuning error), an accumulate-and-dump block (which is combined with the correlator to down-sample its output), and the correlator timing block.

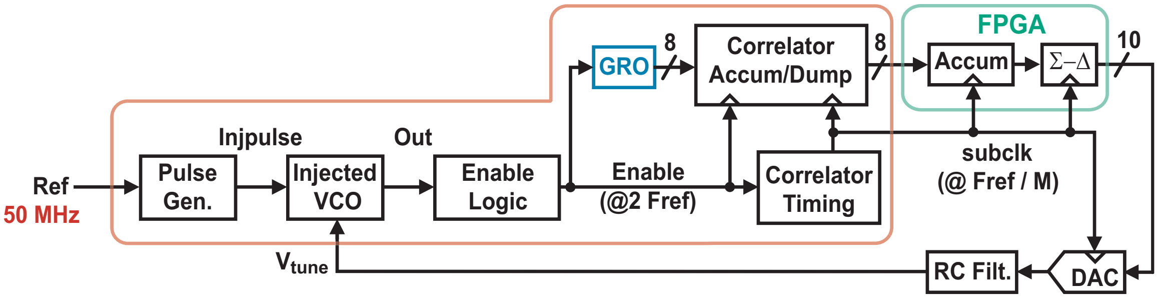


Figure 6-1: Implementation of the Proposed PILO Architecture.

The 8 bit detected error is sent from the IC to the FPGA, which first accumulates the error and then passes the resulting signal into a digital first-order Sigma-Delta modulator in order to reduce the required DAC resolution to 10 bits. The Sigma-

Delta output is fed into the off-chip DAC, and its output is then fed into a passive RC filter (with 500 kHz bandwidth) that feeds into the varactor of the injection-locked LC oscillator. The PILO prototype nominally multiplies a 50 MHz reference frequency to generate a 3.2 GHz output frequency.

The reference signal was generated by an Agilent 8257D ultra low-noise signal source. The signal source generated lower phase noise at higher frequencies, so its sinusoidal output was set to 3.2 GHz, and divided down by an HP pattern generator to a 50 MHz squarewave. The jitter of the divided signal was about 50 fs rms. The sharp edges of the the PRBS source output increased the immunity of the reference signal to added jitter in its way to the PILO on-chip. The PILO IC included a reference divider that allows testing the prototype at different reference frequencies using the same reference input.

One aspect that was different about the PILO test board is that it was electroplated with soft gold (with 12 mircoinch thickness) instead of the immersion gold used in the MDLL test board. The soft gold allowed easier and more robust wirebonding using the ultrasonic wedge bonder that was available for wirebonding.

Figure 6-2 shows the die photo of the PILO IC, which was fabricated in a 0.13 μm CMOS process (IBM mixed-mode process, 8RF-DM). The die active area is approximately 0.4 mm^2 out of a total die size of 1.4 mm × 1.4 mm.

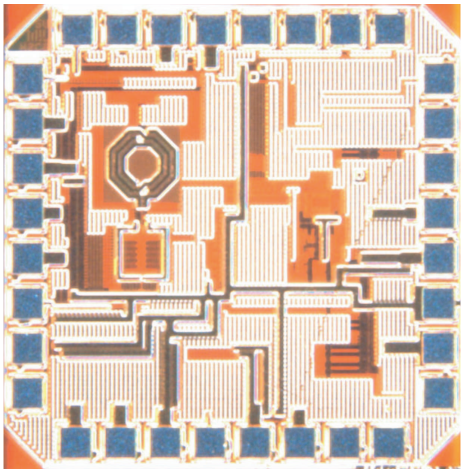


Figure 6-2: PILO IC Die Photo.

Note that the DAC effective resolution for the PILO prototype is 10 bits (instead of 8 bits for the MDLL) to reduce its quantization noise, since the target jitter performance for the PILO is stricter than that of the MDLL. The pole of the RC filter was reduced to 500 kHz (instead of 3 MHz for the MDLL) for the same reason. The FPGA and DAC were clocked by the *subclk* output of the PILO IC, as shown in Figure 6-1.

The setup bits of the PILO IC were controlled using an on-chip bi-directional serial port, which communicated with the FPGA through three pins.

The following sections discuss the prototype blocks in more details.

6.2 Injected VCO and Pulse Generator

Figure 6-3 illustrates the prototype PILO circuit in which injection locking of the LC tank of the oscillator is achieved by periodically shorting the tank with a switch that is driven by a train of narrow pulses. The pulse frequency is set to a sub-multiple of the desired frequency (so that the PILO essentially performs integer-N frequency multiplication), and the width of the pulses is set to be less than about 15 % of the period of oscillation so that the quality factor, Q, of the tank is not severely degraded. To achieve low jitter at the PILO output, the reference source must also have low jitter since its noise will be practically all-passed to the oscillator [2].

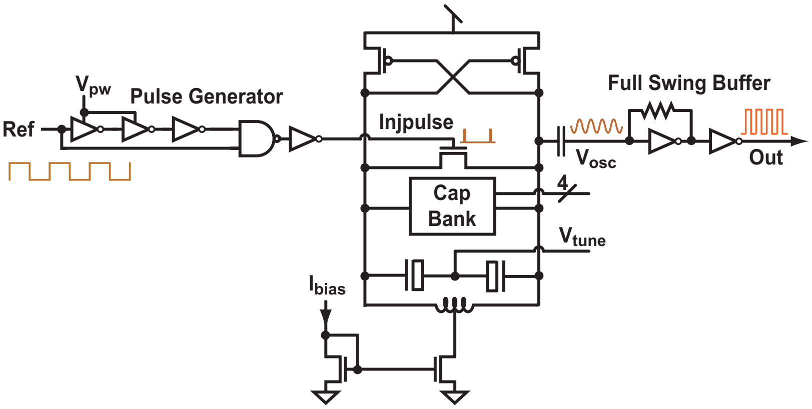


Figure 6-3: Pulse generator and proposed PILO circuit.

As shown in Figure 6-3, the pulse generator consists of digital gates that generate the narrow pulses by performing an AND operation on the reference signal, *Ref*, and a delayed and inverted version of *Ref*. The injected pulse drives an NMOS transistor that shorts the differential output of the VCO. An NMOS is used for switching due to its faster speed and lower resistance compared to PMOS, especially since the common mode voltage of the tank is less than mid-rail.

The VCO is biased from an NMOS current mirror through the center tap of the inductor and includes a varactor made of n-poly/n-well MOSCAP devices and a four-bit MIM capacitor bank that is used to increase the tuning range without a large tuning gain. The VCO-buffer consists of two inverters, the first of which is biased in feedback through a large resistor, which provides a squared version of the VCO output. The VCO was designed by Chun-Ming Hsu [46], a colleague in the same research group as the author.

The supply lines of the first two inverters in the pulse generator have their own power pad. By controlling their supply voltage, V_{pw}, the delay of the inverters and thus the width of the injected pulses is adjusted. The purpose of the pulse width control is to test the effect of the pulse width on the PILO in terms of injection bandwidth, phase noise, and reference spurs.

6.3 Enable Logic

An asynchronous modular divider is used as the Enable Logic for the PILO prototype, as shown in Figure 6-4. Instead of a typical divider output, which has close to 50% duty cycle, the output of the shown divider has a pulse width that nominally corresponds to the output cycle period, and is used as the *Enable* signal. Section 6.3.1 provides additional information about the use of the divider, along with an explanation of a slight modification that allowed disabling of the divider stages that were not used.

