Title
A Multi-loop Calibration-free Phase-locked Loop (PLL) for Wideband Clock Generation

Author
Yang, Dihang

Publication Date
2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

A Multi-loop Calibration-free Phase-locked Loop (PLL) for Wideband Clock Generation

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Dihang Yang

2019

ABSTRACT OF THE DISSERTATION

A Multi-loop Calibration-free Phase-locked Loop (PLL) for Wideband Clock Generation

by

Dihang Yang
Doctor of Philosophy in Electrical Engineering
University of California, Los Angeles, 2019
Professor Asad. A. Abidi, Chair

In a wide-band RF system, the RF channel is located within 50 MHz to 9 GHz. A high-frequency resolution phase-locked loop (PLL) with 100% tuning range oscillator is the core to generate the RF carrier frequency which covers such a wide range. The phase noise and spurs of the PLL are required to be low to avoid degrading RF system performance. A PLL applies $\Sigma\Delta$ modulation to increases its resolution and is known as a fractional-N PLL, but $\Sigma\Delta$ modulation introduces considerable quantization noise into the loop. The nonlinearity of the PLL also converts part of the noise into fractional-N spurs. Noise cancellation is usually applied to eliminate this quantization noise. Calibration, often with long settling time, is necessary to maintain cancellation efficiency. Power intensive calibration is also required to notch spurious tones.

In this thesis, we first investigate the delay-locked loop (DLL) and attempt to use DLL to replace PLL as an RF frequency synthesizer. An LTI model of DLL is established, which indicates the limitation of DLL as a high-performance synthesizer. Then, the thesis focuses on PLL again. A calibration-free triple-loop PLL is introduced. The merits of heterodyne PLL are rediscovered, which applies a mixer in the loop to translate the VCO frequency to a low-frequency feedback signal. By implementing the harmonic mixing concept, the designed prototype effectively reduces the pulling risk of a traditional heterodyne PLL, allowing it to be integrated on a single chip. This PLL provides higher-order noise filtering and can naturally reduce fractional-N PLL noise and spurs. An analytical model for this PLL is also

presented, which allows us to fully appreciate this PLL and optimize the loop design. After this, a sub-sampling PLL-based low-noise frequency extender is introduced, which increases the tuning range of an oscillator from 30% to 100%, and requires only a small chip area. By combining the triple-loop PLL and the frequency extender, a synthesizer which can support a wideband radio system is achieved.

The dissertation of Dihang Yang is approved.

Eric Hudson

Chih-Kong Ken Yang

Sudhakar Pamarti

Asad. A. Abidi, Committee Chair

University of California, Los Angeles

2019

To my family

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor, Prof. Abidi, for his continuous support of my Ph.D. study and related research, and for his patience, motivation, and immense knowledge. His way of appreciating knowledge not only inspires me in the academic research in my own field, but also assists me to absorb knowledge in other fields. His circuit design philosophy, "A simple circuit solution with solid theory! ", will accompany me throughout my entire career and encourage me to keep searching for net solutions. I cannot imagine of having a better advisor and mentor for my Ph.D. study.

My sincere thanks also go to Dr. Hooman Darabi, who provided me with the opportunity to join his team, and gave me invaluable guidance on my Ph.D. project. Without his precious support, it would not have been possible to conduct this research.

I would like to express gratitude to the rest of my thesis committee: Prof. Sudhakar Pamarti, Prof. Chih-Kong Ken Yang, and Prof. Eric Hudson, for their insightful comments and encouragement, but also for their valuable questions, which prompted me to widen my research from various perspectives.

My special thanks go to Dr. Ningyi Wang and Dr. Xiaojing Xu who are my elder brother and sister, my friends, and my colleagues at Broadcom. They give me helpful suggestions on my life, my work, and also my dissertation.

I also thank my friends and colleagues Dr. Wenlong Jiang, Dr. Long Kong, Dr. Xiao Wang, Dr. Hao Xu, and other UCLA peers for their support and insightful discussions. I am also grateful to Dr. David Murphy, Dr. Hao Wu, and other Broadcom engineers for meaningful discussions.

I am sincerely grateful to the Broadcom Foundation, for the two-year Broadcom Fellowship, which enabled me to concentrate on my research without needing to focus on making a living.

I would also like to thank my parents and my parents-in-law for supporting me spiritually throughout my Ph.D. study. I am such a lucky boy to be loved by them. Last, but not least,

I would like to thank my wife for her love, her encouragement, and her support for my study. The Ph.D. study is full of uncertainty and failures which tortures my soul and damps down my passion. Whenever I think about her smile, I can get back my courage and enthusiasm for overcoming the challenges. For me, a Ph.D. study is not only a painful procedure for creating new knowledge but also a beautiful period to understand the meaning of love.

VITA

2009-2013 B.S. (Electrical Engineering), Zhejiang University, Hang Zhou, China

2013-2015 M.S. (Electrical Engineering), UCLA, Los Angeles, CA

2015-2019 Ph.D. (Electrical Engineering), UCLA, Los Angeles, CA

PUBLICATIONS

Dihang Yang, Asad A. Abidi, Hooman Darabi, Hao Xu, David Murphy, Hao Wu, Zhaowen Wang,“ A Calibration-Free Triple-Loop Bang-Bang PLL Achieving 131 fs rms Jitter and -70 dBc Fractional Spurs,” *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, San Francisco, CA, USA, 2019, pp. 266-268.

Alireza Yousefi, Dihang Yang, Asad A. Abidi, Dejan Markovic,“A distance-immune low-power 4-Mbps inductively-coupled bidirectional data link,” *2017 Symposium on VLSI Circuits*, Kyoto, 2017, pp. C60-C61.

CHAPTER 1

Introduction

A frequency synthesizer typically consists of a phase-locked loop (PLL) and a voltage controlled oscillator (VCO) as shown in Fig. 1.1. By selecting the dividing modulus, the loop locks the VCO to the desired frequency, which is known as the carrier. As shown in Fig. 1.2, in the receiver, the carrier downconverts the desired RF channel to the baseband. In the transmitter, the carrier upconverts the baseband signal to an RF channel. The following power amplifier (PA) then transmits it to the outside world.

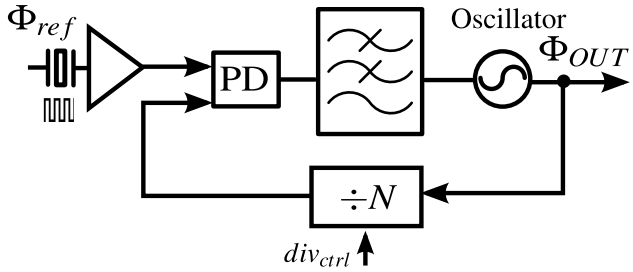


Figure 1.1: A conventional PLL as a synthesizer.

1.0.1 Phase noise, spurs, and their effects

Phase noise is a continuous stochastic process indicating random accelerations and decelerations in phase as an oscillator orbits at a nominally constant frequency in steady-state, as shown in Fig. 1.3.

A pure sinewave $A\cos(\omega t)$ is modulated by the phase noise $\theta_{PN}(t)$ and becomes $A\cos(\omega t + \theta_{PN}(t))$. In the frequency domain, the original impulse of a sinewave is spread out by the random drift of phase, as presented in Fig. 1.4. Since the noise only modulates the phase, the sideband consists of purely PM noise. In both LC oscillator and ring oscillator, the phase

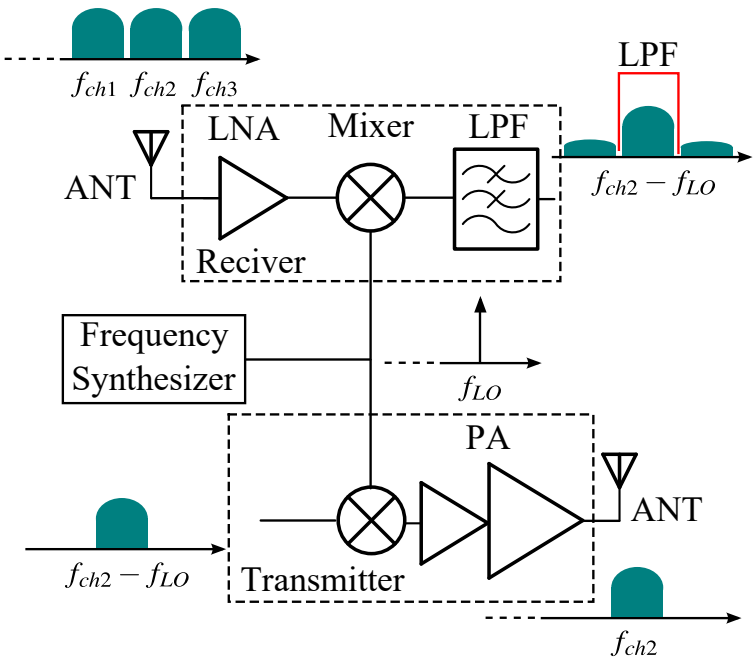


Figure 1.2: In the RF system, the synthesizer generates the carrier to drive the mixer of the transmitter and the receiver.

noise is defined as follows:

$$\mathcal{L}(f) = \frac{K_w}{2(Qf)^2} \tag{1.1}$$

Where K_w is the noise coefficient, which is determined by the device. Q is the quality factor of an oscillator, which is either the quality factor of LC tank in an LC oscillator or $1/\sqrt{2}$ in a ring oscillator.

As shown in Fig. 1.5, when the noisy carrier convolves with the blocker, it skirts into the desired signal and thereby corrupts it. This phenomenon is known as " reciprocal mixing " .

Due to loop non-ideality, the synthesizer outputs PM spurs. As shown in Fig. 1.6, these spurs directly translate the sideband channel to baseband, which degrades the SNR. One of the main challenges of modern synthesizer design is low noise and spurs to reduce " reciprocal mixing ".

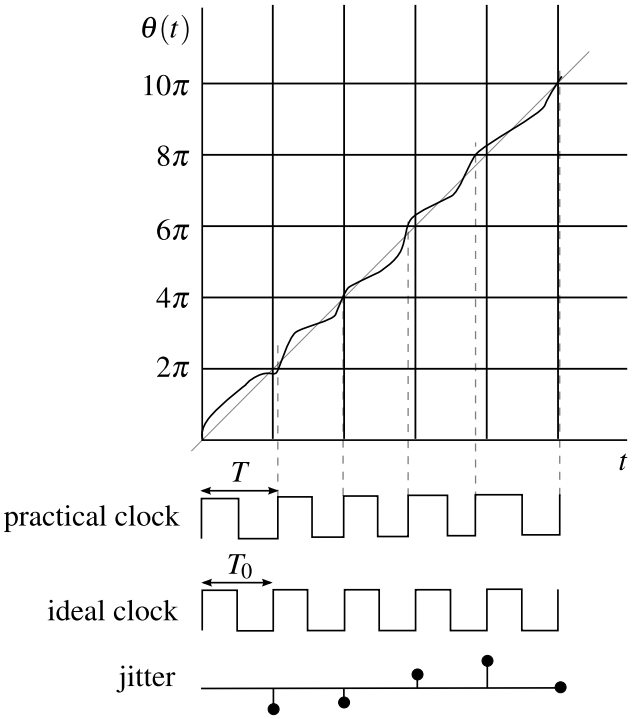


Figure 1.3: The orbit of phase. Due to jitter, practical clock deviate from the ideal clock.

1.0.2 Relation between jitter and phase noise

Modern communication systems implement complicated QAM constellation to increase the data rate. For instance, IEEE 802.11 ax (Wi-Fi 6), the latest iteration of Wi-Fi needs 1024 QAM. In the absence of non-idealities, the signals reside in their designated constellation points. However, in practice, the random phase noise rotates the signal and causes data error, as shown in Fig. 1.7. RMS Jitter σ_τ or integrated phase noise is the metric that quantifies the amount of rotation. They must be controlled to meet the specification of the communication standard. This is another challenge in synthesizer design.

Jitter arises from sampling the orbit at the zero crossing of the waveform, as shown in Fig. 1.3. The clock period T is defined as the interval between successive zero crossings of the waveform at the same transition direction. In the presence of phase noise, T is a set of discrete random variables. Periodic jitter is defined as the standard deviation of this discrete sequence around its mean value, which is the ideal clock period T_0. Jitter is phase noise with the dimensionality of time, $\tau = T_0 \cdot \theta_{PN}/2\pi$. The sampling at the zero-crossing folds back

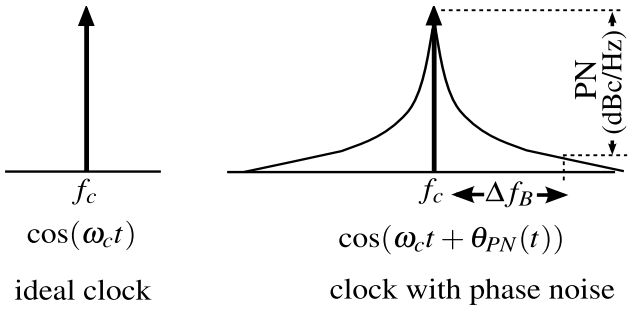


Figure 1.4: Phase noise modulates an ideal single tone to a spectrum.

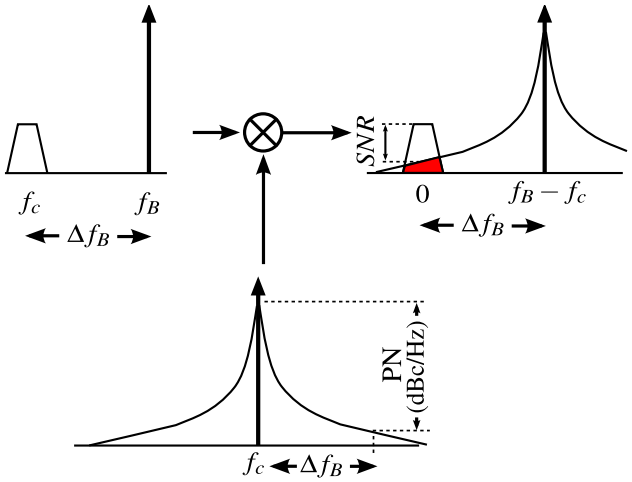


Figure 1.5: Reciprocal mixing degrades SNR.

any noise component at frequencies higher than $f_0/2$. Thus the spectrum of jitter is defined in the Nyquist band $(0, f_0/2)$.

In the presence of noise-folding, RMS jitter is defined as [6]:

$$\sigma_\tau^2 = \int_0^{f_0/2} S_\tau(f)df = \frac{1}{4\pi^2 f_0^2} \int_0^{f_0/2} S_{\theta_{PN}}(f)df \tag{1.2}$$

$S_{\theta_{PN}}$ and S_τ, which are the spectrum of phase noise and jitter, respectively, have already taken the noise-folding into account.

Most of the digital circuits, such as an inverter and a flip-flop, are edge-triggered and thus have intrinsic sampling operation. As demonstrated in [1], the RMS jitter of an inverter can be easily acquired through a window integration. With the knowledge of the RMS jitter, the phase noise of the circuit can be back-calculated through (1.2). This method bypasses the tedious noise-folding analysis and will be used for our DLL analysis.

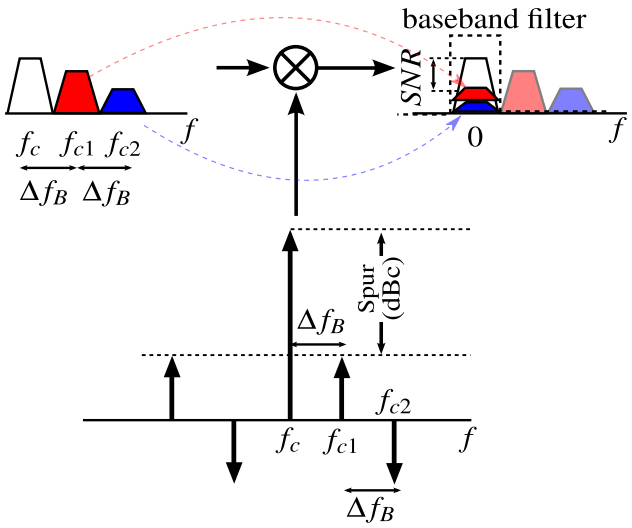


Figure 1.6: Spur reciprocal mixing also degrades SNR. Phase noise is ignored here.

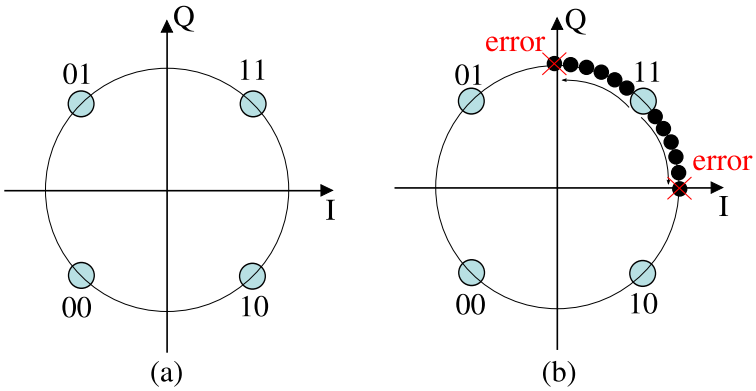


Figure 1.7: (a)4 QAM constellation; (b) Phase noise rotates the signal points.

1.0.3 Integer and fractional synthesizers and the design challenges

The carrier frequency of a free running oscillator drifts with time. An integer-N PLL (Int PLL) is implemented to lock the oscillator frequency to a multiple of its reference. Fig. 1.8 is a simple model of an Int PLL. Its loop gain is:

$$T(s) = K_{PD}H(s)\frac{K_{VCO}}{s}\frac{1}{N} \tag{1.3}$$

The noise of the loop consists of two types, 1) VCO noise and 2) input-referred noise which includes the noise of the phase detector, loop filter, frequency divider, and crystal. The closed-loop transfer functions of the input-referred noise and the VCO noise to the output are shown in (1.4) and (1.5), respectively. Note that (1.4) is lowpass with an in-band gain

of the frequency multiplication factor, whereas (1.5) is highpass.

$$\frac{\Phi_{OUT}}{\mathcal{L}_{in}} = \frac{NT(s)}{1+T(s)} \tag{1.4}$$

$$\frac{\Phi_{OUT}}{\mathcal{L}_{VCO}} = \frac{1}{1+T(s)} \tag{1.5}$$

This inherently contradictory characteristic dictates (1.6). The PLL integrated jitter σ^2_{out} is a convex function of the loop bandwidth f_{BW} and has a well-defined minimum at the optimum loop bandwidth, which is also shown in Fig. 1.8.

$$\sigma^2_{OUT}(f_{BW}) = \int_0^\infty 2\mathcal{L}_{OUT}(f)df = \frac{4K_W}{2Q^2 f_{BW}} + 4N^2 PN_{in} f_{BW} \tag{1.6}$$

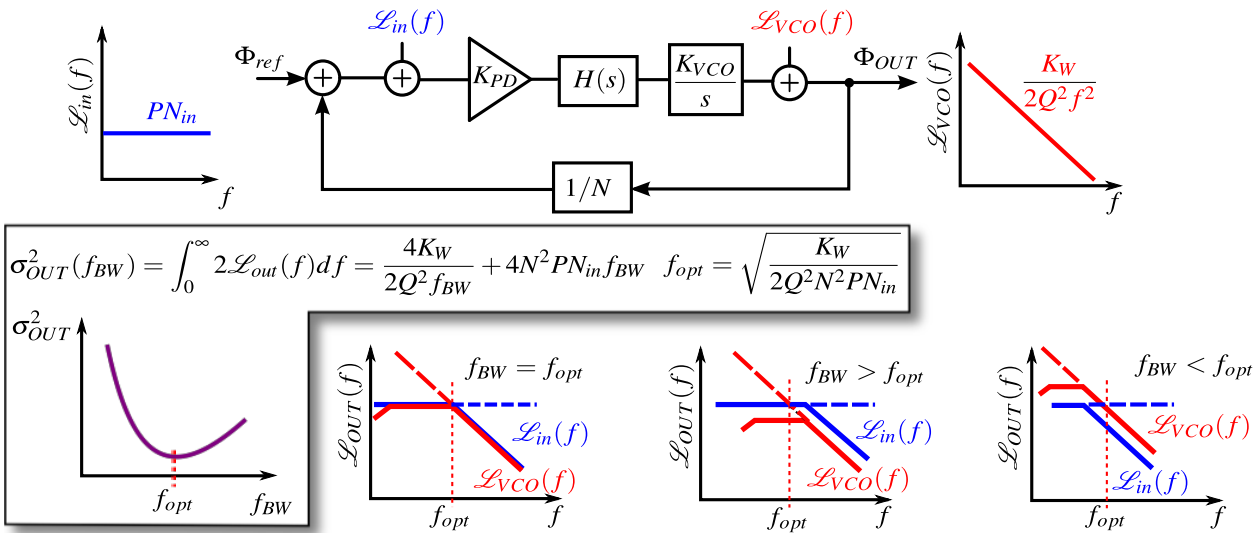


Figure 1.8: Int PLL model.

The salient distinction between a frequency synthesizer and a PLL lies in the formal's ability to change frequency, which is an essential requirement in an RF system to switch channels. A frequency synthesizer changes its output frequency in steps of the reference frequency. This is accomplished by a tunable frequency divider. Since the frequency division ratio is an integer, this type of synthesizer is known as an integer-N synthesizer.

Due to the unpredictability of the incoming signal, an RF channel can exist anywhere in band. Therefore, a fractional-N synthesizer is invented to provide even finer tuning resolution over its integer-N counterpart. it utilizes ΣΔ modulator (SDM) to realize its resolution, as presented in Fig. 1.9. Unfortunately, the SDM also introduces quantization noise into the loop, which is amplified by the PLL and is additive to the output noise. Fractional-N synthesizer has smaller optimum loop bandwidth and larger optimum jitter compared to its integer-N counterpart, as shown in Fig. 1.10.

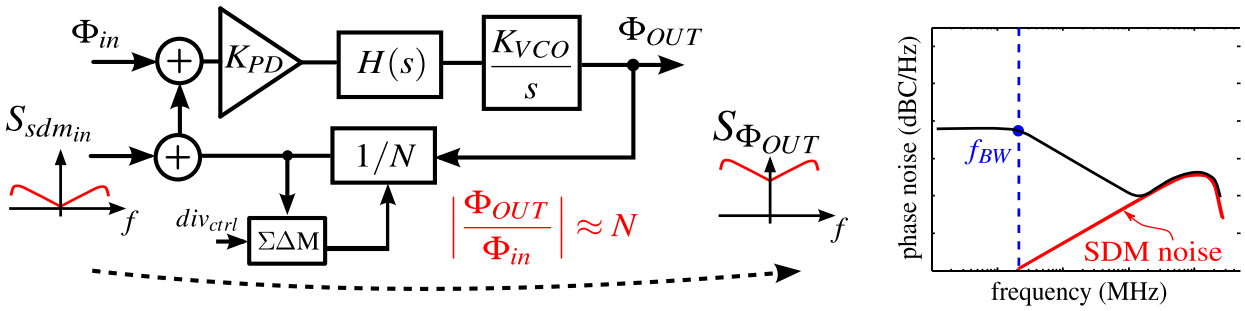


Figure 1.9: FN PLL has a SDM, which introduces quantization noise.

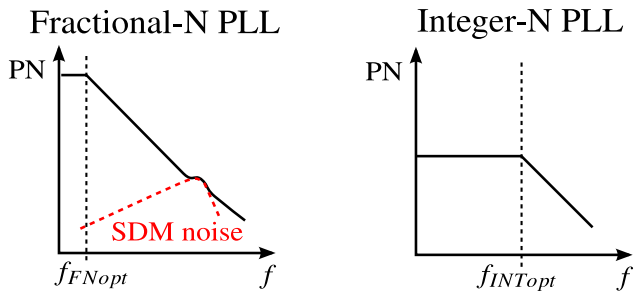


Figure 1.10: Spectrum comparison between FN PLL and Int PLL.

Since PLLs construct synthesizers, in the following contents, we also use PLL to indicate synthesizer.

When an Int PLL is locked, the PD operates in a periodic pattern at the frequency of the reference. It introduces a periodic ripple on the LPF and generates spurs which are offseted from the carrier by f_{ref} and its harmonics. Those spurs are referred to as reference spurs. Because the reference spurs are out of the loop bandwidth and experience sufficient suppression by LPF, they are usually benign to the system.

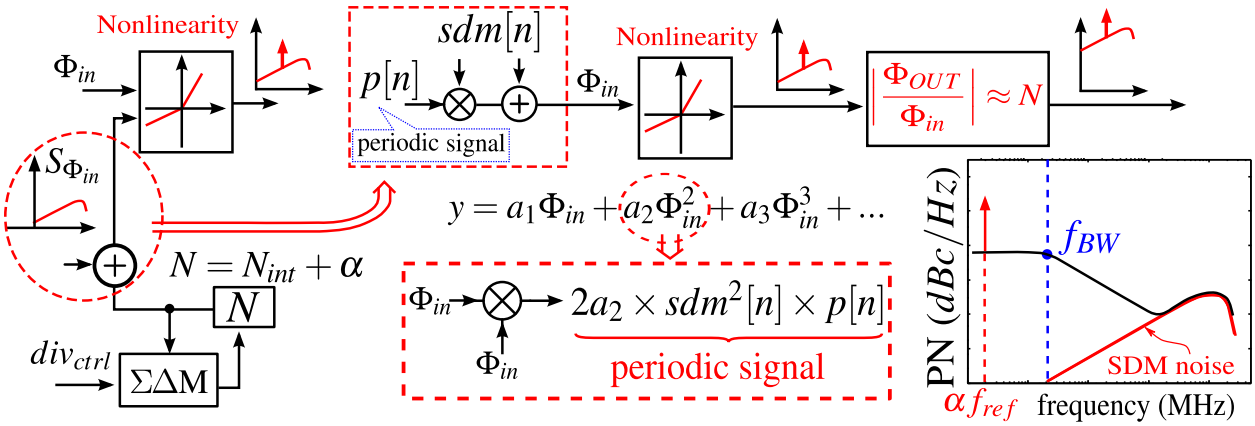


Figure 1.11: Phase detector nonlinearity convert quantization noise to fractional spurs.

In an fractional-N PLL (FN PLL), the nonlinearity of the phase detector converts the quantization noise back to spurs [7, 8], as shown in Fig. 1.11. The $\Sigma\Delta$ modulation spreads the periodic signal $\sin[\alpha\omega_{ref}n]$ in the modulator into widespread noise:

$$SDM_{out}[n] = \sin[\alpha\omega_{ref}n]sdm[n] + sdm[n] \tag{1.7}$$

After experiencing an even-order nonlinearity:

$$PD_{out}[n] = a_2(SDM_{out}[n])^2 \tag{1.8}$$

$$= a_2(\sin[\alpha\omega_{ref}n] + 1)^2 sdm[n]^2 \tag{1.9}$$

$$= a_2 sdm[n]^2(\sin[\alpha\omega_{ref}n]^2 + 2\sin[\alpha\omega_{ref}n] + 1) \tag{1.10}$$

The $sdm[n]^2 \cdot \sin[\alpha\omega_{ref}n]$ is often known as the fractional spur. It becomes significant, after the amplification from the loop. The fractional value of the divider α determines the spur location. For example, if the reference frequency is 20 MHz and α is 1/1000, a fractional spur will occur at the offset frequency of 20 kHz. When α is small, the fractional spur is located within the loop bandwidth, which cannot be filtered by the loop. A conventional FN PLL has a small loop bandwidth to attenuate most of the fractional spurs, lest severe degradation of the clock purity.

In a conventional PLL, all of the noise sources experience the same loop gain $T(s)$. A

single parameter that ties everything together is the loop bandwidth, which is the unity-gain frequency of $T(s)$. This work seeks inspirations from a delay-locked loop (DLL) and a heterodyne PLL to decouple various noise sources from the loop dynamics and therefore enables additional filtering that is independent of the loop bandwidth. The result is a simplification of an FN PLL design.

1.0.4 The challenges of wideband frequency synthesizer

For the implementation of software-defined radio (SDR), the RF system can be used for different communication standards. The RF channel can range from 50 MHz to 9 GHz. The synthesizer must cover a wide frequency range. As shown in Fig. 1.12, a synthesizer with a theoretical 100% tuning range oscillator generates the desired carrier by selecting the correct dividing modulus from the frequency divider array.

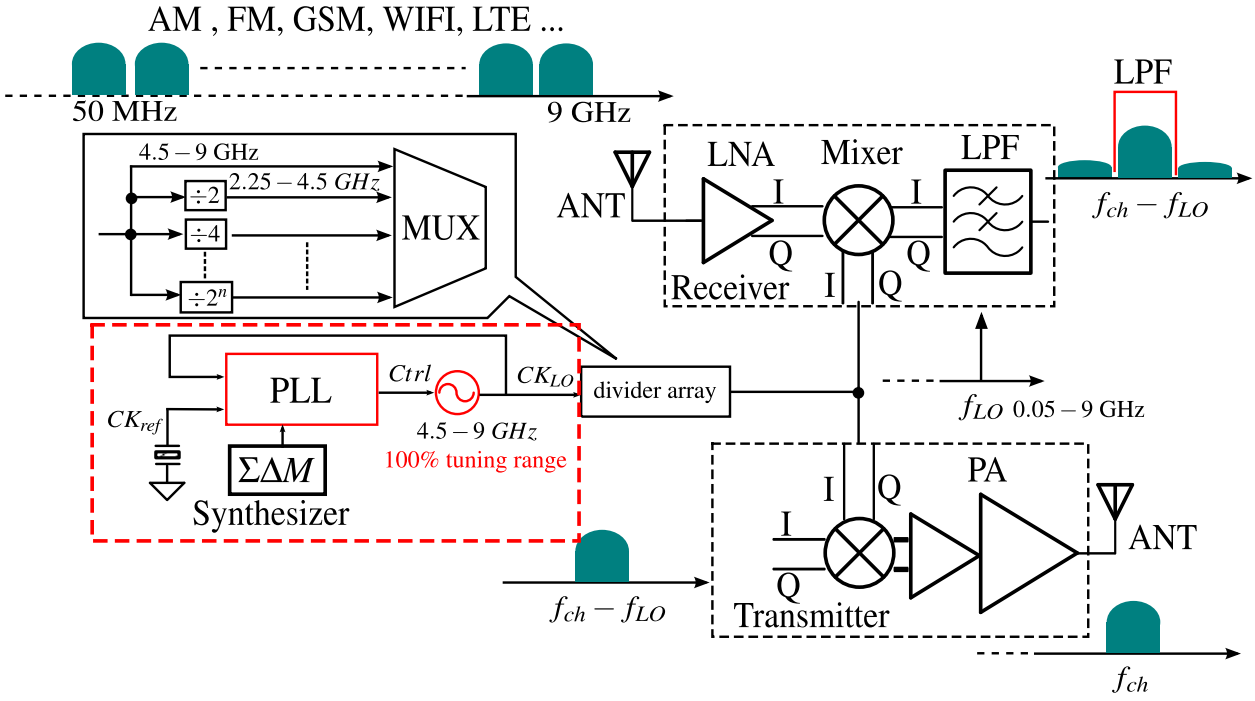


Figure 1.12: A SDR system and its wide tuning range synthesizer.

In addition, the synthesizer needs to meet the stringent specifications of all the various communication standards. Therefore, a synthesizer with low noise, low spur, and high

frequency resolution is required.

In summary, an SDR system requires a 100% tuning range oscillator, and also a power-efficient synthesizer with wide loop bandwidth, high frequency resolution, and high carrier purity.

1.0.5 Thesis organization

Delay locked loops (DLLs) are a variation of ring oscillator based PLLs. The modification of the ring oscillator in the DLL naturally suppresses VCO noise and allows the loop bandwidth to be small to attenuate other noise sources. This breaks the fundamental design trade-off of the loop bandwidth in a conventional PLL. It also has the added benefit of a wide tuning range because it consists of a ring oscillator. This compact architecture is thus an attractive option for modern clock generation. Chapter 2 investigates DLL as a wideband frequency synthesizer. It will later be shown that the application of a DLL, especially as a high-performance RF synthesizer, is limited by spurs and its all-pass nature. Chapter 3 revisits the PLL. Drawing inspiration from a heterodyne PLL, a novel calibration-free triple-loop PLL is designed that breaks the fundamental design trade-off of a synthesizer and achieves our target noise and spur requirements. A complete analysis of this structure and the measurement of a prototype are also included in chapter 3. Chapter 4 presents a PLL-based frequency extender. It improves the tuning range of an oscillator by more than 3 $\times$ without the penalty of area and phase noise. By implementing both the proposed triple-loop PLL and the frequency extender, a wideband synthesizer for the SDR application is achieved. Chapter 5 draws the conclusion of this thesis.

CHAPTER 2

DLL for Frequency Synthesis

2.1 Introduction

The ring oscillator with its compact footprint and scalability is a promising candidate for modern IC clock generation. The ring oscillator has an added benefit of wide frequency tuning range, achieved by changing its current. However, its drawback is poor phase noise. To achieve the same phase noise as an LC oscillator, the ring oscillator consumes approximately $2Q^2$ times the current of the LC oscillator [1]. Two modifications of the ring oscillator are then proposed to improve its noise performance to nearly that of an LC oscillator without prohibitive power consumption. In the first approach, the loop of the ring is broken to form a delay line, and the multi-phase outputs of the delay line are summed through an edge-combining circuitry that generates the desired output frequency. The second approach maintains the oscillation of the ring, but periodically breaks the loop and "refreshes" the noisy oscillator clock edge with a clean one that is in-phase. Both modified architectures require auxiliary feedback circuitries. In the first approach, the feedback ensures the total delay of the delay line equals to the period of the driving reference. In the second approach, it ensures that the oscillator frequency is a multiple of the reference frequency. The two proposed structures, including their auxiliary feedback circuitries (Fig. 2.1), are referred to as edge combining delay lock loops (ECDLLs) [9, 10, 11, 12] and recirculating delay lock loops (RDLLs), respectively [13, 14, 15, 3, 16].

Due to their structural differences, the ECDLL and RDLL architectures have been studied independently. In [17], the loop dynamics of ECDLLs are analyzed by a linear time-invariant (LTI) model. However, the edge combiner, shown in Fig. 2.1(a), is not in the feedback, is

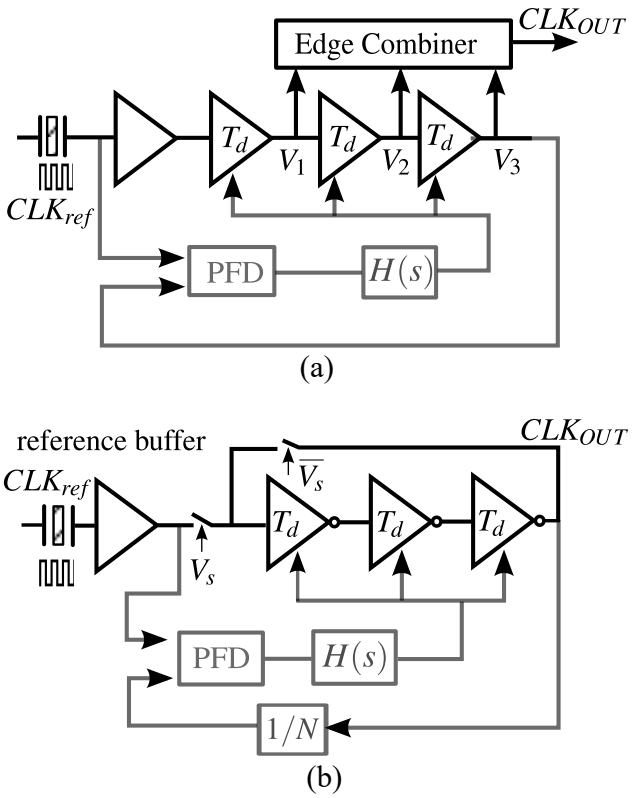


Figure 2.1: Modified ring oscillator architectures: (a) ECDLL; and (b) RDLL.

outside of the scope of [17]. The root mean squared (RMS) jitter of ECDLL, which is the integral of phase noise and spurs, has been analyzed by time-domain state equations [18, 19]. It illustrates the purity of a clock is an essential metric in an edge sampling system, such as an ADC or a CDR which only care the integral noise. However, it is not all-encompassing. For example, in an RF system, reciprocal mixing is also of concern. To quantify the amount of reciprocal mixing, there are specifications on the spot noise of the clock at different offset frequencies. The spectrum of the synthesizer becomes necessary. Fourier transform can be used to obtain the spectrum of ECDLL [20, 21, 22, 2], its lengthy mathematical derivation notwithstanding. Fortunately, since an ECDLL is a nonlinear system similar to a PLL, at steady state, the variables of concerns in the loop, such as loop control voltage and input phase error of PD, experience small fluctuation. We can linearly approximate the loop and build an LTI model for the ECDLL. This sheds much more insight and can obviate the need of Fourier transform.

On the other hand, the study for the RDLLs is more mature. The loop dynamics and

phase noise in RDLLs have been analyzed by a continuous-time LTI model [13] and a discrete-time LTI model that considers white noise-folding [23]. Although [23] is similar to our work, analyzing the noise-folding effect in a digital system requires much more efforts. Here, we relax the effort by using the definition of jitter instead. The authors of [24] provide a closed-form expression of the spur mechanism of RDLL.

The analyses of both types of DLLs are segmented through the literature, making a fair comparison of them difficult. This work analyzed both DLLs on equal footings and reached the following conclusions:

1. The same tapped-delay-line FIR filter model can analyze the modified ring oscillators in both ECDLL and RDLL.

2. The modified oscillators do not have DC poles. Therefore, the noise transfer functions (NTF) of the oscillators no longer follow the Leeson's equation which has the $1/f^2$ shape, and DLLs are more stable than PLLs.

