UNIVERSITY OF CALIFORNIA, SAN DIEGO

Spur Reduction Techniques for Fractional-N PLLs

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

in

Electrical Engineering (Electronic Circuits and Systems)

by

Kevin Jia-Nong Wang

Committee in charge:

Professor Ian Galton, Chair
Professor James Buckwalter
Professor William S. Hodgkiss
Professor Lawrence E. Larson
Professor Thomas T. Liu

2010

The dissertation of Kevin Jia-Nong Wang is approved, and it is acceptable in quality and form for publication on micro-film and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2010

DEDICATION

To my family and friends

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Professor Ian Galton for his many years of encouragement and guidance. His enthusiasm and faith in this work even during the most difficult times led to the results presented here.

I would also like to acknowledge and thank all those who shared the lab with me during my stay. I will miss the serious discussions, the not-so-serious discussions, and the many breaks that we had.

Finally, I want to thank my family and Xixiao for their unending support and patience.

Chapter 1, in full, is a reprint of the material as it appears in the Journal of Solid State Circuits, vol. 43, pp. 2787-2797, Dec. 2008. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis for this paper.

Chapter 2, in full, has been submitted for review to the IEEE Transactions on Circuits and Systems I: Regular papers. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis of this paper.

VITA

1995	Bachelor of Science, Cornell University
1995 – 1996	Teaching Assistant, University of California, Berkeley
1996 – 1998	Research Assistant, University of California, Berkeley
1998	Master of Science, University of California, Berkeley
1998 – 2003	Member of Staff, Silicon Wave
2003 – 2010	Research Assistant, University of California, San Diego
2010	Doctor of Philosophy, University of California, San Diego

ABSTRACT OF THE DISSERTATION

Spur Reduction Techniques for Fractional-N PLLs

by

Kevin Jia-Nong Wang

Doctor of Philosophy in Electrical Engineering (Electronic Circuits and Systems)

University of California, San Diego, 2010

Professor Ian Galton, Chair

Fractional-*N* phase locked loops (PLL) are widely used in modern communication systems to synthesize a highly pure frequency from a lower reference frequency. Stringent requirements are often placed on the spectral purity of the synthesized frequency so that overall system-level requirements are met. Unfortunately, spurious tones are inevitable in the output signals of fractional-*N* PLLs, and in conventional designs they can be attenuated only with design tradeoffs that degrade other aspects

of performance.

This dissertation presents a PLL that utilizes a successive requantizer in conjunction with an offset current technique to suppress both the fractional and reference spurs. A passive, type-II, sampled loop filter (SLF) is also introduced to mitigate the increased reference spurs that result from the use of an offset current.

Chapter 1 describes a phase-noise cancelling, fractional-*N* PLL utilizing the techniques mentioned above. It details the system level and circuit level design and presents measured results.

Chapter 2 presents a discrete-time model for a PLL utilizing the passive SLF described in Chapter 1. A mathematical basis for the model is also presented.

Chapter 3 describes an integer-*N*, realigning PLL utilizing a relaxation oscillator and a calibration scheme to suppress the realignment spur. Realignment can suppress the noise of the voltage controlled oscillator (VCO) and hence finds application in systems utilizing non-LC based VCOs. However, implementation challenges exist with regard to the VCO and many previously published designs suffer from a large realignment spur. The use of a relaxation oscillator and the calibration scheme address these two challenges.

Chapter I:

Spurious Tone Suppression Techniques Applied to a Wide-Bandwidth 2.4GHz Fractional-*N* PLL

Abstract—This paper demonstrates that spurious tones in the output of a fractional-*N* PLL can be reduced by replacing the ΔΣ modulator with a new type of digital quantizer and adding a charge pump offset combined with a sampled loop filter. It describes the underlying mechanisms of the spurious tones, proposes techniques that mitigate the effects of the mechanisms, and presents a phase noise cancelling 2.4GHz ISM-band CMOS PLL that demonstrates the techniques. The PLL has a 975kHz loop bandwidth and a 12MHz reference. Its phase noise has a worst-case reference spur power of −70dBc and a worst-case in-band fractional spur power of −64dBc.

I. Introduction

Most wireless communication systems require local oscillators for up-conversion and down-conversion of their transmitted and received signals. Usually, the spectral purity of the local oscillator is a critical factor in overall transceiver performance, so communication standards explicitly or implicitly stipulate stringent spectral purity requirements on the local oscillators [1-2]. In addition to dictating the maximum acceptable phase noise power in various frequency bands, most standards require that spurious tones in the local oscillator's output be highly attenuated, particularly in critical frequency bands.

Local oscillators in such applications are often implemented as fractional-*N*

phase-locked loops (PLLs). Unfortunately, spurious tones are inevitable in the output signals of fractional-*N* PLLs, and in conventional designs they can be attenuated only with design tradeoffs that degrade other aspects of performance. Generally, spurious tone power can be reduced by increasing the linearity of key circuit blocks such as the charge pump and divider, restricting the choice of reference frequencies, and reducing the loop bandwidth. Unfortunately, increasing linearity tends to increase power consumption and circuit area, restricting the choice of reference frequencies reduces design flexibility, and reducing the loop bandwidth increases in-band phase noise, settling time, susceptibility to oscillator pulling, and loop filter size [1]. Furthermore, these methods of spurious tone reduction become less effective as CMOS circuit technology is scaled into the sub-100 nanometer regime. Therefore, the spurious tone problem negatively affects power consumption, cost, and manufacturability of wireless transceivers, and the problem gets worse as CMOS circuit technology scales with Moore's Law.

This paper presents a 2.4 GHz ISM band fractional-*N* PLL that achieves state-of-the-art spurious tone suppression enabled by techniques that avoid the tradeoffs mentioned above [3]. One of the techniques is the use of a new type of digital quantizer, called a *successive requantizer*, in place of the digital delta-sigma ($\Delta\Sigma$) modulator used in conventional fractional-*N* PLLs [4]. The other technique involves the combination of a charge pump offset and a sampled loop filter.

The paper consists of four main sections. Section II describes the mechanisms by which the two types of spurious tones, reference spurs and fractional spurs, arise in fractional-*N* PLLs. Section III describes the successive requantizer. Section IV describes the charge pump offset and sampled loop filter. Section V presents additional circuit details

and measurement results.

II. Spurious Tones and Their Causes in Fractional-*N* PLLs

A. Fractional-*N* PLL Overview

The purpose of a fractional-*N* PLL is to generate a periodic output signal with frequency $f_{out} = (N + \alpha)f_{ref}$, where N is an integer, α is a fractional value between 0 and 1, and f_{ref} is the frequency of a reference oscillator (e.g., the crystal frequency). As shown in Figure 1, a typical fractional-*N* PLL consists of a phase-frequency detector (PFD), a charge pump, a loop filter, a voltage controlled oscillator (VCO), a frequency divider, and a digital ΔΣ modulator clocked by the divider output [5-7]. The divider output is a two-level signal in which the nth and $(n+1)$th rising edges are separated by $N + y[n]$ periods of the VCO output, where $y[n]$ is an integer-valued sequence from the ΔΣ modulator. As indicated in the figure for the case where the PLL is locked, if the nth rising edge of the reference signal, $v_{ref}(t)$, occurs before that of divider output, $v_{div}(t)$, the charge pump generates a current pulse of nominal amplitude I and a duration equal to the time difference between the two edges. Otherwise, the situation is similar except the polarity of the current pulse is reversed. The PLL's feedback adjusts the output frequency so as to zero the DC component of the charge pump output. This causes the output frequency to settle to f_{ref} times the sum of N and the average of $y[n]$.

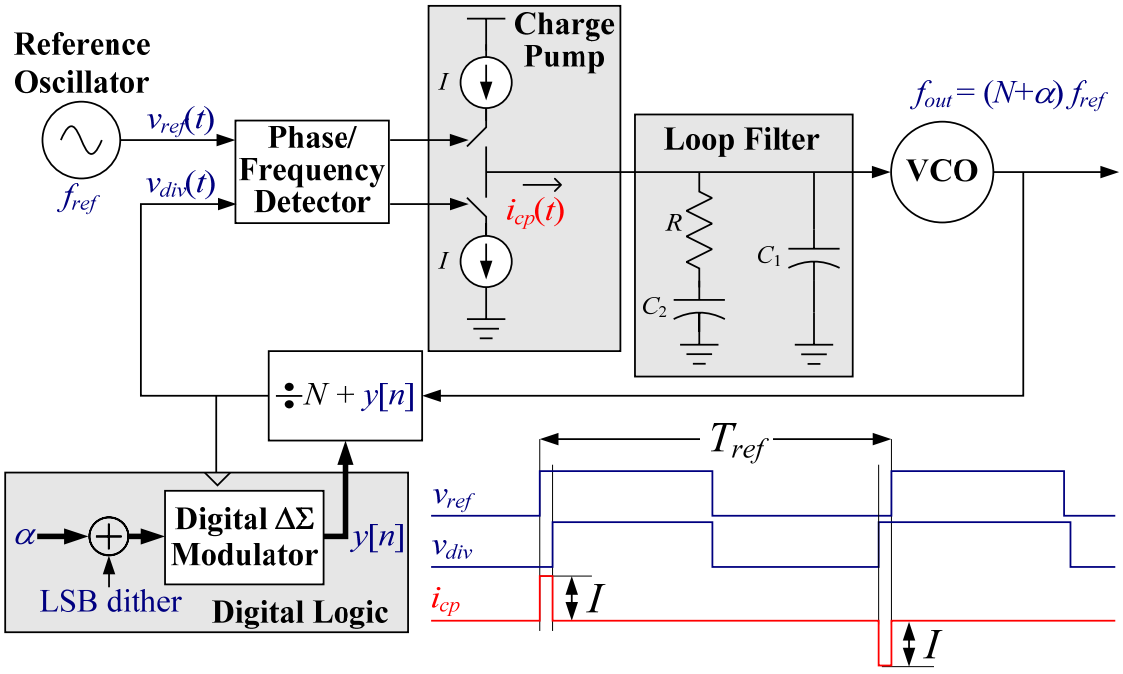


Fig. 1: Block diagram of a typical fractional-N PLL.