Since the *Enable* pulses need to capture the period of specific cycles, the divider is stepped until an *Enable* pulse occurs during the injected cycle. The presence of the

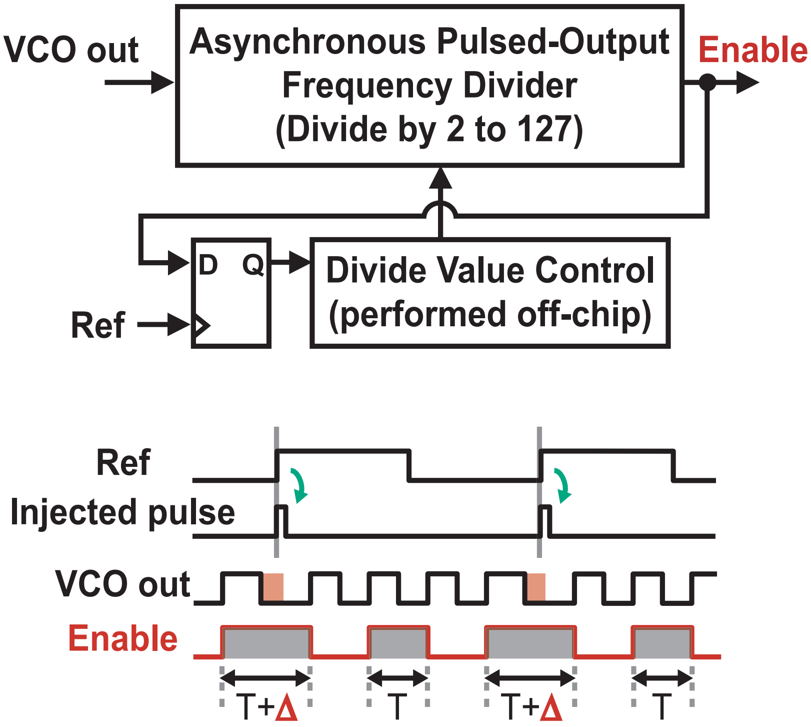

VCO out
Asynchronous Pulsed-Output Frequency Divider (Divide by 2 to 127)
Enable
D
Q
Ref
Divide Value Control (performed off-chip)
Ref
Injected pulse
VCO out
Enable
T+Δ
T
T+Δ
T

Figure 6-4: Enable Logic and associated signals.

Enable pulse is detected by sampling the *Enable* signal using a register that is clocked by the reference signal (or a delayed version of it). A simple algorithm implemented off-chip (using MATLAB) reads the register output and controls the divider division ratio in order to step it. A divider step is accomplished by increasing the division ratio by one for a single divider cycle and then returning it to its original division ratio afterwards. The timing of the stepping operation is assisted by a one-shot monostable circuit (not shown) that receives the stepping command and changes the division ratio for only one divider cycle.

6.3.1 Modular Divider

Figure 6-5 shows the individual modular divider stages that make up the various dividers used in the PILO IC. In addition to the Enable Logic, this divider structure is used in the correlator timing block and the reference divider. Similar to typical dual-modulus divider stages [39, 47], each stage can run as divide-by-2 or divide-by-3 by controlling the p input of the divider stage. The total division ratio is accomplished by the *mod* signals that are passed back by each stage to the one before it to instruct it, if its p is asserted, to swallow one extra cycle of its input during each divider cycle (at the appropriate time). This way, the typical multi-modulus divider can provide a division range that extends from 2^N to 2^{N+1}-1, where N is the number of dual-modulus stages.

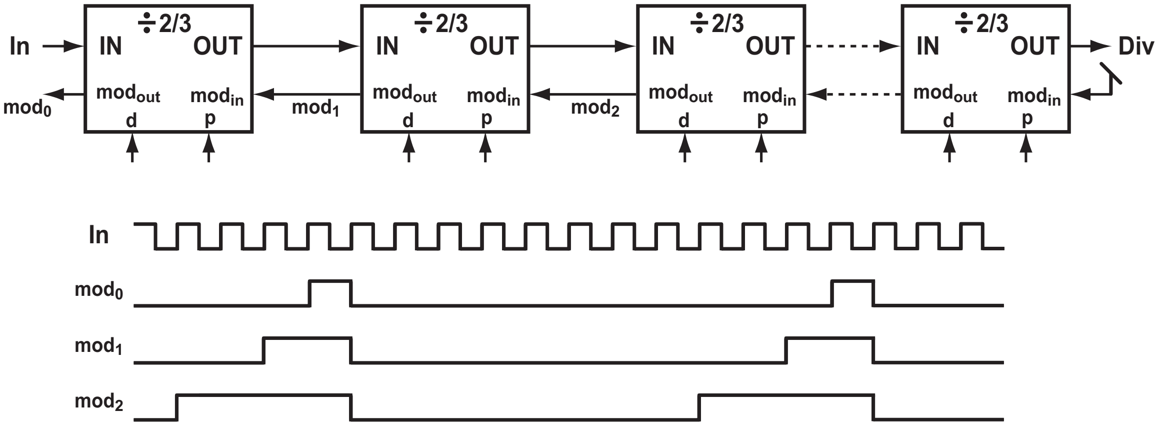


Figure 6-5: Modular divider used in the PILO prototype.

Each *mod* output generates one pulse every divider cycle. The pulse width of each *mod* output is about one cycle of the input to that specific stage, and is retimed by that input. Therefore, as seen in Figure 6-5, mod_0 has a pulse width of one cycle of the input In. Similarly, mod_1 and mod_2 have pulse widths of 2 and 4 periods of In, respectively (assuming all stages are running as divide-by-2). With this property in mind, a multiplexer was used to select one of the *mod* outputs of the first three stages to allow using different *Enable* pulse widths. This is especially useful when the frequency of the VCO is too high for the GRO to be able to measure its short period accurately. Note that due to the internal retiming of the *mod* outputs, mod_0 (which is retimed by In) will have the minimum added jitter compared to all the other *mod* output (and also to the divider output). The inherent retiming, flexible programmability, and robustness of this Enable Logic structure makes it superior to the Enable Logic used for the MDLL prototype (discussed in Section 4.4).

The output of the last divider stage runs at the lowest frequency, i.e. the frequency of Div, so its mod_{in} is always asserted since it only completes one full division cycle every divider cycle. By using an OR gate in the path of any mod signal, the mod_{in} of the stage to the left can be pulled high, thereby making that stage act as if it is the last stage, and hence its output frequency would be the lowest generated by the divider . In doing so, the division ratio can extend all the way down to 2, instead of the typical minimum division ratio for modular dividers (i.e., 2^N)[39].

The divider stages shown in Figure 6-5 have a modification that allows them to be disabled by asserting the input d. The advantage of disabling the stage instead of only bypassing it is the elimination of subharmonic frequencies generated by the bypassed stages which can cause output spurs. In addition, disabling instead of bypassing stages reduces power consumption. Different approaches can be used to disable a divider stage, with varying power consumption and functionality. In the dividers of this prototype, the disabling feature was added by connecting a logic gate in one of the internal paths inside the divider, which makes the input to that stage frozen at one level. Finally, the disabling concept was suggested by Kerwin Johnson, who is a colleague in the same research group as the author.

6.4 Scrambling TDC (GRO)

Similar to the MDLL prototype, the TDC used in the the PILO prototype is a Gated Ring Oscillator (GRO) structure [32] and was also developed by Matt Straayer. The GRO TDC used in the PILO IC has an improved structure that greatly reduces the dead-zone issue encountered in the GRO used in the MDLL prototype. Additionally, its raw resolution at 1.2 V is about 6 ps (instead of 45 ps for the MDLL version). However, the GRO mismatch noise is about 20 ps, which dominates the effective quantization noise, and will be used in noise calculations. The GRO can run at sampling rates up to about 600 MHz and provides an output with a width of 8 bits, allowing it to measure time periods of up to 1.5 ns.

In contrast to the MDLL prototype, the GRO was integrated on-chip with the *Enable* pulse going directly to the *Enable* buffer circuitry and bypassing the set-reset flip-flop that is usually used to generate the *Enable* signal from the reference and input signals (as shown in Figure 4-8).

6.5 Correlator and Accumulator-and-Dump

Since the DAC is implemented off-chip, the accumulator and the Sigma-Delta modulator were implemented on a FPGA off-chip for maximum testing flexibility. However, it was desired to have the option to test the PILO with reference frequencies up to 200 MHz, which might be too fast to be read by the FPGA. Therefore, it was essential to have the ability to lower the data rate before sending it off-chip. The same condition might also be necessary for an all-integrated implementation, since driving a DAC at sampling rates higher than necessary would needlessly complicate its design and increase its power consumption.

Figure 6-6 shows the correlator block, which includes an accumulate-and-dump stage that decimates the correlator output to lower its data rate by up to 8 times. This decimation does not result in any significant aliasing, since the accumulate-and-dump operation is equivalent to a boxcar integrator, followed by sampling. Thus, it low-pass filters high frequencies with a sinc response that has its first null at $\frac{F_s}{L}$, where F_s is the input sampling frequency and L is the downsampling ratio [34]. For the shown figure, $F_s = F_{tdc} = 2F_{ref}$ and $L = 2M$, i.e. the first null of the accumulate-and-dump filter is located at $\frac{F_{ref}}{M}$.