3. For a given noise specification, RDLLs require half the power of ECDLLs, which is more power-efficient.

4. The tapped-delay-line FIR filter can be embedded into the loop to form the complete LTI models that enable the analysis of both DLLs in the frequency domain.

5. The loop of ECDLLs can not suppress the oscillator noise as efficiently as the RDLLs.

6. When the frequency multiplication factor is large, ECDLLs suffer stronger reference spurs than RDLLs do.

7. An RDLL outperforms an ECDLL because it is more power-efficient, has stronger filtering, and suffers lower spurs.

8. The LTI model can also be extended to an injection-locking oscillator.

With the proposed models, the applications of DLLs in a fractional-N synthesizer are analyzed and the following can be concluded:

1. DLLs cannot work as a fractional-N synthesizer for high-performance RF systems that requires very clean clock;

2. A DLL is suitable as an intermediate frequency multiplier to increase the reference frequency of a PLL to reduce the quantization noise of a $\Sigma\Delta$ modulator in the PLL.

3. If the design area is a constraint, DLL is a better choice as the frequency multiplier than a PLL or an injection-locking oscillator. Otherwise, if LC oscillator can be used, the latter two are better.

The following sections build the LTI model for both DLLs, present the closed-loop transfer function of both DLLs, analyze the spurs of both DLLs, and finally, discuss the applications of RDLL. In the appendix, an LTI phase noise model for injection-locking oscillators is presented.

2.2 MODELING THE MODIFIED RING OSCILLATOR

The modified ring oscillators are called the edge-combining frequency multiplier (ECFM) in an ECDLL and the recirculating frequency multiplier (RFM) in an RDLL. They are used to scales up the input frequency by a multiplication factor. All other blocks of the DLL are identical to phase lock loops (PLLs), which are definable by transfer functions. Therefore, once the transfer function of the frequency multiplier is known, the DLL can be analyzed as a conventional linear system. This section analyzes both ECFM and RFM in the frequency domain and discusses their similarities and differences.

2.2.1 Continuous-time finite impulse response (FIR) filter

Before we start the DLL discussion, let's first have a quick review on FIR system. A delay of a signal can be described as

$$y(t) = x(t - T_d) \tag{2.1}$$

Where $T_d = 1/f_d$ is the time of the delay. The impulse response of this system is $\delta(t - T_d)$ which has a Laplace transform of e^{-sT_d}. If a system delays the input $x(t)$ for $N - 1$ times

and adds up all the delayed value, as shown in Fig. 2.2, it has a function of

$$y(t) = \sum_{n=0}^{M-1} x(t - n \cdot T_d) \tag{2.2}$$

Its impulse response is

$$h(t) = \sum_{n=0}^{M-1} \delta(t - n \cdot T_d) \tag{2.3}$$

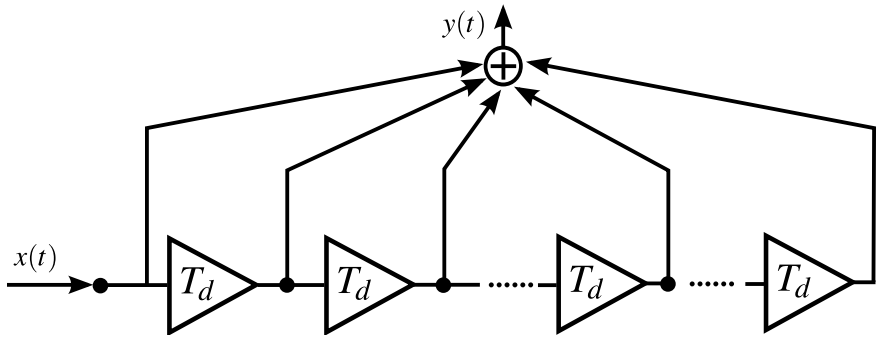


Figure 2.2: A linear system delays the input $x(t)$ and adds up all of the delayed outputs.

Laplace transform of this system is

$$H(s) = \sum_{n=0}^{M-1} e^{-snT_d} = \frac{1 - e^{-sNT_d}}{1 - e^{-sT_d}} \tag{2.4}$$

The frequency response is

$$H(f) = \frac{1 - e^{-j2\pi fNT_d}}{1 - e^{-j2\pi fT_d}} \tag{2.5}$$

It indicates that this system notches the tones at frequency $m \cdot f_d/N$ when m is not the multiple of N. It reaches the peak magnitude when m is the multiple of N, as shown in Fig. 2.3.

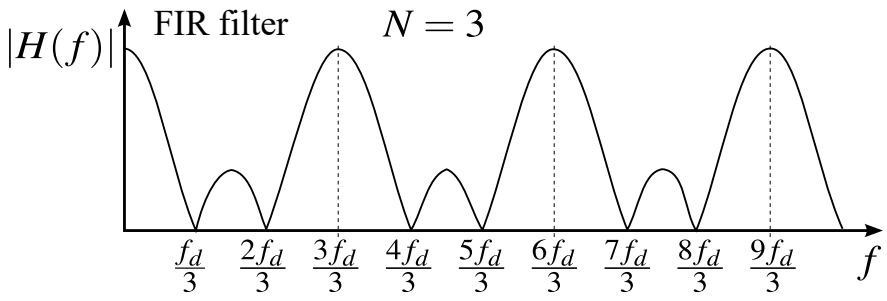


Figure 2.3: An example of FIR filter magnitude. $N = 3$ in this example.

From the theory of discrete-time filter design [25], we can map this continuous system to z domain by $z = e^{sT_d}$ where we assume the discrete-time system has a sampling rate of f_d. Then $H(s)$ is transferred to:

$$H(z) = \frac{1 - z^{-N}}{1 - z^{-1}} \tag{2.6}$$

This corresponds to a discrete-time impulse response:

$$h[n] = \begin{cases} 1 & n = 0, 1, ..., N-1, \\ 0 & otherwise \end{cases} \tag{2.7}$$

Which is a well-known FIR system. Therefore, we regard the original continuous-time system as a continuous-time FIR filter. One may worry if the impulse invariance mapping distorts the Laplace domain model because of aliasing. Fortunately, for this FIR system, when transferring from Laplace domain to z domain, there is no aliasing because the continuous-time FIR system can be regarded as sampling rectangular window and aliasing has already taken place there. As Gardner demonstrated [26], a discrete-time analysis can provide a more accurate prediction for the loop dynamics of a sampling feedback system such as PLLs or the DLLs that we will discuss. In the following chapter, our approach is first to build z domain model of the system to analyze loop stability and then using $z = e^{j2\pi fT_d}$ to convert the transfer function to the frequency domain for the analysis of system frequency response and phase noise.

2.2.2 Edge-combining frequency multiplier

2.2.2.1 Frequency multiplication in ECFM

After breaking the ring, the inverter chain is no longer an autonomous circuit and need to be driven by a reference. Here we have assumed that, by the loop control, the chain is at steady state and its total delay equals to a period of reference T_{ref}. The chain, i.e., the delay-line, propagates the reference and generates multiphase outputs which have the same frequency f_{ref} as the reference. Apparently, to obtain a higher frequency, we must incorporate a circuitry with the function of frequency multiplication. As shown in Fig. 2.4,

from the 2nd inverter in the chain [1], an AND connects the input and output of the inverter and converts the two square waves to a pulse wave with the pulse width of $T_d/2$. Skip one inverter, another AND connects to the fourth inverter and generates the second pulse wave which lags the first one by T_d. Following this procedure, the input and output of each even order inverter in the chain are connected to an AND. Finally, an OR combines all the AND outputs and forms the high-frequency clock. This circuitry is an example of ECFM whose operation can also be recognized as that a reference with a pulse width of $T_d/2$ and frequency of f_{ref} drives the delay line, and an adder linearly sums the even outputs of the line to generate the high-frequency clock as shown in Fig. 2.5. This transformation shifts all the pulse generations which implicitly incorporate nonlinearity to the input of the ECFM, and therefore the delay-line with the edge combiner becomes a linear system which allows LTI analysis.

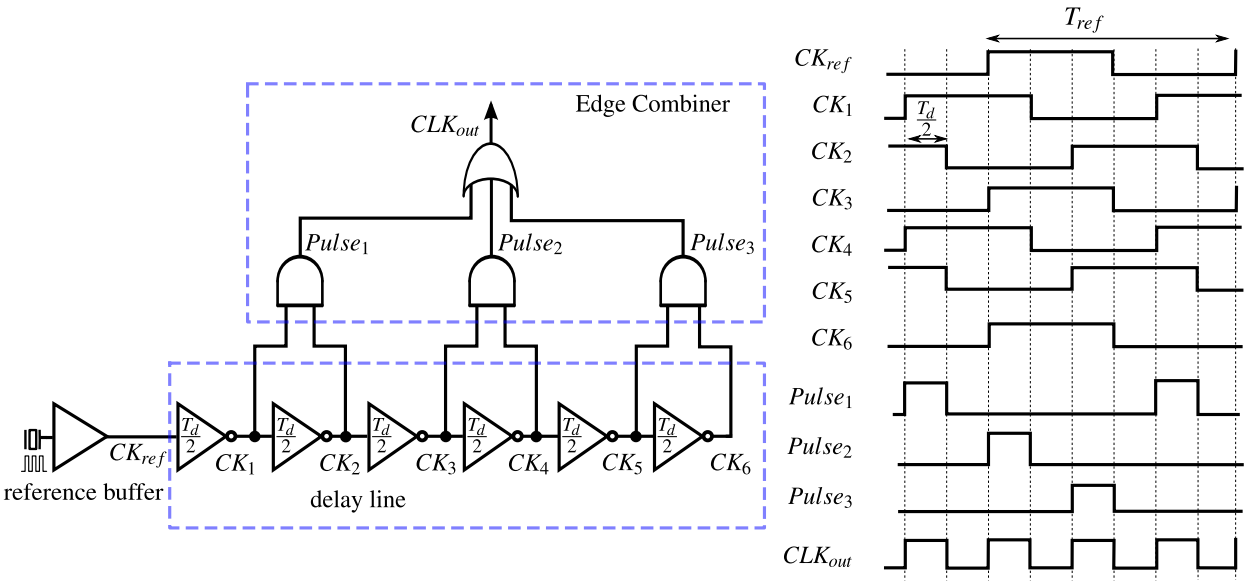


Figure 2.4: The schematic and waveform of ECFM.

Since only the even outputs of the delay line are sent to the adder, we regard the two inverters in between as a buffer. Based on Fig. 2.5, the reference appears at each successive buffer, delayed in time by T_d or in phase by $2\pi/N$. In the frequency domain, each buffer is

[1] normally start from the second inverter to ensure the edges of different AND outputs matches better

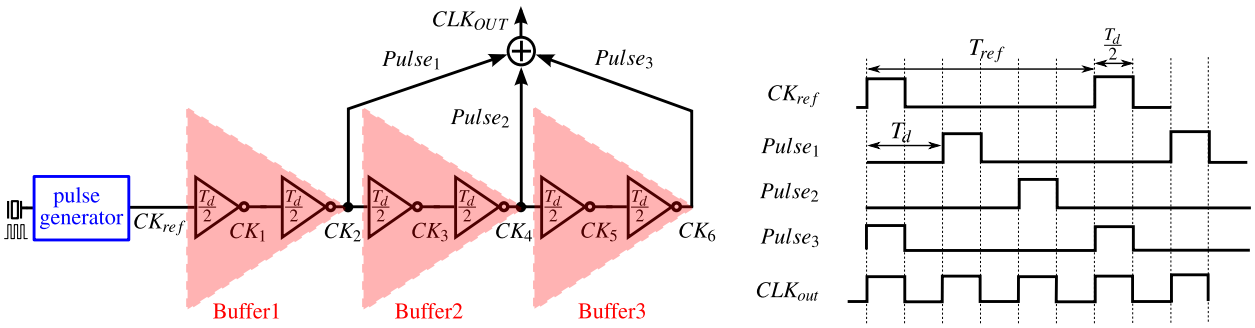


Figure 2.5: An equivalent circuit of ECFM where two cascade inverters form a buffer with the delay of T_d.

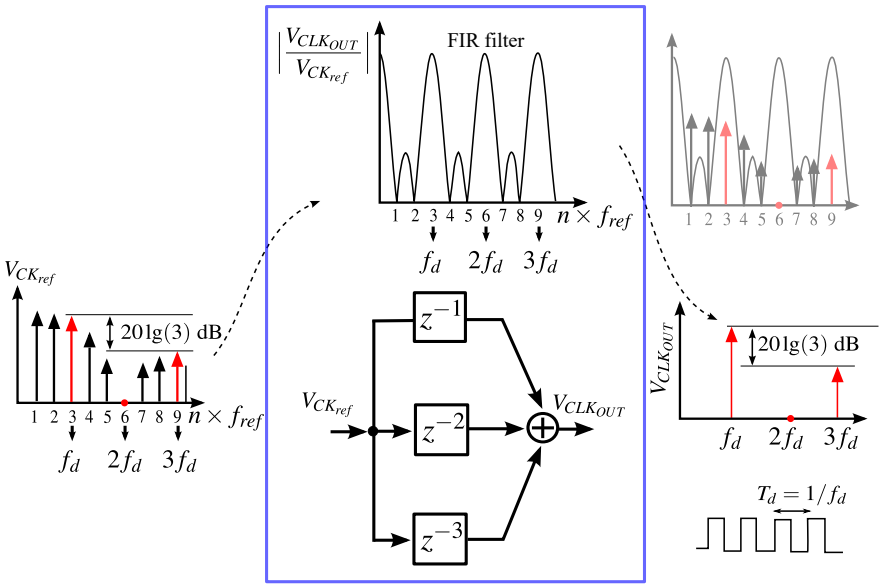


Figure 2.6: Open loop ECDLL structure, in which the FIR filter achieves frequency multiplication. Here $z^{-1} = e^{-s \cdot T_d}$.

a linear delay block $z^{-1} = e^{-sT_d}$. The edge combiner linearly adds up the delayed outputs, forming a classic tapped-delay-line FIR filter that we have discussed. The transfer function of the filter is:

$$H_{FIR}(z) = \sum_{n=1}^{N} z^{-n} = z^{-1} \cdot \frac{1 - z^{-N}}{1 - z^{-1}} \tag{2.8}$$

For its frequency response, based on the discussion of the FIR filter, we know it reaches peak magnitude at DC and the harmonics of $f_d = 1/T_d = Nf_{ref}$ and nullifies the other harmonics of f_{ref} , as shown in Fig. 2.6.

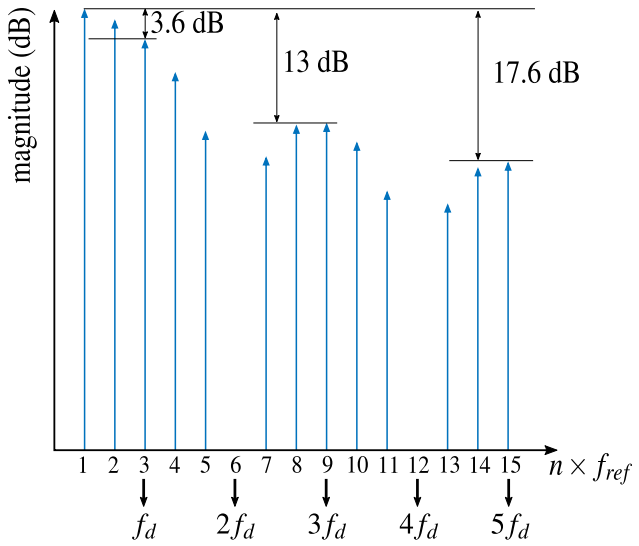


Figure 2.7: The Fourier series of a pulse which has $\frac{T_d}{T_{ref}} = \frac{1}{3}$.

The Fourier series expansion of the reference is:

$$CLK_{ref}(t) = \frac{T_d}{2T_{ref}} + \sum_{n=0}^{\infty} \frac{T_d}{T_{ref}} sinc\left(\frac{\pi n T_d}{2T_{ref}}\right) \cos(2\pi n f_{ref} t) \tag{2.9}$$

Although, as shown in Fig. 2.7, the harmonics of reference are almost equally strong till the harmonic index approaches $2T_{ref}/T_d$, after the selection of the FIR filter, the remaining tones are at DC, f_d and its harmonics. Calculating the strength of the remaining tones through (2.9), we can find they form exactly a square wave with the frequency of f_d, which matches with our observations in Fig. 2.4 and Fig. 2.5.

Now, we understand the mechanism of the ECFM. Each AND generates a narrow pulse to provide a strong tone at the desired frequency of f_d. The delay and the OR act as the FIR filter which selects it and nullifies the unwanted tones.

By shifting the pulse generation process to the input, ECFM is a linear system. In the following discussion, to simplify the analysis, we will always assume the pulse is generated before driving the delay line and the edge combiner is a simple adder. But one must be aware that the practical pulse generation always happens at the output of the delay line since we can easily get multiphase outputs there.

[9] also used the FIR filter model to explain the frequency multiplication in an ECFM but it didn't discuss further for ECDLL noise and transfer function. Here, we will extend

this idea for the analysis of ECFM sideband phase noise and transfer function.

2.2.2.2 Phase noise and transfer function of ECFM

When bandlimited noise is injected at the reference node, the reference clock can be written as:

$$V_{in}(t) = \sin\left(\omega_{ref}t + \theta_{PN}(t)\right) \tag{2.10}$$

, where $\theta_{PN}(t)$ is the phase noise of the sinewave. If this signal drives a nonlinear circuit, such as an inverter-based buffer, the circuit nonlinearity generates harmonics of the sinewave at the output, forming a new waveform as:

$$V_{OUT}(t) = \sum_{n=0}^{\infty} a_n \sin\left(n\omega_{ref}t + n\theta_{PN}(t)\right) \tag{2.11}$$

For the N^{th} harmonic:

$$\begin{aligned} V_{OUTN}(t) &= a_N \sin\left(N\omega_{ref}t + N\theta_{PN}(t)\right) \\ &\approx a_N \sin\left(N\omega_{ref}t\right) + a_N N\theta_{PN}(t)\cos\left(N\omega_{ref}t\right) \end{aligned} \tag{2.12}$$

(2.12) indicates that the PM sideband-to-carrier ratio of the N^{th} harmonic becomes worse by N^2. Therefore, the phase noise of the desired N^{th} harmonic is increased by N^2, as shown in Fig. 2.8.

Because the FIR filter is periodic in the frequency domain, the sideband of the selected N^{th} harmonic experiences the same transfer function as the baseband which is shown in Fig. 2.6. The complete noise transfer function (NTF) model of ECFM is shown in Fig. 2.9. Since the FIR filter selects the N^{th} harmonic of the reference, we focus on the sideband of this tone whose phase noise is N^2 worse than the one of the fundamental tone ($\mathcal{L}_{\phi_{ref}}(f)$). We know for a Gaussian noise passing through an LTI system, its power spectrum density is shaped by the magnitude square of the transfer function. Therefore, the N^2 multiplication on the phase noise power spectrum density corresponds to a gain of N in the noise transfer function. a_N is the amplitude of the input signal. In the transfer function, it converts the

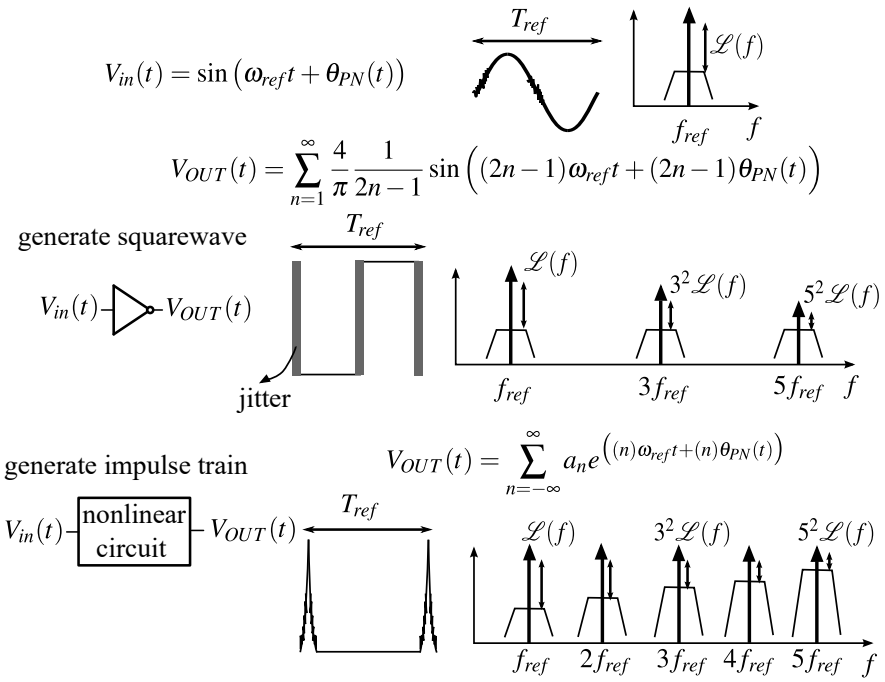


Figure 2.8: Nonlinearity generates the harmonics of the input and raises phase noise correspondingly.

dimensionless phase noise to a dimensioned noise voltage which traverses the FIR filter. At the output, the power spectrum density of the noise voltage is $\mathcal{L}_{\phi_{ref}}\left|a_N \cdot N \cdot \frac{1-e^{-j\omega T_d N}}{1-e^{-j\omega T_d}}\right|^2$. Meanwhile, the desired input signal also passes through the delay line. Since the frequency of the desired tone is f_d, each buffer in the delay line shifts the phase of its input by $2\pi f_d T_d = 2\pi$. Therefore the edge combiner adds up the delayed signal in-phase, scaling up the amplitude of the input by N, as shown in Fig. 2.10. Since phase noise is defined as the single sideband noise normalized to the power of the relative tones [27], the output phase noise is the derived power spectrum density of the noise voltage over the output amplitude square $(N \cdot a_N)^2$ which corresponds to $1/(N \cdot a_N)$ in the noise transfer function. Then the ECFM output phase noise due to the reference is:

$$\mathcal{L}(f) = \mathcal{L}_{\phi_{ref}}\left|\frac{1-e^{-j\omega T_d N}}{1-e^{-j\omega T_d}}\right|^2 \tag{2.13}$$

We can extend this method to other bandlimited input noise, such as the phase noise injected by the k^{th} buffer of the delay line. It is also scaled up by N^2 in the N^{th} harmonic. a_N converts it to voltage power spectrum density. It doesn't modulate the pulse of the previous

buffer. Therefore, the adder counts its effects from the k^{th} buffer. At the output, since the amplitude of the desired tone is set to $N \cdot a_N$, we still divide the power spectrum density of the noise voltage by $(N \cdot a_N)^2$. Then the output phase noise is:

$$\mathcal{L}(f) = \mathcal{L}_{\phi_{DL_k}} \left| \frac{1 - e^{-j\omega T_d (N-k+1)}}{1 - e^{-j\omega T_d}} \right|^2 \tag{2.14}$$

In the ECFM, each buffer must provide a delay of T_d which is the output period. Normally in the applications of DLL, T_d is larger than 200 ps. Therefore the buffers of the delay line experience a wide noise integral window [1] and are the determinant noise contributors. Our discussion focuses on the buffer noise. The transfer functions from each buffer to the output are listed in Fig. 2.11. After superposing all the buffer noise, the complete noise model of ECFM is presented in Fig. 2.12.

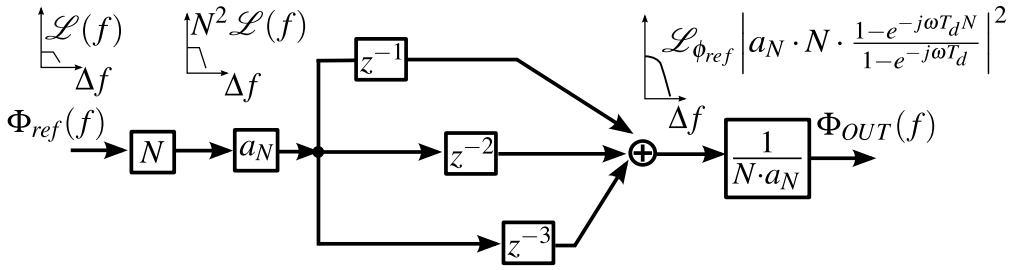


Figure 2.9: Phase noise in an ECFM.

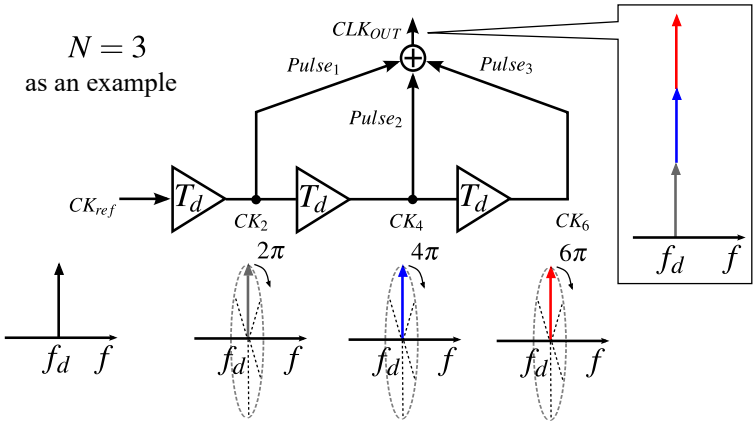


Figure 2.10: The carriers rotate $N \cdot 2\pi$ and are added up in-phase.

The phase noise due to white noise, which possesses a very wide bandwidth, is best

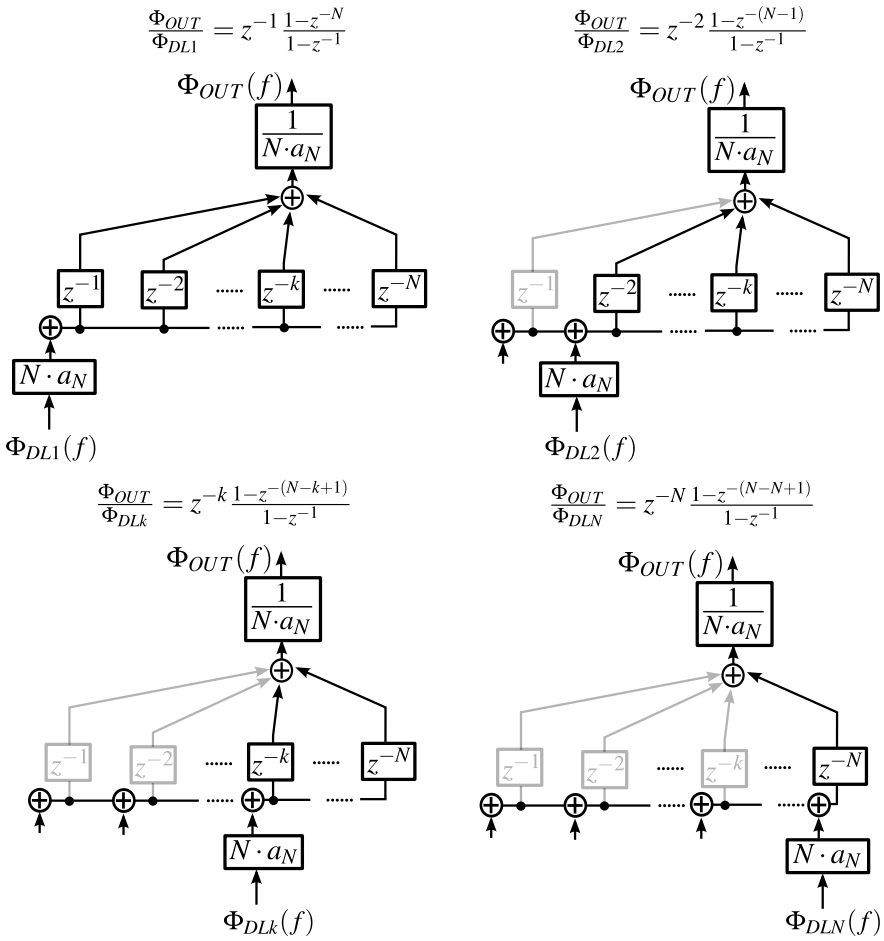


Figure 2.11: The transfer functions from each buffer to the output.

derived from the definition of periodic jitter [6] to avoid dealing with spectrum folding:

$$\sigma_n^2 = \frac{1}{4\pi^2 f_0^2} \int_0^{f_0/2} 2\mathcal{L}(f) df \tag{2.15}$$

where σ_n^2 is the periodic jitter of a buffer, which is thoroughly discussed in [1], and f_0 is the frequency of the buffer output which is also the intrinsic sampling rate. Assuming that only the last buffer of the delay line is noisy, the buffer noise jitters the ECDLL output edge at every N^{th} interval of the output cycle. This interval noise injection is a cyclo-stationary process, which can be treated as a stationary process by time-averaging the statistical parameter in each cycle [28]. At the output, the buffer introduces an average mean square jitter of σ_n^2/N. Therefore, the injected phase noise of the buffer is:

$$\mathcal{L}_{\Phi_{DLn}}(f) = \frac{4\pi^2 \sigma_n^2}{N T_d} \tag{2.16}$$

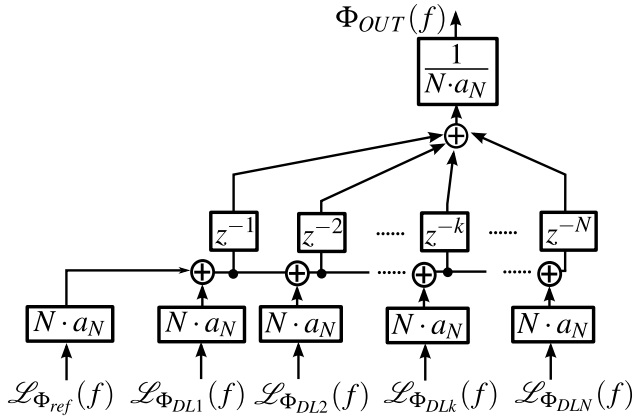


Figure 2.12: Complete model of ECFM for bandlimited noise.

where T_d is the period of the output which is also the intrinsic sampling clock of the ECFM. As all of the buffers along the delay line have the same device parameters, the phase noise injected by any of them is given by (2.16). As shown in Fig. 2.13(a), each pulse wave is modulated by the jitter of the current buffer and also the accumulated jitter propagating from the previous stages. The sequential pulses from each buffer constitute the output. Therefore, their jitter also sequentially adds to the output which, again, forms the discussed FIR filter as shown in Fig. 2.13(b).

Since $N \cdot a_N$ compensates $1/(N \cdot a_N)$, the model of bandlimited noise is identical to the one of the white noise. Our discussed FIR filter can universally analyze both white noise and bandlimited noise. Assume that each buffer injects a total noise of $L_{\Phi_{DLn}}(f)$, including flicker noise and white noise, to the frequency multiplier. Since the noise of each buffer is uncorrelated, based on the transfer functions derived in Fig. 2.11, the total ECFM noise is :

$$\mathcal{L}(f) = \sum_{n=0}^{n=N-1} \mathcal{L}_{\phi_{DLn+1}}(f) \left| \frac{1 - e^{-j2(N-n)\pi f T_d}}{1 - e^{-j2\pi f T_d}} \right|^2 \tag{2.17}$$

So far, we analyze the ECFM by assuming the delay of the buffer in the delay line is exactly the period of the output T_d. In practice, it drifts due to the process, supply, and temperature (PVT) variation. Therefore, ECFM must have tunability to compensate for the delay drift. We usually control the input voltage of the varactor to tune the capacitance and adjust the total delay. In a small tuning range, the delay of the line responses to the control voltage linearly. This allows us to derive the gain of the delay line K_{DL} which is

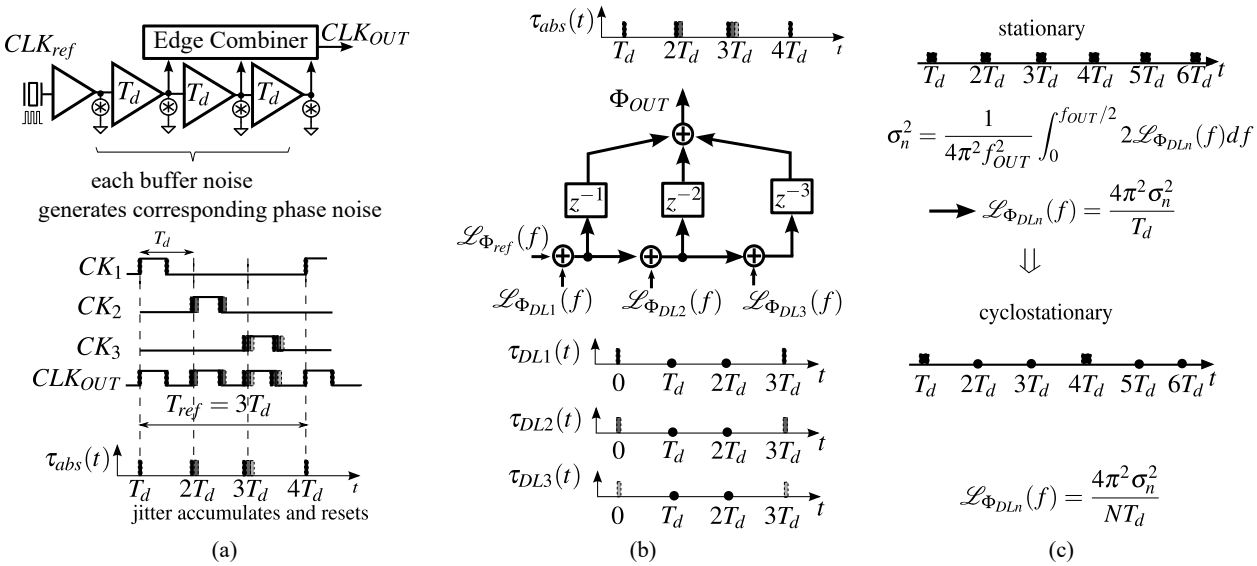


Figure 2.13: The universal ECFM phase noise model.

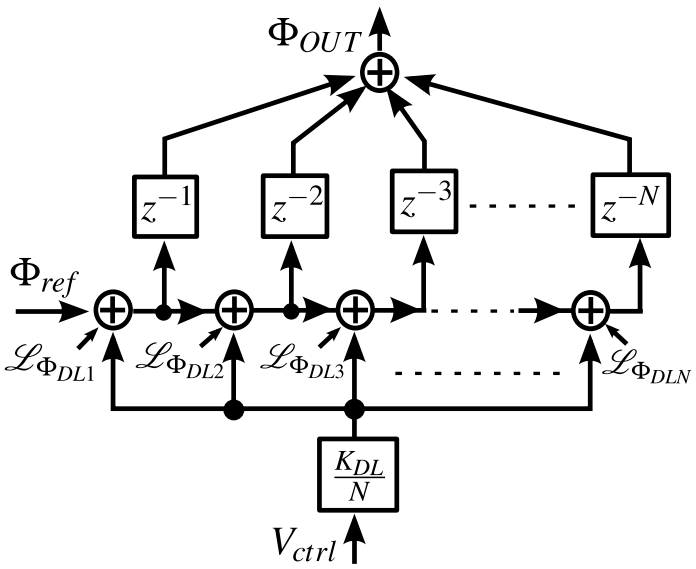


Figure 2.14: Complete model of ECFM.

the derivative of the total delay with respect to the control voltage. Each buffer in the delay line equally contributes to the gain, thus each of them has the gain of K_{DL}/N. Now, if the control voltage changes one unit, the delay of each buffer is increased by K_{DL}/N. Besides, the individual delay contribution from each buffer also propagates to the following stages; therefore, the m^{th} stage pulse is delayed by $m \cdot K_{DL}/N$. Since the pulse of each buffer sequentially constitutes the output waveform, the accumulated delay effect also sends

to the output in sequence, which again forms the thoroughly discussed FIR filter. Then we can establish the complete ECFM model as shown in Fig. 2.14. This model can be used to analyze ECFM phase noise and be embedded into a feedback loop for closed loop analysis, which will be covered in 2.3.

2.2.3 Recirculating frequency multiplier

Frequency multiplication in RDLL is straightforward. As shown in Fig. 2.15 (a), when the multiplexer (mux) closes the delay line to form a loop, the frequency multiplier operates as a ring oscillator. At steady state, this ring oscillates at the frequency $N \cdot f_{ref}$. During every cycle, the devices in the ring inject periodic jitter into the loop [1]. Thus, the jitter in the RFM accumulates, as occurs in a conventional ring oscillator, as shown in Fig. 2.15 (c), until the mux is controlled to break the loop. At this instant, a clean edge of the low-noise reference replaces the noisy edge of the ring, and the reference resets the accumulated jitter. We can also understand the jitter format in RFM by the FIR filter. Before the resetting of the reference, at each cycle of the oscillation, the devices introduce new jitter into the ring. It propagates to the next cycle and with the newly injected jitter together modulates the period of that cycle. The array of the adders in our FIR filter can represent this jitter propagation and accumulation. The number of the adders is determined by $N = f_d/f_{ref}$ which indicates how many cycles of jitter will be accumulated before resetting. The wave of each period sequentially constructs the output waveform. Therefore, the accumulated jitters at each cycle are also added to the output in sequence. The delay and summation in the FIR filter represent this operation. Thus, the jitters in RFM and ECFM follow the same trend which is presented in Fig. 2.15 (b).

In each cycle, since the periodic jitters caused by white noise are uncorrelated. (2.17) can express the RFM noise. The reference jitter is sampled by the RFM and held at the output until the time of the next injection. If its bandlimited, the sampling causes no aliasing. We can regard the reference noise as directly injecting into the RFM. If it's white noise, based on the definition of jitter, the white noise considering spectrum folding can be derived which

still injects into RFM at the reference port. The effect of the reference noise is thus shown in Fig. 2.15 (b).