If $y[n]$ could be set to α directly, then the output frequency of the PLL would settle to $(N + \alpha)f_{ref}$, thereby achieving the goal of the fractional-N PLL. Unfortunately, this is not possible. The divider can only count integer VCO cycles so $y[n]$ is restricted to integer values whereas α is a fractional value. To circumvent this problem $y[n]$ is designed to be a sequence of integers that *average* to α. The input to the ΔΣ modulator is α plus pseudo-random least significant bit (LSB) dither, so its output has the form $y[n] = \alpha + s[n]$, where $s[n]$ is a zero-mean sequence consisting of spectrally shaped ΔΣ quantization noise and LSB dither. As proven in [8], the dither prevents $s[n]$ from containing spurious tones that would otherwise show up as spurious tones in the PLL's output. Hence, the output frequency settles to an average $(N + \alpha)f_{ref}$, as desired, although $s[n]$ introduces phase noise.

The $s[n]$ sequence causes an amount of charge equal to $T_{VCO} \cdot I \cdot t[n]$ to be added to

the nth charge pump pulse, where T_{VCO} is the period of the VCO output (for a given value of α, T_{VCO} is well-modeled as a constant) and

$$t[n] = \sum_{k=0}^{n} s[k] \tag{1}$$

is the *running sum* of $s[n]$. Hence, the PLL's phase noise contains a lowpass filtered version of $t[n]$. The bandwidth of the lowpass filtering operation is called the *loop bandwidth* of the PLL. Usually, the quantization noise transfer function of the ΔΣ modulator is highpass shaped with at least one zero at DC. Therefore, $t[n]$ is bounded and shaped with an order of one less than that of the ΔΣ modulator's quantization noise transfer function. Provided the loop bandwidth is sufficiently low, the resulting phase noise is suppressed below that from other noise sources in the PLL. Alternatively, a DAC can be used to cancel $t[n]$ prior to the loop filter, thereby minimizing its contribution to the PLL's phase noise so that a much larger loop bandwidth can be used [9-13]. Such fractional-N PLL's are called *phase noise cancelling* fractional-N PLLs.

B. Reference Spurs

Reference spurs are spurious tones in the PLL's output that occur at multiples of f_{ref} from f_{out}. They result mainly from periodic disturbances of the loop filter voltage introduced by the charge pump. Therefore, the loop bandwidth and the reference frequency both affect the power of the reference spurs. Widening the loop bandwidth for a given reference frequency or decreasing the reference frequency for a given loop bandwidth both have the effect of reducing the loop filter's attenuation of the disturbances, thereby increasing the power of the reference spurs.

Mismatches between the positive and negative current sources in the charge pump are the primary causes of the disturbances that cause reference spurs. A typical PFD turns on both current sources in the charge pump each reference period for a minimum duration, T_D, where T_D is large enough to ensure that both current sources fully settle before they are turned off. Each reference period the PFD turns on the positive current source when the reference edge occurs and the negative current source when the divider edge occurs, and turns them both off simultaneously T_D seconds after the later of the two edges. The difference between the positive and negative current pulses is the charge pump output current pulse. By ensuring that both current sources have time to settle, a major source of charge pump nonlinearity is avoided [14]. However, inevitable transient and amplitude mismatches between the two current sources give rise to an error component in each charge pump pulse that is constant from period to period. Although the PLL's feedback nulls out the DC component of constant error pulse by adjusting the phase of the VCO, the result is a zero-mean periodic disturbance of the VCO's control voltage which causes a reference spur.

In theory, the disturbance and, therefore, the reference spur could be eliminated by performing an ideal sample-and-hold operation between the loop filter and the VCO once per reference period. The sampled loop filter presented in Section IV provides a practical means of achieving this result to a high degree of accuracy.

C. Fractional Spurs

Fractional spurs are spurious tones in the PLL's output that occur at multiples of

αf_{ref} from f_{out}.[†] Typically, the most significant fractional spurs are the result of disturbances on the loop filter voltage introduced through the charge pump. Therefore, the power of a fractional spur usually depends on both its frequency and the loop bandwidth. In conventional fractional-*N* PLLs, fractional spurs within the loop bandwidth tend to be large, typically well above −60dBc, while fractional spurs at higher frequencies usually are attenuated by the loop filter. Hence, the power of the fractional spur at αf_{ref} can be reduced by reducing the loop bandwidth for any given values of f_{ref} and α. In conventional fractional-*N* PLLs the application's spurious tone suppression requirements typically dictate restrictions on the choice of reference frequency and loop bandwidth so as to ensure that αf_{ref} is sufficiently outside the loop bandwidth for every desired output frequency.

As described in the remainder of this section, fractional spurs arise from two distinct mechanisms. The techniques presented in Sections III and IV respectively address each mechanism to reduce the power of the fractional spurs.

Fractional Spur Mechanism 1

It is well known that nonlinear parasitic coupling between the VCO output signal and harmonics of the reference signal result in fractional spurs. For example, if the *N*th harmonic of the reference signal intermodulates with the VCO output signal through a parasitic coupling path in the circuit, the intermodulation product is a spurious tone at αf_{ref}.

[†] A fractional spur in the the PLL output at a frequency of $f_{out}+f_{spur}$ is often said to *occur at frequency* f_{spur} because it appears at frequency f_{spur} in a phase noise plot. This terminology is used in the remainder of the paper.

The potential for such coupling is greatest in the PFD and charge pump, as these blocks handle signals aligned with the reference signal as well as those aligned with the VCO output [10]. The hard-switching that occurs within these blocks induces disturbances on the local power supply lines because of the bond wire inductance. This modulates the switching threshold of the digital gates powered by these supplies. As illustrated in Figure 2, the two flip-flops in the PFD capture the phase difference between the divider and reference edges. For small phase differences, the disturbance induced by the earlier edge does not have time to die out before the later edge arrives, so it can modulate the delay through the flip-flop of the later edge, thereby corrupting the phase difference measurement. The resulting error contains intermodulation products of the VCO output and reference signal which are injected into the loop filter and cause fractional spurs. Similar coupling effects occur within the charge pump circuitry.

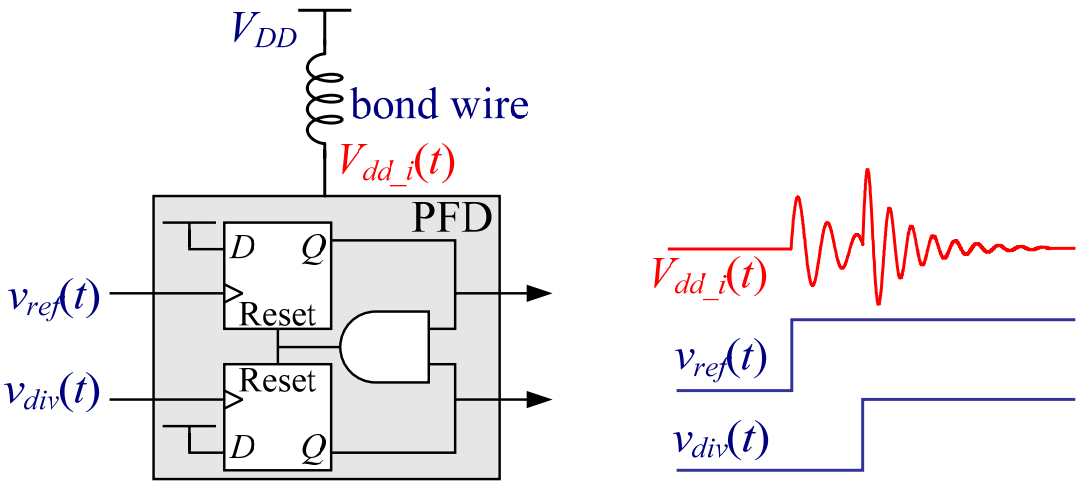


Fig. 2: Example of a nonlinear coupling path in the PFD

Fractional Spur Mechanism 2

Surprisingly, the digital ΔΣ modulator in a fractional-*N* PLL is a fundamental source of spurious tones in the PLL's output [3, 4, 9, 15]. This is true even though dither is used to prevent spurious tones in the ΔΣ modulator's output. Regardless of how dither is applied, spurious tones are induced when the ΔΣ modulator's quantization noise is subjected to nonlinear distortion. This is particularly problematic in fractional-*N* PLLs wherein the output sequence from the ΔΣ modulator is converted to analog form and both $s[n]$ and its running sum, $t[n]$, are subjected to nonlinear operations because of non-ideal circuit behavior.