As already implied, the correlator and accumulator in the figure are clocked at the TDC rate, $F_{tdc} = 2F_{ref}$, while the sampling register and the reset of the accumulator are clocked at a rate of $F_{subclk} = F_{ref}/M$. The XNOR gate, inverter, the signal *sign*, and the C_{in} input of the accumulator are used to implement a 2's-complement multiplication [48] by the stream of alternating zeros and ones described in Section 3.5. Not shown in the path of the *sign* signal in the figure is a NAND gate that can disable the flipping of the *sign* signal. Not shown also is an additional XOR gate after the NAND gate, which is used to selectively invert the *sign* signal in order to select the proper order of substraction in the correlator, as discussed in Section 4.6.1.

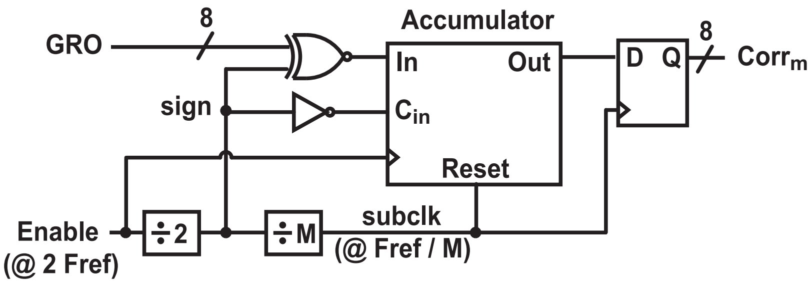


Figure 6-6: Correlator with Accumulate-and-Dump.

The correlator timing circuit is shown in Figure 6-6 in a simplified form of a cascade of divide-by-2 and divide-by-M stages. However, the actual implementation has four divider stages similar to the divider stages of the Enable Logic (which can be individually disabled), with the output of the first stage going to the *sign* signal, and the output of the other three stages going to a multiplexer whose output provides an output clock at various possible rates (depending on the used M). Also, the design includes other circuit blocks , that are not shown, in order to meet the required timing margins, especially the resetting and subsampling signals.

6.6 Calculated Jitter Performance

Using the noise modeling presented in Section 3.6, the phase noise contributions of major noise sources were calculated using MATLAB, as shown in Figure 6-7 [1].

[1] Post-publication note: calculated phase noise and jitter were updated.

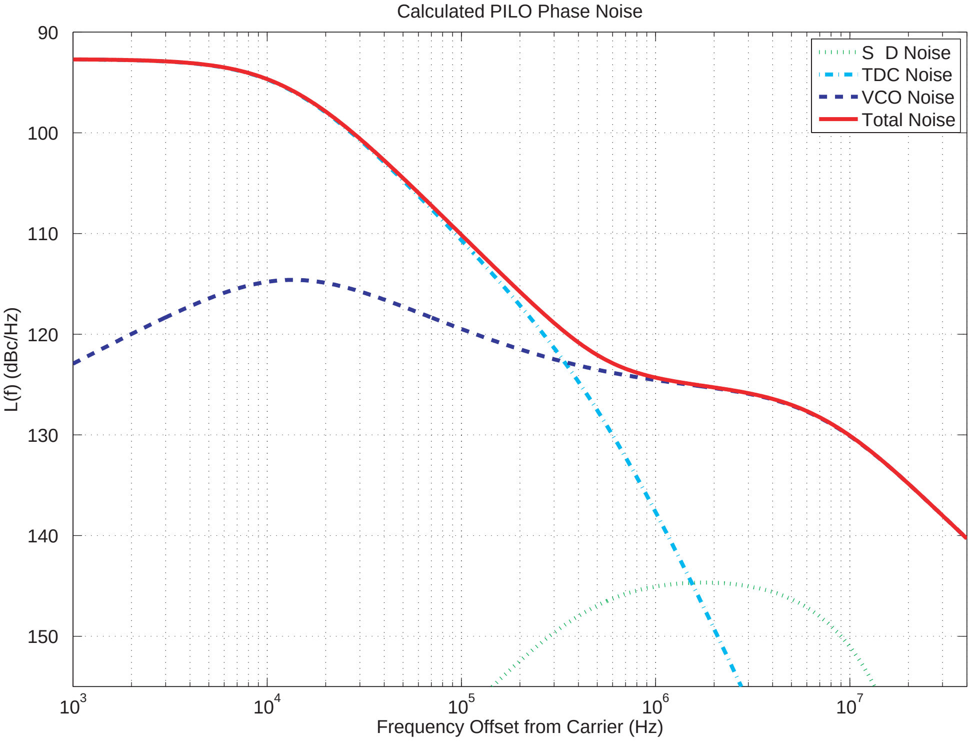


Figure 6-7: Calculated Phase Noise of the PILO Prototype.

The random jitter due to major noise sources and the total random jitter were calculated from the corresponding phase noises using Equation 3.21, and are listed below in Table 6.1. Note that the TDC contributes more jitter than the VCO. However, the TDC jitter contribution can be reduced by lowering the bandwidth, which in the shown case is about 10 kHz.

Table 6.1: PILO Calculated Jitter

	SDM-DAC	TDC	VCO	Total
Calcuated Jitter (fs rms)	11	224	128	258

Listed below are the various parameters used for calculating the PILO phase noise, which were measured or estimated from the prototype and the circuit simulations.

The units for all frequencies are in MHz, and the reference and output frequencies are 50 MHz and 3.2 GHz, respectively. β is estimated to be about 0.6. The phase noise of the free running VCO is about -150 dBc/Hz at 20 MHz offset. However, the phase noise increased to -135 dBc/Hz due to the PILO action, and the reduction in Q due to the periodic shortening of the tank. This increase in phase noise was estimated from the measured phase noise of the PILO [2]. The raw resolution was measured at about 6 ps, however the mismatch noise dominated the TDC noise, and was estimated to be about 20 ps. [3]

Table 6.2: Parameters used for PILO calculations

T_{tdc}	F_{tdc}	F_{acc}	K_{ext}	B	FS	F_{dac}	B_{dac}	F_{lpf}	K_v	$noff$	F_{noff}	$F_{f3corner}$
6 ps	100	50	1	16	1.0	50	10	0.5	80 MHz/V	-135 dBc/Hz	20	0.3

6.7 Implementation Issues

A number of issues were encountered during the prototype testing, some of which are inherent to the architecture and others that can be improved. The following subsections discuss issues related to loop locking and noise coupling.

6.7.1 Loop Locking

As part of the testing routine, the divider of the Enable logic is stepped until the Enable pulse captures the desired output cycle that includes the tuning error, Δ , as discussed in Section 6.3. Additionally, the natural VCO frequency is manually brought close to the desired frequency within the injection-locking range before the tuning loop is enabled. Since the tuning loop is mostly-digital, it is possible to initially

[2] Post-publication note: the increased phase noise at high frequency offset is more plausibly due to jitter added to the reference signal in the buffer and pulse generator paths, as postulated in [Helal, *et al.*, JSSC, May 2009].

[3] Post-publication note: $T_{tdc,raw}$ is used for loop gain calculation, while $T_{tdc,mismatch}$ is used for TDC noise calculation.

tune the frequency of the VCO automatically, either using a PLL configuration, or coarse tuning methods. In one such method, a frequency detector simply compares the output of two counters, one clocked by the reference and the other by the divider output, and infer the frequency offset from the difference between these two counters.

A loop locking issue that is still being investigated is that the loop sometimes locks to a slight frequency offset. In the best performance case, the loop locks such that the spur is about -65 dBc. However, there are instances when the loop locks to locations such that the reference spur level increases to between -55 and -60 dBc, which is more typical, with the worst case being a spur level between -35 to -45 dBc. There are a number of possible culprits, including the ambiguity of the VCO locked edge (discussed in Section 5.5), non-ideal *Enable* pulses, and reference coupling through the opposite edge to the VCO. The last hypothesis was formed due to the observation during one test that the level of the output signal was slightly different around the middle between adjacent injection instances. However, none of the hypotheses have been adequately explored, so further tests are being conducted to determine the causes of the intermittent offset and reduce its effect.