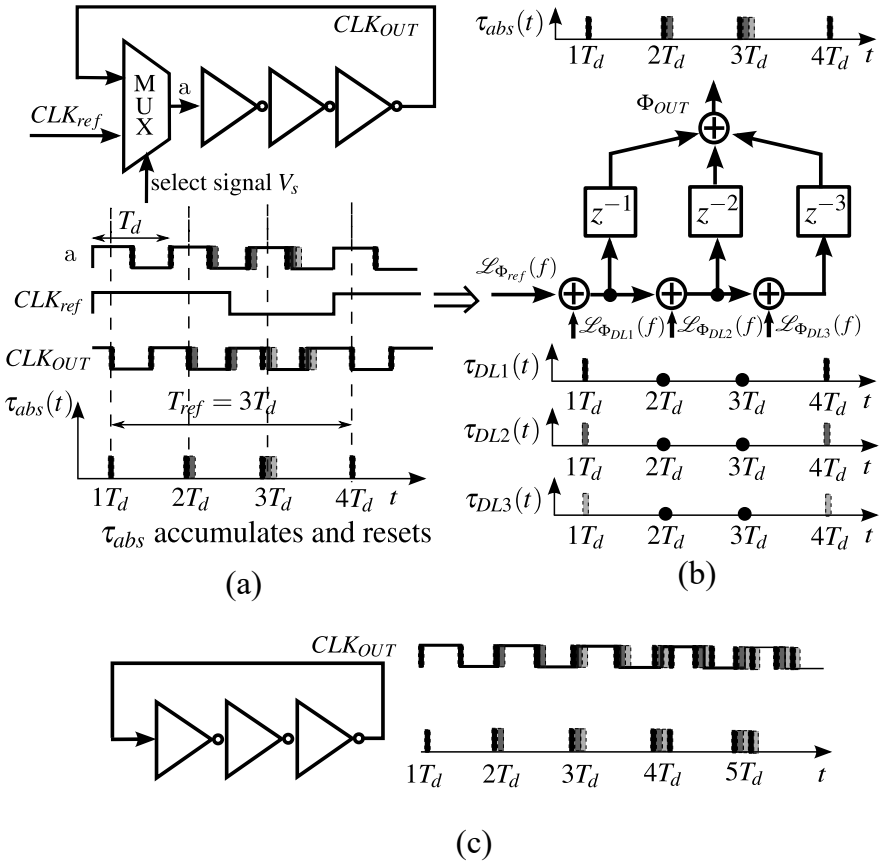


Figure 2.15: (a),(b) RFM jitter format and phase noise model; and (c) jitter of free running ring oscillator.

The flicker noise in each RFM cycle cannot be treated independently, because their phase modulations are correlated between cycles. A better approach, as demonstrated in [1] is to establish the NTF of the RFM flicker noise source. The output flicker noise is then the input noise multiplied by the squared magnitude of the NTF.

Although an RFM does not have an explicit delay function as the delay line in an ECFM where the reference waveform can be stored and propagate in the delay line, its implicit delay can be recognized if we focus on the edge propagation. For example, when the clean edge of the reference injects into the ring, it propagates in the loop until the next injection instance when it just travels back to the injection node and will be replaced by the new injected edge.

The total traveling time of the edge is the delay of RFM which is T_{ref} at steady state. With this knowledge, we can derive the delay gain of RFM. Similar to [13], when the RDLL is locked, the total RFM delay in a reference cycle is $T_{ref} = N \cdot T_d$. The total delay gain K_{DL} is defined as the derivative of the RFM delay on the subject of the control voltage.

$$\begin{aligned} K_{DL} &= \frac{d}{dv_{ctrl}} \left\{ \frac{N}{f_d + K_{vco} v_{ctrl}} \right\} \\ &\approx -\frac{N}{f_d^2} K_{vco} = K_{vco} T_{ref} T_d \end{aligned} \tag{2.18}$$

After converting the dimensions to radian and uniformly distributing the gain among the cycles, the gain per cycle is:

$$K_{RDLL} = K_{DL} \frac{2\pi}{T_{ref}} \frac{1}{N} = \frac{2\pi K_{vco} T_d}{N} \tag{2.19}$$

When we tune the control voltage by one unit, each cycle adds a delay of K_{RDLL}. The delay propagates to the next period and gradually adds up. The output sequentially experiences the delay effect of each cycle. Thus, the complete RFM model is the same as ECFM which is shown in Fig. 2.16(a). The transfer function from the control voltage is:

$$H_{RFM}(z) = K_{RDLL} \left\{ \frac{1 - z^{-N}}{(1 - z^{-1})^2} - \frac{N z^{-N}}{1 - z^{-1}} \right\} \tag{2.20}$$

Within the frequency of interest, by Taylor expansion, $z^{-N} \approx 1 - NsT_d + (NsT_d)^2/2$ and $z^{-1} \approx 1 - sT_d$. (2.20) can be simplified as:

$$H_{RFM}(z) \approx K_{RDLL} \left\{ \frac{NsT_d - (NsT_d)^2/2 - N(1 - NsT_d - (NsT_d)^2/2) sT_d}{(sT_d)^2} \right\} \tag{2.21}$$

$$\approx K_{RDLL} \left\{ \frac{(NsT_d)^2/2}{(sT_d)^2} \right\} \tag{2.22}$$

$$\approx \frac{N^2 K_{RDLL}}{2} = \frac{2\pi K_{vco} T_{ref}}{2} \tag{2.23}$$

As shown in Fig. 2.16(b), the transfer function of our model matches with [13].

In a ring oscillator, the transfer function is K_{vco}/f. The NTF of the flicker noise is K_v/f, as shown in Fig. 2.15(c), where K_v is the gain of the ring oscillator corresponding

to the injected flicker noise current [1, 29]. The two equations are identical, except for the proportionality constant. Thus, by replacing K_{vco} in Fig. 2.15(a) with the new gain factor K_v, we establish the NTF for RFM flicker noise, as shown in Fig. 2.15(d). Knowing this NTF, the RFM output flicker noise is readily derived as:

$$\mathcal{L}_{1/f}(f) = S_i^{1/f} K^2 \left| \frac{1 - z^{-N}}{(1 - z^{-1})^2} - \frac{N z^{-N}}{1 - z^{-1}} \right|^2 \tag{2.24}$$

,

where $S_i^{1/f}$, which has been thoroughly discussed in [1], is the single-sideband total flicker noise current of all inverters in the ring oscillator.

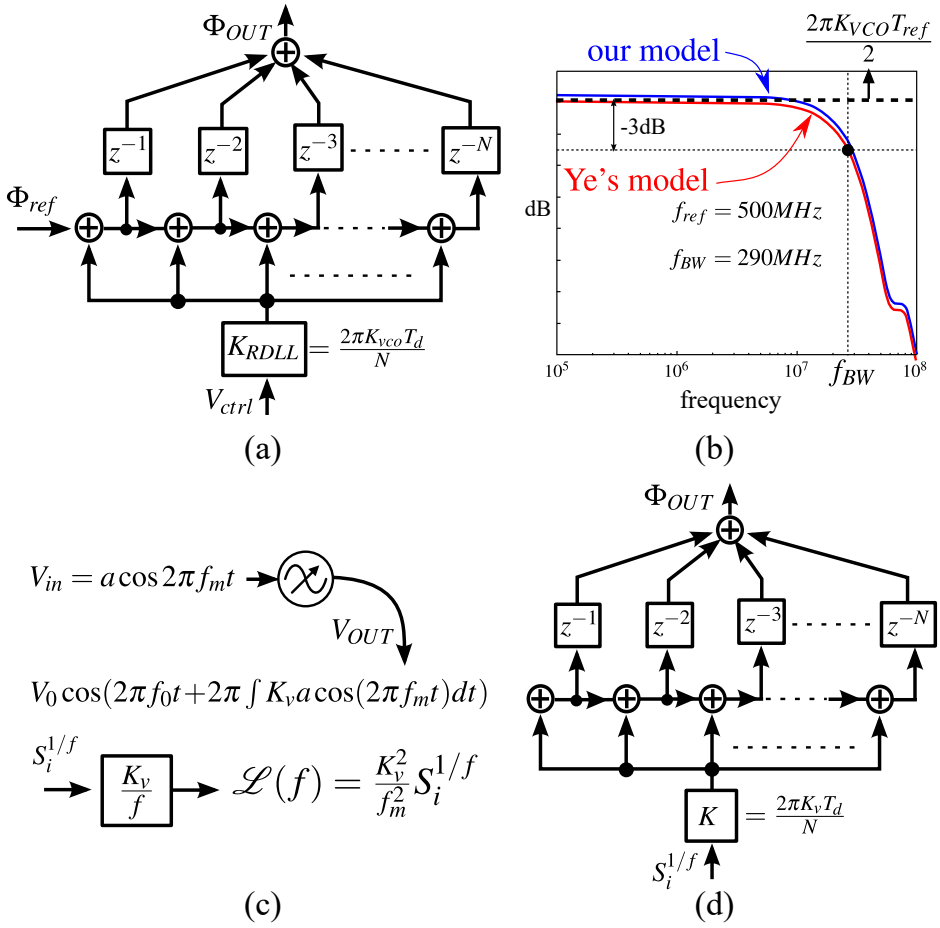


Figure 2.16: (a),(b) RFM model and transfer function comparison; (c) flicker noise model in a ring oscillator; and (d) flicker noise model in an RFM.

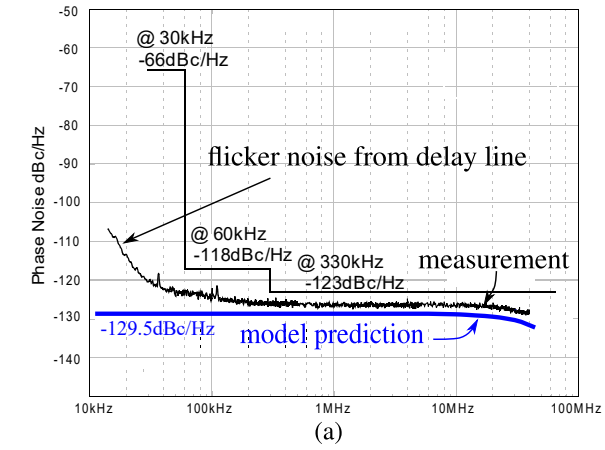
2.2.4 Model verification and comparison of two types of frequency multiplier

An ECDLL phase noise spectrum is reported in [9], and an RDLL phase noise spectrum is reported in [14]. Based on the circuit parameters provided in the two references, we predict the phase noises of the two DLLs by our model and compare them with the reported measurements. As shown in Fig. 2.17, at the moderate offset frequency, the system is dominated by thermal noise, and the model predictions match the measurements for both types of DLLs. For the ECDLL case, at the low offset frequency, as shown in Fig. 2.17(a), the measurements and predictions are discrepant because the flicker noise information is not reported in [9]. For the RDLL case, [14] reported the ring oscillator flicker noise, which allows us to verify our RDLL flicker noise model. [14] minimized the loop bandwidth of the DLL to suppress the quantization noise of bang-bang phase detector [30]. The phase noise of the frequency multiplier is negligibly filtered and thus determines the spectrum of the measurement. As shown in Fig. 2.17(b), the predictions of the open loop RFM model are consistent with the measurements, validating our model.

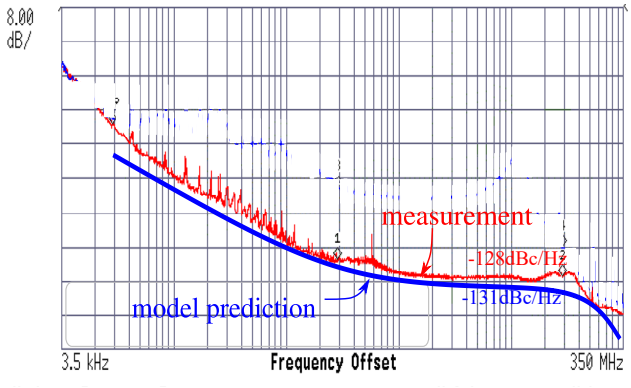
In normal circumstances, thermal noise is the main noise contributor in a DLL. It is uncorrelated between buffers in an ECFM and between cycles in an RFM. Equation (2.17) can be used to predict the thermal noise effect for both frequency multipliers. Within $f_{ref}/3$, where the influence of the delay block z^{-n} is negligible, the frequency multiplier phase noise in both types of DLLs is flat with the level of:

$$\mathcal{L}_{MDL}(f) = \frac{2\pi^2(N+1)(2N+1)}{3}\sigma_n^2 f_d \propto \frac{N^2}{I_{on}} \tag{2.25}$$

where I_{on} is the on-current of a delay cell. This current mainly determines DLL power. As evidenced in (2.25), the phase noise level in DLL depends only on the power and the multiplying ratio $N = f_d/f_{ref}$. Intuitively, a smaller multiplying ratio implies more frequent refreshments of the noisy output clock, which should reduce jitter accumulation or phase noise. Since the system requirement usually sets the DLL output frequency, one of two approaches can lower the phase noise: (1) raising the reference clock; (2) increasing power consumption. Fig. 2.19 presents the phase noise performances of DLLs with different power



(a)



(b)

Figure 2.17: (a) A nine-stage ECDLL with the reference frequency of 100 MHz and output frequency of 900 MHz. The differential pair jitter equation in [1] calculates the periodic jitter, where the effective gate voltage of the differential pair is 0.5 V, the effective gate voltage of the tail FET is 0.5 V, and the output swing is 0.5V; and (b) an RDLL with a multiplying ratio of 4. Its driven frequency is 375 MHz, and output frequency is 1.5 GHz. It consumes 0.45 uW with a supply of 1.1 V. The FET threshold is 0.3 V. The jitter is calculated by the inverter jitter equation in [1].

consumptions and multiplying ratios.

[1] demonstrates that a ring oscillator phase noise is:

$$\mathcal{L}_{ring}(f) = \frac{\sigma_n^2 f_d^3}{f^2} \tag{2.26}$$

In a DLL, the modification of ring provides a first order high-pass filter, which flats $\mathcal{L}_{ring}$

and sets the in-band noise floor to (2.17). As shown in Fig. 2.18, f_{NBW} [2] is defined as the offset frequency, at which the spot noise of the ring equals (2.17):

$$f_{NBW} = \frac{\sqrt{3} f_{ref}}{2\pi} \tag{2.27}$$

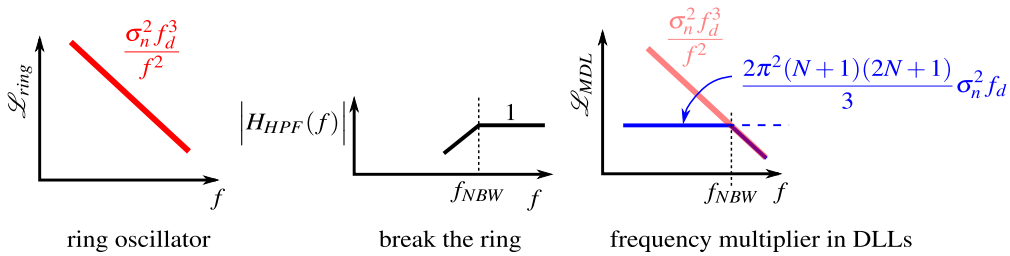


Figure 2.18: DLL noise suppression bandwidth.

With this extra suppression, DLLs do not rely on the loop to filter the VCO noise, decoupling the trade-off between the noise of VCO and other loop components. In other words, we can design a minimal loop bandwidth to suppress the input-referred noise and rely on the modification of the ring to suppress the oscillator noise.

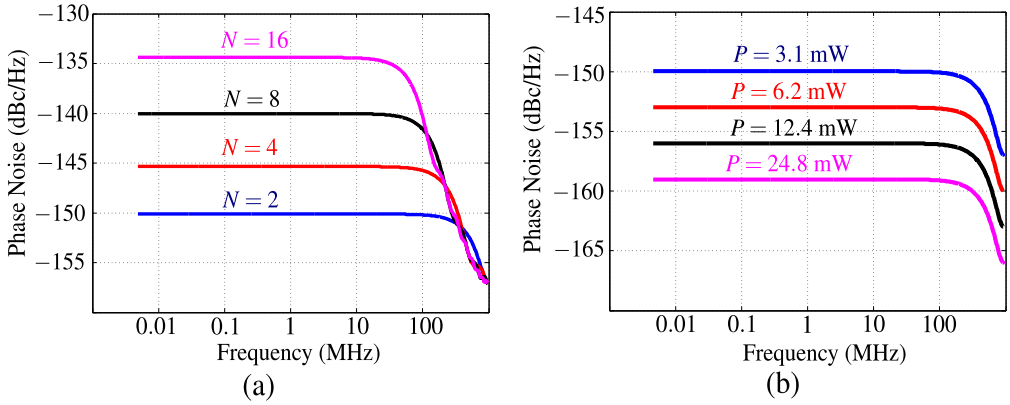


Figure 2.19: (a) Calculated phase noise of a frequency multiplier with different multiplying factors when consuming 3 mW power; and (b) calculated phase noise of a frequency multiplier with different power consumption when the multiplying ratio is 2. The reference frequency in both cases is 1 GHz.

The LTI model of frequency multiplier enables a fair comparison between the two types of DLLs. As shown in Fig. 2.20, the two types of frequency multipliers operate at the same

[2]The frequency multiplication factor is assume to be >> 1

frequency with the same reference. To achieve the same phase noise, as indicated by (2.25), the inverters in both DLLs should have the same on-current I_{on}. This current determines the propagation delay of the inverter, which is denoted by t_{PEC} and t_{PR} in the ECFM and the RFM, respectively. In the ECFM, since the buffers run at the frequency of the reference, the ECFM power is calculated as:

$$P_{EC} = 2NC_{EC}V_{dd}^2 f_{ref} = 4NI_{on}t_{PEC}V_{dd}f_{ref} = 2I_{on}V_{dd} \tag{2.28}$$

On the other hand, each inverter in an RFM operates at the output frequency, and thus the RFM power is given by:

$$P_R = kC_R V_{dd}^2 f_{OUT} = 2kI_{on}t_{PR}V_{dd}f_{OUT} = I_{on}V_{dd} \tag{2.29}$$

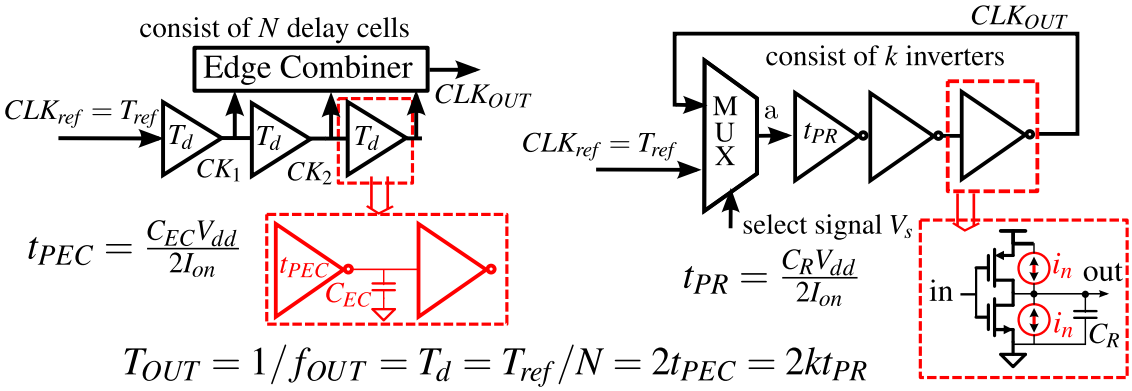


Figure 2.20: ECFM and RFM comparison.

Therefore, for the given noise performance, an RFM is more power-efficient than an ECFM. Intuitively, one understands that, in an RFM, both the rising edge and the falling edge propagation delays of an inverter contribute to the total delay of the RFM. In an ECFM, only one of the edge delays contributes to the delay line. To achieve the same total delay which is a reference cycle, the number of inverters (total number of capacitors) in an ECFM must be doubled, which increases the power consumption.

2.3 PHASE NOISE AND STABILITY ANALYSIS OF CLOSED LOOP DLL

2.3.1 Closed loop ECDLL model

Once the ECFM transfer function is known, the complete ECDLL LTI model can be built, as shown in Fig. 2.21(a). K_{PFD} is the gain of the phase detector. The transfer function of the loop low-pass filter, which is conventionally an integrator in a DLL, is given by:

$$H(z) = \frac{\alpha T_d}{1 - z^{-1}} \tag{2.30}$$

where α is a dimensionless coefficient. Apparently, the edge combiner circuitry is outside of the feedback loop. When considering loop dynamics, the ECDLL model can be simplified to Fig. 2.21(b), which approaches the model derived in [31]. The loop gain $T(z)$, the most important determinant of a feedback system, is simply the product of all transfer functions in one traversal of the loop:

$$T(z) = K_{PFD} \cdot H(z) \cdot K_{DL} \cdot z^{-N} \tag{2.31}$$

As T(z) holds only one dc pole, ascribed to the integrator of the loop low-pass filter, an ECDLL is much more immune to loop instability than a PLL.

In the loop model of Fig. 2.22, three sources of noise from independent contributors are injected into appropriate nodes of an ECDLL. They are the phase noise injected by each buffer of the delay line (Φ_{DL_n}), input referred phase noise associated with the phase detector and the loop filter (Φ_{in}), and the phase noise of the reference (Φ_{ref}). To simplify the noise analysis, two intermediate transfer functions are derived as:

$$H_{Ctrl1}(z) = \frac{V_{ctrl}}{\Phi_{DLn}} = -\frac{T(z)/K_{DL}}{T(z)+1} \tag{2.32}$$

$$H_{Ctrl2}(z) = \frac{V_{ctrl}}{\Phi_{in}} = H_{Ctrl1}(z)z^N \tag{2.33}$$

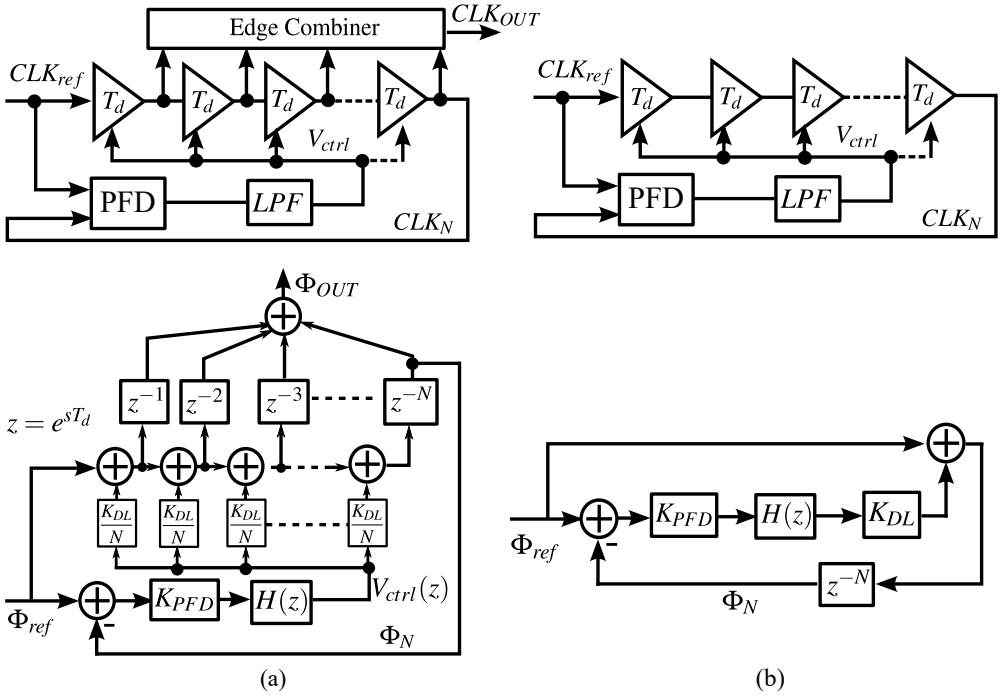


Figure 2.21: Closed loop ECDLL model.

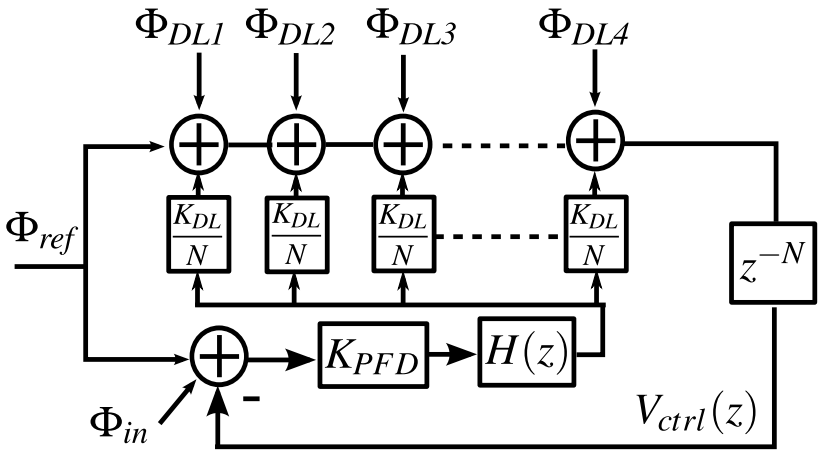


Figure 2.22: Loop model that considers delay cell noise and input feedthrough effect. ϕ_{DLn} represents n^{th} delay cell noise. ϕ_{in} represents input referred noise. ϕ_{ref} is reference phase noise.

The NTF from the j^{th} buffer of the delay line to the ECDLL output is then given by:

$$\begin{aligned} H_j(z) &= \frac{\Phi_{out}}{\Phi_{DLj}} \\ &= \sum_{k=j}^{N} z^{-(k)} + \sum_{k=1}^{N} k \frac{K_{DL}}{N} H_{Ctrl1}(z) z^{-(k)} \end{aligned} \tag{2.34}$$

Hence, the total phase noise contributed by the delay line is:

$$\mathcal{L}_{DL}(f) = \sum_{j=1}^{N} \mathcal{L}_{\phi_{DLj}}(f)|H_j(f)|^2 \tag{2.35}$$

where $\mathcal{L}_{\phi_{DLj}}(f)$ is the phase noise of the j^{th} delay cell.

The NTF from Φ_{in} to the edge combiner output is:

$$H_{input}(z) = \frac{\Phi_{out}}{\Phi_{in}} = \sum_{k=1}^{N} k\frac{K_{DL}}{N} H_{Ctrl2}(z)z^{-(k)} \tag{2.36}$$

The output phase noise corresponding to Φ_{in} is then given by:

$$\mathcal{L}_{IN}(f) = \mathcal{L}_{\phi_{in}}(f)|H_{input}(f)|^2 \tag{2.37}$$

where $\mathcal{L}_{\phi_{in}}(f)$ is the input referred phase noise of the PFD and the low-pass filter.

The NTF from reference to the output is :

$$H_{ref}(z) = \frac{\Phi_{out}}{\Phi_{ref}} = H_1(z) + H_{input}(z) \tag{2.38}$$

Thus, the output phase noise associated with the reference is:

$$\mathcal{L}_{ref}(f) = \mathcal{L}_{\phi_{ref}}(f)|H_{ref}(f)|^2 \tag{2.39}$$

The three sources of noise superpose at the ECDLL output as follows:

$$\mathcal{L}(f) = \mathcal{L}_{DL}(f) + \mathcal{L}_{IN}(f) + \mathcal{L}_{ref}(f) \tag{2.40}$$

2.3.2 Closed loop RDLL model

Unlike an ECDLL, in which the frequency multiplier outputs an open loop signal, the RFM in an RDLL is a part of the feedback control loop. The frequency divider lowers the RFM output to the reference frequency. A PFD compares the divided clock with the reference, and sends the phase error to the LPF to update the frequency control signal. The whole LTI

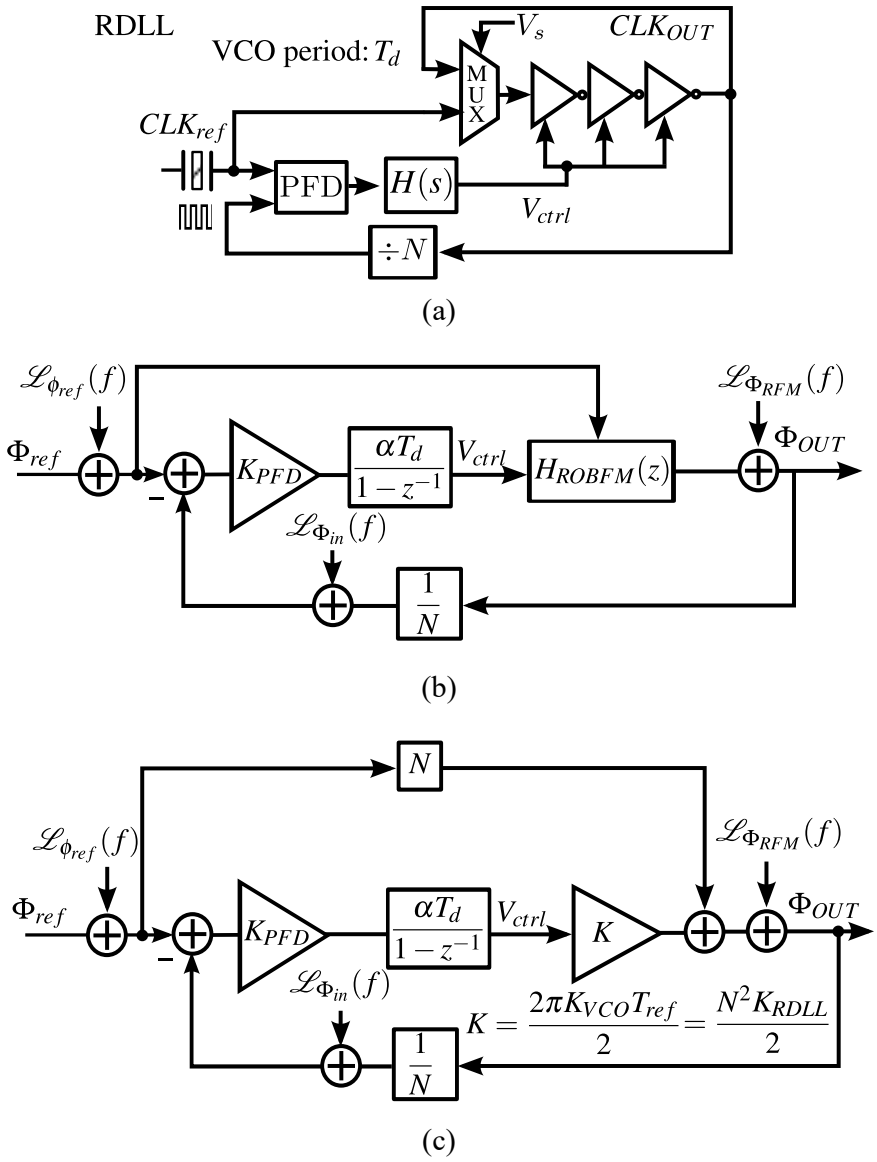


Figure 2.23: Closed loop model of RDLL.

model is shown in Fig. 2.23(b). As presented in Fig. 2.16(b), unlike a ring oscillator, which operates as an integrator, the RFM operates as a constant gain block until $f_{ref}/3$ and then rolls off. Since the high-frequency pole of RFM erodes the loop phase margin negligibly, in practical implementations, we can simplify RFM as $\pi K_{vco} T_{ref}$. A qualitative understanding of this constant is that with a unit control voltage, although the RDLL output phase varies in a sawtooth manner in each reference cycle, as shown in Fig. 2.24, the low-pass filter in the feedback loop ensures that the loop senses only the average phase shift of RFM $\pi K_{vco} T_{ref}$.

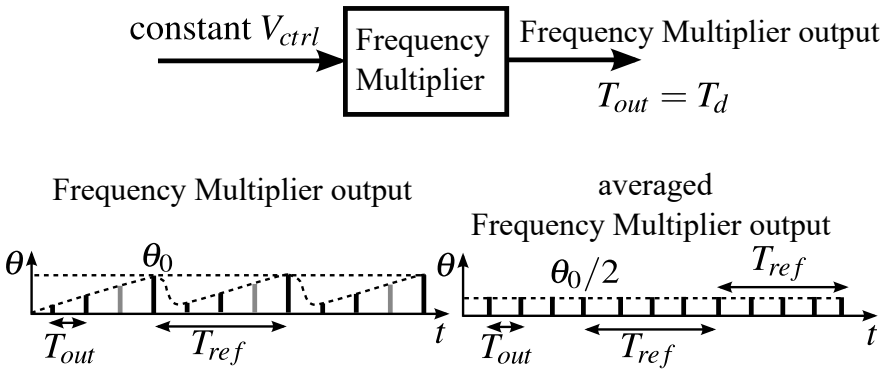


Figure 2.24: Explanation of RFM model simplification.

As the reference jitter is directly injected into the RFM, a feedforward path is established. The simplified feedforward gain is N, which can be obtained by the model in Fig. 2.16(a). Conceptually, N is a frequency multiplication factor.

2.3.3 Closed loop noise comparison in the two type DLLs and model verification

We compared our ECDLL model with the one in [2]. As shown in Fig. 2.25 (b), (c), the proposed transfer functions from the reference and PFD inputs to the ECDLL output matched the results in [2] up to f_{ref}/N. At higher frequencies, discrepancies arise in the transfer functions, but are acceptable since the dominant noise is from the delay line. When considering the transfer function of delay line noise, within the frequency where the loop high-pass filter effect emerges, as shown in Fig. 2.25(a), our proposed NTF gain is around 3 dB lower than [2]. Within this frequency range, the noise is dominated by sources other than the delay line, and thus the discrepancy is acceptable. As the high pass filtering effect fades, the transfer functions of both models coincide. At around f_{ref}/N, the transfer function rolls off as the ECFM FIR filter takes effect. This behavior was well captured by our model, as shown in Fig. 2.17. In contrast, the transfer function in [2] remained constant, causing a discrepancy. From the data in [4], we compared the results of our analysis with the predictions of [2] and the reported measurements. As shown in Fig. 2.27(a), for three different multiplication factors (N= 2, 3, 6), our predictions completely overlap those of [2] and matches well with the measurements.

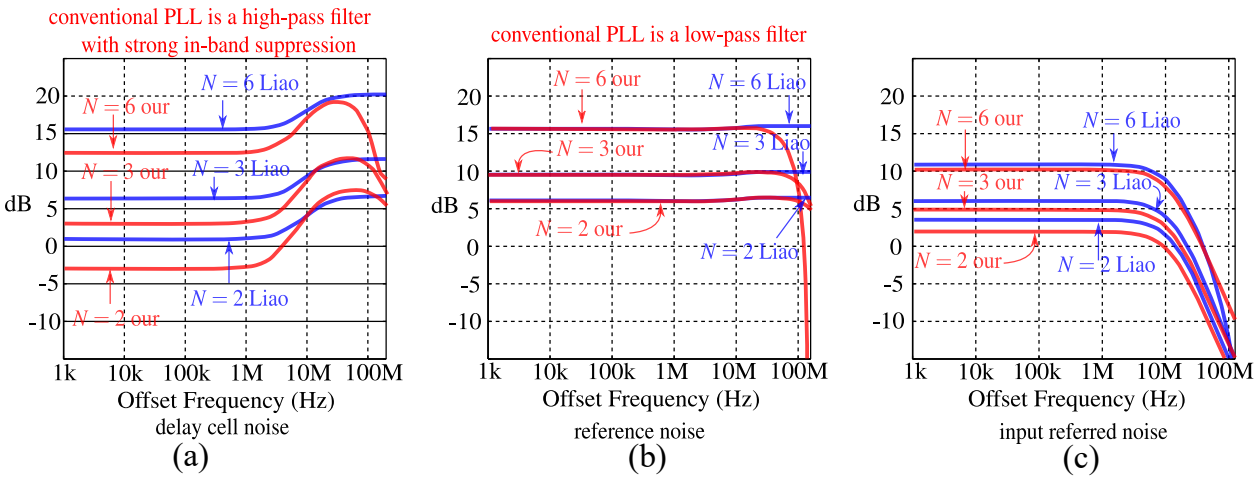


Figure 2.25: (a) Transfer function from delay line noise to output; (b) transfer function from reference phase noise to output; (c) transfer function from input referred noise to output. The data are from [2] where $f_{ref} = 850$ MHz, $N = 6$, $K_{VCDL} = 2.56$ ns/V, $I = 100$ uA, $C = 3$ pF, and $N = 6, 3, 2$ respectively.

Both ECDLL models (our proposed model and that of [2]) confirmed that an ECDLL can merely filter the delay cell noise, as presented in Fig. 2.25 (a), which supports the assertion in [18] that the function of the ECDLL control loop is not to remove the jitter of the voltage controlled delay line (VCDL), but merely to tune the total delay of the VCDL to the desired value. Moreover, if the flicker noise corner is too high, which is the case in modern CMOS technologies [3], the delay line flicker noise, which experiences weak attenuation, contributes significantly to the output.