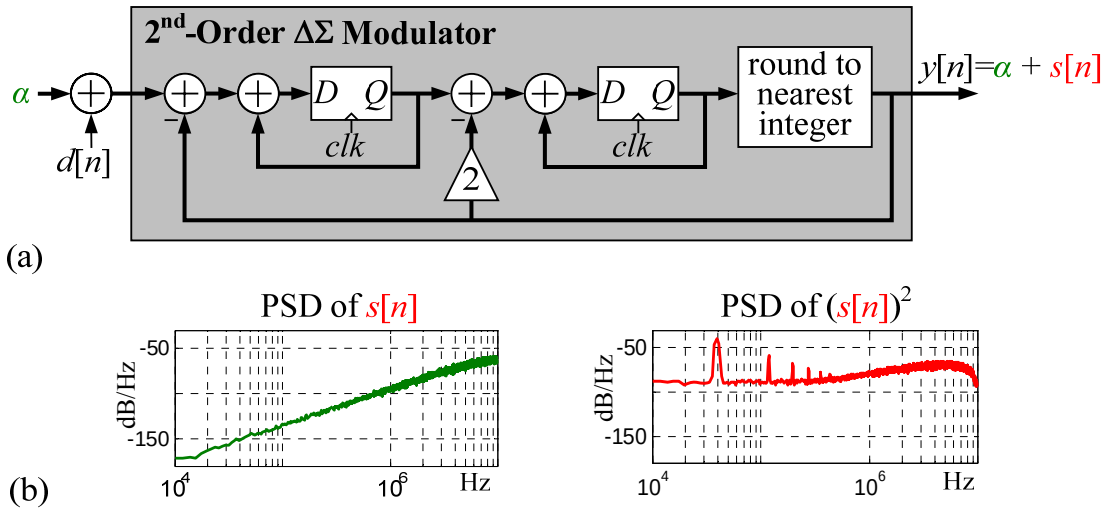


Fig. 3: (a) A second-order digital ΔΣ modulator, and (b) an example in which $s[n]$ is free of spurious tones but a nonlinearly distorted version of $s[n]$ contains spurious tones.

A digital ΔΣ modulator often used in fractional-*N* PLLs is shown in Figure 3a as a demonstration vehicle. It is an all-digital structure consisting of two accumulators, a

round-to-the-nearest-integer quantizer, and two negative feedback paths. It is well known that if the ΔΣ modulator input is kept between 0 and 1, then the output is restricted to the integers: {−1, 0, 1, 2}, and $s[n] = d[n-2] + e_q[n] - 2e_q[n-1] + e_q[n-2]$, where $e_q[n]$ is additive error from the round-to-the-nearest-integer operation of the quantizer. Therefore, $e_q[n]$ is subjected to the equivalent of a three tap FIR filter with a pair of zero-frequency zeros.

As shown in [8], if the dither sequence, $d[n]$, is an equiprobable two-level, white, random sequence of any non-zero magnitude, then $e_q[n]$ is guaranteed to be asymptotically white and zero mean. In this case, $e_q[n]$, and, hence, $s[n]$, are guaranteed to be free of spurious tones. Moreover, the three tap FIR filtering causes the power spectral density (PSD) of the quantization noise component of $s[n]$ to increase at 12 dB per octave in frequency. For example, the simulated PSD of $s[n]$ is shown in the left plot in Figure 3b for the case where $\alpha = 0.002$, $d[n]$ is a white pseudo-random sequence that takes on values of 0 and 2^{-17} with equal probability, and the sample rate is 20 MHz. In this case, the magnitude of the dither is sufficiently small that it is not visible in the PSD plot, yet its presence ensures that spurious tones are avoided in $s[n]$. However, as shown in the right plot of Figure 3b, spurious tones are clearly present in the PSD of $(s[n])^2$. Similar results occur for other types of nonlinear distortion and all other ΔΣ modulators and dither methods known to the authors. For example, the problem occurs even if the dither sequence is white with a triangular probability density function that extends from −1 to 1 and is added directly to the input of the quantizer.

If it seems counter-intuitive that spurious tones can occur when a spur-free sequence is subjected to nonlinear distortion, consider a random sequence given by

$$q[n] = \begin{cases} \pm 1 \text{ (chosen randomly)}, & \text{if } n \text{ is even}, \\ 0, & \text{if } n \text{ is odd}. \end{cases} \qquad (2)$$

It is easy to verify that $q[n]$ is white and, hence, free of spurious tones. However, $(q[n])^2$ is 1 for even values of n and 0 for odd values of n, so $(q[n])^2$ is nothing but a spurious tone at half the sample rate and a constant offset. In this simple case, $q[n]$ has "sufficient randomness" to avoid spurious tones in the absence of nonlinear distortion but not when subjected to even-order nonlinear distortion.

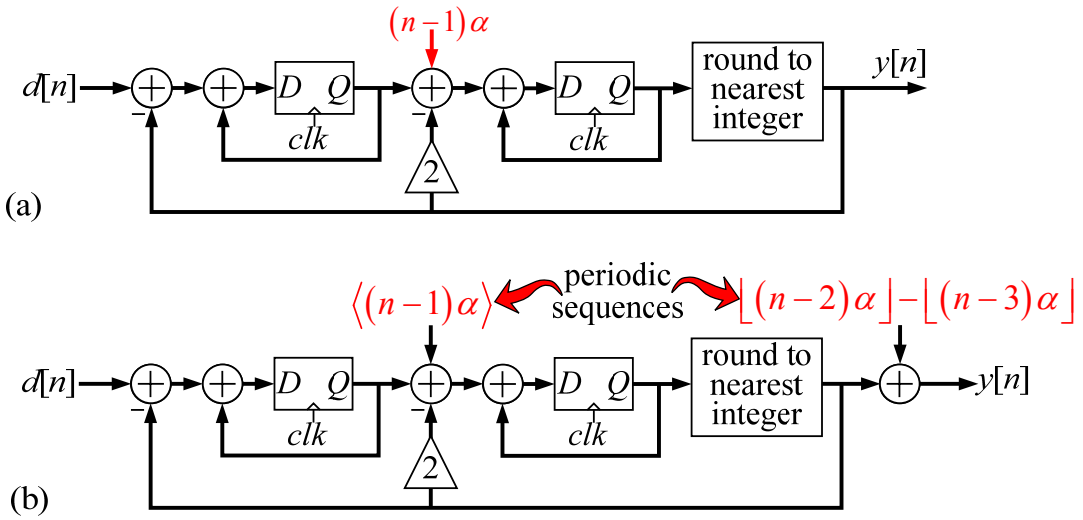


Fig. 4: Structures that are both equivalent to that of Figure 3a.

The situation is conceptually similar, but more complicated, for the case of a ΔΣ modulator. The interaction of the constant input and the first accumulator gives rise to "hidden periodicities" as indicated in Figure 4. Both structures in Figure 4 are equivalent to that of Figure 3a in that they generate the same $y[n]$ sequence. The structure of Figure 4a differs from that of Figure 3a in that in Figure 4a the α input has been replaced by its

delayed running sum, $(n-1)\alpha$, added after the first accumulator. This sequence can be written as

$$(n-1)\alpha = \lfloor (n-1)\alpha \rfloor + \langle (n-1)\alpha \rangle \tag{3}$$

where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x, and $\langle x \rangle$ denotes the fractional part of x. The round-to-the-nearest-integer quantizer has no effect on integer-valued components of its input, and the transfer function from the input of the second accumulator to the output of the ΔΣ modulator is $z^{-1}(1-z^{-1})$ so the integer-valued component of (3) can be moved after the feedback loops as shown in Figure 4b. The significance is that both additive sequences in Figure 4b associated with α are periodic with a period that depends on α, so they are each made up entirely of spurious tones (i.e., their Fourier series components). The dither provides sufficient randomness to avoid spurious tones in $s[n]$ as proven in [8], but not to avoid spurious tones when $s[n]$ is subjected to nonlinear distortion as demonstrated in Figure 3b.

III. A Delta-Sigma Modulator Replacement

The fractional-N PLL presented in this paper uses a successive requantizer in place of a ΔΣ modulator to circumvent fractional spur mechanism 2 [4]. The successive requantizer performs coarse quantization with spectrally shaped quantization noise like a ΔΣ modulator, but its quantization noise is less susceptible to nonlinearity-induced spurious tones as described below.

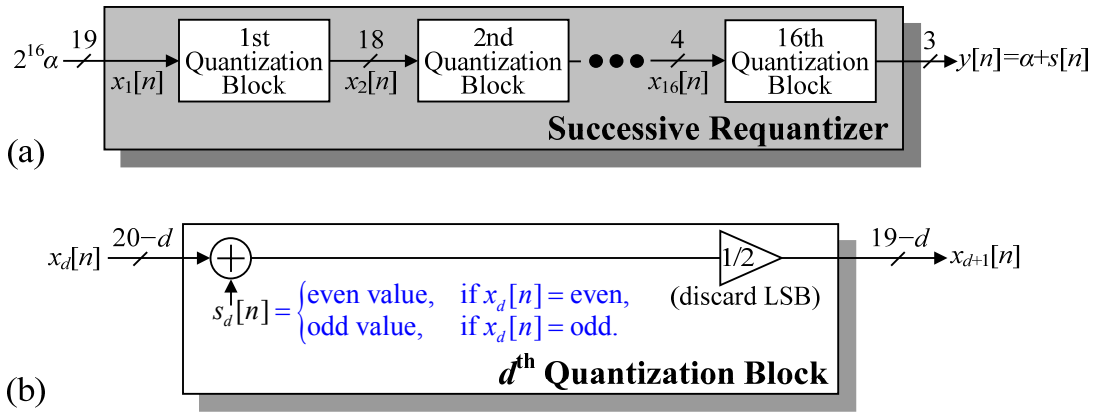


Fig. 5: High-level diagram of an example successive requantizer.