6.7.2 Noise Coupling

Two noise coupling issues were encountered, and remedied, while testing the PILO prototype. One issue was caused on the board, and the other on-chip. The board noise coupling issue was the result of the chip's pad layout. The GRO supply pad was adjacent to the reference input pad, which increased the phase noise of the reference signal due to magnetic coupling between the two wirebonds. One remedy was to extend the reference wirebond an extra distance away from the GRO supply wirebond, since the inductance of the reference wirebond is not as critical as the GRO supply wirebond. Another remedy was to lower the impedance on the board for the reference input line, which would lower the corrupting voltage, since the noise is current-induced. By lowering the impedance of the reference line to about 15 ohms (instead of 50 ohms), the effect of the coupling was significantly reduced. Nonetheless, this issue must be taken into account for future chip implementations.

The on-chip noise coupling issue affected the capacitor bank of the VCO, whose four bits were controlled by the serial register. The source of the noise coupling is the digital output buffer, which shares the same IO supply line as the serial register. The coupling effect was reduced by lowering the IO supply voltage to about 0.9 volts. However, for future chip implementations, care should be taken to use a quiet supply for the serial registers that drive the VCO cap bank. In addition, the control bits should also go through a local buffer that is supplied by the VCO supply or another clean supply.

Chapter 7

Measured Results

Overall jitter is the general performance parameter that the MDLL and PILO prototypes were tested for. However, deterministic jitter is what indicates the effectiveness of the proposed tuning technique in achieving its goal of minimizing deterministic jitter to a level comparable to random jitter, and it will be emphasized for each prototype.

Deterministic jitter will be estimated from reference spurs that is measured using a spectrum analyzer, while random jitter will be estimated from integrated phase noise that is measured using a signal source analyzer. For the MDLL case, overall jitter will be measured using a realtime high performance oscilloscope, since the instrument has a jitter floor less than the expected measurement value. However, for the PILO case, the overall jitter will not be measured directly since the expected jitter falls below most instruments.

A 1-to-2 or 1-to-4 signal splitter was used to measure the prototype jitter using up to four instruments while performing manual adjustments on the prototype (for example the VCO bias current, or the GRO supply voltage if deadzone is a concern). Afterwards, the output of the prototype was connected to each of the individual instruments to measure the performance more accurately without the power attenuation that results from using the power splitter. Performance was measured using different reference frequencies to show its effect on performance and provide sample points to corroborate the noise model developed in Chapter 3.

In addition to jitter performance, power consumption is measured for active blocks of custom IC parts of the prototypes. However, the power consumption of output buffers is not included in the total reported power, since buffers are used only to monitor the functionality of on-chip blocks.

The chapter starts with Section 7.1, which discusses a method to estimate deterministic jitter. Section 7.2 presents the measured results of the MDLL prototype discussed in Chapter 4, proposes a figure of merit (FOM) for MDLLs and other architectures with similar deterministic jitter problem, and compares the measured performance with previous works to illustrate the effectiveness of the proposed techniques. Section 7.3 presents the measured results of the PILO prototype discussed in Chapter 6.

7.1 Estimating Deterministic Jitter

The measured overall jitter includes both random and deterministic jitter components. Since the focus of this thesis is primarily on achieving low deterministic jitter, it is worthwhile to seek a means of measuring it apart from the random component. To do so, it is helpful to look at a frequency domain view of the jitter rather than the time domain view shown by the oscilloscope. In particular, since deterministic jitter occurs periodically at the reference frequency rate, it will show up in the frequency domain as a spurious noise signal with a fundamental frequency offset that corresponds to the reference frequency.

Figure 7-1 shows the phase plot of an imperfectly-tuned MDLL, resulting in a deterministic jitter in the output by Δ , and in periodic steps in the output phase by $\frac{2\pi}{T_{out}}\Delta$, where T_{out} is the ideal output period. Using Fourier analysis, a relationship can be easily derived between Δ and the level of the reference spurs in the output spectrum.

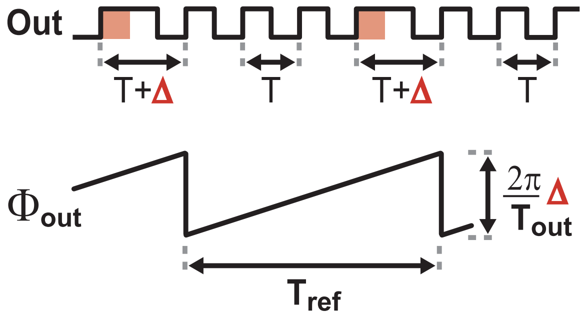


Figure 7-1: Phase plot of an imperfectly-tuned MDLL.

Equation 7.1 provides an expression, based on Fourier analysis, which can be used to estimate deterministic jitter, Δ , from reference spurs in the measured spectrum:

$$\Delta \approx T_{out} \times 10^{\frac{Spur(dBc)}{20}} \tag{7.1}$$

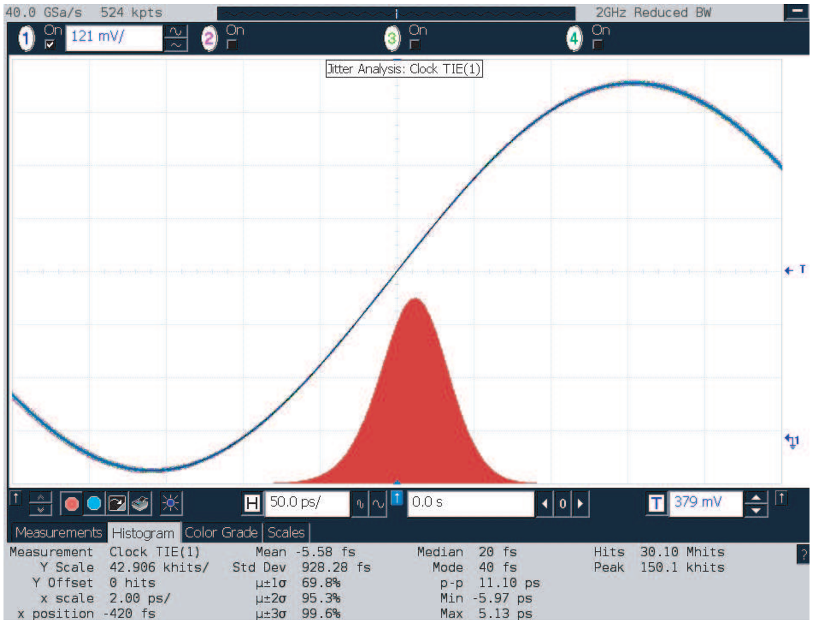
In the above expression, Spur is the level of the reference spur, measured in units of dBc, that corresponds to the difference between the peak of the carrier frequency (at 1.6 and 3.2 GHz for the MDLL and PILO cases, respectively) and the reference spur (at 50 MHz offset, for both the MDLL and PILO cases).

7.2 MDLL Prototype

The prototype was tested at an output frequency of 1.6 GHz and reference frequencies of 12.5, 25 and 50 MHz. However, unless noted, all the discussed results are for the 50 MHz case. At a 1.2V supply, the power consumptions of the MDLL core and the GRO TDC chips (excluding output buffers) are 3.9 and 1.2 mW, respectively. Since the DAC is an off-chip component in this prototype, an estimate of its power and area when integrated on-chip is found by examining recent published work on such components. An 8-bit DAC in a similar 0.13 m process is shown in [49] to consume 3.1 mW with a 100 MHz clock and occupies less than 0.7 mm^2 of area. As for digital functions performed by the FPGA, simulations indicate that they would consume less than 1 mW and occupy less than 0.01 mm^2.