An accurate model that describes the properties of RDLLs was established in [13]. We now compare our simple model with that of [13]. As shown in Fig. 2.26, within $f_{ref}/3$, the predictions of the simple model and [13] are completely matched. The discrepancy at higher frequencies is ascribed to our exclusion of the high-frequency roll-off in RFM. However, in that frequency range, the RDLL phase noise is mainly introduced by the frequency multiplier, which is precisely modeled in 2.2 and under negligible filtering. Thus, our simple model still provides accurate noise prediction. As shown in Fig. 2.27(b), despite some discrepancies in

[3] As an example, in 16nm Finfet, the flicker noise corner of the 5GHz ring oscillator is approximately 10 MHz

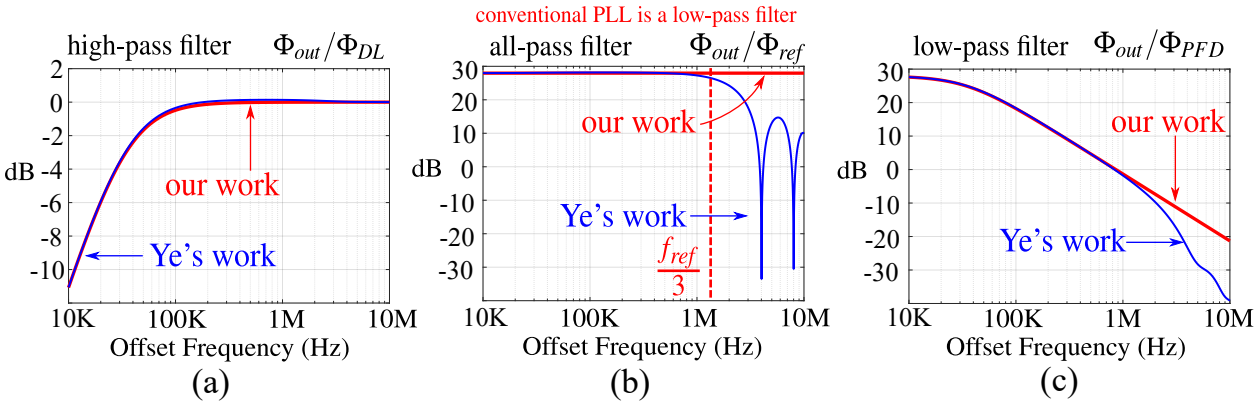


Figure 2.26: (a) RDLL transfer function from VCO noise to output; (b) RDLL transfer function from input referred noise to output; and (c) RDLL transfer function from reference phase noise to output. The data are from [3] where $f_{ref} = 4$ MHz, $N = 25$, $K_{VCO} = 190$ MHz/V, $I = 0.5$ mA, $R = 0$ ohm, and $C = 2.2$ nF, respectively.

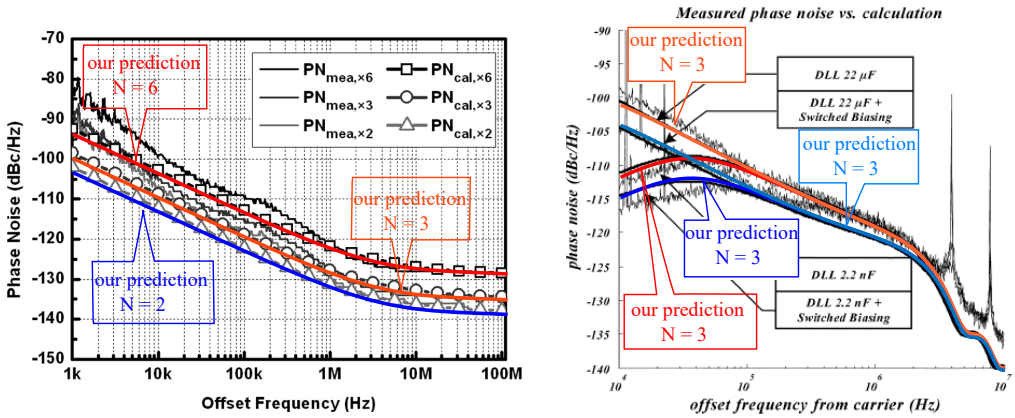


Figure 2.27: (a) ECDLL phase noise prediction vs. measurement. Noise data are obtained from [4]; and (b) RDLL phase noise prediction vs measurement. Noise data are obtained from [5].

the transfer functions, our model predictions mainly overlap those of [13] and closely match the measurements.

Observing Fig. 2.25(a) and Fig. 2.26(a), we find that the RDLL possesses a strong high-pass filter property, which largely suppresses flicker noise. From this perspective, the RDLL outperforms the ECDLL.

Unlike a conventional PLL, neither the ECDLL nor the RDLL can effectively filter the reference noise, as shown in Fig. 2.25(b) and Fig. 2.26(b), respectively. This is a main draw-

back that limits the implementation of DLLs. It will be further discussed in 2.5.

2.4 DLL spurs

2.4.1 ECDLL spurs

In an ideal ECDLL, the high-frequency clock is generated by adding all of the equally delayed clock edges from the delay line outputs. As shown in Fig. 2.28, random mismatches among the delay cells of a real ECDLL cause unequal propagation delays, resulting in duty cycle errors at the output. These errors manifest as spurs in the frequency domain. In an ensemble of ECDLLs, the duty cycle errors should be Gaussian-distributed according to the central limit theorem [28]. The spurs, which are related to the power of the duty cycle error, should then follow the Rayleigh distribution [28]. This statistical property of ECDLL spurs was confirmed in [32].

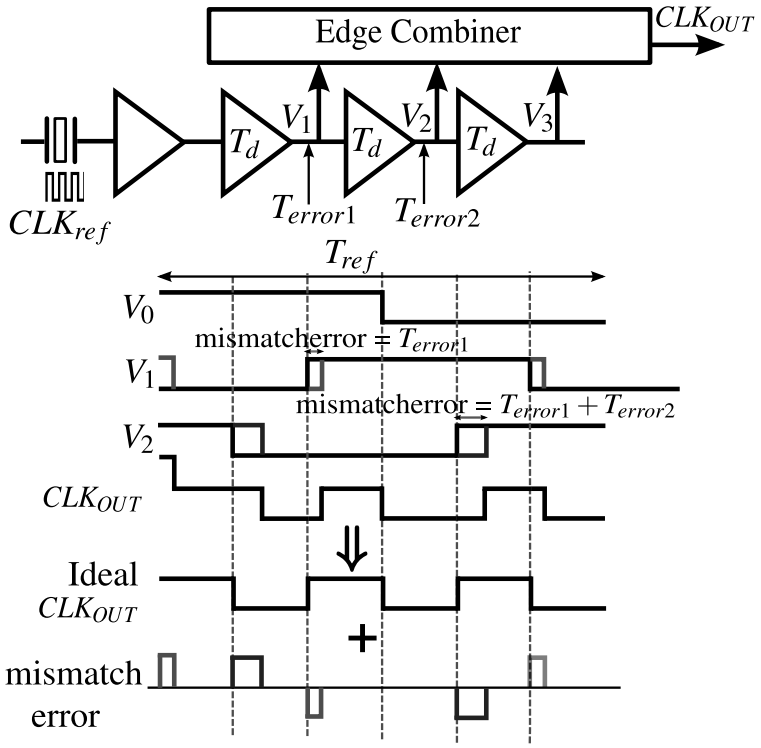


Figure 2.28: Spurs decomposition in ECDLL.

We have mature knowledge of analyzing mismatches in an amplifier. For example, we

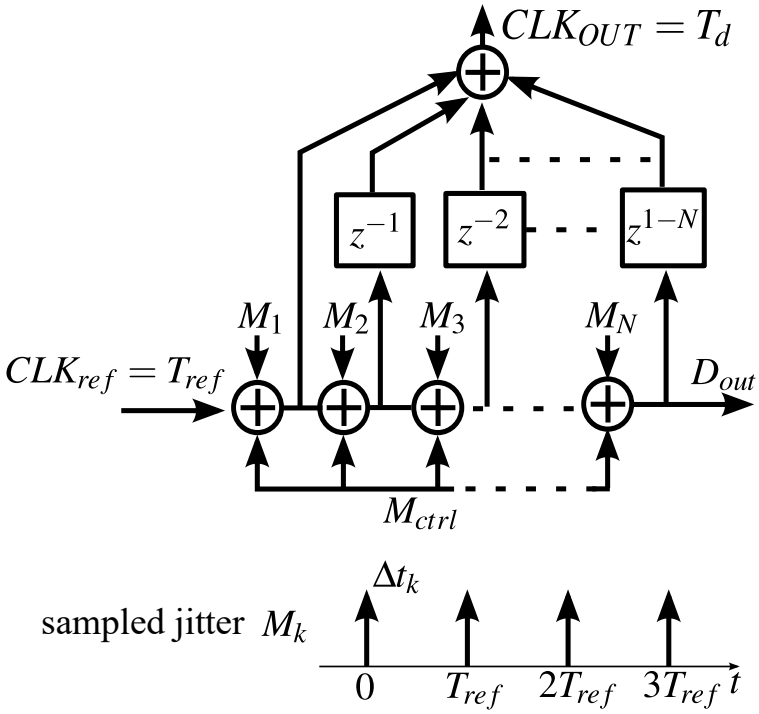


Figure 2.29: The source of spurs in ECDLL. Each delay cell provides mismatch error, which is an impulse train.

model the amplifier unbalances due to small mismatch with differential-mode currents inserted into an otherwise perfectly balanced linear circuit free of mismatch. The benefit is that, even with the existence of mismatch, the amplifier can still be treated as an LTI system.

Here, we follow this concept and regard the small mismatch of each buffer as an extra time error which injects into the mismatch-free ECFM. For a certain delay buffer in a given ECFM [4], the mismatch effect is unchanged and periodically alternates the duty cycle of the edge combiner output. The mismatch error of the k^{th} buffer is captured at the edge-crossing of the buffer, and is represented as a deterministic impulse train $M_n[k]$ with period T_{ref} and constant magnitude Δt_k, as shown in Fig. 2.29. The ensemble of Δt_k is generally assumed to follow a Gaussian distribution with zero expectation and variance $\sigma^2_{\Delta t_k}$ [18, 19]. The perfectly matched ECFM is the discussed linear FIR filter which allows us to easily develop linear transfer functions from each injected mismatch error to the output.

[4]Since spurs are non-ergodic, we start the study of spurs as deterministic signals, and then take their ensemble random property to reach the stochastic conclusion.

At steady state, the final stage of the delay line is aligned with the reference by a loop control signal:

$$M_{ctrl}[k] = -\frac{1}{N}\sum_{n=1}^{n=N} M_n[k] \tag{2.41}$$

Based on [33], the sampled autocorrelation function of the mismatch effect in a certain delay cell is:

$$R_{M_k}(t) = \frac{(\Delta t_k)^2}{T_{ref}}\sum_n \delta(t - nT_{ref}) \tag{2.42}$$

The expected PSD of this impulse train is:

$$\mathcal{E}\{S_{M_k}(f)\} = \frac{\sigma^2_{\Delta t_k}}{T^2_{ref}}\sum_n \delta(f - nf_{ref}) \tag{2.43}$$

From Fig. 2.29, the transfer function from the control signal to the edge combiner output is obtained as:

$$H_{ctrl}(z) = \frac{\phi_{out}(z)}{M_{ctrl}(z)} = \sum_{n=1}^{N} nz^n \tag{2.44}$$

Meanwhile, the transfer function from the k^{th} delay cell to the edge combiner output is:

$$H_k(z) = \frac{\phi_{out}(z)}{M_k(z)} = z^{-(k)}\sum_{n=0}^{N-k} z^n \tag{2.45}$$

Normally, the device mismatch in a buffer is small. Superposition still applies. By linearly adding up all the mismatch effects, the output spurs of the edge combiner are calculated as:

$$spur(z) = \sum_{k=1}^{N} M_k(z)H_k(z) + M_{ctrl}(z)H_{ctrl}(z) \tag{2.46}$$

After simple algebra, (2.46) is:

$$spur(z) = \sum_{k=1}^{N} M_k(z)\left(\frac{z^{-(k)}}{1-z^{-1}} - \frac{1}{N}\frac{z^{-1} - z^{-N-1}}{(1-z^{-1})^2}\right) \tag{2.47}$$

where $z^{-1} = e^{-j2\pi f T_d}$; and T_d is the ECDLL output period, which equals T_{ref}/N. Because the mismatch effects from different delay cells are uncorrelated, the expected PSD of ECDLL spurs is:

$$S_{spur} = \sum_{k=1}^{N} \mathcal{E}\{S_{M_k}(z)\}\left|\frac{z^{-(k-1)}}{1-z^{-1}} - \frac{1}{N}\frac{1-z^{-N}}{(1-z^{-1})^2}\right|^2 \tag{2.48}$$

Spurious tones appear at offset frequency of f_{ref} and its harmonics. At these frequencies, z^{-N} is always 1. Therefore, (2.48) further simplifies to:

$$S_{spur} = \frac{\sigma^2_{\Delta t_k}}{T^2_{ref}} \times \frac{N^3}{4\pi^2} \tag{2.49}$$

To coordinate (2.49) with phase, we must multiply it by $(2\pi/T_d)^2$ [34] [35]. As the mismatch effect continuously influences the clock, we need another coefficient T_d^2 that back-transfers the sampled mismatch effect to a continuous-time effect:

$$S_{spur} = \frac{\sigma^2_{\Delta t_k}}{T^2_{ref}} \times \frac{N^3}{4\pi^2} \frac{4\pi^2}{T_d^2} T_d^2 = \frac{\sigma^2_{\Delta t_k} N^3}{T^2_{ref}} \tag{2.50}$$

(2.50) is identical to the result of [36]. Our proposed LTI model approaches the same conclusion as the direct Fourier analysis, demonstrating that the proposed ECDLL model captures the essence of the circuit.

2.4.2 RDLL spurs

Spurs in an RDLL are mainly sourced from PFD mismatches, CP mismatches, and injection switch delays. However, the spur mechanism in the RDLL differs from that in the PLL, where the mismatches generate ripples on the control signal, thereby modulating the oscillator frequency. As shown in Fig. 2.30, the mismatches in an RDLL generate a static phase error between the reference and the RDLL output. To compensate the static phase errors in steady state, the RDLL must deviate from its desired frequency, introducing a static frequency error related to the ideal frequency. Such a frequency error induces a continuously accumulating phase error, which is eventually reset by the reference clock injection. Therefore, for a given chip with certain static phase error, the RDLL output phase error always follows a sawtooth shape, which can be analyzed by an LTI system, as shown in Fig. 2.31.

Through simple algebra, which is similar to the analysis of ECDLL spur, the first spurious tone of RDLL is:

$$\mathcal{E}\{S_{spur}\} = \frac{\sigma^2_{SPO} N^2}{T^2_{ref}} = \frac{\sigma^2_{SPO}}{T_d^2} \tag{2.51}$$

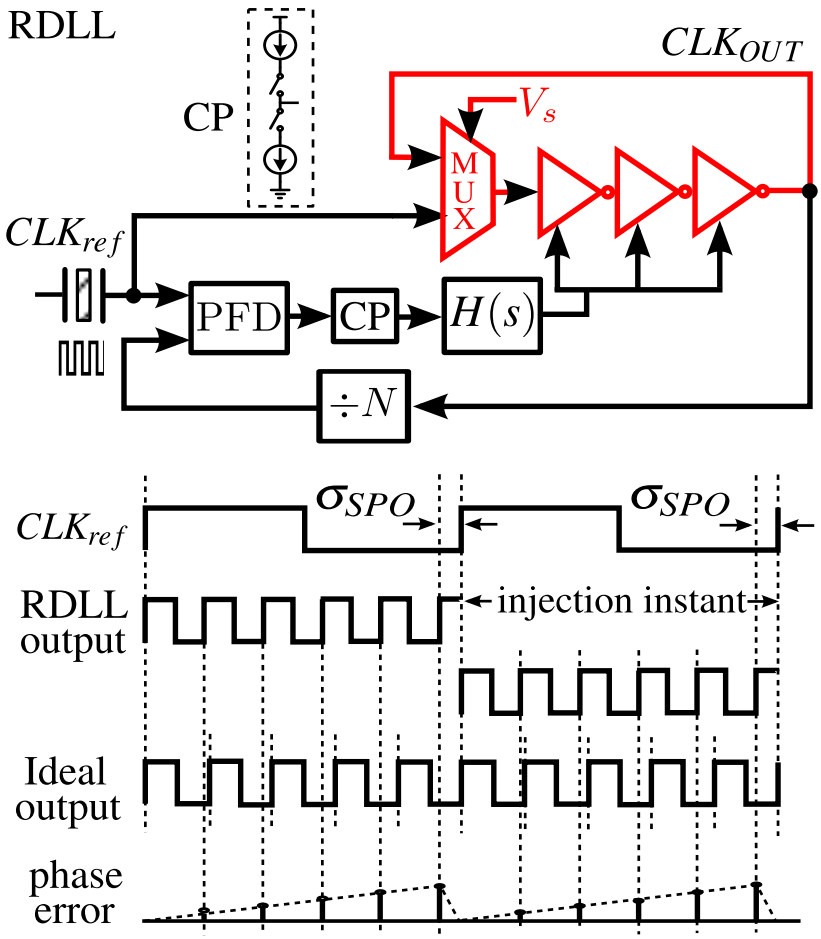


Figure 2.30: RDLL spur mechanism.

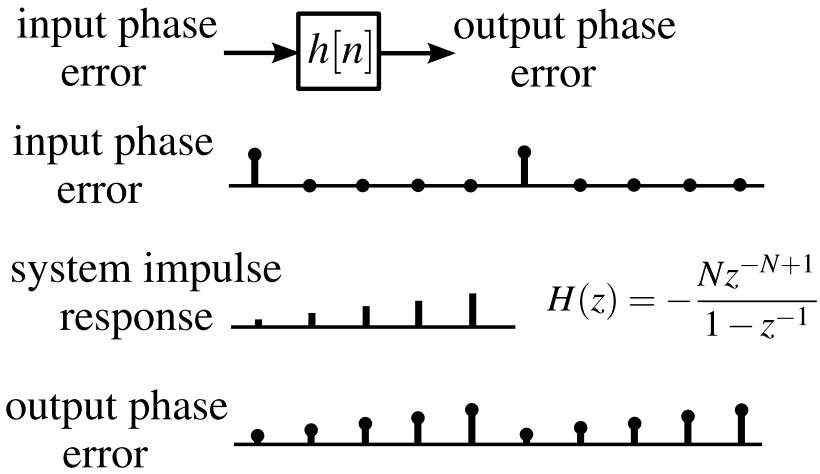


Figure 2.31: An LTI model to analyze RDLL spurs.

, where T_d is again the DLL output period. An identical equation to (2.51) was presented in [24]. The spurs in RDLL are less amplified by the frequency multiplication factor than the ones in ECDLL. Thus, when the multiplication factor is large, an RDLL is a better choice than an ECDLL.

2.5 Discussion of DLL implementations

As the RDLL outperformed the ECDLL in all of the above analyses, it is the primary choice of DLL architecture.

The most general implementation of RDLL is a local clock generator for a digital system, in which RMS jitter is an issue of concern and only coarse frequency tuning resolution is required. On the other hand, wireless implementation requires an RF synthesizer for high-resolution frequency tuning and low spot phase noise. Implementing the DLL into the RF system remains an ongoing research topic. Based on the proposed theory, this section explores the possibilities of the DLL in RF implementations.

2.5.1 RDLL as a fractional-N synthesizer

The main challenge in designing a fractional-N DLL is the fractional injection of the reference clock. In [37], the reference injection instance was adjusted by a digital-to-time converter (DTC), as shown in Fig. 2.32. The DTC output and frequency multiplier output were then adaptively aligned by calibration, achieving precise clock injection.

However, such a structure has several disadvantages. First, although DTC cancels the quantization noise of the sigma-delta modulator (SDM), due to the limited time resolution, there will be residual noise. A PLL removes the residue by loop filtering, whereas a DLL directly injects it to the output through the same path as the reference noise, which, as shown in Fig. 2.33, experiences no filtering.

As reported in [37], the DTC calibration canceled 52 dB of quantization noise, but the remaining quantization noise, experiencing no further attenuation, contributes a noise level

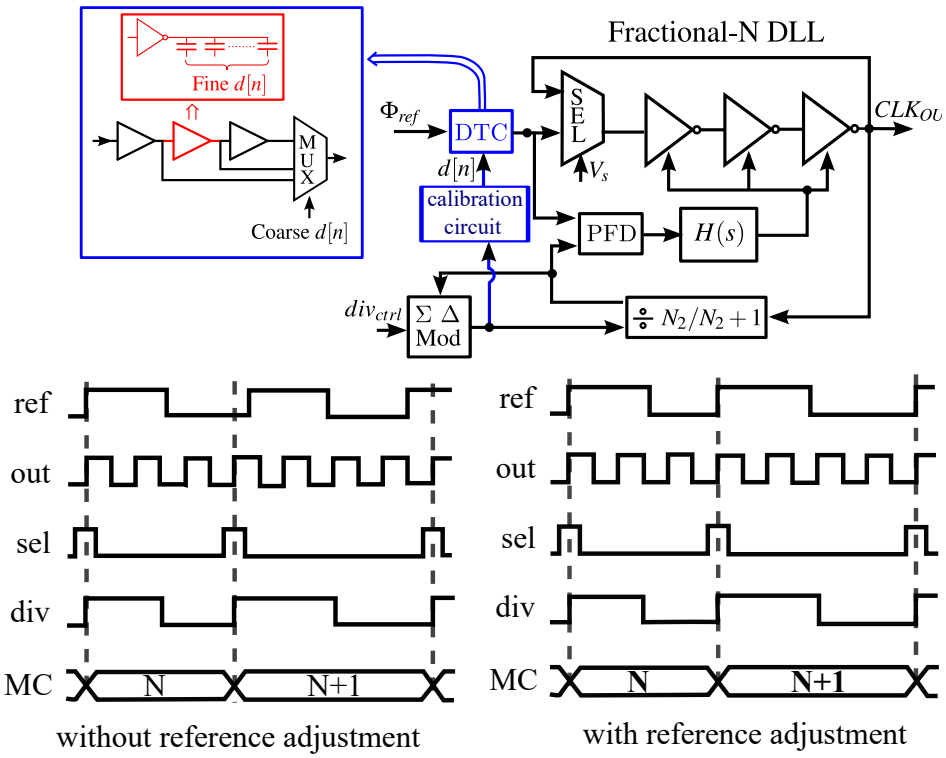


Figure 2.32: Architecture of a fractional-N DLL.

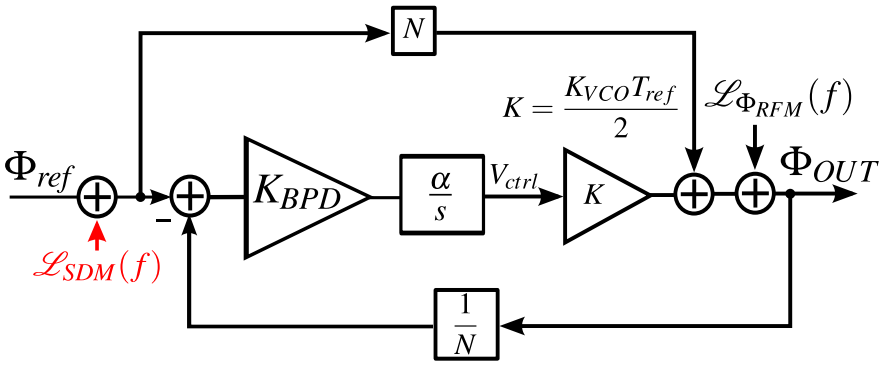


Figure 2.33: Fractional-N DLL model. The remaining SDM noise is injected at the reference ports and cannot be filtered by the DLL loop.

of approximately -115 dBc/Hz at the DLL output. Besides, the DTC nonlinearity folds the quantization noise and generates at worst -47 dBc fractional spurs. All of these spurs are directly injected into the DLL output, again enjoying no attenuation. Moreover, the reported -55 dBc reference spurs remain problematic. Above all, the implementation of fractional-N DLL is limited to Bluetooth and some wireless standards with relaxed noise and spur requirements.

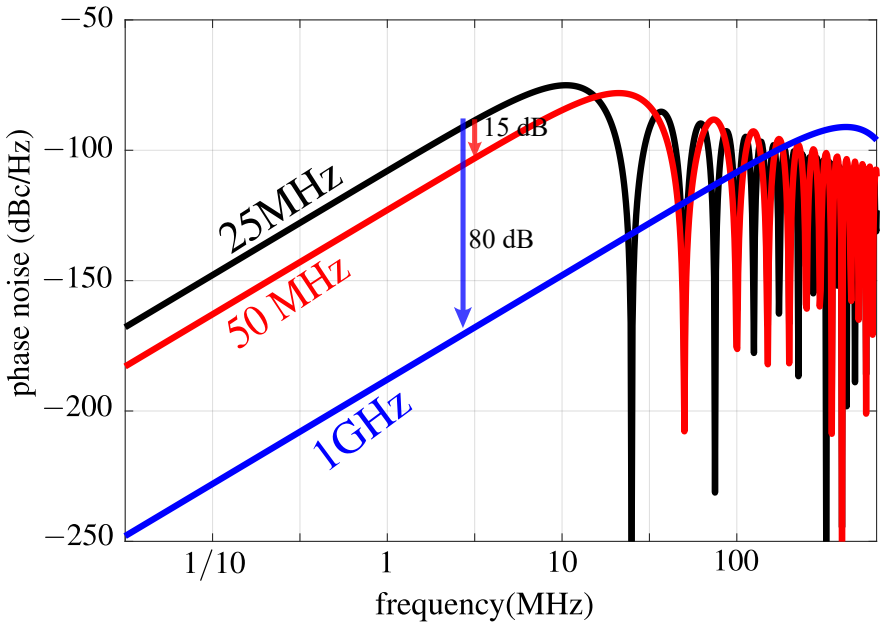


Figure 2.34: Increasing SDM driving frequency can lower quantization noise.

2.5.2 RDLL as a reference clock frequency multiplier

For wireless standards, such as GSM and Wi-Fi, a synthesizer needs high purity of spectrum. In such a case, the phase noise performance is often limited by the SDM quantization noise (2.52), which was modeled in [38]:

$$S_{sdm} = \frac{4\pi^2 T_{ref}}{12} \frac{2\sin^{2m}(\pi f T_{ref})}{(fT_{ref})^2} \approx \frac{T_{ref}^{2m-2}(\pi f)^{2m-2}}{6} \tag{2.52}$$

where $T_{ref} = 1/f_{ref}$ is the SDM driving period; and m is the order of SDM. The SDM noise can be efficiently reduced by increasing the synthesizer reference frequency. The Increment of the reference frequency accelerates the SDM operation, broadening the bandwidth of the quantization noise. As the total variance in the quantization noise is unchanged, the widely distributed noise has lower noise floor. For instance, as shown in Fig. 3.18, doubling the driving frequency lowers the in-band noise floor of the third-order SDM by 15 dB. This noise can be efficiently filtered by a 1 MHz-bandwidth PLL. If the driving frequency can be increased to 1 GHz, the quantization noise level is decreased by 80 dB, which can be filtered easily.

An RDLL is a good choice for reference frequency multiplication to increase the updating

rate of SDM. It needs only to ensure that its own phase noise is lower than:

$$\mathcal{L}_{DLL} = \mathcal{L}_{\phi_{ref}} + 20log_{10}N_{DLL} \tag{2.53}$$

where $\mathcal{L}_{\phi_{ref}}$ usually represents the noise floor of the crystal and N_{DLL} is the multiplication factor of the DLL. Then, the noise contribution of the DLL-based frequency multiplier is acceptable. The cascaded PLL can efficiently filter the DLL spurs. For instance, without any filtering, a DLL reference spurs are at the level of -40 dBc when its carrier frequency is normalized to 2.4 GHz. Normally the reference frequency is around 20 to 40 MHz. By designing the PLL with a 1 MHz loop bandwidth and a third-order filter strength, the cascaded loop can lower the DLL spur to -100 dBc, which meets most of the wireless standards requirements. Therefore, RDLL is suitable for intermediate frequency multiplication in synthesizer design Fig. 2.35 [23, 39, 32].

Except for a DLL, either an injection-locking oscillator [40, 13] or an integer-N PLL can work as the pre-multiplier. Which of the three is most suitable as a pre-multiplier? To answer this question, the design variable that we need to look into is the previous defined noise suppression bandwidth f_{NBW}. As demonstrated, by breaking the ring, a DLL provides a first-order filtering to the ring oscillator. The noise suppression bandwidth reaches around:

$$f_{NBW_{DLL}} = \frac{\sqrt{3}f_{ref}}{2\pi} \approx \frac{f_{ref}}{3.6} \tag{2.54}$$

For an integer-N PLL, the control loop is a first-order high-pass filter for the VCO[5]. The unity-gain bandwidth of the loop determines its noise suppression bandwidth, which however is limited by the loop dynamics. Based on the report [41] and our next chapter analysis, the largest noise suppression bandwidth of a PLL is :

$$f_{NBW_{PLL}} = \frac{f_{ref}}{10} \tag{2.55}$$

An injection-locking oscillator can also be regarded as a DLL, except the injection strength is not as strong as the one in a DLL because the periodic noise resetting cannot

[5]For a type-II PLL, since the zero is usually very small, we can also approximate it as a first-order filter

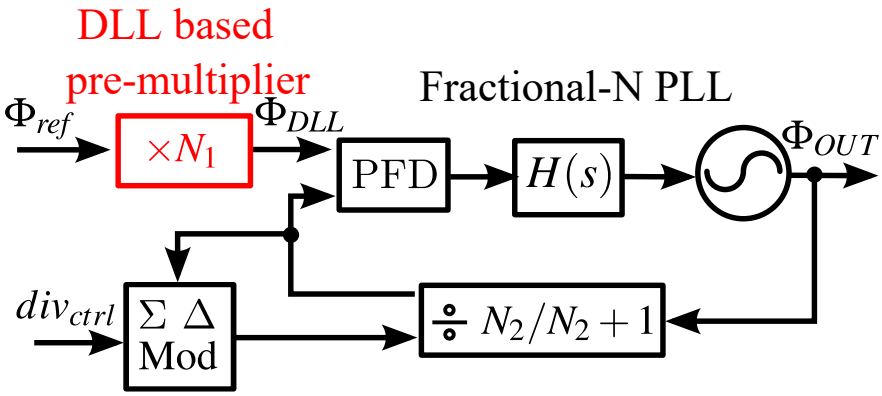


Figure 2.35: A fractional-N synthesizer with a DLL-Based frequency pre-Multiplier.

completely erase the integrated noise in the oscillator. Therefore, the suppression bandwidth is smaller than a DLL:

$$f_{NBW_{inj}} = \sqrt{\frac{f_{ref}}{\pi^2\left(\frac{1}{3}+\left(\frac{2}{\beta}-1\right)^2\right)}} \tag{2.56}$$

We use β to define the injection-locking strength. The appendix demonstrates the derivation of $f_{NBW_{inj}}$. When β equals one, the injection-locking oscillator merges to a DLL.

Among the three architectures, a DLL has the most significant noise suppression and thus is the best choice for the pre-multiplier.

However, if we can sacrifice the area to replace ring oscillator with an LC oscillator, an injection-locking oscillator or a PLL is a better choice since they can provide roughly extra $2Q^2$ noise suppression [1].

In the future, new architectures will be proposed as the frequency multiplier. No matter how the structures are, the essence is still to quantify the noise suppression bandwidth. Then we can use $f_{NBW_{DLL}}$ as a benchmark to determine the improvement from the new architecture.

CHAPTER 3

Harmonic Mixing Technique for Low Noise Clock Generation

3.1 Introduction

The previous study indicates that DLL is not suitable for the target wideband frequency synthesizer. PLL solutions are investigated next. As mentioned previously, the trade-off between the filtering of VCO noise and the attenuation of sigma-delta modulator quantization noise is the primary constraint for low noise synthesizer design. The fractional spurs, which is introduced by the loop non-linearity, is another cause of spectrum purity degradation.

In the early 2000s, it is prevalent to design a narrow bandwidth PLL with a power intensive VCO to meet the design targets [42, 43]. The watershed came when [44, 45] proposed the concept of SDM noise cancellation that seemingly overcame the trade-off in fractional-N synthesizer design for the first time. This cancellation is accomplished by injecting inverted SDM bits into the loop low pass filter (LPF) via a current-based DAC. This injected current cancels the SDM noise stored on the LPF. The drawback of this approach is the nonlinearity along the injection path causes unwanted noise-folding and in-band spurs. All-digital PLLs (ADPLLs) [46, 47, 48, 49], which convert the phase error into bits through a time-to-digital converter (TDC), can directly cancel the SDM noise in the digital domain, obviates the need of DAC. The main design challenge shifts to the linearity and quantization noise of the TDC. A gated ring oscillator (GRO) TDC with first-order noise shaping [49, 5] achieves a sub-ps resolution and therefore low quantization noise, but it requires gain calibration and suffers from nonlinearity caused by transistor leakage during the hold mode. The bang-bang phase detector, a one-bit TDC, has been demonstrated to be linear with low noise when its input

jitter is under a certain level [50, 30]. In the fractional mode, however, the PD gain is lowered by significant phase fluctuations from the fractional divider. Calibration can restore the gain, but it also creates unwanted spurs from its non-linearity [51, 52]. Subsampling PLLs [53] that utilize the harmonic of the reference to directly downconvert the VCO spectrum, avoid the frequency divider and thus achieve a unity closed loop transfer function. However, it also requires calibration for fractional operation [36].

Spurious tones are another factor that degrades the synthesizer spectrum purity. The reference spur, which is essentially phase disturbance at the reference clock frequency f_{ref}, usually experiences strong attenuation by the PLL. The fractional spur, which is mainly caused by phase detector nonlinearity [7, 8], is located at αf_{ref}. As mentioned in chapter 1, α can be arbitrarily small due to channel location requirement, which moves the fractional spurs in band. To cope with this, [7] proposes a new SDM code that is less susceptible to circuit nonlinearity and injects a static phase error into the loop that pushes the charge pump into the linear region. [4] demonstrates a digital calibration circuit that nullifies the spurs.

The prevalent trend in the literature is to use calibration to deal with synthesizer nonideality. However, this work begs to differ. First, the converging time of a calibration algorithm is not well-defined. Based on the study of [49, 54, 51], calibration takes at least 20 us to converge. In applications that require fast convergence, calibration is not suitable. Second, as the digital circuit complexity increases, the timing difficulty increases as well. To this end, a fast device with a lower threshold voltage is required. This, in turn, increases the leakage current with today's deep sub-micron process and increases the total power consumption. Last, as evident from [4], it takes 2mA to cancel one spurious tone of a PLL. If there are multiple spurious tones, the calibration power becomes prohibitive.

As shown in Fig. 3.1, a conventional PLL has an in-band closed-loop transfer function of N, significantly amplifying fractional spurs and SDM noise. Therefore, it requires complicated cancellation and calibration circuitry. The heterodyne PLL [55], a structure often found in non-monolithic high-performance instruments, is an inspiration of our novel solution that naturally achieves calibration-free fractional operation and has a unity closed-loop

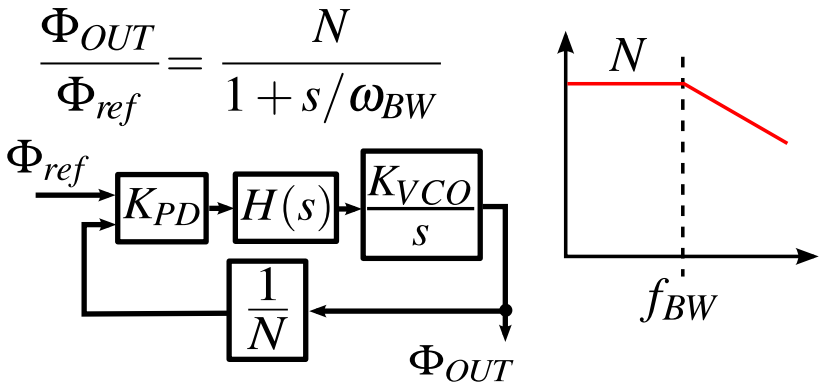


Figure 3.1: Transfer function of a conventional PLL.

transfer function. The presented prototype amplifies no fractional spur and quantization noise of SDM. Thus, it achieves state-of-the-art 131 fs RMS jitter and less than -70 dBc fractional spurs.

The remainder of this paper is organized as follows. 3.2 introduces the mechanism of heterodyne PLL and presents the primary constraint to integrate it on-chip. 3.3 introduces the driving force behind integrating the heterodyne architecture by adding harmonic mixing into the picture. 3.4 to 3.7 discuss the design of the various key building blocks. 3.8 presents the measurement results.

3.2 Heterodyne PLL

A conceptual architecture of heterodyne PLL is shown in Fig. 3.2. Two auxiliary synthesizers generate a fractional auxiliary clock (FAC) f_{FAC} for a high resolution and an integer auxiliary clock (IAC) f_{IAC} with coarse tuning steps. In the main PLL, instead of obtaining a feedback signal by dividing down the oscillator (VCO), the mixer with a cascaded low-pass filter, passing only the frequency difference, generates the desired IF signal. The PD senses the phase error of the two inputs and generates the corresponding control for the main oscillator to make f_{IF} track f_{FAC}. At steady state, the main PLL output frequency is the sum of two auxiliary frequencies.

As shown in Fig. 3.3, a mixer directly translates the main VCO spectrum to IF without

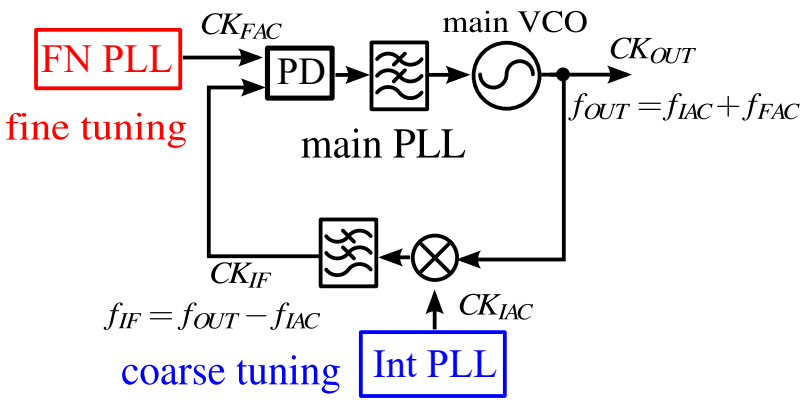


Figure 3.2: Conceptual structure of heterodyne PLL.