A high-level view of successive requantizer is shown in Figure 5. It quantizes a 19-bit input sequence by 16 bits to generate a 3-bit output sequence [3-4]. By design convention, the input and output of the successive requantizer are integer-valued. For the fractional-*N* PLL application, the goal is to quantize α, which is a fractional value between 0 and 1, and in this design α is taken to be a constant multiple of 2^{-16}. Therefore, α is scaled by 2^{16} prior to the successive requantizer to convert it into an integer. As explained below, the 3-bit integer-valued output of the successive requantizer is $y[n]=\alpha+s[n]$, where $s[n]$ is quantization noise.

As shown in Figure 5a the successive requantizer consists of 16 *quantization blocks*, each of which simultaneously halves its input and quantizes the result by one bit every sample period. The general form of each quantization block is shown in Figure 5b wherein all variables are integer-valued two's complement numbers. The output of the dth quantization block is $x_{d+1}[n]=(x_d[n]+s_d[n])/2$, where $s_d[n]$ a sequence generated within the quantization block. At each time n, $s_d[n]$ is chosen such that $x_d[n]+s_d[n]$ does not ex-

ceed the range of a (20−d)-bit two's complement integer, and the parity of $s_d[n]$ is the same as that of $x_d[n]$. The parity restriction ensures that $x_d[n]$+$s_d[n]$ is an even number so its LSB is zero. Discarding the LSB simultaneously halves the quantization block's input value and quantizes the result by one bit. The resulting quantization noise is $s_d[n]/2$, so the successive requantizer's overall quantization noise is

$$s[n] = \sum_{d=1}^{16} 2^{d-17} s_d[n]. \tag{4}$$

Therefore, $s[n]$ is a linear combination of the $s_d[n]$ sequences, so it inherits the properties of the $s_d[n]$ sequences.

A key feature of the successive requantizer is that the properties of its quantization noise can be engineered by appropriate design of the $s_d[n]$ sequences. So far, the only restriction on the $s_d[n]$ sequences is that they must be chosen such that $x_d[n]$+$s_d[n]$ is a (20−d)-bit two's complement even integer for each n and d. This leaves considerable flexibility in the design of the $s_d[n]$ sequences which is exploited to achieve the desired quantization noise properties.

The successive requantizer partially exploits this flexibility to ensure that the running sum of each $s_d[n]$ sequence, i.e.,

$$t_d[n] = \sum_{k=0}^{n} s_d[k], \tag{5}$$

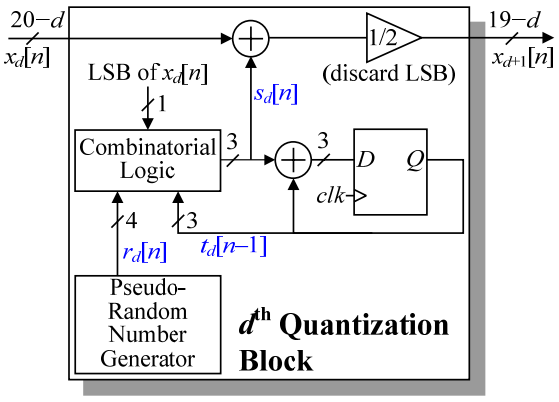
is bounded for all n, and each $s_d[n]$ has a smooth PSD that increases monotonically with frequency. This implies that $s[n]$ is highpass shaped quantization noise that is free of spurious tones and the PSD of $s[n]$ is zero at $\omega = 0$.

This still leaves flexibility in the design of the $s_d[n]$ sequences which is exploited

as described below to ensure that the sequences

$$\left(s[n]\right)^p \quad \text{for} \quad p = 1,2,3,4,5, \qquad \text{and} \qquad \left(t[n]\right)^p \quad \text{for} \quad p = 1,2,3, \tag{6}$$

are free of spurious tones, where $t[n]$ is the running sum of $s[n]$ given by (1). The objective is to ensure that the successive requantizer's quantization noise does not introduce significant spurious tones when subjected to the degree of nonlinear distortion expected from the analog circuits within the PLL. Circuit simulations were used during the PLL's design to verify that preventing spurious tones from occurring in the sequences given by (6) is sufficient to achieve this objective.



Combinatorial Logic Truth Table:

LSB of $x_d[n] = 0$			LSB of $x_d[n] = 1$		
$t_d[n-1]$	$r_d[n]$	$s_d[n]$	$t_d[n-1]$	$r_d[n]$	$s_d[n]$
2	≥ 0 and ≤ 3	0	2	≤ −1 or ≥ 4	−1
2	≤ −1 or ≥ 4	−2	2	≥ 0 and ≤ 3	−3
1	≤ −1 or ≥ 6	0	1	≥ 1 and ≤ 3	1
1	≥ 0 and ≤ 5	−2	1	≤ −1 or ≥ 4	−1
0	0 or 1	2	1	0	−3
0	≤ −1 or ≥ 4	0	0	≥ 0	1
0	2 or 3	−2	0	≤ −1	−1
−1	≤ −1 or ≥ 6	0	−1	≥ 1 and ≤ 3	−1
−1	≥ 0 and ≤ 5	2	−1	≤ −1 or ≥ 4	1
−2	≥ 0 and ≤ 3	0	−1	0	3
−2	≤ −1 or ≥ 4	2	−2	≤ −1 or ≥ 4	1
			−2	≥ 0 and ≤ 3	3

Fig. 6: Implementation of each quantization block for a successive requantizer with $s^p[n]$, p = 1, 2, 3, 4, 5, and $t^p[n]$, p = 1, 2, 3, that are free of spurious tones.

The register transfer level details of the dth quantization block are shown in Figure 6. Each value of $s_d[n]$ is calculated via the combinatorial logic shown in the figure as a function the previous value of $t_d[n]$, the parity of the current value of $x_d[n]$, and the current value of a 4 bit pseudo-random sequence, $r_d[n]$, where $\{r_d[n], d = 1, 2, \ldots, 16, n = 0, 1, 2, \ldots\}$ well-approximate independent identically distributed random variables. For this

design the range of values taken on by $s_d[n]$ and $t_d[n]$ are

$$s_d[n] \in \{-3, -2, -1, 0, 1, 2, 3\}, \qquad \text{and} \qquad t_d[n] \in \{-2, -1, 0, 1, 2\}, \tag{7}$$

It can be verified that $t_d[n]$ is a discrete-valued Markov random sequence conditioned on the parity of $x_d[n]$. Whenever $x_d[n]$ is odd the one-step state transition matrix for $t_d[n]$ is given by

$$\mathbf{A_o} = \left[P\{t_d[n] = T_j \mid t_d[n-1] = T_i, o_d[n] = 1\} \right]_{5\times 5} \tag{8}$$

and whenever $x_d[n]$ is even the one-step state transition matrix for $t_d[n]$ is given by

$$\mathbf{A_e} = \left[P\{t_d[n] = T_j \mid t_d[n-1] = T_i, o_d[n] = 0\} \right]_{5\times 5} \tag{9}$$

where $P\{X \mid Y\}$ denotes the conditional probability of event X given event Y, $o_d[n]$ is the LSB of $x_d[n]$, and $T_1 = -2$, $T_2 = -1$, $T_3 = 0$, $T_4 = 1$, $T_5 = 2$. The specific state transition matrices corresponding to the quantization block shown in Figure 5 are

$$\mathbf{A_o} = \begin{bmatrix} 0 & 3/4 & 0 & 1/4 & 0 \\ 3/16 & 0 & 3/4 & 0 & 1/16 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 1/16 & 0 & 3/4 & 0 & 3/16 \\ 0 & 1/4 & 0 & 3/4 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{A_e} = \begin{bmatrix} 1/4 & 0 & 3/4 & 0 & 0 \\ 0 & 5/8 & 0 & 3/8 & 0 \\ 1/8 & 0 & 3/4 & 0 & 1/8 \\ 0 & 3/8 & 0 & 5/8 & 0 \\ 0 & 0 & 3/4 & 0 & 1/4 \end{bmatrix}. \tag{10}$$

As derived in [4], these state transition matrices ensure that the sequences in (6) are free of spurious tones because each is a random process whose autocorrelation function converges to a constant as its time spread increases. Furthermore, the PSD of $s_d[n]$ has a zero at $\omega = 0$ and increases at 6 dB per octave as ω increases from zero. In this respect, the quantization noise shaping of this version of the successive requantizer is comparable to that of a first-order $\Delta\Sigma$ modulator.

Successive requantizers with higher than first-order quantization noise shaping

can also be designed. For example, second-order quantization noise shaping can be achieved by quantization blocks that calculate $s_d[n]$ as a function the running sum of $t_d[n]$ in addition to $t_d[n]$, a random sequence, and the parity of $x_d[n]$. However, the fractional-N PLL in this work is a phase noise cancelling fractional-N PLL, so higher than first-order shaping is not necessary because most of the quantization noise is removed prior to the loop filter via a DAC.