7.2.1 Measured Performance

To demonstrate the sub-picosecond jitter performance of the MDLL prototype, the overall jitter was measured using an Agilent DSO81204B high performance oscilloscope. Figure 7-2 shows a measured overall jitter of 928 fs (rms) and 11.1 ps (peak-to-peak) based on 30.1 million samples and a reference frequency of 50 MHz. Measured overall jitter for the case of reference frequencies equal to 25 and 12.5 MHz were 1.23 and 1.92 ps (rms), respectively.



```
Measurement  Clock TIE(1)      Mean -5.58 fs      Median  20 fs      Hits  30.10 Mhits
    Y Scale  42.906 khits/  Std Dev  928.28 fs      Mode  40 fs      Peak  150.1 khits
   Y Offset  0 hits           μ±1σ  69.8%            p-p  11.10 ps
    x scale  2.00 ps/         μ±2σ  95.3%            Min -5.97 ps
 x position -420 fs           μ±3σ  99.6%            Max  5.13 ps
```

Figure 7-2: Measured Overall Jitter (1.6 GHz MDLL output with a 50 MHz reference).

Even though the physical bandwidth of the oscilloscope is 12 GHz, its bandwidth was set to 2 GHz (implemented internally by DSP) in order to lower the jitter floor of the instrument as much as possible. By doing so, the higher harmonics of the measured input are also rejected. However, the measured jitter is still accurate, since the fundamental carries the same phase noise and jitter information as the full signal. In addition, the jitter floor of the instrument contributes to the measured jitter. To estimate the jitter floor of the oscilloscope, a low-jitter signal source (Agilent E8257D) was used to generate a test signal of the same frequency and amplitude as the prototype. The signal source output jitter was estimated from its phase noise to be less than 100 fs. However, when the jitter of the signal source was measured using the oscilloscope, it was dominated by the instrument's internal jitter and measured about 500 fs. This measured jitter floor agrees with the published specification of the instrument. Using the measured jitter floor, a more accurate estimate of the jitter can be obtained by subtracting the jitter floor from the measured jitter (as variances). In the the MDLL case, the resulting overall jitter was about 800 fs rms.

As shown in Figure 7-3, measurement of the MDLL output with a HP8595E spectrum analyzer reveals a reference spur of -58.3 dBc. Using Equation 7.1, the corresponding deterministic jitter is estimated to be 0.76 ps (peak-to-peak). This result validates the proposed technique's ability to achieve sub-picosecond deterministic jitter.

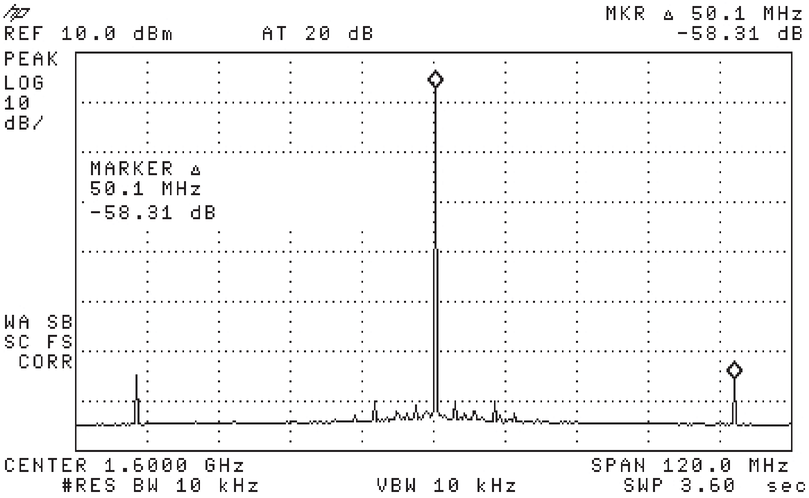
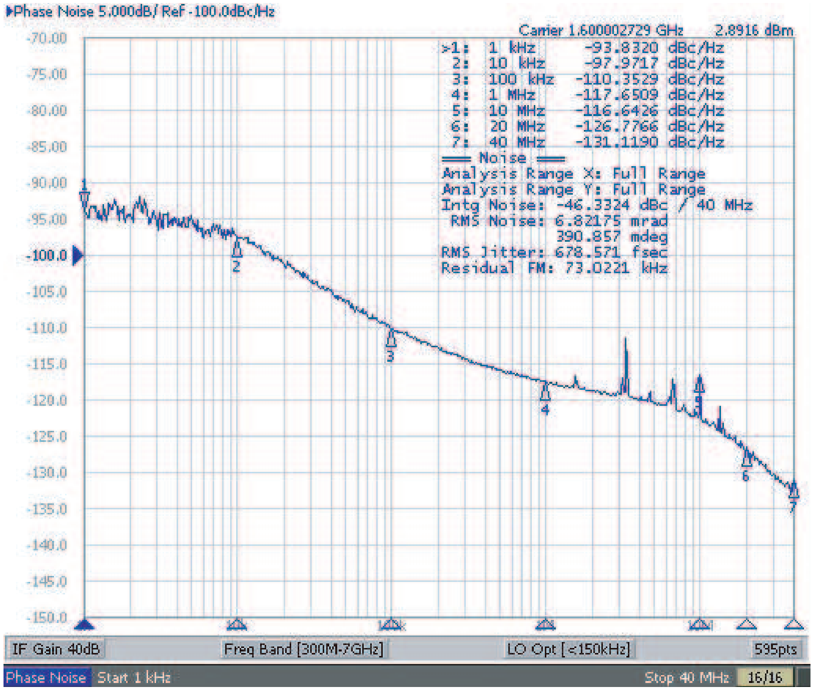


Figure 7-3: MDLL Measured Reference Spur.

As an additional measure of the performance, the phase noise of the MDLL output was measured using an Agilent E5052A signal source analyzer, as shown in Figure 7-4. The random jitter was estimated by integrating the measured phase noise from 1 kHz to 40 MHz, and was found to be 679 fs (rms).



```
Carrier 1.600002729 GHz     2.8916 dBm
>1:  1 kHz      -93.8320 dBc/Hz
 2:  10 kHz     -97.9717 dBc/Hz
 3:  100 kHz   -110.3529 dBc/Hz
 4:  1 MHz     -117.6509 dBc/Hz
 5:  10 MHz    -116.6426 dBc/Hz
 6:  20 MHz    -126.7766 dBc/Hz
 7:  40 MHz    -131.1190 dBc/Hz
══ Noise ══
Analysis Range X: Full Range
Analysis Range Y: Full Range
Intg Noise: -46.3324 dBc / 40 MHz
 RMS Noise: 6.82175 mrad
            390.857 mdeg
RMS Jitter: 678.571 fsec
Residual FM: 73.0221 kHz
```

```
IF Gain 40dB          Freq Band [300M-7GHz]          LO Opt [<150kHz]          595pts
Phase Noise  Start 1 kHz                                          Stop 40 MHz  16/16
```

Figure 7-4: MDLL Measured Phase noise.

7.2.2 Comparison to Previous Work

Table 7.1 compares the performance of the proposed MDLL architecture to previous works. The comparison is limited to edge-multiplexing MDLL architectures, which some sources refer to as recirculating DLLs. The key figure of merit we propose for this type of architecture is the deterministic jitter as estimated from the measured reference spurs, using Equation 7.1. The comparison clearly shows that the proposed architecture achieves the lowest jitter - both random and deterministic - compared to previous works. In addition, the proposed architecture is unique in its highly digital tuning approach as compared to the primarily analog approaches used in previous works.

Table 7.1: MDLL Measured Performance Comparison

	ISSCC 2002 [11]	CICC 2006 [14]	CICC 2006 [15]	[This work]
Output Frequency (GHz)	2.0	1.216	0.176	1.6
Reference Frequency (MHz)	250	64	8	50
Reference Spur (dBc)	-37	-46.5	-70 (estimated)	-58.3
Deterministic Jitter (ps pp) estimated from meas. Spurs (Figure-of-merit)	7.06 (reported DJ:12)	3.89	1.8	0.76
Random Jitter (ps rms) from integrated phase noise	N/A	N/A	5 (1.8 simulated) (1 kHz to 10 MHz)	0.68 (1kHz to 40 MHz)
Overall Jitter	1.62 ps (rms) 13.11(p-p) 25 khits	(@2.16 GHz) 1.6ps(rms) 12.9 (p-p) 12.2 khits	N/A	0.93 ps (rms) 11.1 ps (p-p) 30.1 Mhits
Technology (CMOS)	0.18 μm	0.18 μm	0.18 μm	0.13 μm