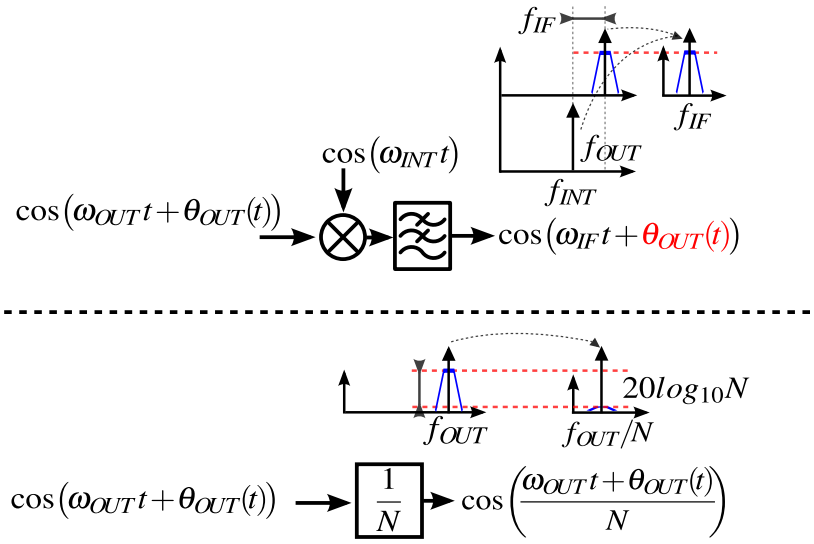


Figure 3.3: Frequency mixer vs. frequency divider.

a change in SNR. A frequency divider, which only divides phase, suppresses the input PM sideband, i.e., phase noise, and PM spurs while maintaining the carrier strength. The noise-to-carrier ratio is reduced after division. As shown in Fig. 3.4, a time domain explanation is that a $1/N$ divider triggers the edge transition every N cycles of the oscillator. It decimates the noise of the oscillator and therefore causes the phase noise reduction. As for the frequency translation in a mixer, the edge of IF is constructed by interpolating the held voltage after downconversion. Every cycle of the oscillator contributes noise to the IF. Thus, there is no noise reduction. Therefore, the feedback factor is 1 in the main PLL whereas $1/N$ in a conventional PLL. The s-domain model of the main PLL is shown in Fig. 3.5. For both

auxiliary clocks, the main PLL has a unity closed-loop transfer function, which is the property that we are searching for. This merit can also be understood from time domain. As shown in Fig. 3.6, assuming the loop gain is large enough, at steady state, the IF is locked to f_{FAC}, and the input phase error is zero. Therefore, θ_{OUT} tracks θ_{FAC}. The FAC noise is directly translated to the output without amplification.

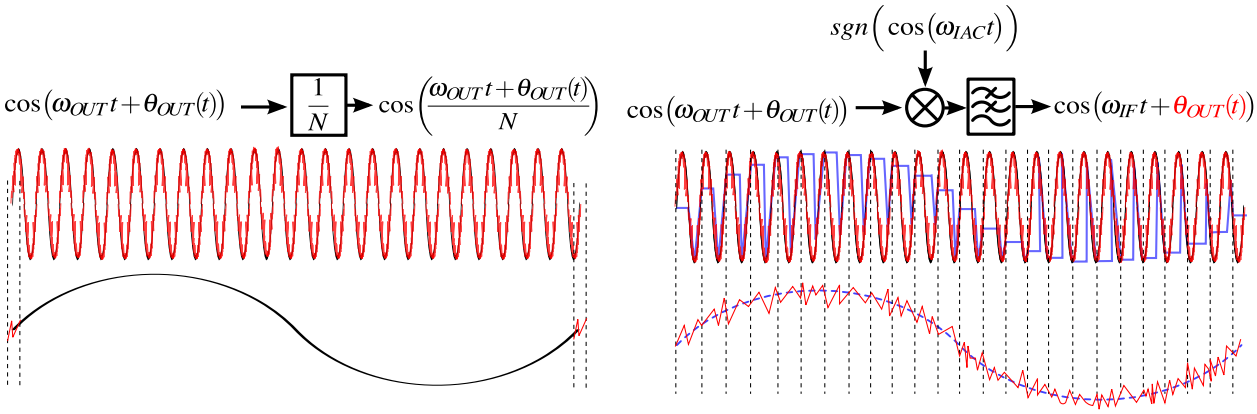


Figure 3.4: The waveform of frequency division and frequency translation.

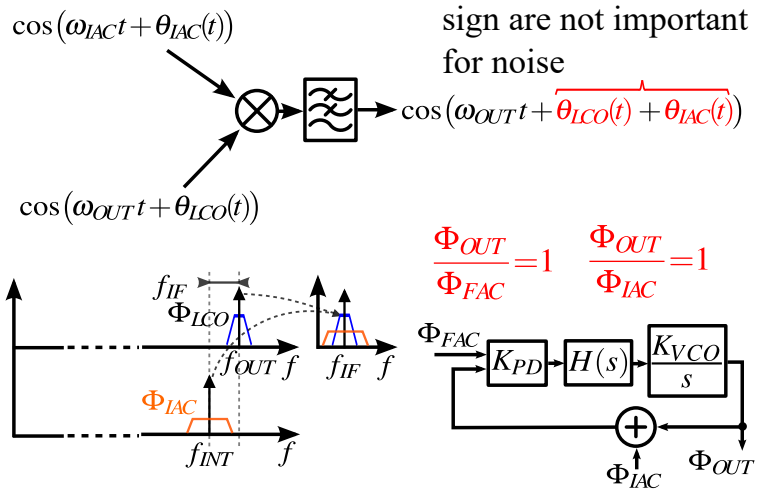


Figure 3.5: Main PLL model in the heterodyne architecture.

As shown in Fig. 3.7, the fractional PLL (FN PLL) phase noise and spurs can be reduced by adding the outside loop divider N_{FAC}. The divider attenuates the phase noise and spurs of the fractional PLL by $20log_{10}N_{FAC}$. The attenuation remains unaltered at the main PLL output due to the unity closed-loop transfer function. Therefore, when choosing a large

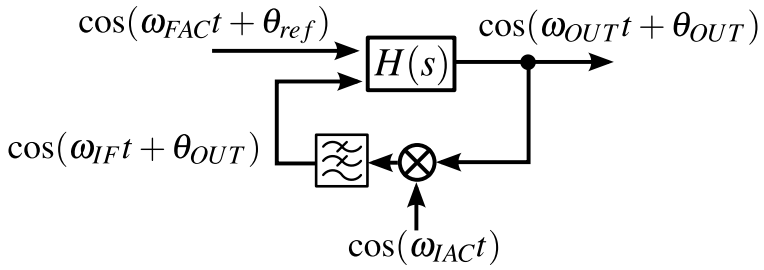


Figure 3.6: Main PLL time-domain operation.

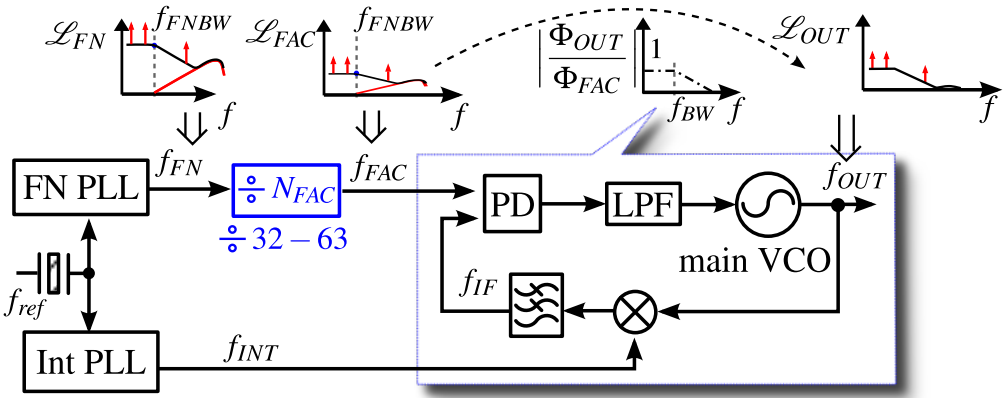


Figure 3.7: Applying heterodyne PLL for our implementation. The N_{FAC} reduces the FN PLL noise. The main loop has no noise amplification.

N_{FAC}, this modified heterodyne PLL achieves substantial fractional PLL noise and spur reduction without any calibration.

Besides, main loop further attenuates the high frequency noise and spurs of the divided clock. This cascade higher order filtering is another advantage of our multi-loop structure. As shown in Fig. 3.9, while each reference loop has their own filtering, the cascaded main loop provides extra attenuation. In total, they exhibit a higher-order filtering, which cannot be achieved by a single loop.

The crystal and the loop oscillator are the primary noise sources of the integer PLL (Int PLL) [56, 41]. By increasing the loop bandwidth, this PLL can suppress the oscillator noise to be much lower than the scaled-up crystal. Then this integer loop works like an ideal frequency multiplier which increases the frequency and noise of the crystal by the multiplication factor. In the main loop, since there is no noise amplification, the primary

noise source is the main VCO which is high-pass filtered there. As shown in Fig. 3.8, since the FAC noise is negligible, the dominant noise contributors of the whole heterodyne synthesizer are the main loop VCO, and the Int PLL, whose noise, by design, is mainly determined by the scaled-up crystal. The output jitter σ^2_{OUT} is a convex function of the main loop bandwidth f_{BW}, which again has a well-defined minimum like a simple integer-N PLL.

$$\sigma^2_{OUT}(f_{BW}) = \int_0^\infty 2\mathcal{L}_{OUT}(f)df = \frac{4K_W}{2Q^2 f_{BW}} + 4PN_{in} \cdot f_{BW} \tag{3.1}$$

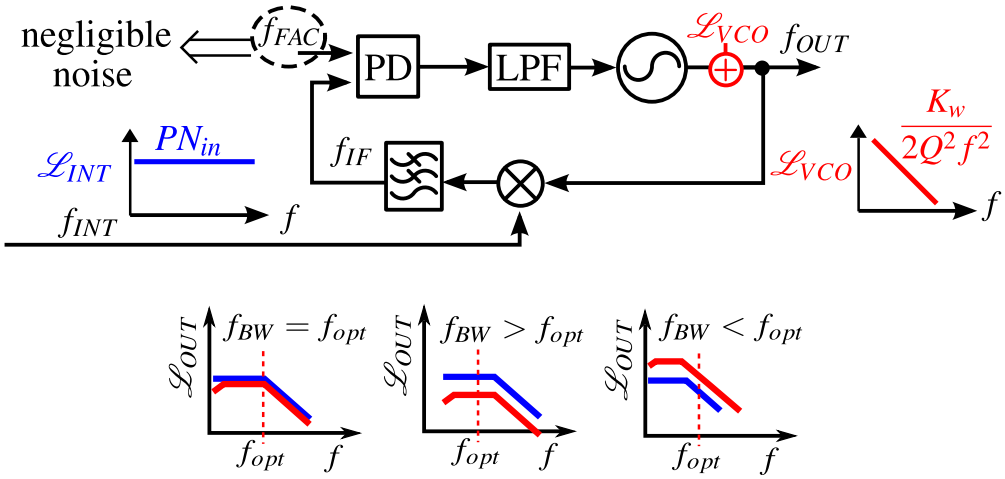


Figure 3.8: The noise of heterodyne PLL.

In this design, as presented in Fig. 3.10, the output frequency is 7 to 9 GHz. Because of the large ratio of N_{FAC}, f_{FN} is designed to around 3 GHz. Then N_{FAC} outputs a clock with a tunable frequency around 50-87 MHz which allows the main PLL to have a wide loop bandwidth without stability problem. Because of the small f_{FAC}, f_{INT} has an operating frequency similar to the output. This triple-loop PLL used in instruments [4] has been historically realized by shielded discrete modules, enjoying small signal crosstalk. When implemented on the same chip, since f_{INT} is close to f_{OUT}, the VCO of the integer PLL and the main VCO can suffer from strong mutual pulling [57, 58], devastating the function of the entire architecture.

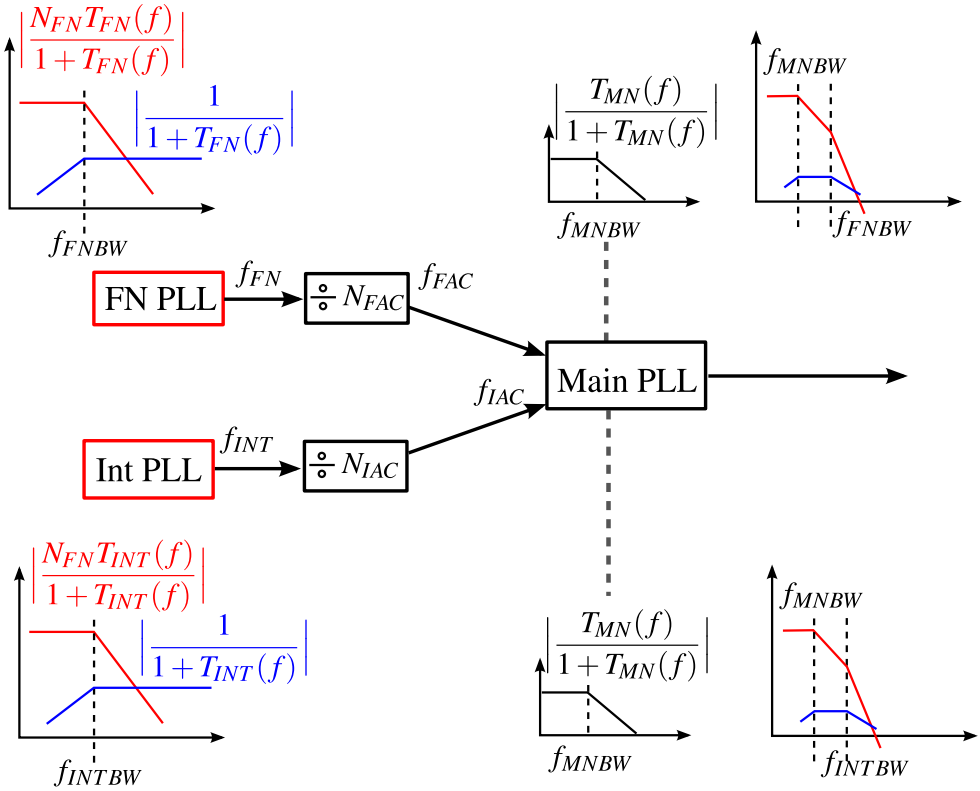


Figure 3.9: Multi-loop architecture provides cascade filtering strength.

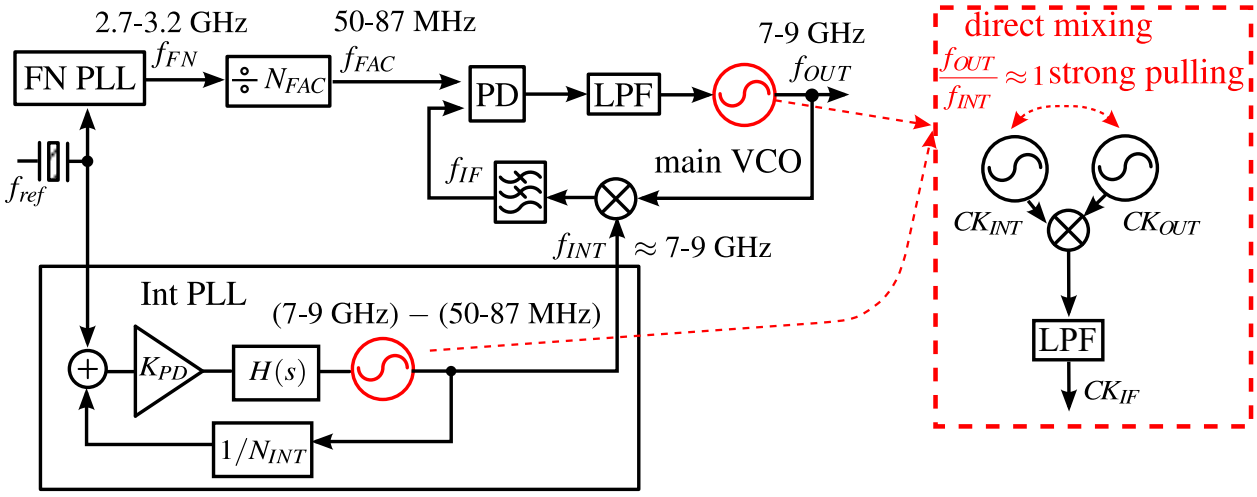


Figure 3.10: Pulling in heterodyne PLL.

3.3 Harmonic mixing to avoid pulling

In 1964, a harmonic mixing PLL, as shown in Fig. 3.11, is invented for a sampling oscilloscope [59]. The VCO oscillates at a moderate frequency f_{VCO} and samples the high-frequency

input. The K^{th} harmonic of f_{VCO} is close to the input, which translates the input to a low-frequency output, which is known as an aliasing signal. The loop locks the carrier of the output to f_{ref}. The essential inspiration here is that we can use the harmonic of the integer PLL to downconvert the main VCO. In this way, the frequencies of the two oscillators in the heterodyne PLL are widely separated, which substantially reduces the mutual pulling risk.

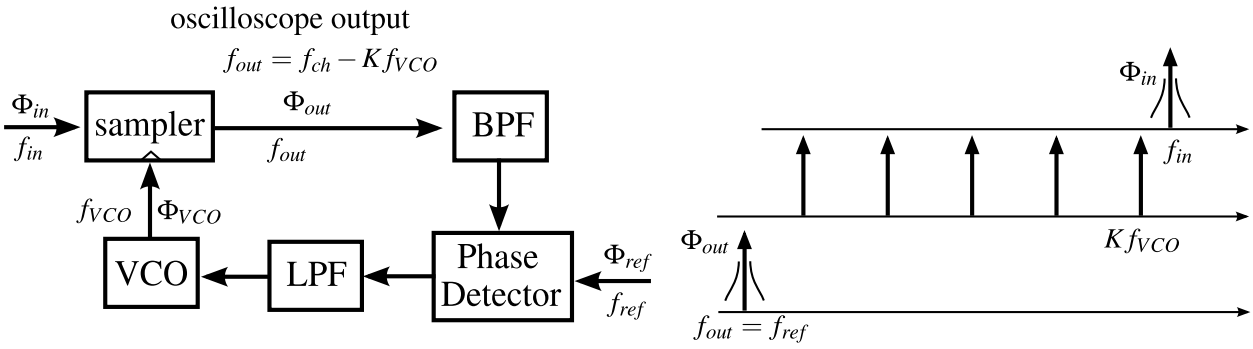


Figure 3.11: Harmonic mixing PLL for sampling oscilloscope.

If a square-wave clock drives the mixer, it only provides odd harmonics, and the mixing gain of the harmonics is much lower than the one of the fundamental tone. When applying it for harmonic mixing, the small amplitude IF will be susceptible to the noise of the following blocks. As shown in Fig. 3.12, a sample-and-hold (SH) replaces the mixer in the main loop. The edge of the clock samples the input. It has an aperture time with an order of magnitude of 10 ps and can be regarded as an ideal sampling impulse. Therefore, this edge sampling provides harmonics with equal strength as the fundamental, and thus equal mixing gain. We choose the 5^{th} harmonic of the sampling clock f_{IAC} to downconvert the main VCO to IF. Instead of directly generating f_{IAC} through the Int PLL, f_{INT} is divided down by 8 or 9 to form f_{IAC}. In this way, the frequency ratio of the two VCOs is a rational fraction which is either 5/8 or 5/9. The potential pulling risks only happens between the high-frequency harmonics of the two oscillators. Since the output of an LC oscillator is close to a sinewave, the energy of the high-frequency harmonics is small. The harmonic pulling risks are negligible. The final PLL architecture, as shown in Fig. 3.13, can now be integrated. Because of our frequency plan, the Int PLL works around 12-14 GHz and thus requires an LC oscillator. Running at this high-frequency, however, stresses the design of the two dividers

on the IAC path. Later, the measurement will show that it causes extra power consumption.

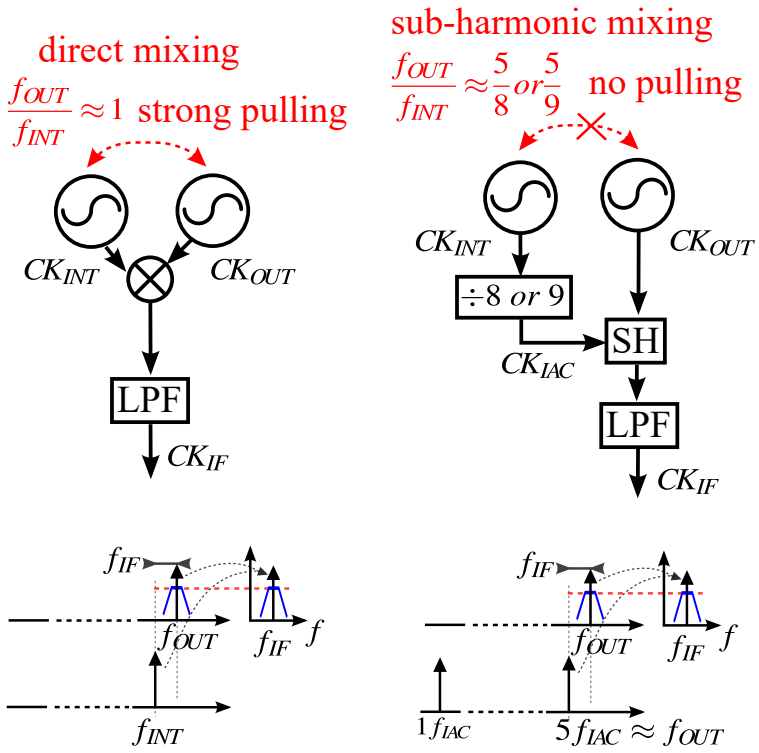


Figure 3.12: Sample-and-hold, which provides strong harmonics for mixing, avoids pulling of two VCOs in the mixer-based architecture.

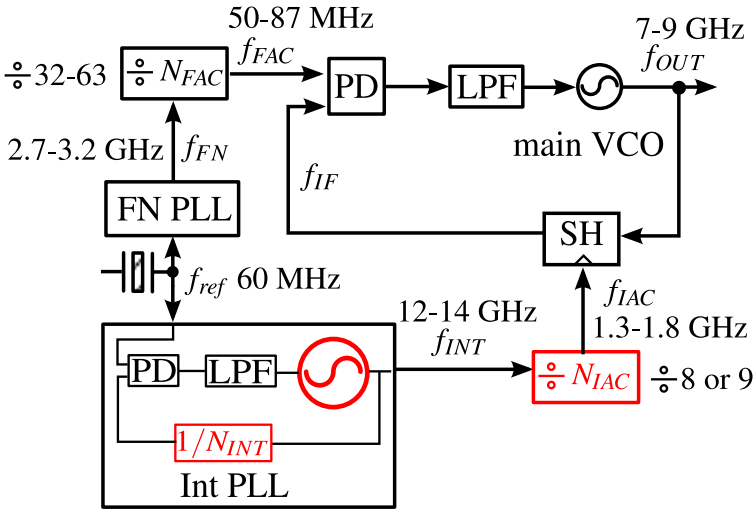


Figure 3.13: Final heterodyne PLL with harmonic mixing, which can be integrated on a single chip.

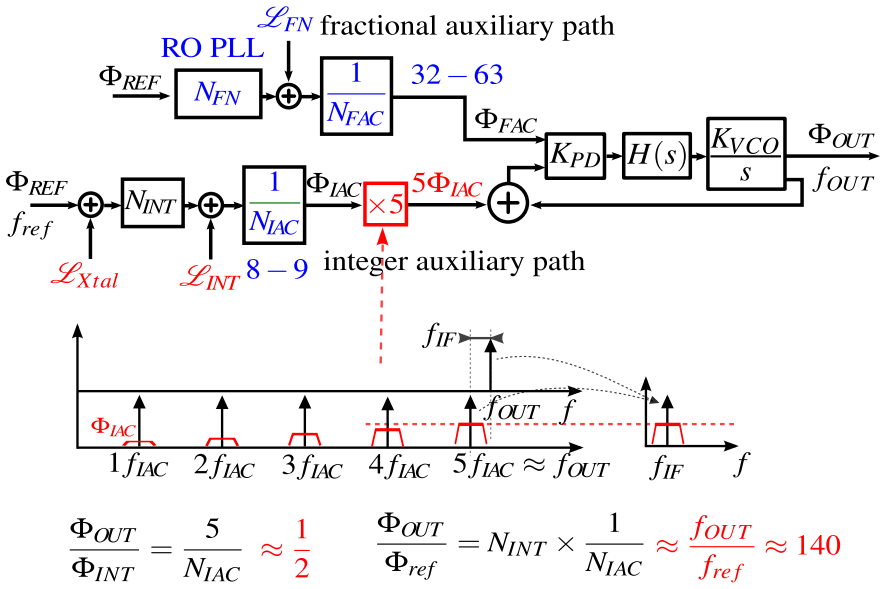


Figure 3.14: Simplified triple-loop PLL model.

The simplified triple-loop model is shown in Fig. 3.14. On the FAC path, the large dividing ratio substantially suppresses the FN PLL phase noise, relaxing the noise requirement for the FN PLL. A compact ring oscillator is chosen for its simplicity. Moreover, the ring oscillator has negligible magnetic couplings with the other two oscillators. On the IAC path, the divided integer PLL phase noise is amplified by 5^2 because we utilize the 5^{th} harmonic of f_{IAC} to downconvert the main VCO, and the phase noise of a harmonic is equal to the phase noise of the fundamental multiplied by the square of the harmonic index. The gain from the Int PLL output to the final output is around f_{OUT}/f_{INT}, which normalizes the integer PLL phase noise to the output frequency. There is no substantial noise reduction. Therefore, this integer loop must be of low noise. The total gain on IAC path is around f_{OUT}/f_{ref} which is also the gain of a conventional PLL, hence this triple-loop structure still amplifies the reference noise in the same way like a conventional PLL.

3.4 FN PLL design

Since the noise requirement of the FN PLL is relaxed, the main consideration is power and simplicity. Although a calibration-free bang-bang PLL is simple, the sigma-delta modulation in the divider significantly reduces the bang-bang phase detector gain and considerably

increases its noise level [30, 51]. To reduce the PD noise, this loop must implement calibration which contradicts with our goal. A simple TDC is usually an inverter chain. Assuming a moderate TDC resolution τ_{res} of 20 ps [60], the reference frequency $f_{ref} = 1/T_{ref}$ is 60 MHz, and the loop multiplication ratio N is 50, based on (3.2) from [49], the TDC quantization noise level at the output is -96.9 dBc/Hz. Since it will be reduced by the cascade divider, it is acceptable for our design.

$$\mathcal{L}_{TDC} = 10log_{10}\Big(\frac{(2\pi N)^2}{T_{ref}}\frac{\tau_{res}^2}{12}\Big) \tag{3.2}$$

However, to cover the input variation range, the TDC requires a long inverter chain which is power intensive. The mismatch of different inverters also results in unwanted nonlinearity, which is difficult to resolve. Moreover, a high-resolution linear TDC is complicated, which contradicts to our aim of simplicity.

An analog charge pump (CP) consists of a PMOS current source and an NMOS current source. Based on [61], the PLL output noise due to CP is:

$$\mathcal{L}_{CP} = 10log_{10}\Big(\frac{8KT}{I_{cp}V_{eff}}\frac{\tau_{reset}}{T_{ref}}(2\pi N)^2\Big) \tag{3.3}$$

When the overdrive voltage of the current source V_{eff} is around $0.1V$, under the same dividing ratio and reference frequency, and assuming the resetting time of the PFD τ_{reset} is 1/5 of the reference, we can obtain -97.8 dBc/Hz in-band noise with 100 uA CP on-current I_{CP}. Therefore, the CP is very power-efficient. Consequently, We choose a CP PLL for this fractional loop.

As shown in Fig. 3.15, the mismatch of the up and dn current source causes an inconsistent slope for the transfer function of CP [44, 7]. When the phase error toggles between positive and negative, nonlinearity arises.

Injecting a static phase error into the PLL can bias the CP to its linear operation region. One method to generate the static phase is by applying the offset tri-state PFD [62]. As shown in Fig. 3.16, the offset PFD separates the operation of the up and dn CP source. A constant window determined by the delay t_{del} controls the up current which bias the loop to its linear region. The phase error sent from the divider modulates the pulse width of

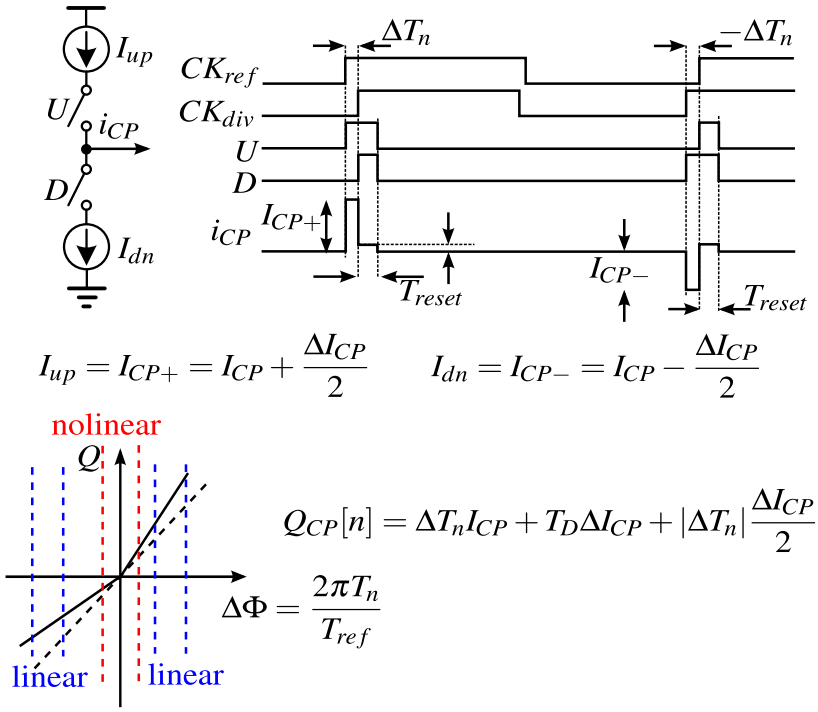


Figure 3.15: CP nonlinearity.

the dn current to tune the loop control. Although this PFD generates the static phase, the separate operation of the up and dn current introduces large ripples on the LPF and might push the current source of CP into the triode region, which causes nonlinearity, as well. As shown in Fig. 3.17 (a), [7, 45] proposed a periodic injection of an offset pulse into the LPF to force the static phase error. The injection pulse starts from the rising edge of the divider output and extends for an integer number of VCO periods to ensure synchronicity with the SDM operation. We find that with the injected pulse current, it is unnecessary to keep the up current of the original CP. The PFD-CP can be simplified, as shown in Fig. 3.17 (b). When PLL is locked, the offset pulse and the dn pulse turn on simultaneously. Compared with offset PFD, less net current flows out of the CP. The ripple at the CP output is much smaller. The current sources of CP have less risk to enter the triode region.

The LPF is modified in the same way as [7]. The capacitor, which is directly connected to CP, is split into two parallel half-sized capacitors separated by a CMOS transmission gate switch. The switch turns on when the CP and pulse current finish the charge injection. It avoids the LPF sensing the ripple caused during the CP tracking phase. After approximately

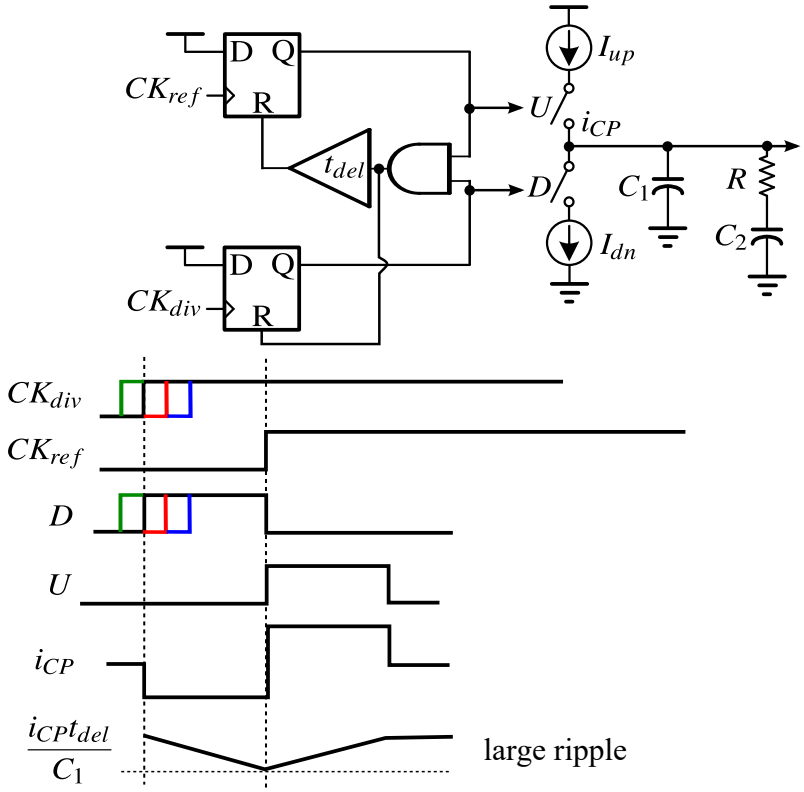


Figure 3.16: Operation of offset tri-state PFD, obtaining static phase error, but with large ripple.

1ns, the on switch turns off and is ready for the next cycle operation.

$$\mathcal{L}_{sdm} = \frac{1}{12 f_{SDM}} (2\pi)^2 \Big(2\sin(\pi f / f_{SDM})\Big)^{2(m-1)} \tag{3.4}$$

Based on (3.4) from [63], a higher operating frequency f_{SDM} can lower the SDM in-band noise by $(1/f_{SDM})^{2m-3}$ where m is the order of the SDM. As shown in Fig. 3.18, for a divider with 2nd order SDM, when f_{SDM} is increased from 60 MHz to 120 MHz, the in-band noise is lowered by 9dB.

Fig. 3.19 is the complete FN PLL. The frequency doubler, similar to [48], doubles f_{SDM} and thus lowers the SDM in-band phase noise. Therefore, we can widen the loop bandwidth to further attenuate the ring oscillator phase noise, while maintaining the low SDM noise. The cascaded divider N_{FAC} is tunable from 32 to 63, which provides at least 30 dB noise and spur reduction. After the frequency division, the FN loop noise is lowered to around

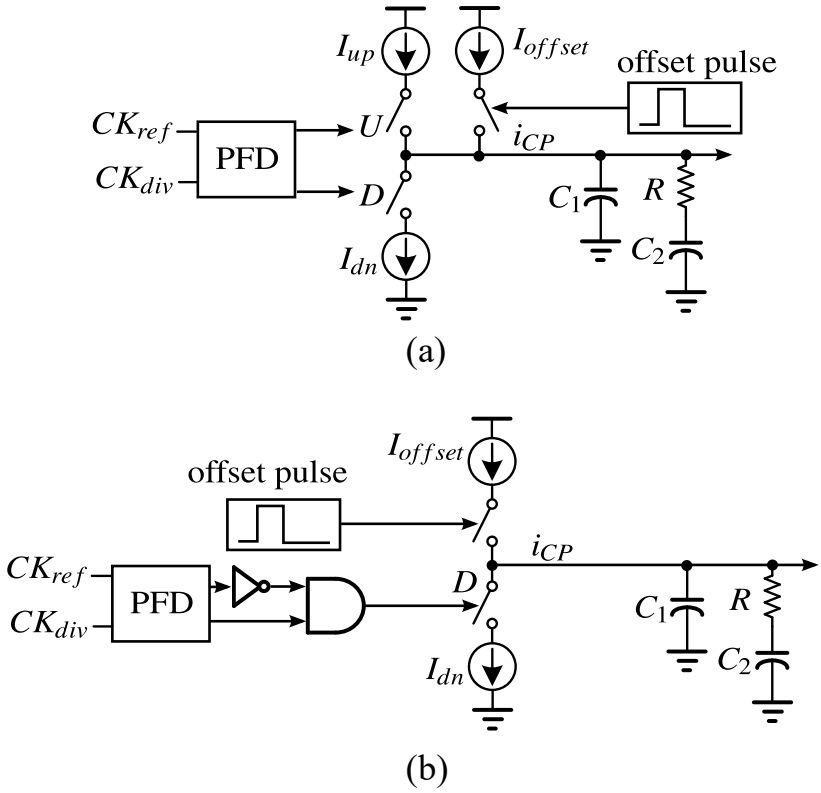


Figure 3.17: (a) Original offset pulse current injection; and (b) proposed offset pulse current injection.

-130 dBc/Hz. Moreover, the main PLL filters the remaining noise further. At the main PLL output, the jitter contribution from this fractional PLL becomes negligible.

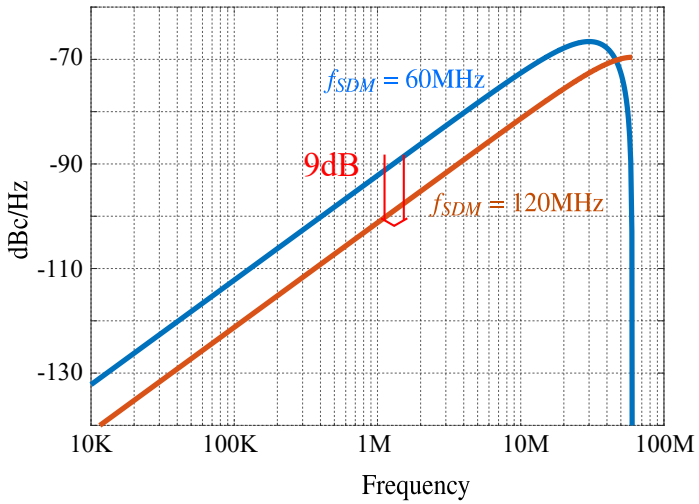


Figure 3.18: SDM noise at different operating frequencies.