A drawback of the quantization block shown in Figure 6 is that its reduced susceptibility to nonlinearity-induced spurious tones comes at the expense of increased quantization noise power. For example, if it is desired to have quantization noise with a first-order highpass spectral shape, but it is not necessary to prevent nonlinear distortion from inducing spurious tones in the quantization noise and its running sum, a quantization block that implements

$$s_d[n] = \begin{cases} 0, & \text{if } x_d[n] = \text{even} \\ p_d[n], & \text{if } x_d[n] = \text{odd and } t_d[n-1] = 0, \\ 1, & \text{if } x_d[n] = \text{odd and } t_d[n-1] = -1, \\ -1, & \text{if } x_d[n] = \text{odd and } t_d[n-1] = 1, \end{cases} \tag{11}$$

can be used, where $p_d[n]$ is an independent random sequence that takes on the values 1 and –1 with equal probability. In this case $s_d[n]$ takes on values of –1, 0, and 1, whereas the $s_d[n]$ generated by the quantization block of Figure 6 takes on values of –3, –2, –1,…, 3. Consequently, the power of the quantization noise from a quantization block based on (11) is significantly lower than that from the quantization block of Figure 6.

This example suggests what is likely to be a fundamental tradeoff: reduced susceptibility to nonlinearity-induced spurious tones comes at the expense of increased quantization noise power. The tradeoff has yet to be proven theoretically, but it is exhi-

bited by all variants of the successive requantizer developed to date by the authors. In each case, generating $s_d[n]$ sequences with reduced susceptibility to nonlinearity-induced spurious tones has required choices to be made that increase the power of the $s_d[n]$ sequences. This is not a significant problem in phase noise cancelling fractional-*N* PLLs, but it is likely to be an issue in fractional-*N* PLLs without phase noise cancellation. Analytical quantification of the tradeoff and its effect on the performance of fractional-*N* PLLs without phase noise cancellation are ongoing subjects of research.

IV. A Charge Pump Offset and Sampled Loop Filter

The fractional-*N* PLL presented in this paper injects a constant current pulse into the loop filter each reference period as a means of mitigating fractional spur mechanism 1 [10]. As shown in Figure 7, an offset pulse generator in parallel with the charge pump introduces a positive current pulse of amplitude *I* starting from the rising edge of the divider output and extending for 8 VCO periods. The offset current pulses cause a fixed VCO phase shift such that in each reference period the divider edge always occurs at least 6 VCO periods prior to the reference edge. Separating the edges in this fashion gives the power supply disturbance described in Section III time to die out between the edges, thereby alleviating the coupling problem.

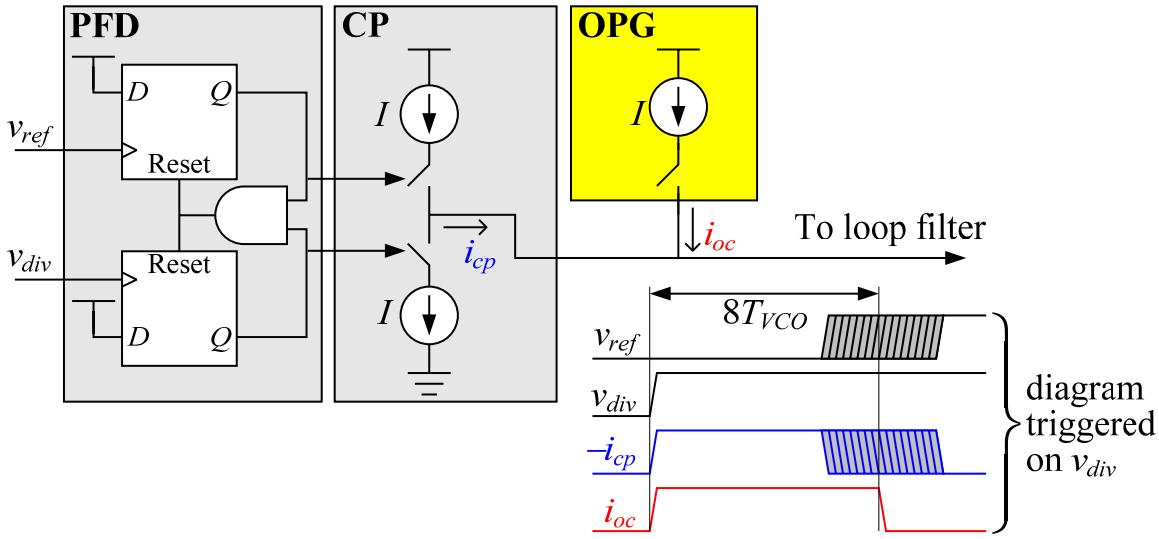


Fig. 7: The phase-frequency detector, charge pump, offset pulse generator and the associated timing diagram.

Unfortunately, the offset current pulse technique has a severe side-effect if used with a conventional loop filter: transient and amplitude mismatches between the current source in the offset pulse generator and the negative current source in the charge pump add significant power to the reference spur. The effect is more severe than that caused by mismatches between the positive and negative charge pump current sources in a conventional configuration because of the increased duration of the pulses.

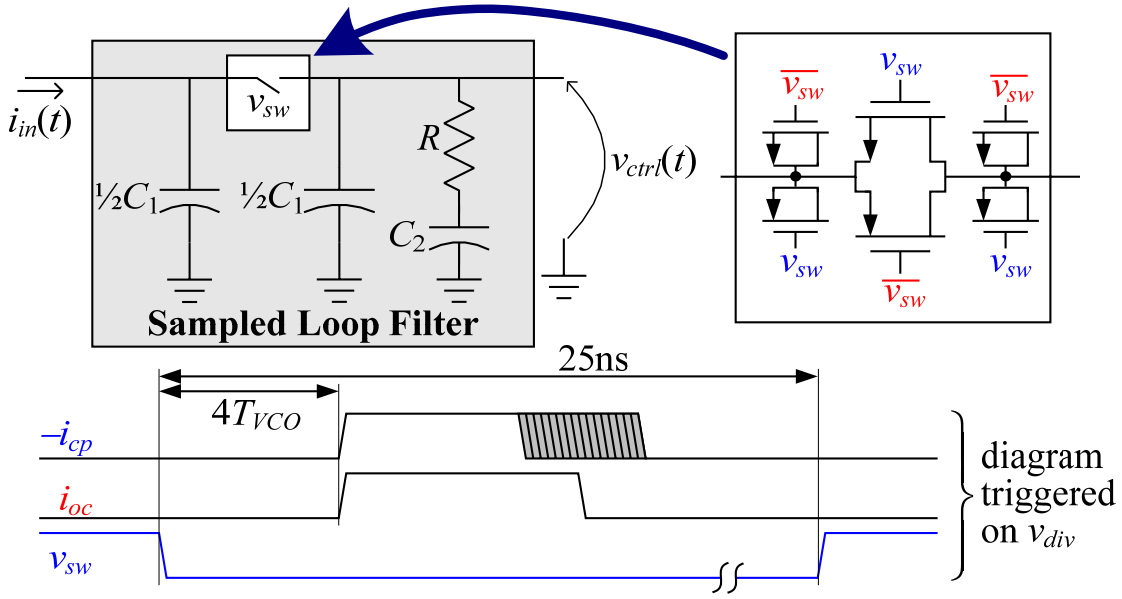


Fig. 8: The sampled loop filter and the associated timing diagram.

The side-effect is avoided in this work by the sampled loop filter shown in Figure 8. It differs from the conventional loop filter shown in Figure 1 only in that the C_1 capacitor has been split into two parallel half-sized capacitors separated by a CMOS transmission gate switch. Thus, it reduces to a conventional loop filter when the switch is closed. As indicated in Figure 8, the switch is opened once per reference period for a duration of approximately 25 ns starting 4 VCO periods prior to the rising edge of the divider. This ensures that it is open whenever the loop filter's input current is non-zero. Once the PLL has settled, the voltage across the switch just before it closes each reference period depends only on circuit noise and quantization noise from the successive requantizer. Therefore, to the extent that the switch is ideal, closing the switch each period does not inject periodic disturbances at the reference frequency so reference spurs are avoided. As with other sampled loop filter designs, this design also eliminates reference spurs caused

by mismatches between the current sources in the charge pump [11,16].

The switch is implemented as a transmission gate with half size dummy transmission gates on either side as shown in Figure 8. The dummy transmission gates are shorted and driven in opposite polarity to the main transmission gate. Their purpose is to cancel charge injection from the main transmission gate that would otherwise cause a reference spur.

One way to ensure precise cancellation of the charge injection in such a switch configuration is to design the loop filter and surrounding circuitry so the impedances from the two switch terminals to ground are equal. This could have been achieved by placing a series resistance of $2R$ and capacitance of $C_2/2$ from each side of the switch to ground instead of the series resistance of R and capacitance of C_2 on just the right side of the switch as shown in Figure 8. However, doing so would have prevented the voltage on the left side of the switch from settling to a constant each reference period prior to closing the switch, thereby negating the reference frequency suppression property of the sampling process.

Fortunately, the charge injection is well cancelled despite the asymmetry from the series combination of R and C_2. The edges of the signals that control the transmission gates are sharp, so the charge injected by each MOS transistor is in the form of short-duration, and, hence, high-bandwidth pulses of current. For such a pulse, the impedance of the $\frac{1}{2}C_1$ capacitors is much lower than that of the resistor except over a small low-frequency portion of its bandwidth. Therefore, the resistor acts approximately like an open circuit with respect to charge injection pulses, so the series combination of R and C_2 has little effect with respect to charge injection.