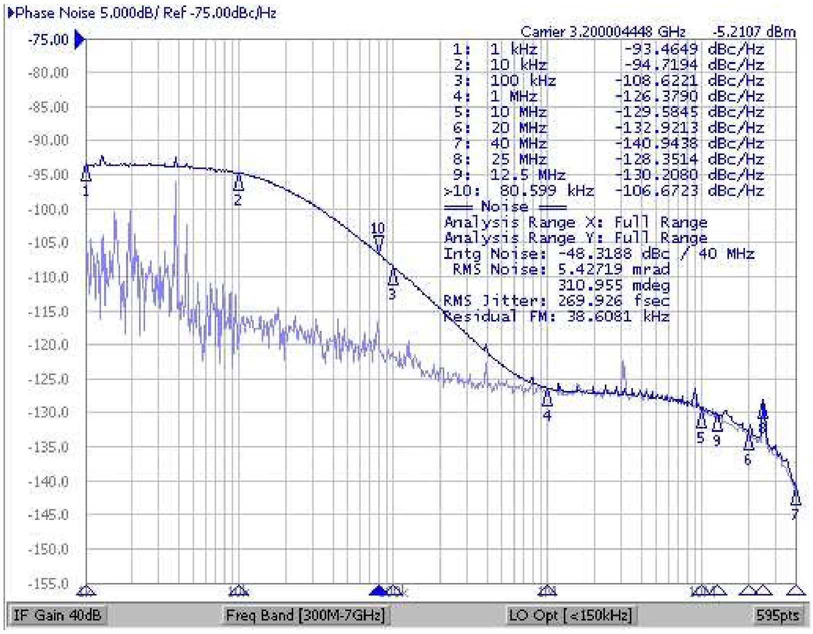
7.3 PILO Prototype

7.3.1 Measured Performance

The prototype was tested at an output frequency of 3.2 GHz and a reference frequency of 50 MHz. The power dissipation of the PILO chip, not counting the output buffers, is 29.2 mW out of which the VCO consumed 21.3 mW and the GRO consumed 6.1 mW. The supplies of the VCO and GRO were set at 1.4 and 1.45 V, respectively, while the supply of the digital and IO pins were set at 0.95 V to reduce the coupling issues discussed in 6.7.

Figure 7-5 displays the measured phase noise from an Agilent E5052A signal source analyzer under the conditions of open-loop tuning of the PILO (where its natural frequency was manually tuned to match the injection frequency), and closed-loop tuning of the PILO using the feedback technique described above.

The integrated phase noise (from 1 kHz to 40 MHz) is 120 fs (rms) for the open-loop tuned case, and 270 fs (rms) for the closed-loop tuned case. As observed in Figure 7-5, the open-loop tuned PILO achieves low phase noise at low frequencies due to the fact that injection-locking leads to suppression of the low frequency VCO phase noise [20]. Note that the open-loop tuned PILO exhibits -10 dB/decade roll-off at lower frequencies, which is the result of suppressing the -30 dB/decade roll-off of the free-running VCO at those frequencies. In contrast, the closed-loop tuned PILO has



```
Carrier 3.200004448 GHz     -5.2107 dBm
  1:  1 kHz             -93.4649 dBc/Hz
  2:  10 kHz            -94.7194 dBc/Hz
  3:  100 kHz          -108.6221 dBc/Hz
  4:  1 MHz            -126.3790 dBc/Hz
  5:  10 MHz           -129.5845 dBc/Hz
  6:  20 MHz           -132.9213 dBc/Hz
  7:  40 MHz           -140.9438 dBc/Hz
  8:  25 MHz           -128.3514 dBc/Hz
  9:  12.5 MHz         -130.2080 dBc/Hz
>10:  80.599 kHz       -106.6723 dBc/Hz
══ Noise ══
Analysis Range X: Full Range
Analysis Range Y: Full Range
Intg Noise: -48.3188 dBc / 40 MHz
 RMS Noise: 5.42719 mrad
            310.955 mdeg
RMS Jitter: 269.926 fsec
Residual FM: 38.6081 kHz
```

Figure 7-5: Measured phase noise for the open-loop and closed-loop tuned PILO.

higher noise due to the measurement noise induced by the GRO TDC. However, this represents a more practical implementation in which the natural oscillator frequency is automatically locked to the desired multiple of the injected frequency. Note that the bandwidth of the closed-loop tuning is digitally set, and lowering it to reduce measurement noise does not result in the increased area or leakage encountered in analog loops. However, the loop bandwidth needs to be set high enough to track thermal variations, and is digitally set to about 10 kHz in this case.

Figure 7-6 shows the -60 dBc reference spur that was measured by an Agilent 8595E spectrum analyzer. Based on Fourier series analysis, the deterministic jitter is estimated at approximately 310 fs (peak-to-peak)[1].

[1] Post-publication note: later results demonstrated estimated random and deterministic jitter of 130 fs rms and 200 fs peak-to-peak, respectively, as reported in [Helal, *et al.*, JSSC, May 2009].

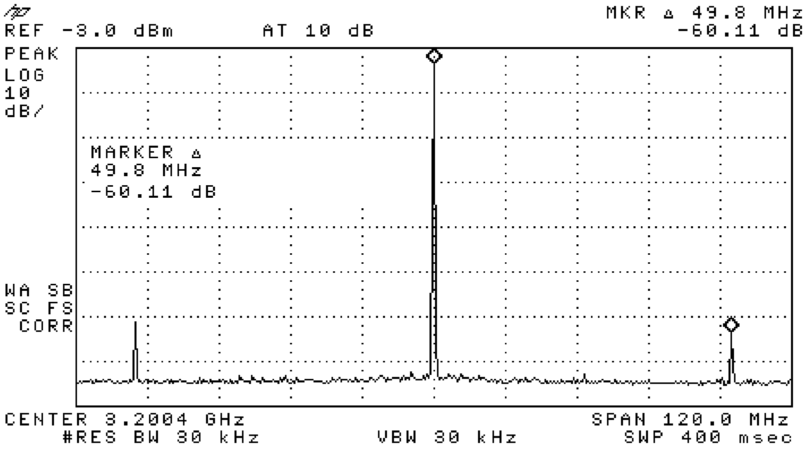


Figure 7-6: Measured spurious performance of the close-loop tuned PILO.

Chapter 8

Conclusion

This chapter starts with Section 8.1 which provides a thesis summary, leading to the discussion of thesis contributions in Section 8.2. Finally, the chapter and the thesis end with Section 8.3 which discusses possible future research.

8.1 Summary

The first motivation for this thesis was to propose an architecture that is suitable for digital ICs, where PLLs traditionally are used for clock multiplication. In contrast to PLLs, the aim of the proposed architecture is to have as much digital content as possible. This is would enable its compatibility with digital design flows, which PLLs do not fit in well due to their large analog content. At the same time, the intended architecture should have jitter performance that meets, if not exceeds, the performance of typical PLLs with comparable power and area requirement. A Multiplying Delay-Locked Loops (MDLL) is a plausible candidate architecture due to its ability to suppress the high phase noise of ring oscillators, which are typically used for digital ICs due to their small area. However, classical MDLL architectures suffer from deterministic jitter due to path mismatch and other analog nonidealities in the tuning loop.

This thesis proposed a digital detection technique that used an available scrambling time-to-digital converter and a digital correlator to practically eliminate the path mismatch problem. In addition, the proposed digital period correlation technique allowed the use of a digital loop filter that only required a digital-to-analog converter and a simple lowpass filter to close the loop. By using a highly-digital tuning path, the analog nonidealities of the classical tuning path were avoided, and the goal of an architecture that is suitable for digital ICs was achieved. Measurement results confirmed the success of achieving the intended performance goal by demonstrating a sub-picosecond overall jitter performance.

The thesis then demonstrated the proposed detection technique in the context of Injection-Locking techniques. The benefit of injection-locking, in terms of suppression of VCO phase noise, is similar to MDLLs, but is mainly applied to LC oscillators. Clock multiplying architectures that use injection locking can suffer from similar challenges as the classical MDLLs. If a subharmonic injection-locked oscillator is tuned using a PLL path, mismatch between the injection and PLL paths causes deterministic jitter. Nonidealities of the analog loop play the same role in exacerbating this problem.

The thesis first proposed a Pulse Injection-locked Loop (PILO) structure that is amenable to the proposed detection and tuning technique, and that also offers a linearized and more intuitive analysis of its operation. A PILO prototype that used the proposed highly-digital tuning path demonstrated random and deterministic jitter of about 300 fs (rms, and peak-to-peak, respectively).