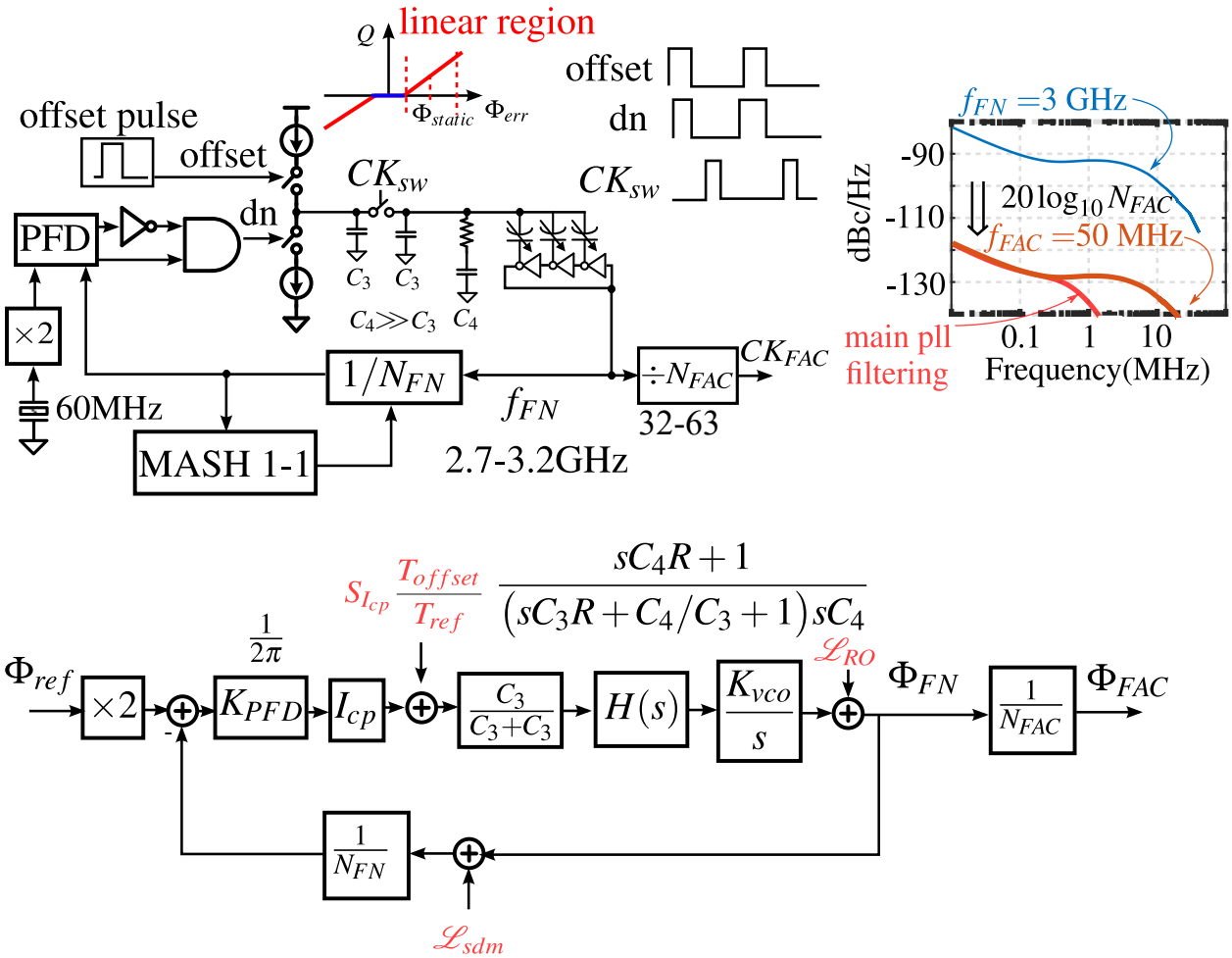


Figure 3.19: Fractional auxiliary clock circuit, model, and phase noise prediction.

3.5 Int PLL design

For the integer loop, we prefer a simple and low noise PLL. A subsampling PLL [53], having no amplification to the phase detector (PD) and CP noise by eliminating the loop divider, is an architecture suitable for our low noise target. However, the complexity of the PD and CP design, and the requirement for an extra frequency acquisition loop increase the design effort. A type-I sampling integer-N PLL is reported in [41], which can have a wide loop bandwidth, low phase noise, and low reference spurs. Unlike a conventional type-I PLL, where the phase error modulates the XOR output pulse width, and the LPF extracts the average of the pulse as the VCO control voltage, the reported PLL uses a switch-cap filter to extract the phase information by sampling the finite falling edge of the XOR. As shown in Fig. 3.20, a replacement of the XOR with an inverter causes no functional difference. As shown in Fig. 3.21, without phase error, the switch-cap samples a constant DC voltage V_{DC}. When a phase error occurs, the falling edge either leads or lags the sampling instance. The sampled voltage varies accordingly which indicates the phase error. The PD gain therefore is:

$$K_{PD} = -\frac{V_{swing} \cdot SR \cdot T_{ref}}{2\pi} \tag{3.5}$$

The phase detector circuit is shown in Fig. 3.22. C_2 holds the charge until the next sampling instance, forming a sinc() frequency response. The reference frequency and multiples lie at the nulls of the sinc(). Therefore, this PLL has inherited low reference spurs which allows the loop bandwidth to be broadened to suppress VCO noise.

Due to the high gain of PD, the input referred noise of the switch-cap is negligible. The PD noise is contributed by an inverter, which is also very small. By retiming the divider output, the divider phase noise can be controlled much lower than the crystal reference. The reference crystal and the LC oscillator are the primary noise contributors in this loop.

We design a class-A VCO as shown in Fig. 3.23. Tail resistors, which have no flicker noise, control the power of the VCO. By designing the single-side swing to 150 mV, the

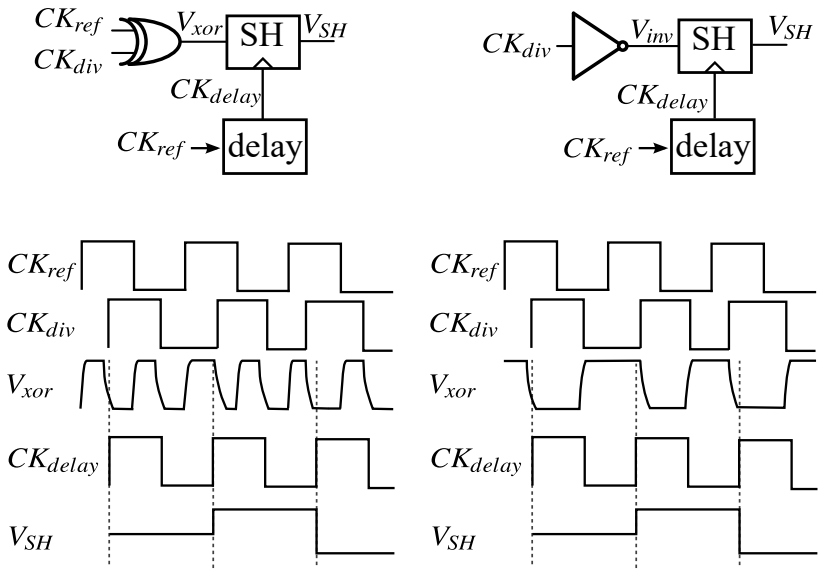


Figure 3.20: Phase detector operation: The left circuit is the original structure, while the right one is the proposed structure.

four transistors of the VCO are kept in the saturation region to maintain a high loaded Q. The VCO oscillates at 12-14 GHz where the inductor occupies a small area. Although, as resonant frequency increases, both the proximity effect and the skin effect increase the ac resistance [64, 65], the inductor still obtains a Q of 20 at the planned frequency. Therefore, the design of the inductor is relaxed.

When designing a VCO, the main concerns are the gm cell and the LC tank parallel resistance. The kickback noise from the VCO buffer is often neglected. In a type-I PLL which only provides first order filtering, the loop cannot eliminate flicker noise. Any effect that worsens the flicker noise should be taken into account. As shown in Fig. 3.24, the switch operation of the buffer upconverts the flicker noise to the carrier frequency [1]. Through the capacitive path, this noise kicks back into the VCO. Around the carrier frequency, the LC tank is almost an open circuit. The kickback noise concentrates on one side of the VCO, modulating the current from the other side of the gm cell. The modulated PMOS and NMOS generate a differential noise current, which flows into the LC tank in the same way as the gm cell noise, forming $1/f^3$ noise at the VCO output. In our simulation, without the noise buffer, the VCO noise at 100 kHz offset is -81 dBc/Hz. When connected with a VCO

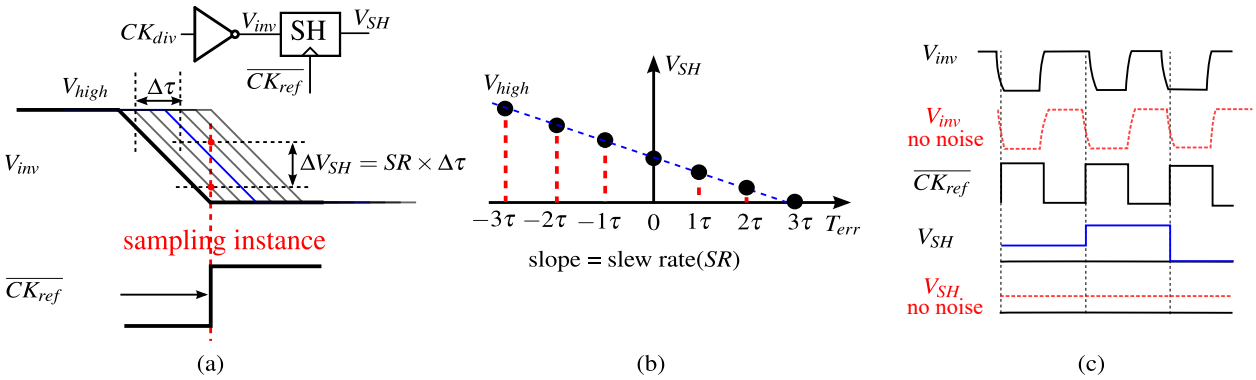


Figure 3.21: (a) Switch-cap samples different voltage when falling edge varies; (b) transfer function between phase error and the sampled voltage; and (c) time domain waveform of sampling operation.

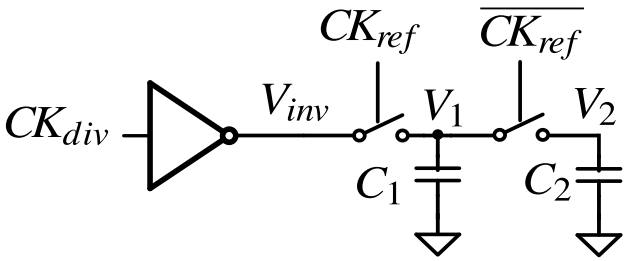


Figure 3.22: The circuit of the switch-cap filter.

buffer, the VCO noise at the same offset increases to -76 dBc/Hz. As the offset frequency increases, the LC tank impedance reduces. The kickback noise is more uniformly distributed to both sides of the VCO as a common mode signal, which contributes less differential noise current and therefore less phase noise. Making the kickback noise a common mode signal is the key to reduce the kickback effect. By duplicating the coupling path and connecting the kickback noise source to both sides of the VCO, we convert the differential injecting noise to a common mode one, as shown in Fig. 3.25 (b). Practically, buffer noise kicks back to the VCO through the parasitics of the buffer. As shown in Fig. 3.25 (c), a duplicated buffer connects the other side of the VCO with the node Out_{buffer}. It provides a desired parasitic path for the kickback noise to flow. The large resistors on the supply and ground of the duplicated buffer choke the current and make the buffer an open circuit with no noise contribution and power consumption. Simulation indicates that with this duplicated path,

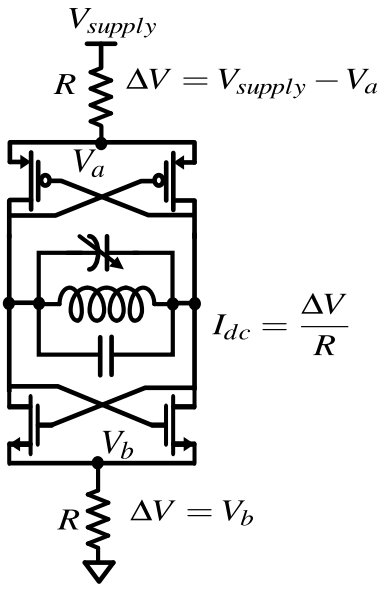


Figure 3.23: Integer PLL VCO structure.

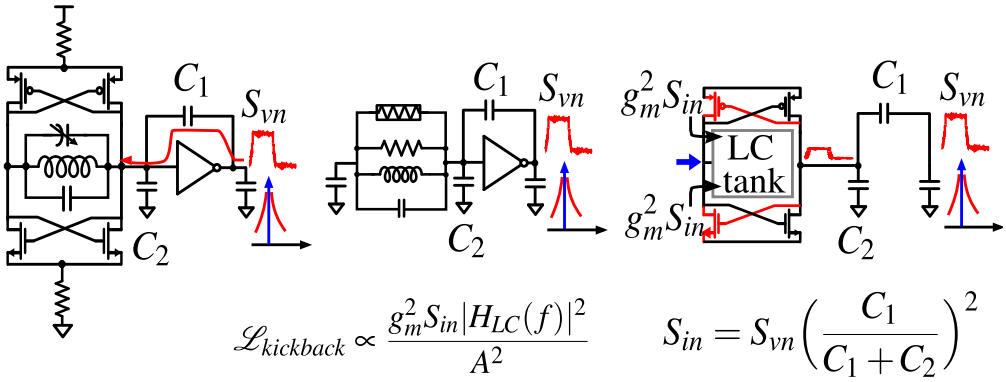


Figure 3.24: Buffer noise kicks back into the VCO.

the VCO noise at 100kHz is -79 dBc/Hz, exhibiting a 3 dB improvement.

The schematic of the Int PLL is shown in Fig. 3.26. When the reference clock is low, C_1 shares the sampled charge with C_2.

$$\Big(C_1 + C_2\Big)V_b[n] = C_2 V_b[n-1] + C_1 V_a[n] \tag{3.6}$$

When the reference is high, C_1 resets the remaining charge and tracks the inverter output. The transfer function of the switch cap is:

$$\frac{V_b}{V_a}(z) = \frac{C_1}{C_1 + C_2 - C_2 z^{-1}} \tag{3.7}$$

When approximating in continuous-time frequency-domain, the transfer function is:

$$\frac{V_b}{V_a}(f) = \frac{C_1/C_2}{\frac{C_1 f_{ref}}{C_2} + j\omega} \tag{3.8}$$

This coinsides with the transfer function in [41]. The impulse response of the hold operation in C_2 is rectangular window with a width of T_{ref}, which introduces a sinc() as mentioned previously. It also results in a half period delay. In addition to the switch-cap pole and the delay which erode the loop phase margin, the tracking phase of the switch-cap filter also introduces a high frequency pole which potentially degrades the loop stability. At the tracking phase, the first switch turns on and forms a RC filter with C_1 [66]. The pole of this filter is determined by C_1 and the switch on-resistance. It normally is located at frequency higher than 1 GHz which introduces negligible effect for a conventional loop with less than 10 MHz bandwidth. Therefore, here we ignore its effect, but in the next chapter where we will design a PLL with a loop bandwidth larger than 100 MHz, this pole effect emerges.

The model of the PLL is presented in Fig. 3.27 (a). The loop bandwidth is $f_{ref}/10.5$, which can effectively suppress the oscillator noise without eroding the phase margin. As shown in Fig. 3.27 (b), after loop filtering, the oscillator noise contribution is much lower than the crystal. The crystal determines the noise floor. This loop indeed works like an ideal frequency multiplier. Moreover, the main loop attenuates the output noise further, which is another advantage of our multi-loop structure.

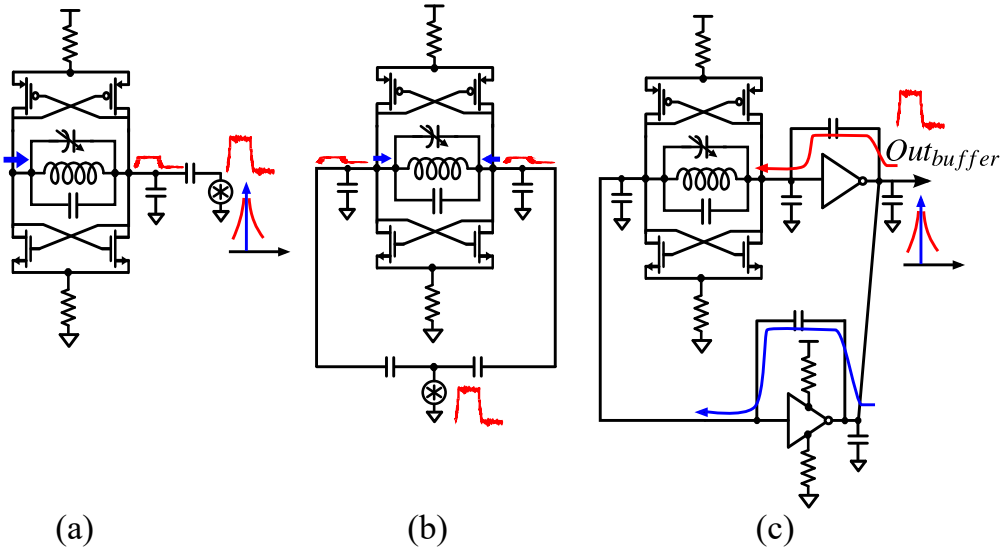


Figure 3.25: (a) Noise kicks back through capacitor path; (b) Noise kicks back as a common mode signal through the capacitor path and its duplication; and (c) practical generation of the duplicated path.

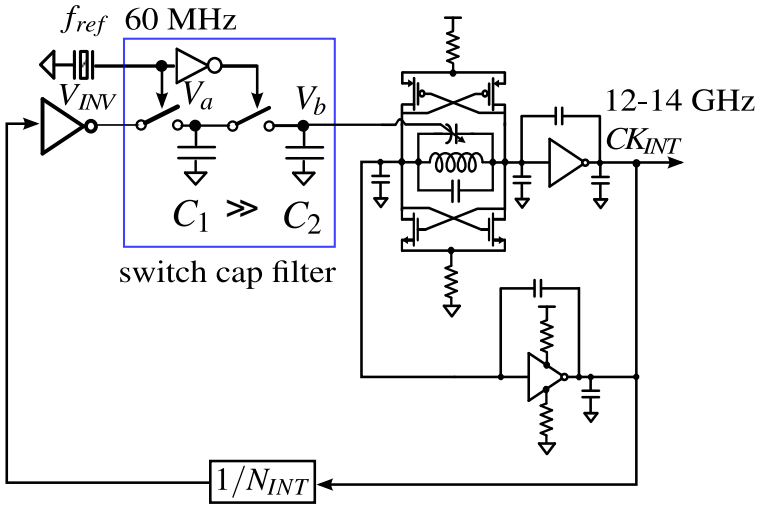


Figure 3.26: The schematic of the Int PLL.

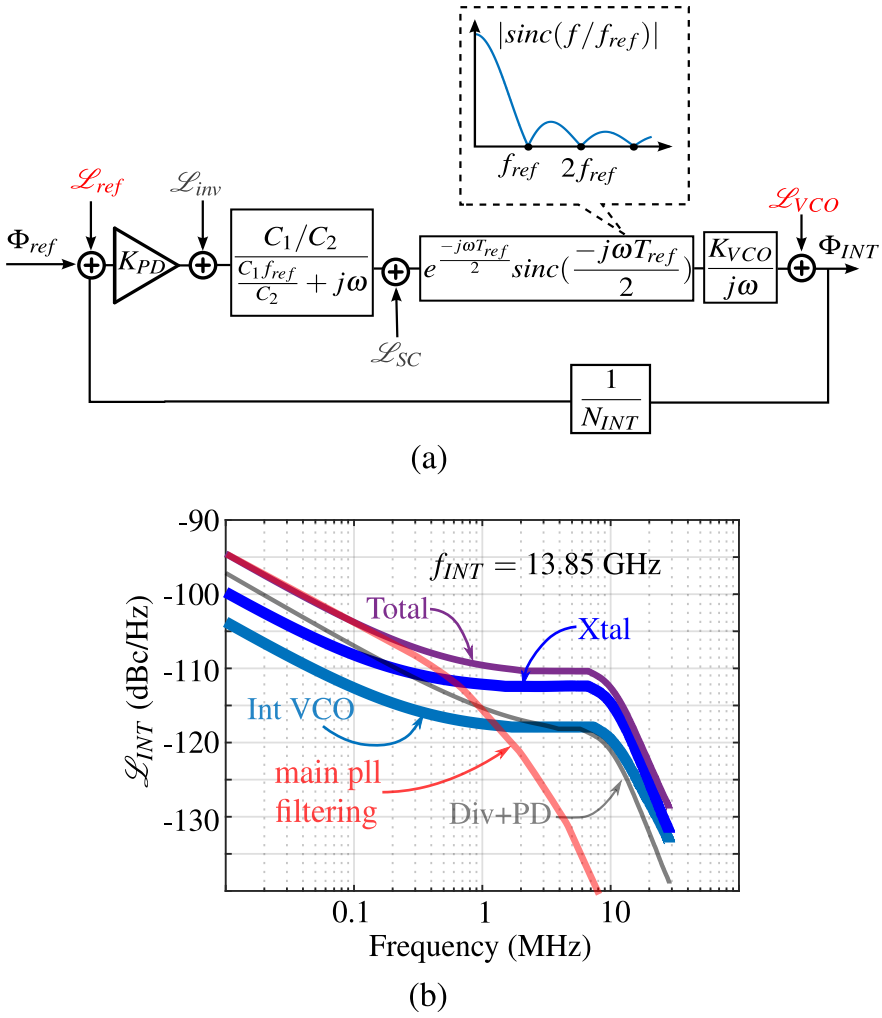


Figure 3.27: (a) Int PLL model; (b) phase noise.

3.6 Main PLL design

The design concerns for the main loop are bang-bang PD design, SH circuit design, and loop locking speed.

3.6.1 Bang-bang PD

Because of its simplicity, bang-bang PD, as shown in Fig. 3.28, is implemented in our main loop. [30] demonstrated that the PD gain, which is determined by the input jitter σ_{Φ_e}, is:

$$K_{BPD} = \sqrt{\frac{2}{\pi}} \frac{1}{\sigma_{\Phi_e}} \tag{3.9}$$

.

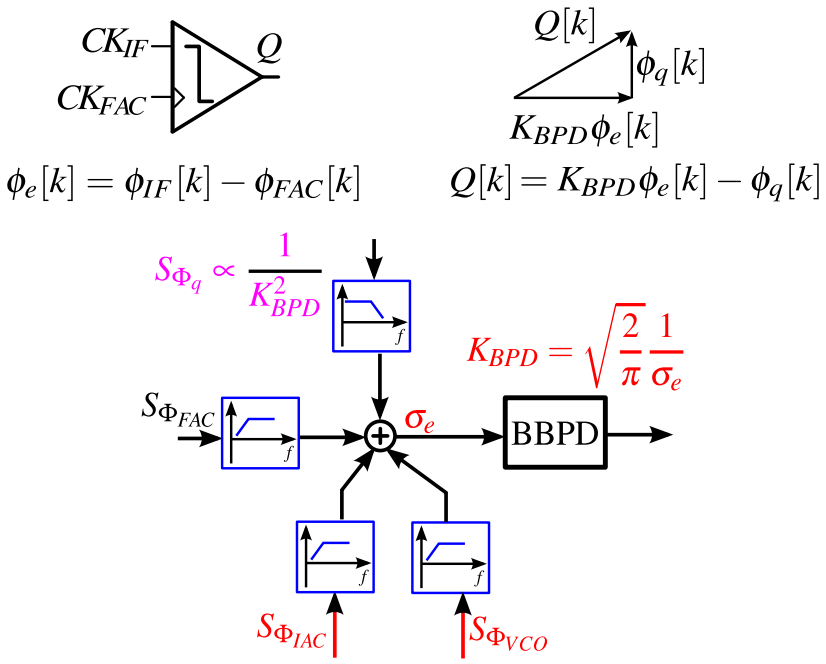


Figure 3.28: Bang-bang PD model. The gain is correlated to the input jitter.

The PD input referred quantization noise is:

$$\mathcal{L}_{BPD}(f) \approx \left(1 - \frac{2}{\pi}\right) \frac{1}{f_{ref}} / K_{BPD}^2 \tag{3.10}$$

.

In our PLL, the noise of the IAC path $S_{\Phi_{IAC}}$ and the main VCO $S_{\Phi_{VCO}}$ mainly determine the input jitter. They can be designed low to maintain the PD gain K_{BPD}. Previous studies [30, 50, 51] illustrated that, despite a proper regulation of PD input jitter, PD quantization noise remains the main noise contributor at the output because of the amplification in the conventional high gain loop. In our design, the unity closed-loop transfer function has no amplification to the quantization noise. Therefore at the main PLL output, the in-band noise contributed from PD is still determined by (3.10). With a large K_{BPD}, the PD contributes only a small portion of phase noise.

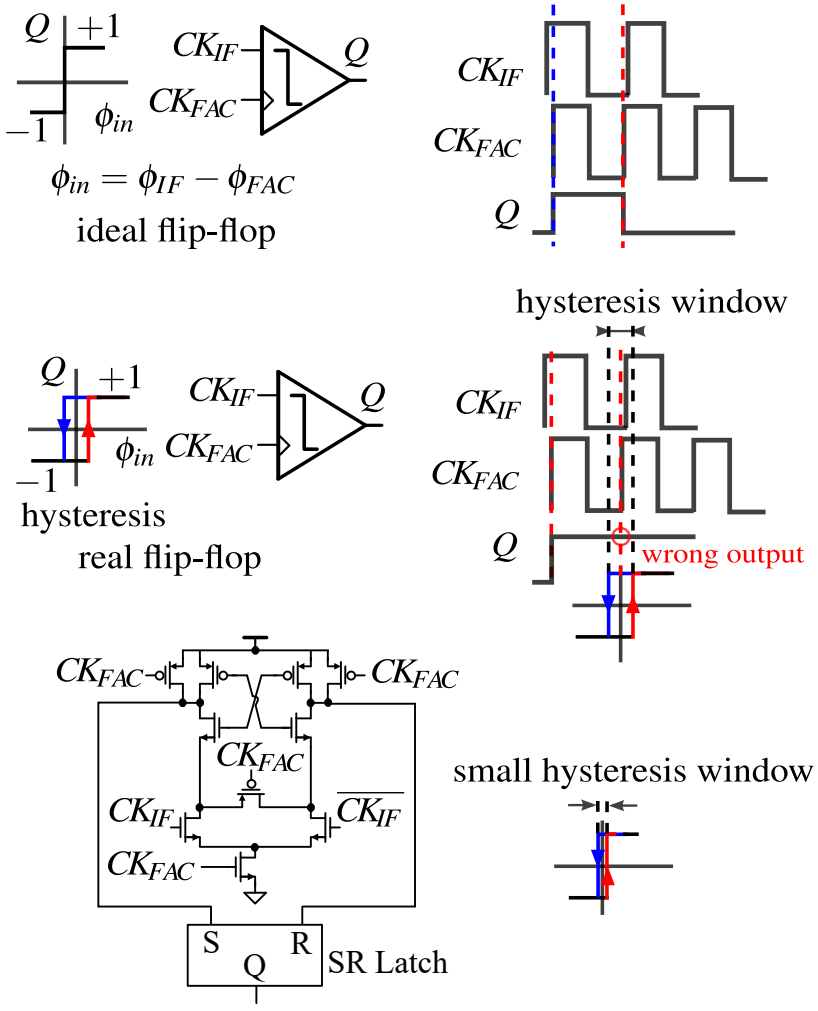


Figure 3.29: Flip-flop hysteresis causes nonlinearity.

As shown in Fig. 3.29, the bang-bang PD, a flip-flop, suffers from hysteresis, which is caused by the previous memory. For example, after the bang-bang PD senses a low-to-high

transition, at the next cycle, a high-to-low transition occurs. If the new transition is within the hysteresis window, the flip-flop output remains high instead of falling low, failing to respond to the input. This nonlinearity results in unwanted noise-folding and spurs. The strongarm latch [67] resets itself every cycle. It cleans up the previous memory, and thus has the minimum hysteresis window. Consequently, it is suitable as the bang-bang PD.

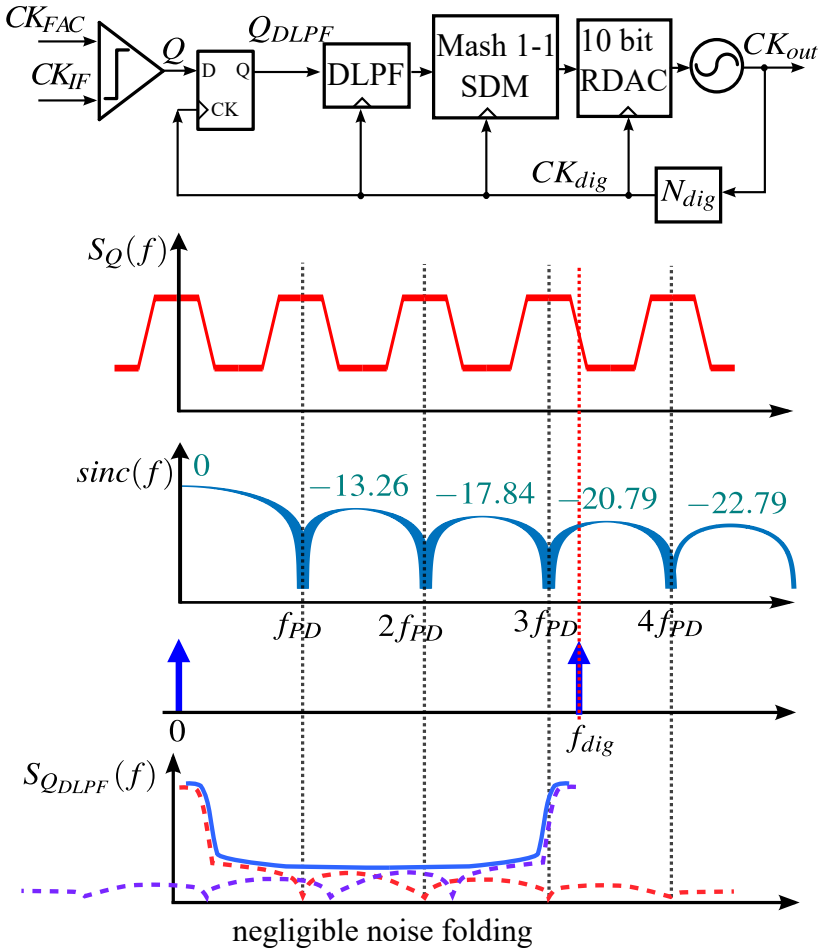


Figure 3.30: Oversample induces negligible noise-folding.

As shown in Fig, 3.30, the clock of the digital circuit CK_{dig} is divided from the main VCO. This arrangement prohibits the harmonic of CK_{dig} from pulling the main VCO [68]. The asynchronous operation between digital LPF (DLPF) and PD causes unwanted noise-folding, potentially degrading the PLL performance. The bang-bang PD output has an intrinsic sinc() [30], which suppresses the high-frequency noise. Thus, oversampling the bang-bang PD only folds the suppressed image noise, which reduces the folding effect. In

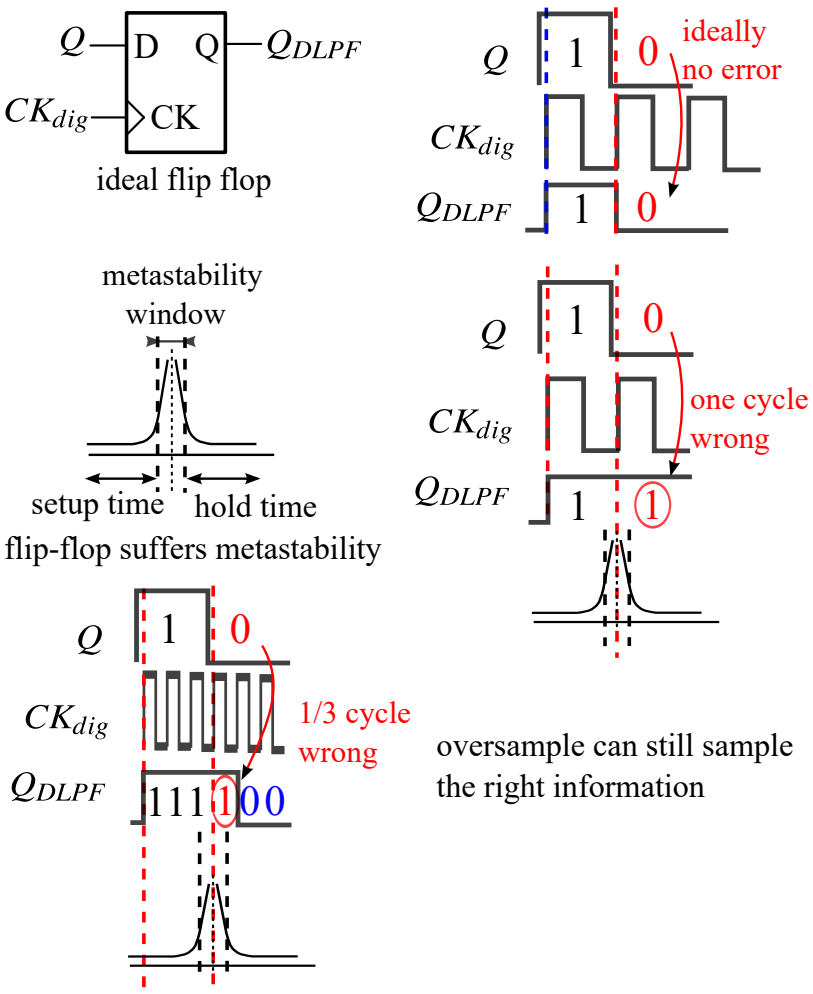


Figure 3.31: Oversample mitigates the metastability issue.

our design, CK_{dig} is 300 MHz while f_{FAC} is less than 90 MHz. The sinc() provides 20dB reduction to the images.

As shown in Fig. 3.31, the flip-flops of the DLPF might encounter metastability when capturing the bang-bang PD data due to the asynchrony. Despite the occurrence of metastability, oversampling ensures that the average sampled value per reference cycle is close to the bang-bang PD output, preventing serious nonlinearity.

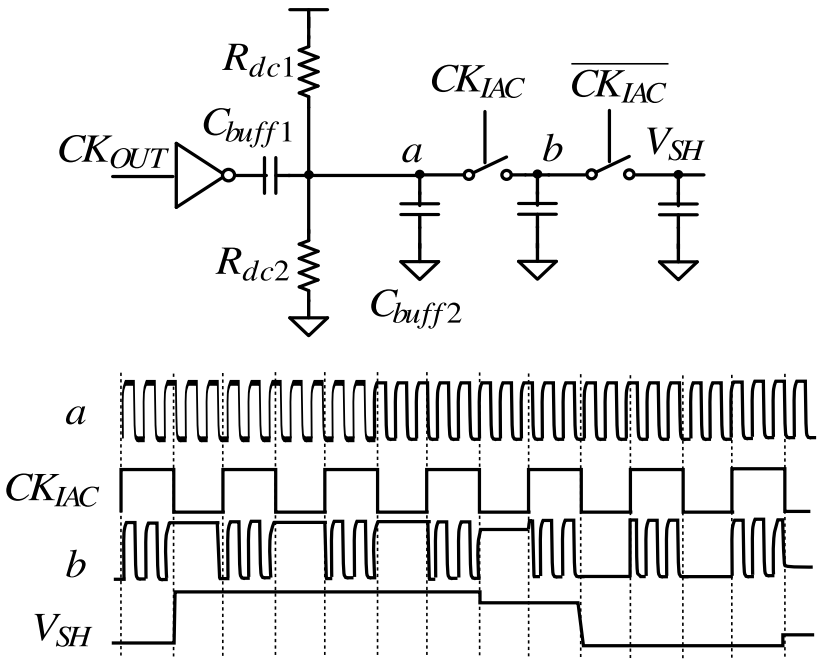


Figure 3.32: Sample-and-hold circuit and its operation waveform.

3.6.2 Sample-and-hold design

As shown in Fig. 3.32, the sample-and-hold (SH) is similar to the switch-cap filter in our sampling PLL. The SH input buffer is a simple inverter. This nonlinear buffer generates harmonics of the main VCO, which also mix with the sampling impulses. As shown in Fig. 3.33, through the SH, the fundamental tone of the input is translated to IF by $5f_{IAC}$. The harmonics of the input are also translated to the harmonics of IF. Fortunately, when the loop is locked, IF tracks f_{FAC} which is the reference of the loop. The harmonics of IF lie at the multiples of f_{FAC}. When sampled by the PD that has a sampling rate of f_{FAC}, they only induce a static phase error. Due to parasitic coupling, frequency upconverting, and switch nonlinearity [69], SH also generates unwanted high-frequency tones. When experiencing nonlinearity from the following circuit, they might generate beat frequencies close to the IF signal and cause unwanted in-band spurs. Those high-frequency tones are at frequencies higher than f_{IAC} which is larger than 1 GHz. Thus, a 4th-order linear low-pass filter (LPF) with a dominant pole at 100 MHz is utilized to attenuate these unwanted tones and pass the IF which is between 50-87 MHz. The resistor divider at the RF port provides the DC

supply for the LPF buffer. The two capacitors C_{buff1} and C_{buff2} form an ac voltage divider to lower the swing of the LPF input, which ensures superior linearity.