V. Additional Circuit Details and Measurement Results

A simplified functional diagram of the phase noise cancelling fractional-*N* PLL IC prototype is shown in Figure 9 and a die photograph of the IC is shown in Figure 10. Its reference frequency is 12 MHz, and its output frequency range covers the 2.4 GHz ISM band. The phase noise cancellation enables a loop bandwidth of 975 kHz which is close to the $f_{ref}/10$ loop bandwidth upper limit for stability [17].

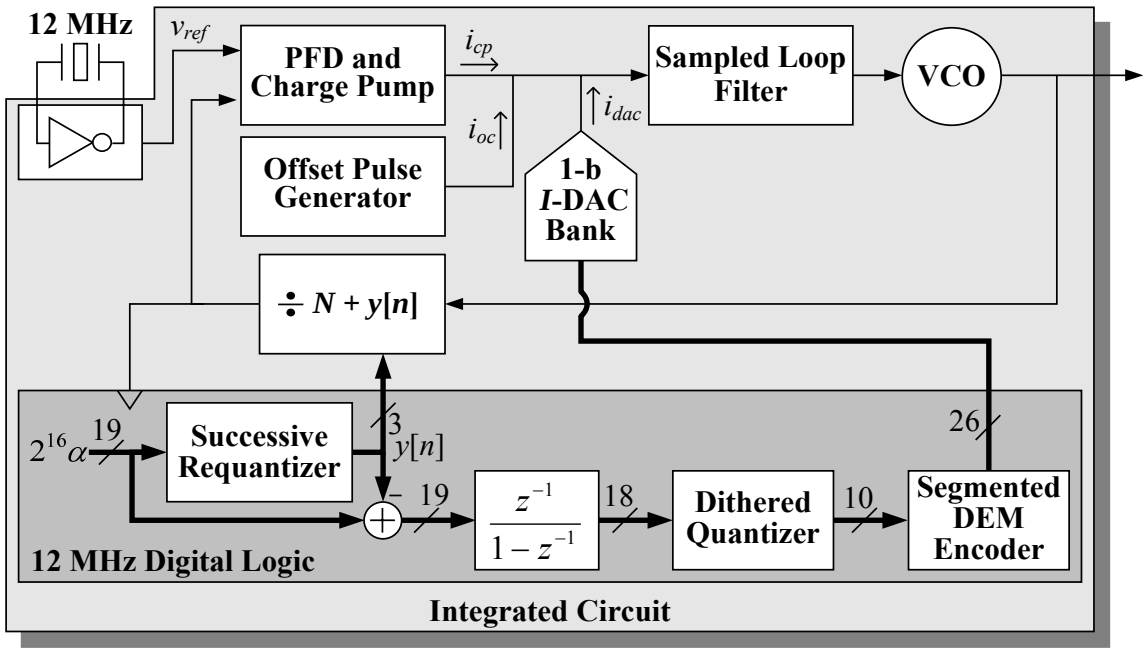


Fig. 9: High-level diagram of the integrated circuit prototype.

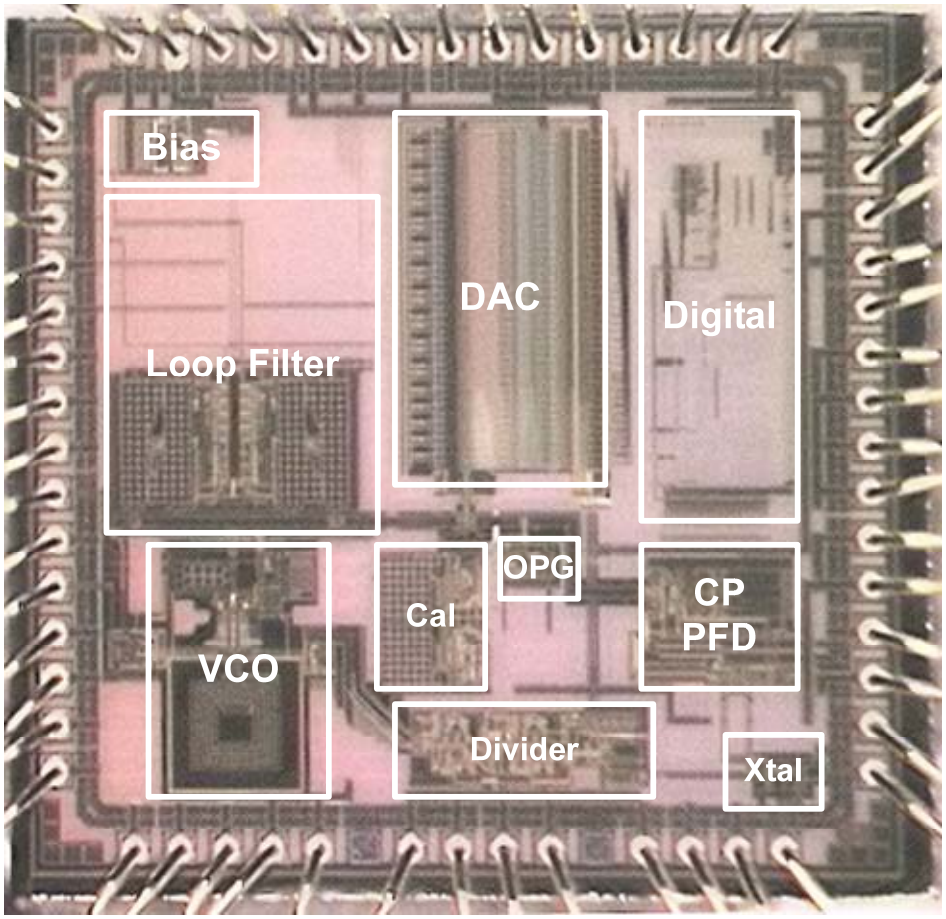


Fig. 10: Die photograph.

The IC is a modified version of that presented in [13]. The primary modifications are that the successive requantizer shown in Figures 5 and 6, the offset pulse generator shown in Figure 7, and the sampled loop filter shown in Figure 8 have been included. The other circuit blocks of the PLL described in [13] have been reused with relatively minor changes. For comparison, the PLL includes the ΔΣ modulator shown in Fig. 2a which can optionally be used instead of the successive requantizer, the offset pulse generator can be enabled or disabled, and the loop filter's sampling can be enabled or dis-

abled. With sampling disabled, the loop filter reduces to a conventional loop filter.

The divider is similar to that presented in [13] except with minor changes to provide timing signals that control the offset current generator and open the loop filter switch each reference period. As described in [13] the necessary timing signals are obtained by a chain of flip-flops clocked at half the VCO frequency. The timing signal used to close the loop filter switch each reference period could similarly have been derived within the divider block, but an RC one-shot circuit with a nominal duration of 25 ns is used instead for simplicity because the length of time the switch is left open is not critical. Provided the switch is open when the loop filter's input current is non-zero, the PLL dynamics are relatively insensitive to the length of time it is open.

A representative close-in PSD plot of the PLL's output with the successive requantizer, offset pulse generator, and sampled loop filter enabled and α chosen such that αf_{ref} = 50 kHz is shown in Figure 11. As expected fractional spurs occur at multiples of 50 kHz. Although the fractional spurs are well inside the 975 kHz loop bandwidth, they are all below −70 dBc in power.

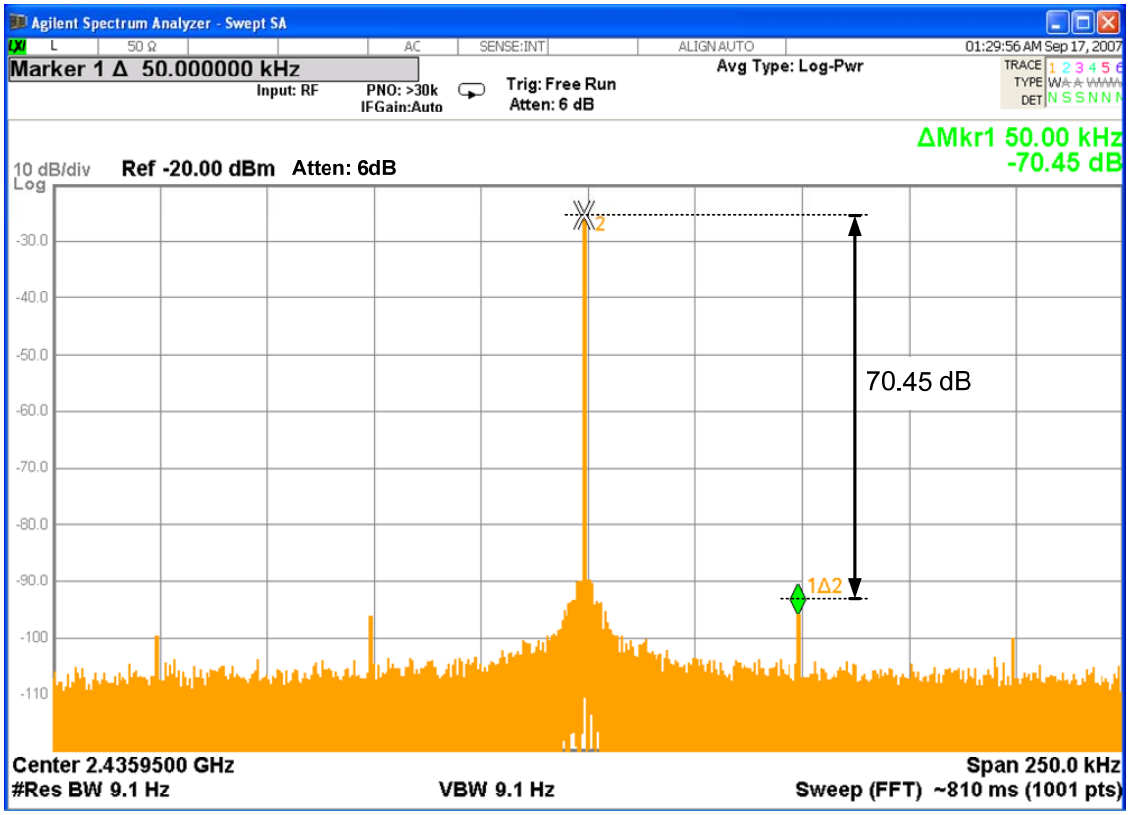


Fig. 11: Representative measured close-in output spectrum for the case of $\alpha f_{ref} = 50$ kHz.