Thus, the proposed detection and tuning technique was applied successfully in architectures that use both ring oscillators and LC oscillators that, while using a clean reference source, demonstrated jitter performance better than typical PLLs of comparable power and area.

8.2 Thesis Contributions

There are four major contributions of this thesis in the context of clock multiplication techniques that uses either ring oscillators or LC oscillators to achieve low jitter performance. These contributions are summarized in the following list:

1. A digital period correlation technique that eliminates path mismatch encountered in typical phase detectors.

2. A highly-digital tuning path that uses the proposed detection technique and eliminates analog nonidealities that degrade the performance of analog tuning paths.

3. An MDLL prototype that uses the proposed tuning technique, which demonstrated sub-picosecond jitter performance. In addition, the highly-digital nature of the prototype makes it a top candidate for digital ICs, due to its easier inclusion in digital design flows that allows its porting between technologies

4. A Pulse Injection-Locked Oscillator structure, whose operation allowed it to use the proposed tuning technique. A PILO prototype demonstrated random and deterministic jitter performance of about 300 fs.

8.3 Future Research

There are several extensions envisioned for this thesis. From a practical perspective, one extension is to implement an all-integrated MDLL architecture that uses a ring oscillator with multi-bit tuning ports and a built-in DAC and lowpass filters. However, from a research point of view, one venue that begs for investigation is a clock multiplier based on an optical PILO, in which the injection source is optical instead of electrical. This idea is further explained in the following subsection.

8.3.1 Optical PILO

This thesis demonstrated the feasibility and effectiveness of using pulses to directly inject-lock LC oscillators. However, the ultimate jitter performance of a PILO architectures strongly depends on the availability of a clean reference source. Mode-Locked Lasers (MLL) can provide such a clean reference source. Mode-locked lasers generate a train of very narrow pulses that exhibit ultra-low jitter in the range of tens of femtoseconds to even sub-femtosecond [50]. Injecting a PILO with the output of a MLL requires the use of a photodiode to convert the optical pulse into electrical current that can shift the phase of the LC oscillator. There are two issues involved with this idea. The first is the need for a photodiode that has a high enough quantum efficiency to generate current pulses that can effectively injection-lock the oscillator. The second issue is the feasibility of using a photodiode, or any other possible integrated optical structure, as a differential shorting switch.

Photodiode Issues

The quantum efficiency of photodiodes integrated on silicon is typically low. However, a PIN photodiode has been recently shown to enjoy a dramatically increased quantum efficiency by using an intentionally-damaged intrinsic silicon region [51]. Such a device can be used as a current injector for a PILO based on a single-ended LC oscillator. However, it can not be used as a shorting switch for differential oscillators for two reasons. The first is that the diode will be forward biased during parts of the cycle, causing a decrease in the quality factor, Q, of the tank and a significant increase in the phase noise and distortion of the output.

The second reason is that the quantum efficiency of the photodiode will be severely reduced at a zero-voltage bias (which is the locking point for injection by a shorting switch). The reduction in quantum efficiency precludes minimizing the forward-bias problem by stacking several diodes in series to reduce the bias voltage on individual diodes. Using the damaged silicon region on its own as a shorting switch, would cause the same problem, since at zero bias there would not be enough electric field to effectively move the generated carriers across the switch. Therefore, investigation is needed to develop or find a suitable device that acts as an optically-triggered electric switch, if a differential optical PILO is desired.

Applications

Recently, there has been a number of studies on generating an electrical clock using optical pulses from MLLs. For example, [52] shows an example of leveraging the extremely low-jitter of a MLL to generate an electric clock with sub-100 fs jitter, which is about the same performance that can be expected from an optical PILO. One possible application is the use in distributed optical clocks [53]. However, a PILO based on a ring oscillator structure would be needed in this case, since distributed clocks are needed for digital chips, which can not afford the area required for the inductors of LC oscillators.

The other application is the use of the RF output of an optical PILO to synchronize other lasers. RF output with jitter in the range of tens of femtoseconds or less can be generated from a MLL output using discrete bandpass filters and electronic amplifiers. However, such a method is not programable, since a bandpass filter is usually designed for a single frequency. In addition, the filters are expensive and some needed frequencies might not be readily available off the shelf. On the other hand, an optical PILO can be programmed to any multiple of the MLL repetition rate within the frequency range of the LC oscillator (which is typically 10 to 20 % of the center frequency). Such an application is not suitable for state-of-the-art laser research. However, it might be used in commercial applications, where a jitter of tens to hundreds of femtoseconds is acceptable.

Bibliography

[1] R. Farjad-Rad, W. Dally, H.-T. Ng, R. Senthinathan, M.-J. Lee, R. Rathi, and J. Poulton, “A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips,” *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1804–1812, Dec. 2002.

[2] S. Ye, L. Jansson, and I. Galton, “A multiple-crystal interface PLL with VCO realignment to reduce phase noise,” *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1795–1803, Dec. 2002.

[3] A. Waizman, “A delay line loop for frequency synthesis of de-skewed clock,” in *Proc. IEEE International Solid-State Circuits Conference, ISSCC*, 1994, pp. 298–299.

[4] P. Maulik and D. Mercer, “A DLL-Based Programmable Clock Multiplier in 0.18-μm CMOS With -70 dBc Reference Spur,” *IEEE J. Solid-State Circuits*, vol. 42, no. 8, pp. 1642–1648, Aug. 2007.

[5] Q. Du, J. Zhuang, and T. Kwasniewski, “A Low-Phase Noise, Anti-Harmonic Programmable DLL Frequency Multiplier With Period Error Compensation for Spur Reduction,” *IEEE Trans. Circuits Syst. II*, vol. 53, no. 11, pp. 1205–1209, Nov. 2006.

[6] H. Ahmed, C. DeVries, and R. Mason, “A digitally tuned 1.1 GHz subharmonic injection-locked VCO in 0.18 μm CMOS,” in *Proc. 29th European Solid-State Circuits Conference, ESSCIRC*, 2003, pp. 81–84.

[7] F. M. Gardner, *Phaselock Techniques*, 3rd ed. Wiley, 2005.

[8] M. Perrott, M. Trott, and C. Sodini, “A modeling approach for $\Sigma-\Delta$ fractional-N frequency synthesizers allowing straightforward noise analysis,” *IEEE J. Solid-State Circuits*, vol. 37, no. 8, pp. 1028–1038, Aug. 2002.

[9] B. Kim, T. Weigandt, and P. Gray, “PLL/DLL system noise analysis for low jitter clock synthesizer design,” in *Proc. IEEE International Symposium on Circuits and Systems, ISCAS*, vol. 4, 1994, pp. 31–34.

[10] J. McNeill, “Jitter in ring oscillators,” *IEEE J. Solid-State Circuits*, vol. 32, no. 6, pp. 870–879, Jun. 1997.

[11] R. Farjad-Rad, W. Dally, H.-T. Ng, J. Poulton, T. Stone, R. Rathi, E. Lee, D. Huang, and R. Nathan, “A 0.2-2 GHz 12 mW multiplying DLL for low-jitter clock synthesis in highly-integrated data communication chips,” in *Proc. IEEE International Solid-State Circuits Conference, ISSCC*, vol. 1, 2002, pp. 76–77.

[12] S. Ye, L. Jansson, and I. Galton, “A multiple-crystal interface pll with vco realignment to reduce phase noise,” in *Proc. IEEE International Solid-State Circuits Conference, ISSCC*, vol. 1, 2002, pp. 78–447.

[13] G.-Y. Wei, J. Stonick, D. Weinlader, J. Sonntag, and S. Searles, “A 500MHz MP/DLL clock generator for a 5Gb/s backplane transceiver in 0.25 μm CMOS,” in *Proc. IEEE International Solid-State Circuits Conference, ISSCC*, 2003, pp. 464–465.

[14] Q. Du, J. Zhuang, and T. Kwasniewski, “An Anti-Harmonic Locking, DLL Frequency Multiplier with Low Phase Noise and Reduced Spur,” in *Proc. IEEE Custom Integrated Circuits Conference, CICC*, 2006, pp. 761–764.