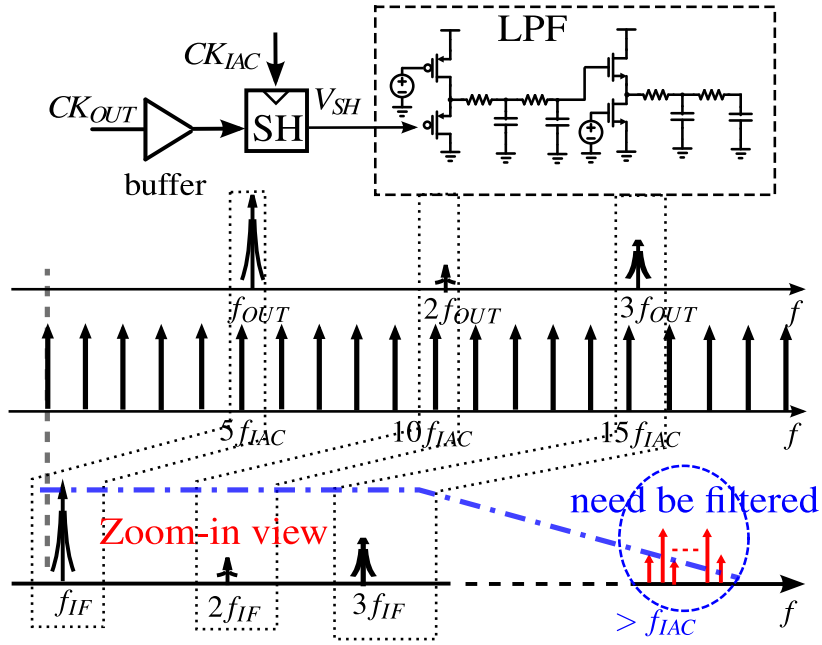


Figure 3.33: LPF filters the unwanted high-frequency spurs at SH output. The in-band spurs only cause static phase error.

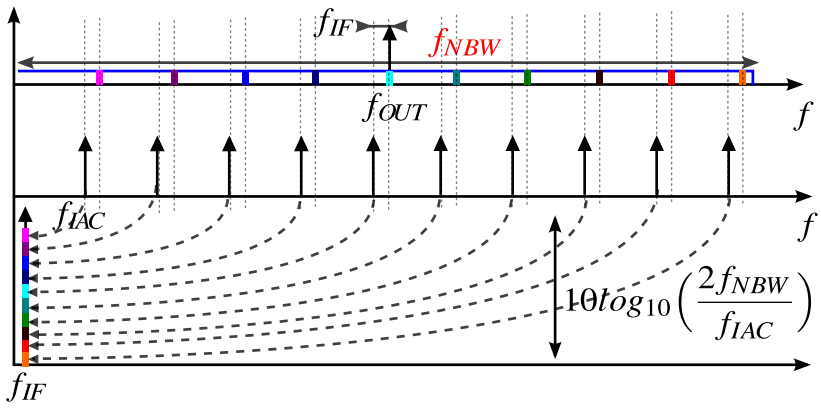


Figure 3.34: Noise-folding in SH circuit.

The SH shifts the flicker noise of the input buffer to IF without a change in SNR. Since the thermal noise of the input buffer has a wide bandwidth, sampling causes noise-folding. Because the switch-cap has a limited bandwidth f_{NBW} which shapes the buffer white noise [66], the folding strength is approximately equal to $2f_{NBW}/f_{IAC}$, as shown in Fig. 3.34. The

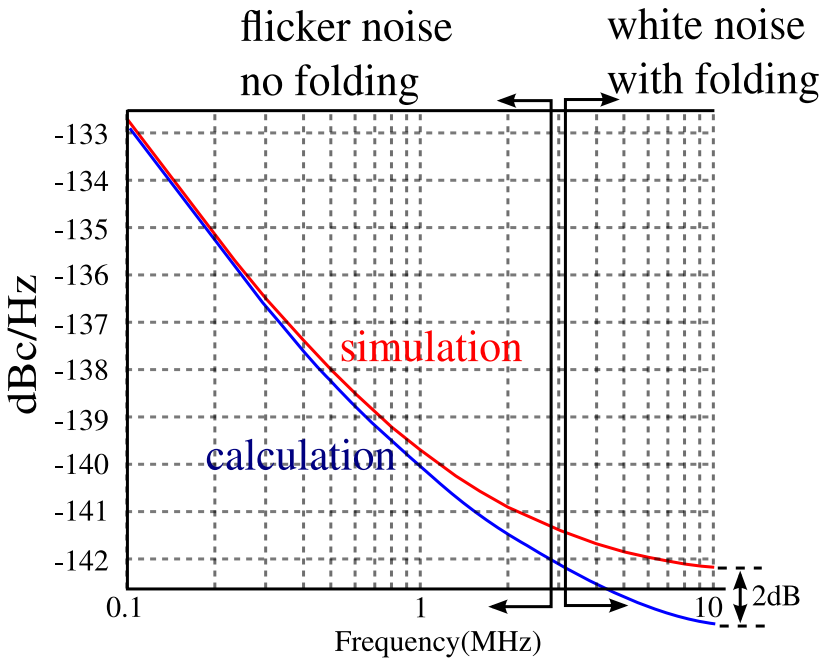


Figure 3.35: SH output phase noise: Prediction and simulation

predicted SH output phase noise matches well with the simulation, as shown in Fig. 3.35. Even with noise-folding, the SH output noise floor is still less than -140 dBc/Hz, which is acceptable in our design. The closed-loop phase noise of the main PLL is band-limited due to loop filtering. It experiences negligible noise-folding when sampled by the SH circuit. In the above, the SH nonideality has negligible effects on the main PLL.

3.6.3 Main PLL locking

Although linear with high gain, a bang-bang PD has a limited locking range because its linear region is very small, and a moderate phase error can completely saturate the PD gain. Thus, a coarse TDC with wide capture range is necessary to accelerate the locking of the main loop.

As shown in Fig. 3.36, the TDC consists of a tri-state PFD and two digital counters. The counter clock T_{cnt}, tunable from 500 MHz - 800 MHz, is divided from our FN PLL. It also determines the TDC resolution. When the leading signal triggers one pulse of the PFD, the pulse enables the corresponding counter until the other pulse, which is triggered by the

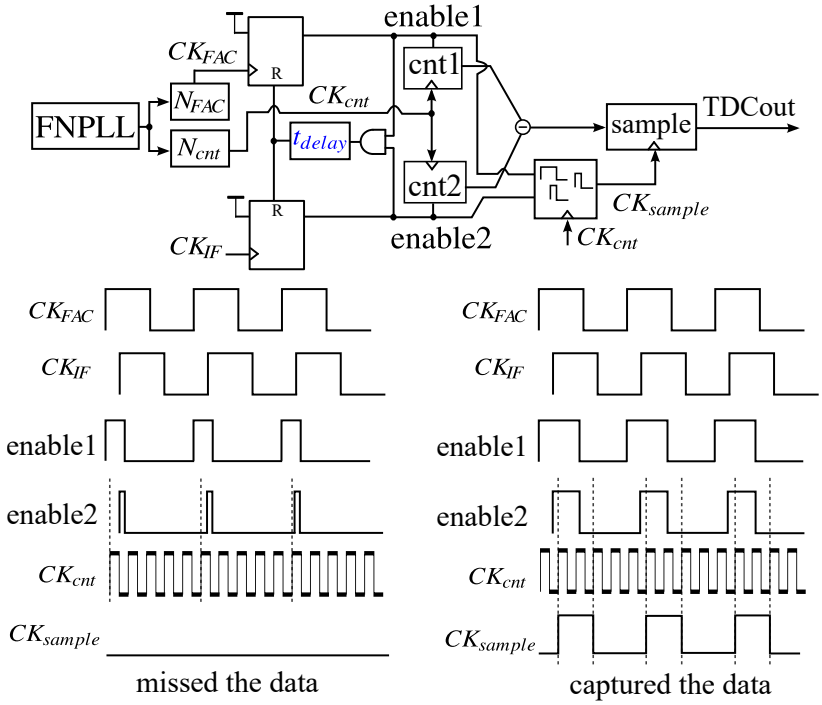


Figure 3.36: Coarse TDC: Tri-state PFD with a wide resetting window controls the on/off of the two counters, which is clocked by the divided FN PLL.

lagging signal, is detected by the TDC. Then, the TDC sends the subtraction of the two counter outputs to the DLPF and resets both counters. The subtraction of the two counters indicates the phase error. The counter clock and the PFD inputs are asynchronous signals. If the lagging pulse is too narrow, the TDC might miss it, leading to incorrect operation. Therefore, the PFD resetting delay is set to approximately 3 ns to ensure that the counter clock can always sample the lagging pulse.

As shown in Fig. 3.37, if the subtraction of the two counters is successively zero for the programmed N reference cycles, it indicates that, during the N cycles of operation, the accumulated phase error due to the frequency error f_{err} between the IF signal and f_{FAC} is still smaller than the resolution of the TDC:

$$2\pi f_{err} N T_{FAC} < \frac{2\pi T_{cnt}}{T_{ref}} \tag{3.11}$$

Then we switch the phase detector to bang-bang PD. The remaining frequency error is:

$$f_{err} < \frac{T_{cnt} f_{ref}^2}{N} \tag{3.12}$$

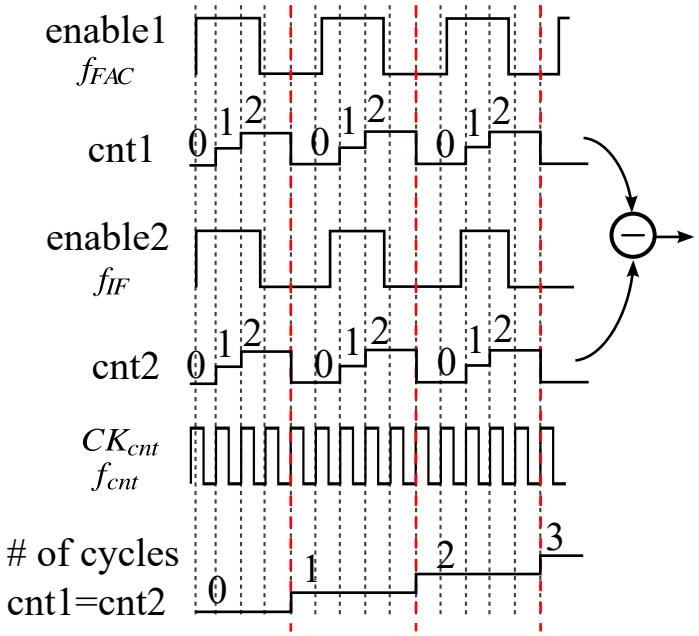


Figure 3.37: When it is sensed that both enabled signals are low, the counters are reset, and the system evaluates if switched from TDC to BBPD.

.

When transferred to bang-bang PD, the loop gain T_{PLL} should compensate the remaining frequency error and ensure further reduction of the remaining phase error:

$$T_{PLL} = G\beta K_{VCO} > f_{err} \tag{3.13}$$

As shown in Fig. 3.38, when the bang-bang PD output starts to periodically toggle, the PLL limit cycle occurs, which indicates that the loop gain is too large. Then, we slightly reduce G and wait for the next toggle. This process is repeated until G is 1. Then, PLL enters its steady state operation. With this auxiliary frequency acquisition, the simulated locking time of the main PLL is within 15 us, as shown in Fig. 3.39.

3.6.4 Main PLL schematic

The main PLL schematic is shown in Fig. 3.40. The DAC is a 10-bit r-ladder. The main VCO, oscillating at 7-9 GHz, has a tail inductor to boost the noise performance. After the 4th order LPF, there will be a sinewave. The digital buffer converts the sinewave to pseudo-

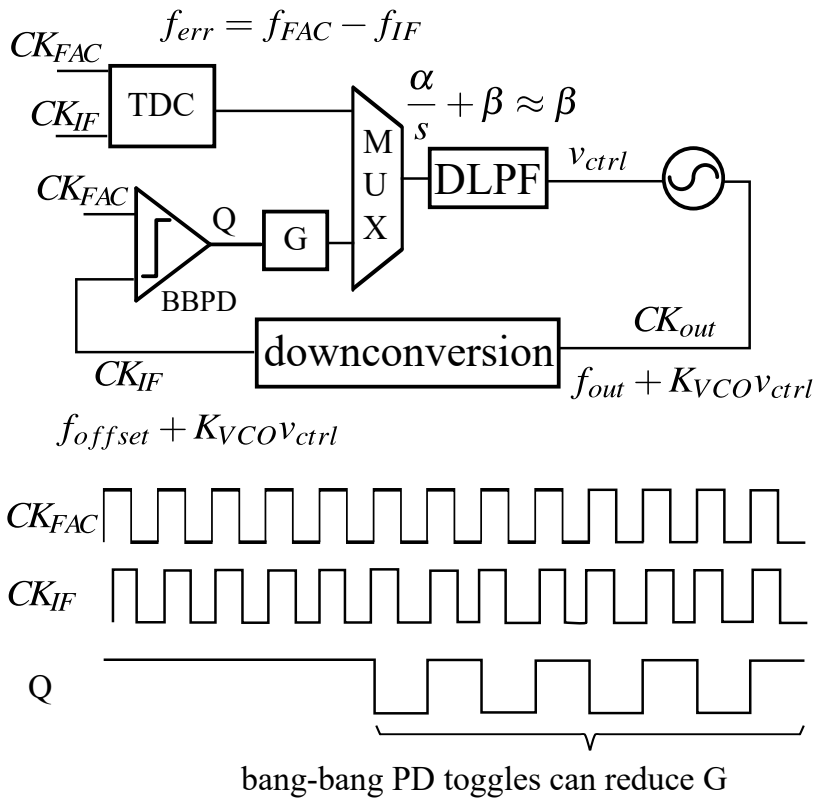


Figure 3.38: When bang-bang PD output periodically toggles, PLL limit cycle occurs. The G needs to be reduced.

differential square waves, which drives the regenerative bang-bang PD. The bang-bang PD sends digital bits into the digital LPF. By the operation of digital LPF and the RDAC, the control voltage is generated to tune the VCO.

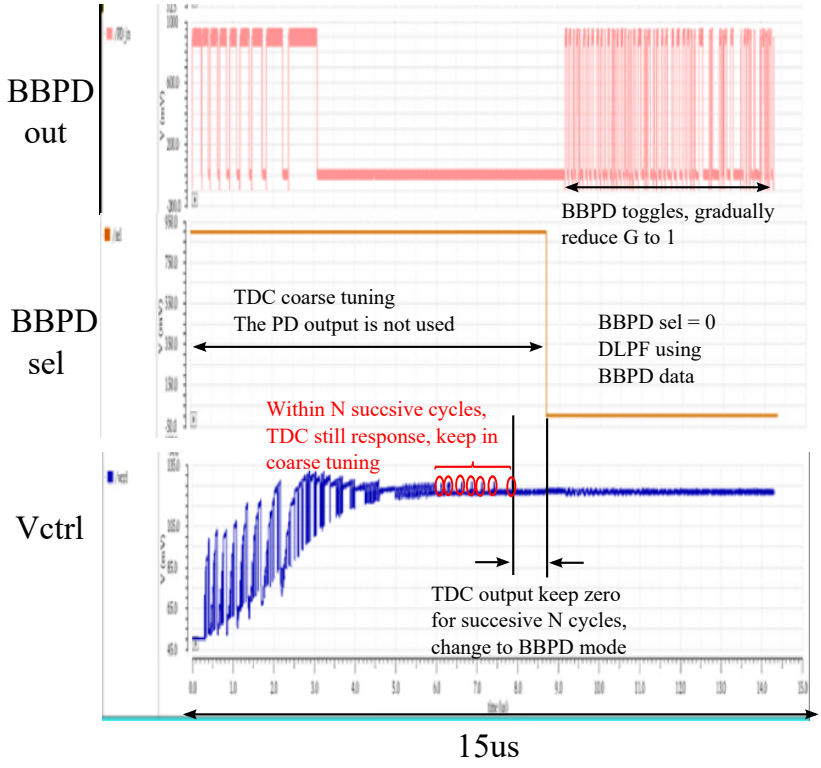


Figure 3.39: Transient simulation waveform to illustrate the main PLL locking procedure.

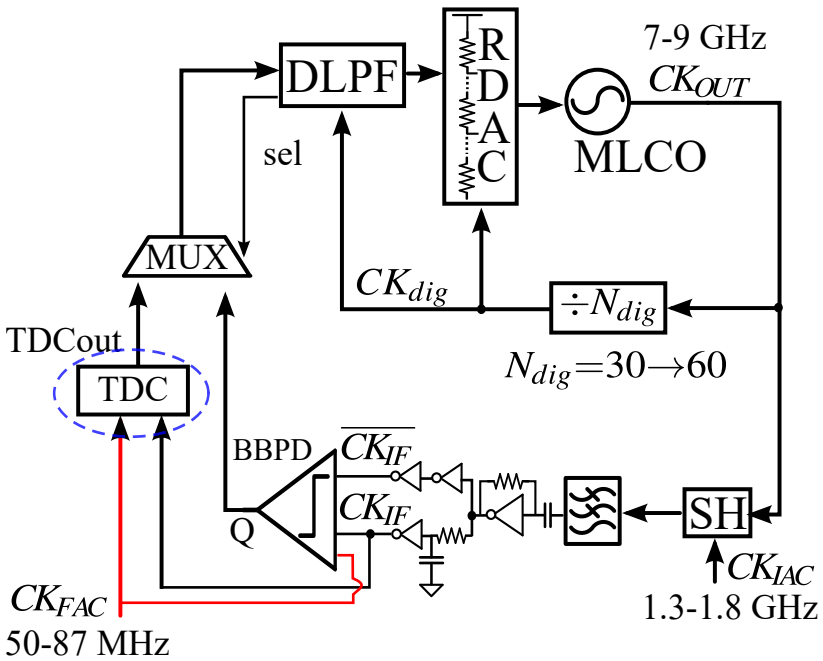


Figure 3.40: Main PLL architecture.

3.7 Spur reduction techniques

The deep N-well layer is buried underneath the DLPF to prevent the noise of DLPF from coupling to other critical blocks. LDOs are allocated to blocks with different operating frequencies to avoid supply couplings. Three PLLs and the crystal reference have separated supply and ground pads. These pads are star-connected on the PCB board. The supply on the board is close to an ac short circuit, prohibiting the supply fluctuation of each block from bouncing to other blocks, as shown in Fig. 3.41.

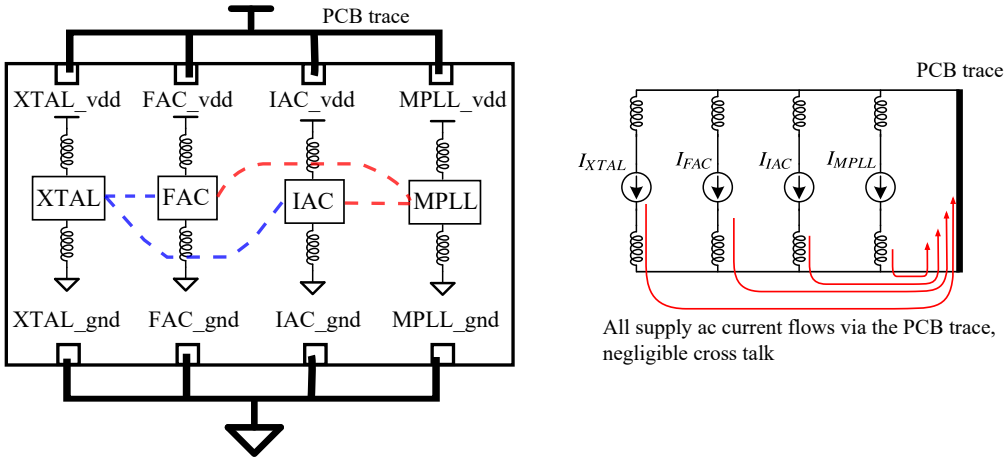


Figure 3.41: Separate on-chip power and ground for three PLLs, and connect them off-chip through PCB low resistance trace.

Although our architecture reduces the pulling of the two VCOs, the magnetic coupling between the two inductors still cause high frequency spurs. Moreover, the switching operation and nonlinearity of the other blocks might fold these spurs in-band. To further reduce the magnetic coupling, the VCO of the integer-N PLL uses a bowtie inductor. EMX simulation is utilized to identify the best location of two inductors. As shown in Fig. 3.42, when two inductors are in-phase, magnetic shielding is most manifest, providing -80 dB shielding; on the other hand, when two inductors are orthogonal, the shielding is worst, which is only -66 dB. This observation matches with [70]. A simple explanation for this is that when two inductors are in-phase, the two circles of the bowtie inductor generate opposite eddy currents with the equal strength, which cancel each other. The whole inductor thus suffers negligible coupling.

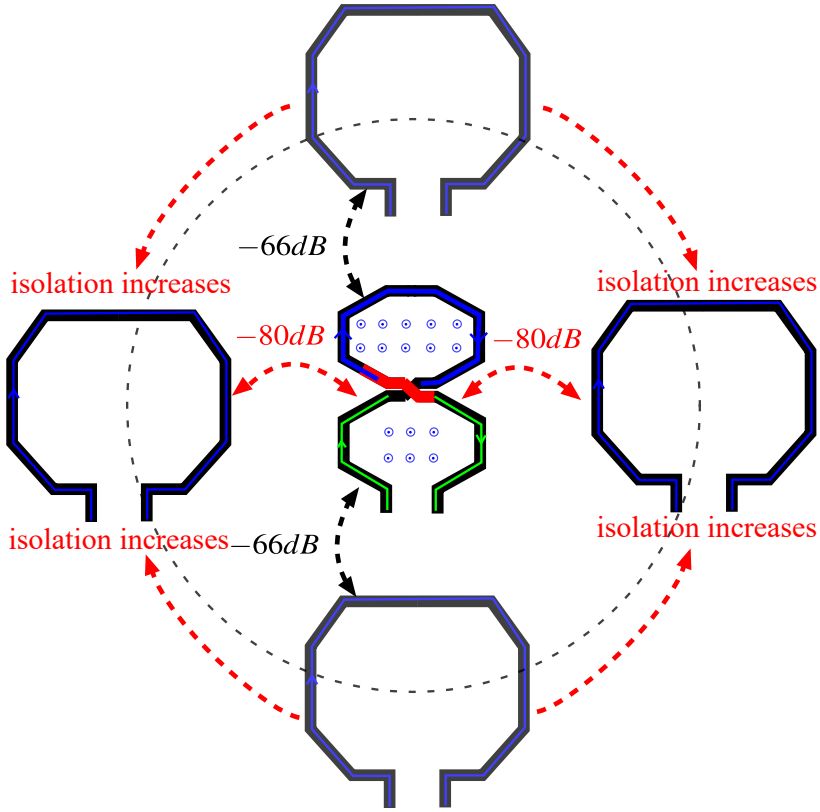


Figure 3.42: Horizontal alignment of two VCOs can obtain maximum isolation.

3.8 Measurement

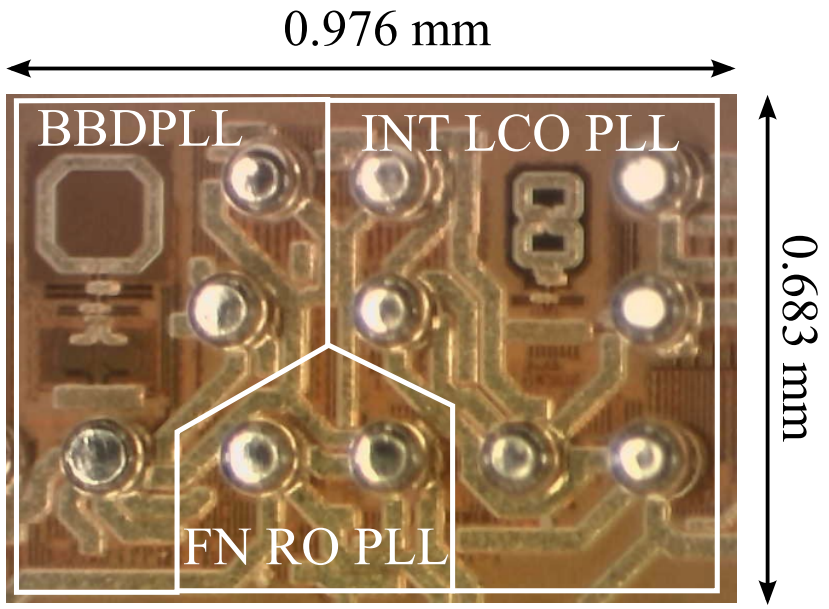


Figure 3.43: Chip die photo.

Our triple-loop PLL is fabricated in the TSMC 16nm CMOS technology. The die is shown in Fig. 3.43. The core area of the circuit is 0.25 mm^2, while the total chip occupies 0.6 mm^2. We use an Agilent E4440A PSA Series Spectrum Analyzer to measure the PLL output. An on-chip 60 MHz crystal oscillator serves as the reference. Fig. 3.44 shows the PLL noise performance at around 7.7 GHz and 8.7 GHz. At both frequencies, with the noise being integrated from 10 kHz to 10 MHz, the PLL achieves 131 fs RMS jitter. After a 2/3 divider, which centers the frequency to a 5.805 GHz 802.11a channel, the jitter corresponds to 0.27° RMS integrated phase noise.

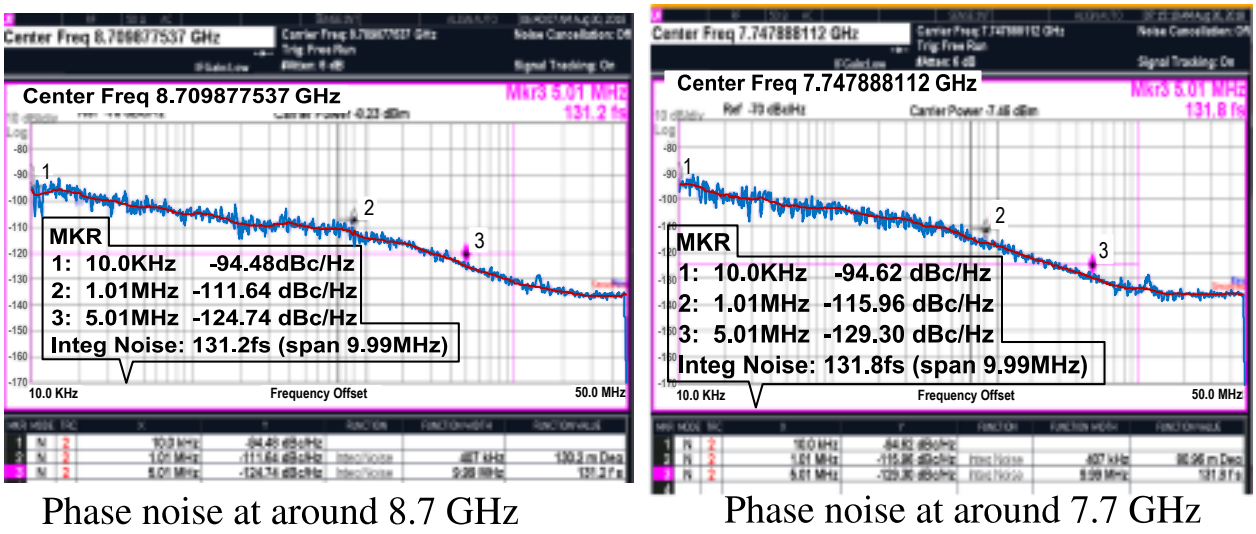


Figure 3.44: Measure phase noise at approximately 7.7 GHz and 8.7 GHz.

Fig. 3.45 plots the worst case spurious tones when sweeping the fractional dividing ratio of the FN PLL. The worst fractional spur is less than -70 dBc. The swept fractional spurs at two operating frequencies have similar levels, which implies that the main PLL indeed has a unity closed-loop transfer function. At a higher frequency, the spurs are attenuated due to the main PLL filtering. Since f_{FAC} determines the main PLL reference frequency, when sweeping it from 50 MHz to 87 MHz, the corresponding offset frequency and level of the main loop reference spur change accordingly. The worst-case reference spur is -66 dBc. We believe that this spur mechanism is the coupling from the N_{IAC} and its routine to the the main VCO. Because of strong filtering from three PLLs, no spur is observed at the frequency of the crystal, i.e., 60 MHz. Fig. 3.46 are two examples of measured spectrum.

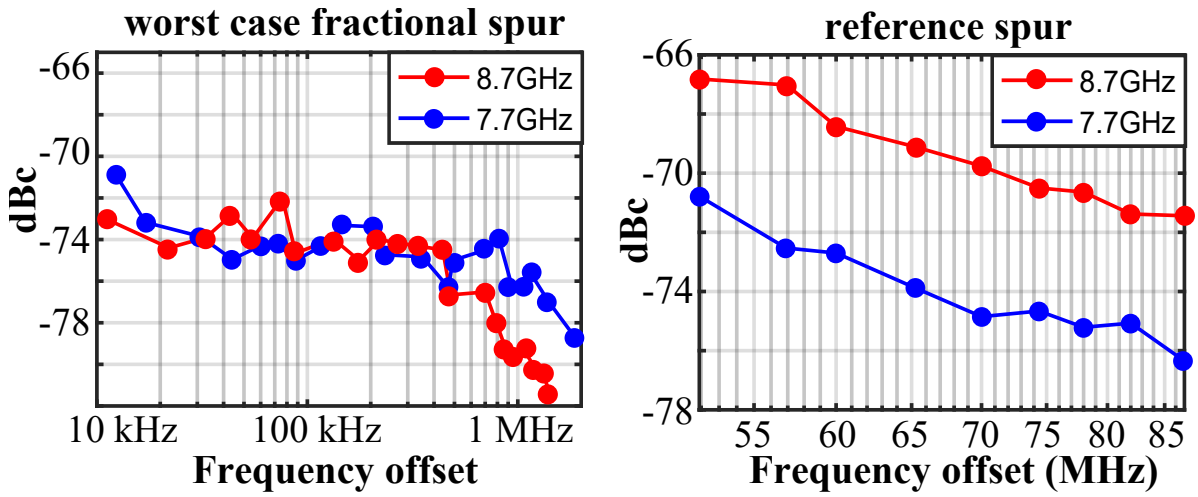


Figure 3.45: Swept worst-case fractional spurs and reference spurs at around 7.7 GHz and 8.7 GHz.

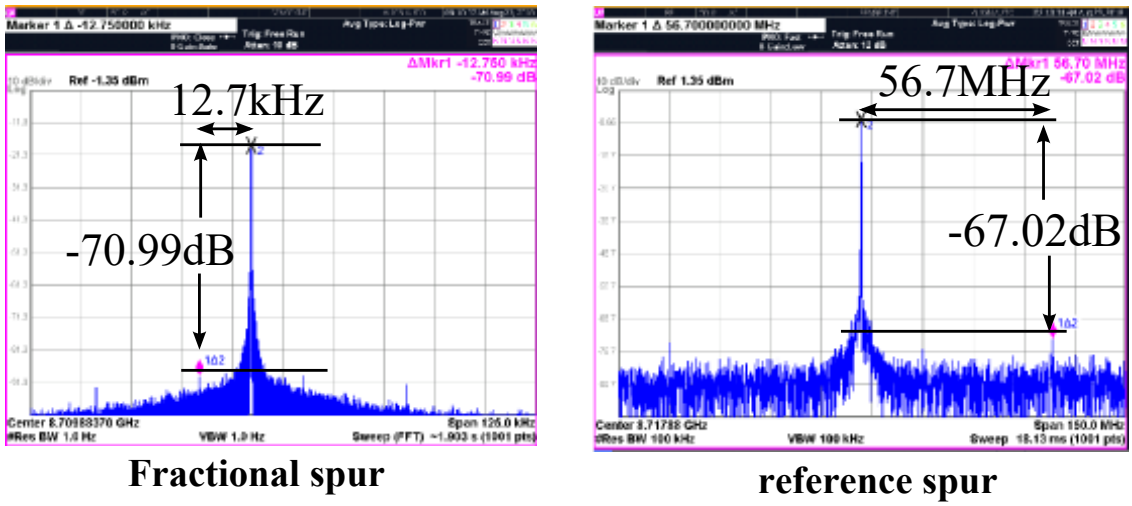


Figure 3.46: Examples of measurement.

We also compared the measurement with our analytical model, as shown in Fig. 3.47. They match very well. Based on our analysis, the main VCO contribute 45 % of the total jitter, integer PLL and the crystal each contributes roughly 20 %. The FN PLL contributes only 1 %, which indicates that our structure indeed can efficiently suppress the FN loop noise.

The supply of this prototype is 0.9V. The fractional PLL with the cascaded divider consumes 1.9mA; the integer PLL consumes 2.8mA; the divider cascaded with the integer

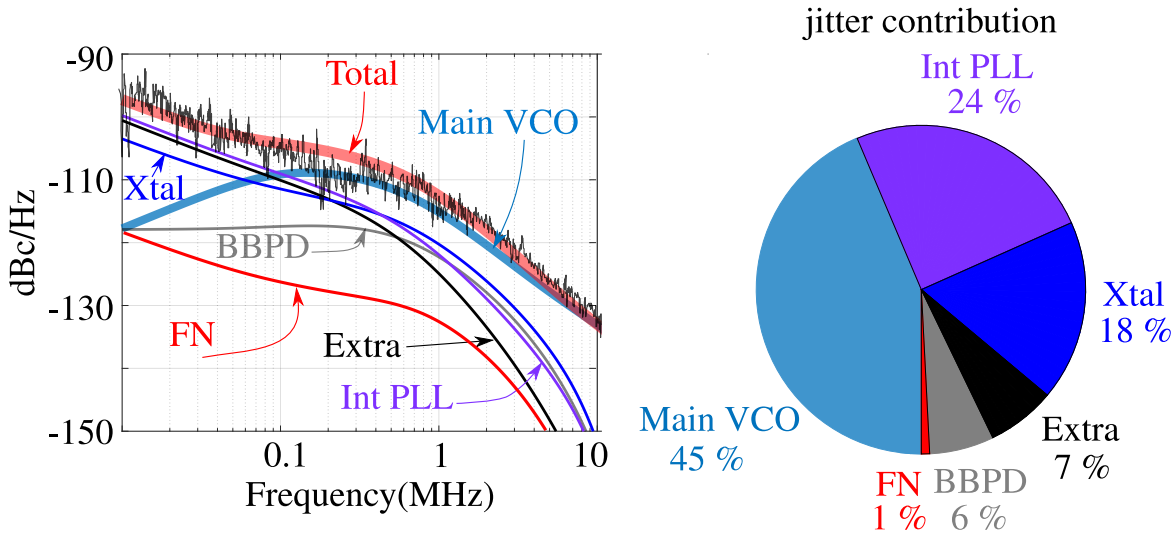


Figure 3.47: Model predictions and jitter contribution.

PLL consumes 2.1mA due to its high input frequency and long routing wires; the main PLL consumes 6.6mA of which the main VCO takes 4.9mA, leading to total power consumption of 13.4mA. The FOM of the proposed PLL is - 246.4dB. Performance summary and comparison are shown in 3.1.

	Our Design	**Yao ISSCC17**	**Gao ISSCC16**	**Gao ISSCC15**	**Bertulessi ISSCC18**	**Chen ISSCC15**
Architecture	Triple-loop BBDPLL	DPLL	DPLL	DPLL	BBDPLL	Subsampling DPLL
Calibration	**no**	yes	yes	yes	yes	yes
Power(mW)	13.4	13.4	8.2	9.5	5.28	11.5
Reference (MHz)	60	26	40	40	52	49.15
Output (GHz)	7-9	2.7	2.7-4.33	3.2-2.8	3.7-4.1	2.6-3.9
RMS jitter (fs)	**131**	137	159	173	183	226
In-band PN (dBc/Hz)[1]	**-108.5**	-103.4	-104	-103.5	-99.5	-100.3
Ref. Spur (dBc)[1]	<-66	-77.4	-74.5	-72	N/A	-50
Frac. Spur (dBc)[1]	**<-70**	-68.4	-50.5	N/A	-43	-52
FOM (dB)[2]	-246.4	-245.9	-246.8	-245.5	-247.5	-241.8
Area (mm^2)	0.25	0.257	0.3	0.3	0.61	0.23
Process (nm)	16	14	28	28	65	65

Table 3.1: Comparison with the state of art design.

[1]The phase noise and spurs are normalized to 8.7 GHz for fair comparison.

[2]$FoM = 10\log_{10}\left[(\sigma^2_{jitter}) \times Power/1mW\right]$

CHAPTER 4

Frequency tuning-range extension

4.1 Frequency tuning-range extension

The previous chapter only described a low noise and low spur synthesizer with a 30% frequency tuning range. However, a 100% tuning range oscillator is necessary for the SDR system. Normally, the oscillator consists of a bank of resonators to cover the frequency range [71]. However, such a design occupies a large chip area, opposing the modern IC design trend. A coupled LC resonator has both even and odd resonant modes. Exploiting both modes can realize a compact 100% tuning range LC oscillator [72] but maintaining low noise at both resonant modes is still very challenging.

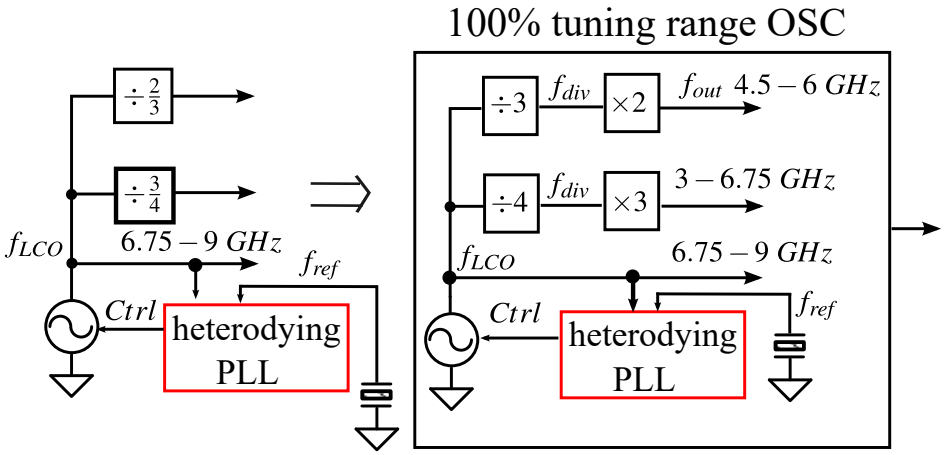


Figure 4.1: Fractional divider to achieve frequency extension.