To evaluate the fractional spur performance of the PLL comprehensively it is necessary to perform the measurement shown in Figure 11 for many values of α ranging between 0 and 1. Figure 12 presents the results of such measurements for four cases: 1) the ΔΣ modulator enabled and the offset pulse generator disabled, 2) the successive requantizer enabled and the offset pulse generator disabled, 3) the ΔΣ modulator enabled and the offset pulse generator enabled, and 4) the successive requantizer enabled and the offset pulse generator enabled. For each case, the figure shows the measured power of the largest spurious tone in the PLL's phase noise for each of 100 values of α ranging between 0 and 1.

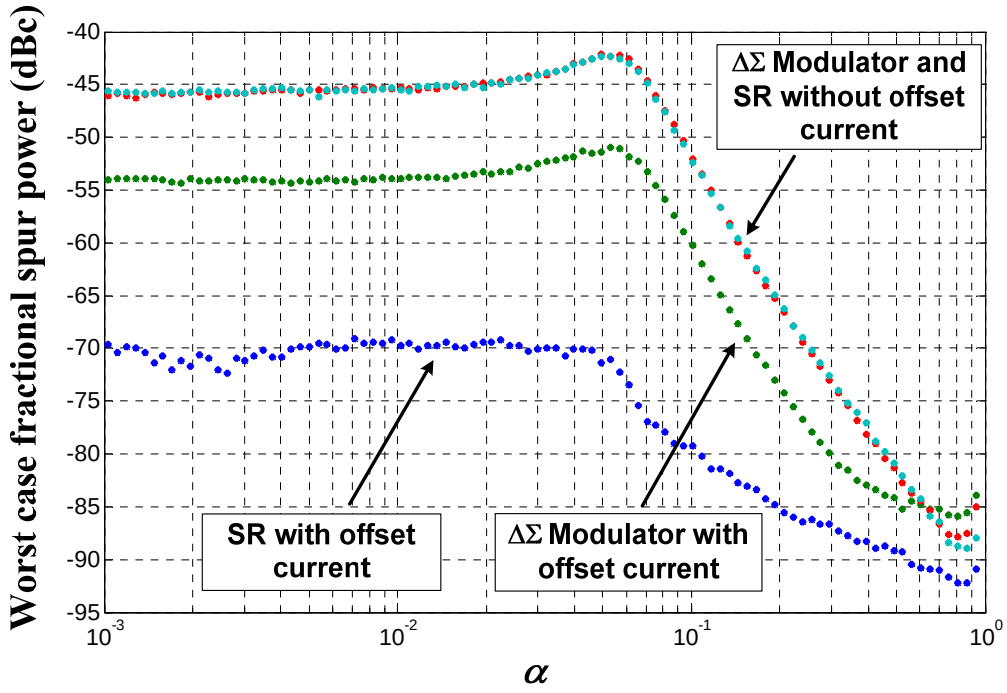


Fig. 12: Power levels of the largest measured fractional spurs with and without the enhancements enabled for 100 PLL frequency offsets in the range $0 < \alpha f_{ref} < 12\text{MHz}$.

As shown in Figure 12, the fractional spur powers for the two cases in which the offset pulse generator is disabled are almost identical, and are much higher than the corresponding fractional spur powers for the two cases in which the offset pulse generator is enabled. This suggests that fractional spur mechanism 1 is dominant over fractional spur mechanism 2. With the ΔΣ modulator, enabling the offset pulse generator reduces the fractional spur powers by a maximum of 9 dB, and with the successive requantizer, enabling the offset pulse generator reduces the fractional spur powers by a maximum of 27 dB. This suggests that once fractional spur mechanism 1 is circumvented, fractional spur mechanism 2 becomes significant. By circumventing fractional spur mechanism 2, the successive requantizer results in a maximum fractional spur power reduction of 18 dB relative to the ΔΣ modulator case.

As indicated in Figure 12, in each case the fractional spur powers are relatively constant for small values of α but decrease as α increases above about 0.055. This is expected because the frequencies of the fractional spurs increase with α, so after a point they move outside the loop bandwidth and are attenuated. An unusually large loop bandwidth has been used in this work to provide a worst-case scenario in which to demonstrate the spurious tone suppression techniques presented in the paper. The roll-offs shown in Figure 12 would start at smaller values of α if the loop bandwidth were decreased.

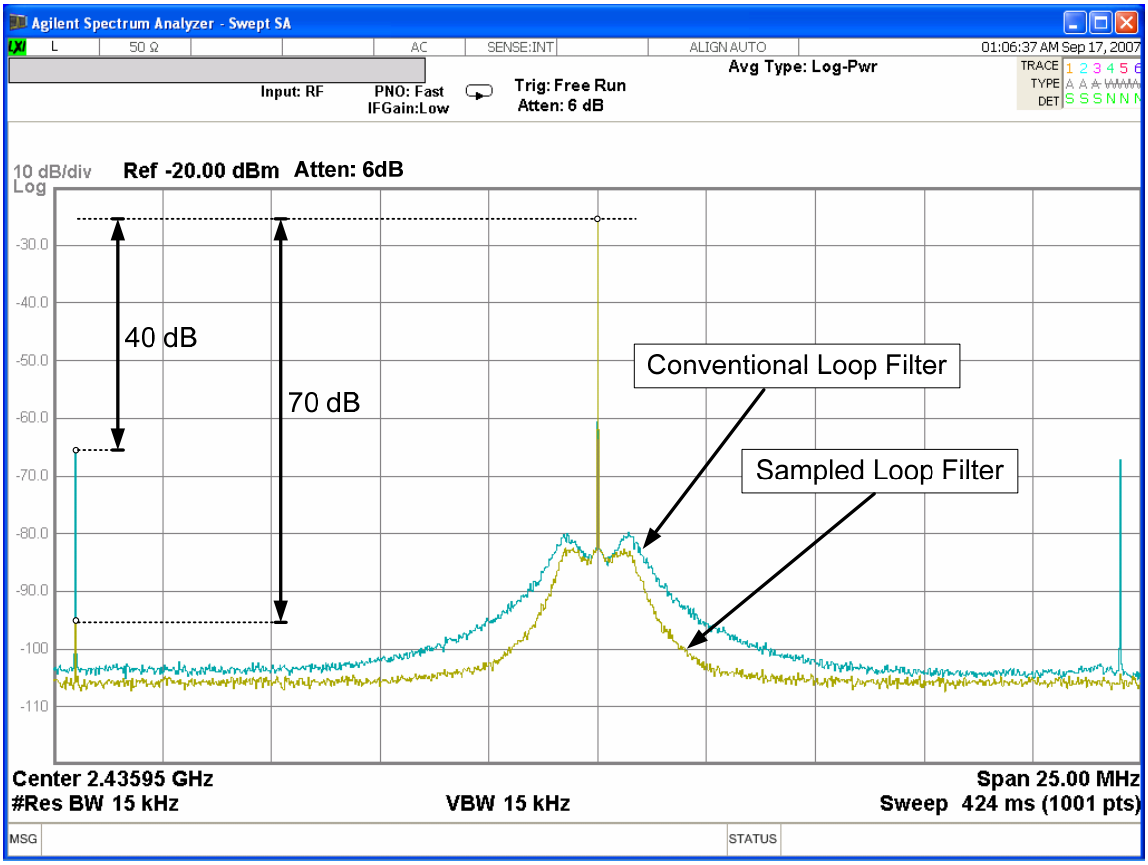


Fig. 13: Representative measured spectra with the sampled loop filter enabled and disabled.

Representative measured PSD plots of the PLL output over a 25 MHz span are

shown in Figure 13 for the PLL with the sampled loop filter and the PLL with the conventional loop filter. With the conventional loop filter the reference spur power is −40 dBc, which is large because of the large loop bandwidth and low reference frequency. With the sampled loop filter, the reference spur drops to −70 dBc.

Furthermore, it can be seen in Figure 13 that the phase noise away from the carrier is lower for the case of the sampled loop filter than for the case of the conventional loop filter. This is expected [18]. As described in [9], practical circuit limitations dictate that the current pulses from the charge pump have a fixed amplitude but variable widths whereas those from the DAC have a fixed width but variable amplitudes. Hence, even with perfect matching the component of the voltage corresponding to quantization noise at the node where the DAC and charge pump are connected can only be cancelled perfectly between the DAC and charge pump current pulses. When the pulses are non-zero, imperfectly cancelled current associated with quantization noise disturbs the node. Without sampling, the disturbance modulates the VCO, thereby increasing the phase noise. With sampling, the VCO is shielded from the disturbance.