[15] P. Maulik and D. Mercer, “A 150MHz-400MHz DLL-Based Programmable Clock Multiplier with -7OdBc Reference Spur in 0.18um CMOS,” in *Proc. IEEE Custom Integrated Circuits Conference, CICC*, 2006, pp. 757–760.

[16] A. E. Siegman, *Lasers.* University Science Books, 1986.

[17] B. V. D. Pol, "Forced oscillator in a circuit with nonlinear resistance," *Phil. Mag.*, vol. 3, pp. 65–80, Jan. 1927.

[18] R. Adler, "A Study of Locking Phenomena in Oscillators," *Proc. IRE*, vol. 34, no. 6, pp. 351–357, Jun. 1946.

[19] K. Kurokawa, "Injection locking of microwave solid-state oscillators," *Proc. IEEE*, vol. 61, no. 10, pp. 1386–1410, Oct. 1973.

[20] B. Razavi, "A study of injection locking and pulling in oscillators," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1415–1424, Sep. 2004.

[21] E. Shumakher and G. Eisenstein, "On the noise properties of injection-locked oscillators," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 5, pp. 1523–1537, May 2004.

[22] K. Yamamoto and M. Fujishima, "A 44-μW 4.3-GHz injection-locked frequency divider with 2.3-GHz locking range," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 671–677, Mar. 2005.

[23] P. Mayr, C. Weyers, and U. Langmann, "A 90GHz 65nm CMOS Injection-Locked Frequency Divider," in *Proc. IEEE International Solid-State Circuits Conference, ISSCC*, 2007, pp. 198–596.

[24] P. Kinget, R. Melville, D. Long, and V. Gopinathan, "An injection-locking scheme for precision quadrature generation," *IEEE J. Solid-State Circuits*, vol. 37, no. 7, pp. 845–851, Jul. 2002.

[25] J. Lee and M. Liu, "A 20Gb/s Burst-Mode CDR Circuit Using Injection-Locking Technique," in *Proc. IEEE International Solid-State Circuits Conference, ISSCC*, 2007, pp. 46–586.

[26] S. Kudszus, M. Neumann, T. Berceli, and W. Haydl, “Fully integrated 94-GHz subharmonic injection-locked PLL circuit,” *IEEE Microw. Guided Wave Lett.*, vol. 10, no. 2, pp. 70–72, Feb. 2000.

[27] F. Plessas and G. Kalivas, “Locking techniques for RF oscillators at 5-6 GHz frequency range,” in *Proc. 10th IEEE International Conference on Electronics, Circuits and Systems, ICECS*, vol. 3, 2003, pp. 986–989.

[28] M. White, D. Lampe, F. Blaha, and I. Mack, “Characterization of surface channel CCD image arrays at low light levels,” *IEEE J. Solid-State Circuits*, vol. 9, no. 1, pp. 1–12, Jan. 1974.

[29] C. Enz and G. Temes, “Circuit techniques for reducing the effects of op-amp imperfections: autozeroing, correlated double sampling, and chopper stabilization,” *Proc. IEEE*, vol. 84, no. 11, pp. 1584–1614, Nov. 1996.

[30] I. Nissinen, A. Mantyniemi, and J. Kostamovaara, “A cmos time-to-digital converter based on a ring oscillator for a laser radar,” in *Proc. 29th European Solid-State Circuits Conference, ESSCIRC*, 2003, pp. 469–472.

[31] R. Staszewski, J. Wallberg, S. Rezeq, C.-M. Hung, O. Eliezer, S. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, N. A10 Barton, M.-C. L. A. M.-C. Lee, P. Cruise, P. A12 Cruise, M. Entezari, M. A13 Entezari, K. Muhammad, K. A14 Muhammad, and D. Leipold, D. A15 Leipold, “All-digital PLL and transmitter for mobile phones,” *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2469–2482, Dec. 2005.

[32] B. M. Helal, M. Z. Straayer, G.-Y. Wei, and M. H. Perrott, “A Low Jitter 1.6 GHz Multiplying DLL Utilizing a Scrambling Time-to-Digital Converter and Digital Correlation,” in *Proc. IEEE Symposium on VLSI Circuits*, Jun. 2007, pp. 166–167.

[33] ——, "A Highly-Digital MDLL-based Clock Multiplier that leverages a Self-Scrambling Time-to-Digital Converter to Achieve sub-picosecond Jitter Performance," *IEEE J. Solid-State Circuits*, to be published, Apr. 2008.

[34] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*, 2nd ed. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1998.

[35] T. C. Weigandt, "Low-Phase-Noise, Low-Timing-Jitter Design Techniques for Delay Cell Based VCOs and Frequency Synthesizers," Ph.D. dissertation, University of California, Berkeley, Spring 1998.

[36] V. Stojanovic, "Channel-limited high-speed links: Modeling, analysis and design," Ph.D. dissertation, Stanford University, September 2004.

[37] L. Dai and R. Harjani, "Comparison and analysis of phase noise in ring oscillators," in *Proc. IEEE International Symposium on Circuits and Systems, ISCAS*, vol. 5, May 2000, pp. 77–80.

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

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

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

[41] E. Alon, V. Stojanovic, and M. Horowitz, "Circuits and techniques for high-resolution measurement of on-chip power supply noise," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 820–828, Apr. 2005.

[42] X. Zhang, B. Aliener, and A. Daryoush, "A study of subharmonic injection locking for local oscillators," *IEEE Microw. Guided Wave Lett.*, vol. 2, no. 3, pp. 97–99, Mar. 1992.

[43] H. Moyer and A. Daryoush, "A unified analytical model and experimental validations of injection-locking processes," *IEEE Trans. Microw. Theory Tech.*, vol. 48, no. 4, pp. 493–499, Apr. 2000.

[44] S. Verma, H. Rategh, and T. Lee, "A unified model for injection-locked frequency dividers," *IEEE J. Solid-State Circuits*, vol. 38, no. 6, pp. 1015–1027, Jun. 2003.

[45] A. Hajimiri and T. H. Lee, *The Design of Low Noise Oscillators.* Kluwer Academic Publishers, 1999.

[46] C.-M. Hsu, M. Straayer, and M. Perrott, "A Low-Noise Wide-BW 3.6GHz Digital ΔΣ Fractional-N Frequency Synthesizer with a Noise-Shaping Time-to-Digital Converter and Quantization Noise Cancellation," in *Proc. IEEE International Solid-State Circuits Conference, ISSCC*, to be presented, Feb. 2008.

[47] M. H. Perrott, "6.976, High Speed Communication Circuits and Systems, Spring 2003, OCW." [Online]. Available: http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-976High-Speed-Communication-Circuits-and-SystemsSpring2003/LectureNotes/index.htm

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

[49] N. Ghittori, A. Vigna, P. Malcovati, S. D'Amico, and A. Baschirotto, "1.2-V Low-Power Multi-Mode DAC+Filter Blocks for Reconfigurable (WLAN/UMTS, WLAN/Bluetooth) Transmitters," *IEEE J. Solid-State Circuits*, vol. 41, no. 9, pp. 1970–1982, Sep. 2006.

[50] D. Yoshitomi, Y. Kobayashi, M. Kakehata, H. Takada, and K. Torizuka, "Synchronization of Ti:sapphire and Cr:forsterite mode-locked lasers with 100-

attosecond precision by optical-phase stabilization," *Optics Express*, vol. 14, no. 13, pp. 6359–6365, Jun. 2006.

[51] M. W. Geis, S. J. Spector, M. E. Grein, R. T. Schulein, J. U. Yoon, D. M. Lennon, S. Deneault, F. Gan, F. X. Kärtner, and T. M. Lyszczarz, "CMOS-Compatible All-Si High-Speed Waveguide Photodiodes With High Responsivity in Near-Infrared Communication Band," *IEEE Photon. Technol. Lett.*, vol. 19, no. 3, pp. 152–154, Mar. 2007.

[52] J. Kim, F. X. Kärtner, and M. H. Perrott, "Femtosecond synchronization of radio frequency signals with optical pulse trains," *Optics Letters*, vol. 29, no. 17, pp. 2076–2078, Sep. 2004.

[53] M. Karami, A. Afzali-Kusha, R. Faraji-Dana, and M. Rostami, "Quantitative Comparison of Optical and Electrical H, X, and Y clock Distribution Networks," in *Proc. IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, 2007, pp. 488–489.