Instead of brutally designing a 100% tuning range LC oscillator, we propose a frequency extender, which expands a 30% tuning range LC oscillator to 100%. As shown in Fig. 4.1, a fractional divider with a dividing ratio of 2/3 and 3/4 can perform frequency extension. A miller divider [73] is a traditional fractional divider, but requires an inductor for sharp

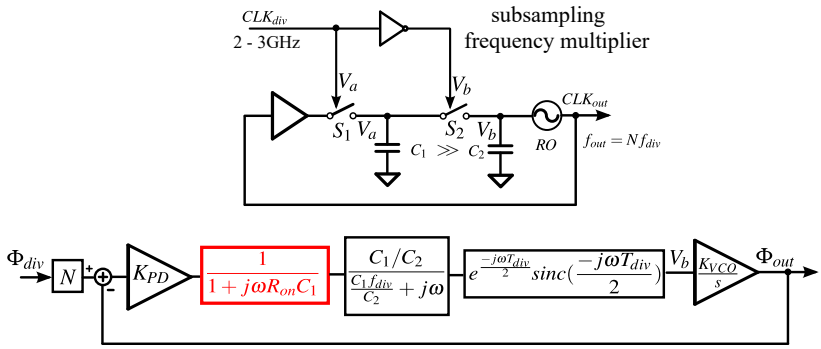


Figure 4.2: The architecture and model of the proposed subsampling PLL modified from the sampling PLL for better performance. R_{on} is the on resistance of the switch.

filtering, occupying a large area. Our proposed fractional divider consists of an integer divider as the numerator and an integer multiplier as the denominator. The design challenge of such a structure is a low-spur and low-noise frequency multiplier. The previously discussed Int PLL is promising for this implementation. Since the input frequency of the multiplier is higher than 1.68 GHz, the PLL can have a bandwidth on the order of 100 MHz. Such a wide bandwidth PLL efficiently suppresses the oscillator phase noise, allowing the implementation of a ring oscillator to conserve area. With a high reference frequency, the signal coupling in the switch-cap filter manifests, degrading the null strength of the filter. The noise contributed by the phase detector and divider becomes comparable with the oscillator's, and experiences negligible attenuation from the wide bandwidth PLL. In such a case, the closed-loop gain induced by the loop divider matters. Even a divide-by-2 causes a 6 dB incremental on the output spurs and the output noise from PD and divider. Subsampling, as mentioned previously, with no loop divider, provides us with at least 6 dB of extra spur and noise reduction. Directly connecting the ring oscillator to the phase detector modifies our sampling PLL to a subsampling one. The model of the modified PLL is shown in Fig. 4.2. Without the loop divider, the new PLL has a unity closed-loop gain.

Transistor-level simulation has been performed to compare the architectures. The sampling PLL consumes 50 mA to approach a similar noise performance to the subsampling one, which only consumes 33 mA. Around 6.6 GHz, the spurs in the sampling PLL is -46 dBc, while in subsampling PLL is only -61 dBc. Consequently, the subsampling PLL is preferable

as a frequency multiplier. Since the loop bandwidth is larger than 100 MHz, the previously mentioned pole introduced at the tracking phase of switch cap can not be ignored. As shown in Fig. 4.2, with the switch-cap pole and the half period delay together, it degrades the loop phase margin and causes the noise peaking at 300 MHz offset in Fig. 4.3.

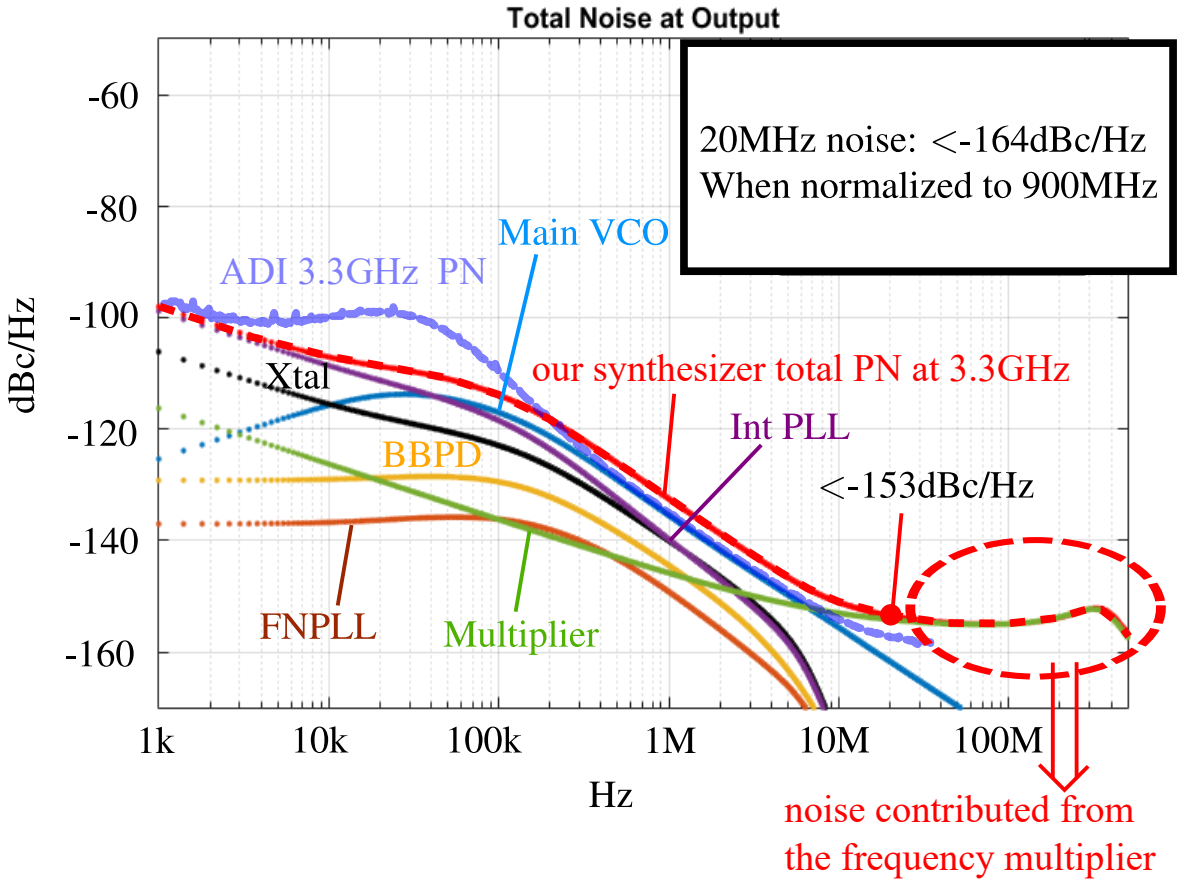


Figure 4.3: Noise comparison between our architecture and ADI product.

With the extender and our proposed triple-loop PLL, we can achieve a 100% tuning range synthesizer, which is the essence of the SDR clock generation. The GSM transmitter standard, which requires -162 dBc/Hz phase noise at 20 MHz offset is the most challenging specification for synthesizer design. As shown in Fig. 3.47, Because of the cascade loop filtering, the noises of the two reference are substantially suppressed at 20 MHz offset. To meet GSM requirement, we only need to scale up the power of the main VCO in our heterodyne synthesizer. Based on the measured FOM of the main VCO, which is -190 dB, we need to burn 50 mA in the VCO to meet the GSM requirement. With such a low-noise oscillator,

the loop bandwidth is optimized to 200 kHz which leads to less than 90 fs integral jitter. The design of the two auxiliary is unaltered which still only consumes around 6.8 mA. The frequency multiplier consumes 33 mA. Based on the ADI data sheet [71], The integer divider needs another 6 mA. In total, the synthesizer consumes 104 mA to meet the GSM standard and also achieve 100% tuning range. We compared our design with the ADI4355 [71] which is the only standalone PLL product for a wideband radio system. The comparison in 4.1 illustrates that our complete synthesizer architecture can more efficiently meet the requirements of the market. Fig. 4.3 is the predicted noise of our synthesizer at 3.3 GHz. When normalized to 900 MHz, the 20 MHz noise is less than -162 dBc/Hz, meeting the GSM specification. The 300 MHz noise peaking is caused by the frequency multiplier.

	Our Design	**ADI4355[71]**
Loop BW(kHz)	200	35
Power(mW)	104	117
Fout(GHz)	0.07-9	0.05-6.6
Number of inductor	2	4
RMS jitter (fs)	90	400
Supply (V)	0.9	3.3

Table 4.1: Comparison with ADI 4335.

CHAPTER 5

Conclusion

This thesis conducted a thorough analysis of DLLs and provided simple but accurate LTI models for both edge-combining and recirculating-delay types of DLLs. The RDLL was found to outperform the ECDLL regarding spur amplification, phase noise, and output power. The modification of the ring oscillator sufficiently suppressed the oscillator phase noise. Therefore, a DLL does not rely on the loop to suppress VCO noise. There is no design trade-off, unlike the case of PLL. However, a DLL all-pass filters its input noise and spurs. It cannot filter SDM noise and reference spurs. This bottleneck limits its implementation as a high-performance wideband synthesizer.

Then, a novel triple-loop PLL, which enables the generation of high spectrum purity clock for high-performance RF system implementation, is designed. Inspired from the heterodyne PLL, this triple-loop structure relaxes the fundamental trade-off between phase noise and bandwidth in a conventional fractional PLL and does not require calibration. It also provides a straightforward spur reduction solution without the necessity of calibration. The implementation of the harmonic mixing allows this triple-loop PLL to be integrated on a single chip. An accurate analytical model is established, which allows design optimization and shortens the R&D cycle.

A compact frequency extender, which is based on a subsampling PLL, is designed to extend the tuning range of an oscillator from 30% to 100%. By combining the frequency extender and the triple-loop PLL, the targeted wideband frequency synthesizer is achieved.

In summary, this dissertation investigated many kinds of clock generation techniques, including DLL, analog PLL, digital PLL, bang-bang PLL, sampling PLL, subsampling PLL, and heterodyne PLL. It then provides a solution, which combines the merit of different loops

to generate a high-performance synthesizer which is suitable for a wideband radio system.

CHAPTER 6

Appendix

6.1 Noise model of injection-locking oscillator

Injection-locking can be implemented in both LC oscillator [40] and a ring oscillator [13]. As shown in Fig. 6.1 (a), in an LC oscillator, a clean reference clock periodically turns on the switch across the LC tank at the edge crossing of the oscillator. The switch redistributes the remaining charge at both sides, converting the differential noise to a common mode signal. As the single-side model illustrates, this process is equivalent to erase the stored noise in the capacitor. As shown in Fig. 6.1 (b), during every reference cycle, the injected signal pulls the edge of the oscillator and forces it to be aligned with the reference. The noise stored in the oscillator, therefore, will also be reset.

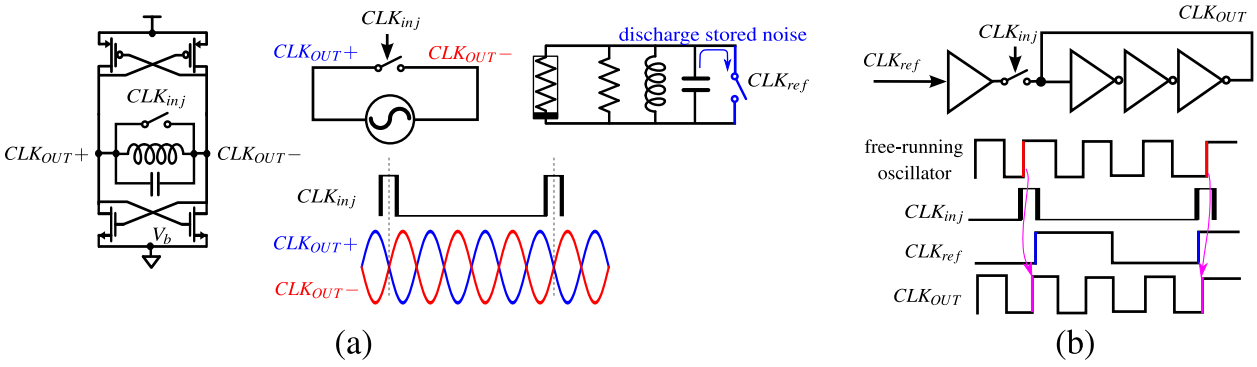


Figure 6.1: (a) Injection-locking LC oscillator; (b) Injection-locking ring oscillator.

As shown in Fig. 6.2, before the injection pulse, the jitter in the oscillator accumulates every cycle. At the injection instance, the limited discharging time and constraint pulling strength prevent the injection signal from completely erasing the noise in the oscillator. The

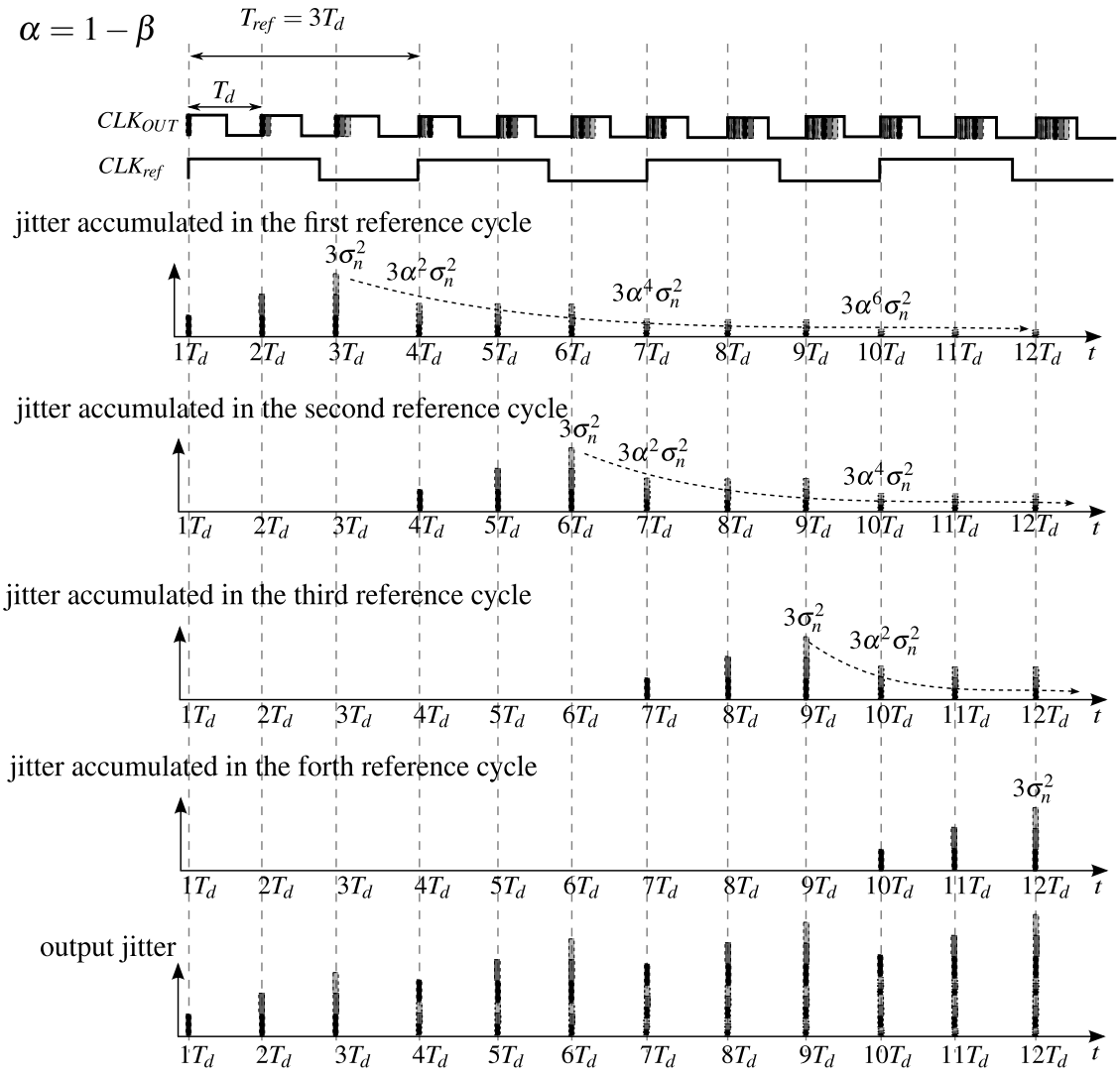


Figure 6.2: The jitter of an injection-locking oscillator.

remaining noise is held and still perturbs the phase of the oscillator. At the next injection instance, the injection signal reduces the new accumulated noise and further attenuates the previous residual noise. All the residue noise from the previous operation perturbs the current phase of the oscillator, which is a lossy integral function. Meanwhile, the current accumulated noise also modulates the oscillator. Therefore, the model of an injection-locking oscillator is shown as Fig. 6.3. The oscillator phase noise is:

$$\mathcal{L}_{inj}(f) = \sum_{n=0}^{n=N-1} \mathcal{L}_{\phi_{DLn}}(f) \left| \frac{1 - e^{-j2(N-n)\pi f T_d}}{1 - e^{-j2\pi f T_d}} + \frac{1 - z^{-N}}{1 - z^{-1}} \cdot \frac{(1-\beta)z^{-N}}{1 - (1-\beta)z^{-N}} \right|^2 \tag{6.1}$$

Where β is the injection locking strength which has a value between 0 to 1. For the in-band noise, (6.1) can be simplified as:

$$\mathcal{L}_{inj}(f) \approx \sum_{n=0}^{n=N-1} \mathcal{L}_{\phi_{DLn}}(f)\left|(N-n)^2 + N^2\frac{(1-\beta)}{\beta}\right|^2 \tag{6.2}$$

Combined with (2.16), the in-band noise of an injection-locking oscillator is:

$$\mathcal{L}_{inj} = \pi^2\sigma_n^2\left(\left(\frac{2}{\beta}-1\right)^2 + \frac{1}{3}\right) \tag{6.3}$$

(6.3) compeletly matches with the conclusion in [74]. But here, we arrive the same conclusion based on a simple LTI model.

Then the noise suppression bandwidth of an injection-locking oscillator is:

$$f_{NBW_{inj}} = \sqrt{\frac{f_{ref}}{\pi^2\left(\frac{1}{3}+\left(\frac{2}{\beta}-1\right)^2\right)}} \tag{6.4}$$

When β is 1, this oscillator works as a DLL. The suppression bandwidth is enlarged to (2.27).

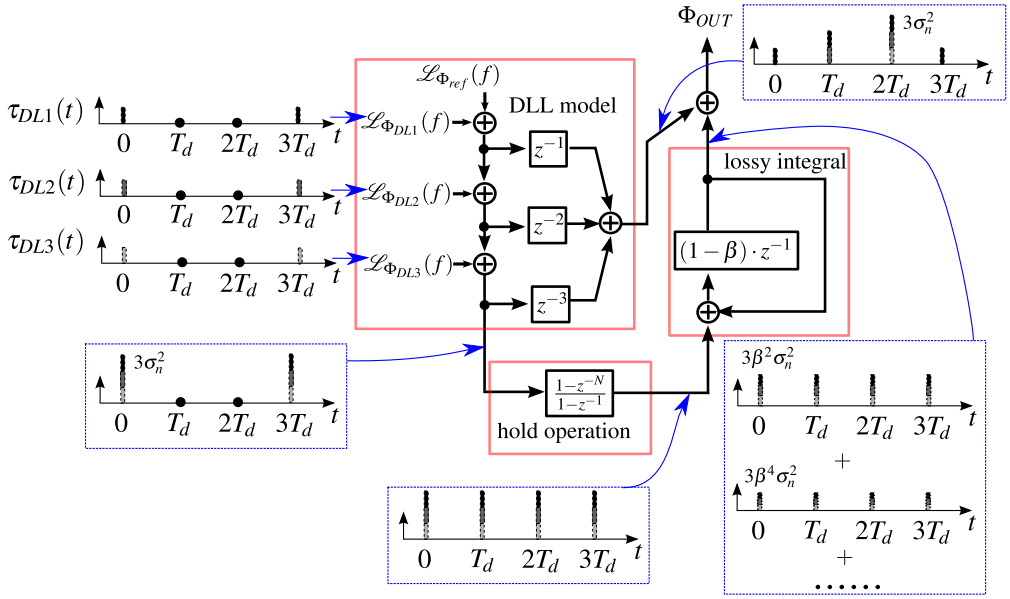


Figure 6.3: Model of injection-locking oscillator.

REFERENCES

[1] A. A. Abidi, "Phase Noise and Jitter in CMOS Ring Oscillators," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp. 1803–1816, Aug 2006.

[2] F.-R. Liao and S.-S. Lu, "A Waveform-Dependent Phase-Noise Analysis for Edge-Combining DLL Frequency Multipliers," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 60, no. 4, pp. 1086–1096, April 2012.

[3] S. Ye, L. Jansson, and I. Galton, "Techniques for In-band Phase Noise Suppression in Re-circulating DLLs," in *Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE 2003*, Sept 2003, pp. 297–300.

[4] C. Ho and M. S. Chen, "A Digital PLL With Feedforward Multi-Tone Spur Cancellation Scheme Achieving < -73 dBc Fractional Spur and < -110 dBc Reference Spur in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 12, pp. 3216–3230, Dec 2016.

[5] M. Z. Straayer and M. H. Perrott, "A Multi-Path Gated Ring Oscillator TDC With First-Order Noise Shaping," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1089–1098, April 2009.

[6] S. Levantino, L. Romano, S. Pellerano, C. Samori, and A. L. Lacaita, "Phase Noise in Digital Frequency Dividers," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 5, pp. 775–784, May 2004.

[7] K. J. Wang, A. Swaminathan, and I. Galton, "Spurious Tone Suppression Techniques Applied to a Wide-Bandwidth 2.4 GHz Fractional-N PLL," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2787–2797, Dec 2008.

[8] K. Hosseini, B. Fitzgibbon, and M. P. Kennedy, "Observations Concerning the Generation of Spurious Tones in Digital Delta-Sigma Modulators Followed by a Memoryless Nonlinearity," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 11, pp. 714–718, Nov 2011.

[9] G.Chien, "Low-Noise Local Oscillator Design Techniques using a DLL based Frequency Multiplier for Wireless Applications," Ph.D. dissertation, University of California, Berkeley, Berkeley, Spring 2000.

[10] T.-C. Lee and K.-J. Hsiao, "The Design and Analysis of a DLL-Based Frequency Synthesizer for UWB Application," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 6, pp. 1245–1252, June 2006.

[11] F. Liao and S. Lu, "A Programmable Edge-Combining DLL With a Current-Splitting Charge Pump for Spur Suppression," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 12, pp. 946–950, Dec 2010.

[12] K. Ryu, J. Jung, D.-H. Jung, J. Kim, and S.-O. Jung, "High-Speed, Low-Power, and Highly Reliable Frequency Multiplier for DLL-Based Clock Generator," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,*, vol. PP, no. 99, pp. 1–1, 2015.

[13] S. Ye, L. Jansson, and I. Galton, "A Multiple-Crystal Interface PLL With VCO Realignment to Reduce Phase Noise," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1795–1803, Dec 2002.

[14] A. Elshazly, R. Inti, B. Young, and P. Hanumolu, "Clock Multiplication Techniques Using Digital Multiplying Delay-Locked Loops," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 6, pp. 1416–1428, June 2013.

[15] B. Helal, M. Straayer, G.-Y. Wei, and M. Perrott, "A Highly Digital MDLL-Based Clock Multiplier That Leverages a Self-Scrambling Time-to-Digital Converter to Achieve Subpicosecond Jitter Performance," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4, pp. 855–863, April 2008.

[16] 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 Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1804–1812, Dec 2002.

[17] M.-J. Lee, W. Dally, T. Greer, H.-T. Ng, R. Farjad-rad, J. Poulton, and R. Senthinathan, "Jitter Transfer Characteristics of Delay-Locked Loops - Theories and Design Techniques," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 4, pp. 614–621, Apr 2003.

[18] R. C. H. van de Beek, E. A. Klumperink, C. S. Vaucher, and B. Nauta, "Low-Jitter Clock Multiplication: A Comparison Between PLLs and DLLs," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,*, vol. 49, no. 8, pp. 555–566, Aug 2002.

[19] X. Gao, E. Klumperink, and B. Nauta, "Advantages of Shift Registers Over DLLs for Flexible Low Jitter Multiphase Clock Generation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 3, pp. 244–248, March 2008.

[20] O. Casha, I. Grech, F. Badets, D. Morche, and J. Micallef, "Analysis of the Spur Characteristics of Edge-Combining DLL-Based Frequency Multipliers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 2, pp. 132–136, Feb 2009.

[21] A. Ojani, B. Mesgarzadeh, and A. Alvandpour, "Modeling and Analysis of Harmonic Spurs in DLL-Based Frequency Synthesizers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 11, pp. 3075–3084, Nov 2014.

[22] H. Guo, X. Wang, and T. Kwasniewski, "Spur Analysis and Reduction of Edge Combining DLL-based Frequency Multiplier," in *2013 26th Annual IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2013, pp. 1–4.

[23] G. Tak and K. Lee, "A Low-Reference Spur MDLL-Based Clock Multiplier and Derivation of Discrete-Time Noise Transfer Function for Phase Noise Analysis," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 2, pp. 485–497, Feb 2018.

[24] S. L. J. Gierkink, "Low-Spur, Low-Phase-Noise Clock Multiplier Based on a Combination of PLL and Recirculating DLL With Dual-Pulse Ring Oscillator and Self-Correcting Charge Pump," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2967–2976, Dec 2008.

[25] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, New Jersey: Prentice-Hall, 2000.

[26] F. Gardner, "Charge-pump phase-lock loops," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1849–1858, November 1980.

[27] D. Murphy, J. J. Rael, and A. A. Abidi, "Phase Noise in LC Oscillators: A Phasor-Based Analysis of a General Result and of Loaded Q," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 6, pp. 1187–1203, June 2010.

[28] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 1991.

[29] A. Homayoun and B. Razavi, "Relation Between Delay Line Phase Noise and Ring Oscillator Phase Noise," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 2, pp. 384–391, Feb 2014.

[30] H. Xu and A. A. Abidi, "Design Methodology for Phase-Locked Loops Using Binary (Bang-Bang) Phase Detectors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 7, pp. 1637–1650, July 2017.

[31] M.-J. Lee, W. Dally, T. Greer, H.-T. Ng, R. Farjad-rad, J. Poulton, and R. Senthinathan, "Jitter Transfer Characteristics of Delay-Locked Loops - Theories and Design Techniques," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 4, pp. 614–621, Apr 2003.

[32] D. Park and S. Cho, "A 14.2 mW 2.55-to-3 GHz Cascaded PLL With Reference Injection and 800 MHz Delta-Sigma Modulator in 0.13um CMOS," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 12, pp. 2989–2998, Dec 2012.

[33] M. Corinthios, *Signals, Systems, Transforms, and Digital Signal Processing with MATLAB*, 1st ed. Boca Raton, FL: CRC Press, 2009.

[34] U.-K. Moon, K. Mayaram, and J. T. Stonick, "Spectral Analysis of Time-Domain Phase Jitter Measurements," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 49, no. 5, pp. 321–327, May 2002.

[35] "Understanding and Characterizing Timing Jitter," *Tektronix*, 2003 [Online]. Available: http://www.tektronix.com/jitter.

[36] Z. Chen, Y. Wang, J. Shin, Y. Zhao, S. A. Mirhaj, Y. Kuan, H. Chen, C. Jou, M. Tsai, F. Hsueh, and M. F. Chang, "Sub-sampling All-digital Fractional-N Frequency Synthesizer with −111 dBc/Hz In-band Phase Noise and an FOM of −242 dB," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, Feb 2015, pp. 1–3.

[37] S. Levantino, G. Marucci, G. Marzin, A. Fenaroli, C. Samori, and A. L. Lacaita, "A 1.7 GHz Fractional-N Frequency Synthesizer Based on a Multiplying Delay-Locked Loop," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 11, pp. 2678–2691, Nov 2015.

[38] M. Perrott, M. Trott, and C. Sodini, "A Modeling Approach for Σ-Δ Fractional-N Frequency Synthesizers Allowing Straightforward Noise Analysis," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 8, pp. 1028–1038, Aug 2002.

[39] H. Kim, J. Sang, H. Kim, Y. Jo, T. Kim, H. Park, and S. H. Cho, "A 5GHz −95dBc-Reference-Spur 9.5mW Digital Fractional-N PLL Using Reference-Multiplied Time-to-Digital Converter and Reference-Spur Cancellation in 65nm CMOS," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, Feb 2015, pp. 1–3.

[40] A. Elkholy, M. Talegaonkar, T. Anand, and P. Kumar Hanumolu, "Design and Analysis of Low-Power High-Frequency Robust Sub-Harmonic Injection-Locked Clock Multipliers," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 12, pp. 3160–3174, Dec 2015.

[41] L. Kong and B. Razavi, "A 2.4 GHz 4 mW Integer-N Inductorless RF Synthesizer," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 3, pp. 626–635, March 2016.

[42] B. D. Muer and M. S. J. Steyaert, "A CMOS Monolithic ΔΣ-Controlled Fractional-N Frequency Synthesizer for DCS-1800," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 7, pp. 835–844, July 2002.

[43] W. Rhee, B. Song, and A. Ali, "A 1.1-GHz CMOS Fractional-N Frequency Synthesizer with a 3-b Third-order ΔΣ Modulator," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 10, pp. 1453–1460, Oct 2000.

[44] S. Pamarti, L. Jansson, and I. Galton, "A Wideband 2.4-GHz Delta-Sigma Fractional-N PLL With 1-Mb/s In-Loop Modulation," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 49–62, Jan 2004.

[45] E. Temporiti, G. Albasini, I. Bietti, R. Castello, and M. Colombo, "A 700-kHz Bandwidth ΣΔ Fractional Synthesizer with Spurs Compensation and Linearization Techniques for WCDMA Applications," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1446–1454, Sept 2004.

[46] C. Yao, W. F. Loke, R. Ni, Y. Han, H. Li, K. Godbole, Y. Zuo, S. Ko, N. Kim, S. Han, I. Jo, J. Lee, J. Han, D. Kwon, C. Kim, S. Kim, S. W. Son, and T. B. Cho, " A 14nm Fractional-N Digital PLL with 0.14ps rms jitter and −78dBc Fractional Spur for Cellular RFICs," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2017, pp. 422–423.

[47] X. Gao, O. Burg, H. Wang, W. Wu, C. Tu, K. Manetakis, F. Zhang, L. Tee, M. Yayla, S. Xiang, R. Tsang, and L. Lin, "A 2.7-to-4.3GHz, 0.16ps rms-jitter, −246.8dB-FOM, Digital Fractional-N Sampling PLL in 28nm CMOS," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan 2016, pp. 174–175.

[48] X. Gao, L. Tee, W. Wu, K. Lee, A. A. Paramanandam, A. Jha, N. Liu, E. Chan, and L. Lin, "A 28nm CMOS Digital Fractional-N PLL with −245.5dB FOM and a Frequency Tripler for 802.11abgn/ac Radio," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, Feb 2015, pp. 1–3.

[49] C. Hsu, M. Z. Straayer, and M. H. Perrott, "A Low-Noise Wide-BW 3.6-GHz Digital $\Delta\Sigma$ Fractional-N Frequency Synthesizer With a Noise-Shaping Time-to-Digital Converter and Quantization Noise Cancellation," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2776–2786, Dec 2008.

[50] T. Kuan and S. Liu, "A Digital Bang-Bang Phase-Locked Loop with Automatic Loop Gain Control and Loop Latency Reduction," in *2015 Symposium on VLSI Circuits (VLSI Circuits)*, June 2015, pp. C138–C139.

[51] D. Tasca, M. Zanuso, G. Marzin, S. Levantino, C. Samori, and A. L. Lacaita, "A 2.9-4.0-GHz Fractional-N Digital PLL With Bang-Bang Phase Detector and 560-fs_{rms} Integrated Jitter at 4.5-mW Power," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 12, pp. 2745–2758, Dec 2011.

[52] L. Bertulessi, L. Grimaldi, D. Cherniak, C. Samori, and S. Levantino, "A Low-Phase-Noise Digital Bang-Bang PLL with Fast Lock Over a Wide Lock Range," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb 2018, pp. 252–254.

[53] X. Gao, E. A. M. Klumperink, M. Bohsali, and B. Nauta, "A 2.2GHz 7.6mW Sub-sampling PLL with −126dBc/Hz In-band Phase Noise and 0.15ps rms Jitter in 0.18m CMOS," in *2009 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*, Feb 2009, pp. 392–393,393a.

[54] A. Swaminathan, K. J. Wang, and I. Galton, "A Wide-Bandwidth 2.4 GHz ISM Band Fractional-N PLL With Adaptive Phase Noise Cancellation," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 12, pp. 2639–2650, Dec 2007.

[55] D. Scherer, B. S. Chan, F. H. Ives, C. William, Jr., and D. W. Mathiesen, "Low-Noise RF Signal Generator Design," in *Hewlett Packard Journal*, vol. 32, no. 2, Feb 1981.

[56] X. Gao, E. A. M. Klumperink, M. Bohsali, and B. Nauta, "A Low Noise Sub-Sampling PLL in Which Divider Noise is Eliminated and PD/CP Noise is Not Multiplied by N^2," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 12, pp. 3253–3263, Dec 2009.

[57] A. Mirzaei, M. E. Heidari, R. Bagheri, S. Chehrazi, and A. A. Abidi, "The Quadrature LC Oscillator: A Complete Portrait Based on Injection Locking," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 9, pp. 1916–1932, Sept 2007.

[58] A. Mirzaei and H. Darabi, "Mutual Pulling Between Two Oscillators," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 2, pp. 360–372, Feb 2014.

[59] C. Yen, "Phase-Locked Sampling Instruments," *IEEE Transactions on Instrumentation and Measurement*, vol. IM-14, no. 1 and 2, pp. 64–68, March 1965.

[60] R. B. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P. T. Balsara, "1.3 V 20 ps Time-to-Digital Converter for Frequency Synthesis in 90-nm CMOS," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 3, pp. 220–224, March 2006.

[61] S. Levantino, D. Tasca, G. Marzin, M. Zanuso, C. Samori, and A. L. Lacaita, "A Wideband Fractional-N PLL with Suppressed Charge-pump Noise and Automatic Loop Filter Calibration," in *2012 IEEE Radio Frequency Integrated Circuits Symposium*, June 2012, pp. 177–180.

[62] S. E. Meninger and M. H. Perrott, "A 1-MHZ Bandwidth 3.6-GHz 0.18 μm CMOS Fractional-N Synthesizer Utilizing a Hybrid PFD/DAC Structure for Reduced Broadband Phase Noise," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 966–980, April 2006.

[63] M. H. Perrott, M. D. Trott, and C. G. Sodini, "A Modeling Approach for $\Sigma - \Delta$ Fractional-N Frequency Synthesizers Allowing Straightforward Noise Analysis," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 8, pp. 1028–1038, Aug 2002.

[64] C. Wang, H. Liao, C. Li, R. Huang, W. Wong, X. Zhang, and Y. Wang, "A Wideband Predictive "Double-π Equivalent-Circuit Model for On-Chip Spiral Inductors," *IEEE Transactions on Electron Devices*, vol. 56, no. 4, pp. 609–619, April 2009.

[65] G. Antonini, A. Orlandi, and C. R. Paul, "Internal Impedance of Conductors of Rectangular Cross Section," in *1998 IEEE EMC Symposium. International Symposium on Electromagnetic Compatibility. Symposium Record (Cat. No.98CH36253)*, vol. 1, Aug 1998, pp. 94–99 vol.1.

[66] T. Iizuka and A. A. Abidi, "FET-R-C Circuits: A Unified TreatmentPart II: Extension to Multi-Paths, Noise Figure, and Driving-Point Impedance," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 9, pp. 1337–1348, Sept 2016.

[67] A. A. Abidi and H. Xu, "Understanding the Regenerative Comparator Circuit," in *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*, Sept 2014, pp. 1–8.

[68] P. T. B. Robert Bogdan Staszewski, "All-Digital Frequency Synthesizer in Deep-Submicron CMOS," 2006.

[69] T. Iizuka, T. Ito, and A. A. Abidi, "Comprehensive Analysis of Distortion in the Passive FET Sample-and-Hold Circuit," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 4, pp. 1157–1173, April 2018.

[70] N. M. Neihart, D. J. Allstot, M. Miller, and P. Rakers, "Twisted Inductors for Low Coupling Mixed-Signal and RF Applications," in *2008 IEEE Custom Integrated Circuits Conference*, Sept 2008, pp. 575–578.

[71] Analog Devices Inc., "Microwave Wideband Synthesizer with Integrated VCO," *ADF4355-3 datasheet*, 2015.

[72] G. Li, L. Liu, Y. Tang, and E. Afshari, "A Low-Phase-Noise Wide-Tuning-Range Oscillator Based on Resonant Mode Switching," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 6, pp. 1295–1308, June 2012.

[73] R. L. Miller, "Fractional-Frequency Generators Utilizing Regenerative Modulation," *Proceedings of the IRE*, vol. 27, no. 7, pp. 446–457, July 1939.

[74] P. Maffezzoni and S. Levantino, "Phase Noise of Pulse Injection-Locked Oscillators," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 10, pp. 2912–2919, Oct 2014.