Table 1: Performance table. Spur measurements represent the worst case results over the four ICs tested.

Design Details		
Technology	0.18 um 1P6M CMOS	
Package and Die area	32 pin TQFN, 2.2 mm × 2.2 mm	
Vdd	1.8 V	
Reference frequency	12 MHz	
Output frequency	2.4 – 2.5 GHz	
Measured loop bandwidth	975 kHz	
Measured Current Consumption		
VCO and Divider Buffer	5.9 mA	Core 27.1 mA
Divider	7.3 mA	
Charge Pump, PFD, and Buffers	8.6 mA	
Offset Current	0.6 mA	
Digital	1.9 mA	
DAC	2.8 mA	
Bandgap Bias Generator	5.4 mA	9.8 mA
Crystal Buffer	2.7 mA	
External Buffer	1.7 mA	
Measured Integer-N Performance		
Phase Noise at 100 kHz	-103 dBc/Hz	
Phase Noise at 3 MHz	-125 dBc/Hz	
Reference spur without sampling enabled	-58 dBc	
Reference spur with sampling enabled	-70 dBc	
Measured Fractional-N Performance		
Phase Noise at 100 kHz	-98 dBc/Hz	
Phase Noise at 3 MHz	-121 dBc/Hz	
Worst case in-band fractional spur with ΔΣ modulator	-45 dBc	
Worst case in-band fractional spur with SR	-64 dBc	
Reference spur without sampling enabled	-40 dBc	
Reference spur with sampling enabled	-70 dBc	

Four copies of the IC were tested. Table 1 shows the worst-case measurements taken from the four ICs. The fractional spur results for one of the ICs are shown in Figure 12, and two other of the ICs exhibited very similar results. However, one of the ICs exhibited a worst case fractional spur power of −64 dBc at a small number of frequencies

near the edge of the loop bandwidth. At all other frequencies, it behaved similarly to the other three ICs.

An IC wiring mistake disabled the DAC calibration circuitry described in [13], so the measurements described above were made after a one-time manual adjustment of the DAC gain. To confirm the diagnosis of the mistake, it was corrected in one copy of the IC by FIB microsurgery, but with the anticipated side effect of a coupling path that increased the measured in-band phase noise, 3MHz phase noise, and largest in-band fractional spur by 10dB, 3dB, and 3dB, respectively, above those shown in Table 1.

Table 2: Comparison of reference spur performance to the previously published state-of-the-art.

Reference frequency (MHz)	Loop bandwidth (kHz)	Reference spur magnitude (dBc)	Normalized reference spur (dBc)	Reference
12	975	-70	-70	This work
8	120	-81	-52	[19]
50	1000	-74	-50	[11]
12	730	-53	-48	[13]
1	40	-62	-50	[16]

Table 2 compares the PLL's reference spur performance to the previously published state-of-the-art. To a good approximation, the loop filter disturbance that causes reference spurs in a PLL is attenuated by −40dB per decade in frequency above the loop bandwidth. Therefore, to compare the reference spur powers of any two PLLs meaningfully, the difference between their reference-frequency-to-loop-bandwidth ratios must be considered. For each PLL, Table 2 shows both the measured reference spur power as well as the *normalized reference spur power*, which is the power that the reference spur would have had had the reference frequency-to-loop-bandwidth ratio been 12 MHz/975 kHz as in this paper. As shown the in table, the reference spur performance of the PLL

presented in this paper exceeds the previous state-of-the-art by 18 dB.[†]

Table 3: Comparison of fractional spur performance to the previously published state-of-the-art.

Reference frequency (MHz)	Loop bandwidth (kHz)	Reported fractional spur		Equivalent in-band fractional spur (dBc)	Reference
		Frequency (kHz)	Magnitude (dBc)		
12	975	50	-70		This work
50	1000	"In-band"	-45		[11]
35	700	8.5	-60		[10]
50	500	400	-42		[20]
50	390	98	-48		[21]
12	730	1000	-47	-42	[13]
25	1000	3125	-55	-36	[22]
33	200	257	-40	-36	[23]
26	35	2080	-100	-20	[15, 24, 25]

Table 3 compares the PLL's fractional spur performance to the previously published state-of-the-art. Unfortunately, comprehensive fractional spur measurement results such as shown in Figure 12 are rare in the previously published literature. In most cases, fractional spur powers are only reported for a small number of frequencies, often above the loop bandwidth. In cases where the power of a fractional spur within the loop bandwidth has been reported, the value is shown in Table 3 and it is assumed to be representative of all fractional spurs within the loop bandwidth. In cases for which the power of a fractional spur within the loop bandwidth is not reported, Table 2 provides an *equivalent in-band fractional spur power* obtained by adding the attenuation imposed by the PLL on the fractional spur given its position relative to the loop bandwidth. As in the case of the reference spur, the attenuation is taken to be −40dB per decade in frequency above the

[†] A JSSC paper by A. Maxim reports a PLL with a normalized reference spur of −73dBc. However, it has recently been determined by the Editor-in-Chief of the JSSC that this result is fraudulent, so it has not been included in the table.

loop bandwidth. As shown in the table, the fractional spur performance of the PLL presented in this paper exceeds the previous state-of-the-art by 10 dB.

ACKNOWLEDGEMENT

Chapter 1, in full, is a reprint of the material as it appears in the Journal of Solid State Circuits, vol. 43, pp. 2787-2797, Dec. 2008. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis for this paper.

REFERENCES

[1] B. Razavi, *Phase-Locking in High-Performance Systems: From Devices to Architectures,* Wiley-Interscience, 2003.

[2] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, Second Edition, Cambridge University Press, 2003.

[3] K. Wang, A. Swaminathan, I. Galton, "Spurious-tone suppression techniques applied to a wide-bandwidth 2.4GHz fractional-N PLL", *IEEE International Solid-State Circuits Conference*, Digest of Technical Papers, February 2008.

[4] A. Swaminathan, A. Panigada, E. Masry, I. Galton, "A digital requantizer with shaped requantization noise that remains well behaved after non-linear distortion," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5382-5394, November 2007.

[5] B. Miller, B. Conley, "A multiple modulator fractional divider," *Annual IEEE Symposium on Frequency Control*, vol. 44, pp. 559-568, March 1990.

[6] B. Miller, B. Conley, "A multiple modulator fractional divider," *IEEE Transactions*

on Instrumentation and Measurement, vol. 40, no. 3, pp. 578-583, June 1991.

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

[8] S. Pamarti, J. Welz, I. Galton, "Statistics of the quantization noise in 1-bit dithered single-quantizer digital delta-sigma modulators," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, vol. 54, no. 3. pp. 492-503, March 2007.

[9] S. Pamarti, L. Jansson, I. Galton. "A wideband 2.4GHz ΔΣ fractional-N PLL with 1 Mb/s in-loop modulation," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 49-62, January 2004.

[10] E. Temporiti, G. Albasini, I. Bietti, R. Castello, M. Colombo. "A 700kHz 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, September 2004.

[11] S. E. Meninger, and M. H. Perrott. "A 1MHz bandwidth 3.6GHz 0.18um 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.

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

[13] A. Swaminathan, K. Wang, I. Galton, "A wide-bandwidth 2.4GHz ISM-band fractional-*N* PLL with adaptive phase-noise cancellation," *IEEE Journal of Solid-State Circuits*, pp. 2639-2650, Dec. 2007.

[14] B. Razavi, *Design of Analog CMOS Integrated Circuits*, McGraw Hill, 2001.

[15] B. De Muer, M. Steyaert. "A CMOS Monolithic ΔΣ-Controlled Fractional-N Frequency Synthesizer for DCS-1800," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 7, July 2002.

[16] B. Zhang, P. E. Allen, J. M. Huard, "A fast switching PLL frequency synthesizer with an on-chip passive discrete-time loop filter in 0.25-µm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 855 – 865, June 2003.

[17] F. M. Gardner, "Charge-pump phase-lock loops," *IEEE Transactions on Communi-*

cations, vol. COM-28, pp. 1849-1858, November 1980.

[18] L. Liu, B. Li, "Phase Noise Cancellation for a Σ-Δ Fractional-*N* PLL Employing a Sample-and-hold Element," Asia-Pacific Microwave Conference, December 2005.

[19] C. Liang, H. Chen, S. Liu, "Spur-Suppression Techniques for Frequency Synthesizers," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 54, no. 8, pp. 653-657, Aug. 2007.

[20] C.-M. Hsu, M. Z. Straayer, M. H. Perrott., "A Low-Noise, Wide-BW 3.6Ghz Digital ΔΣ Fractional-N Frequency Synthesizer with a Noise-Shaping Time-to-Digital Converter and Quantization Noise Cancellation," *IEEE International Solid-State Circuits Conference*, Digest of Technical Papers, pp. 340-341, 617, February 2008.

[21] K. Tajima, R. Hayashi, T. Takagi, "New suppression scheme of ΔΣ fractional-N spurs for PLL synthesizers using analog phase detectors," *Microwave Symposium Digest, 2005 IEEE MTT-S International*, pp. 4, 12-17 June 2005.

[22] C. Park, O. Kim, B. Kim, "A 1.8-GHz self-calibrated phase-locked loop with precise I/Q matching," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 5, pp.777-783, May 2001.

[23] Y.-C. Yang, S.-A. Yu, Y.-H. Liu, T. Wang, S.-S. Lu, "A Quantization Noise Suppression Technique for ΔΣ Fractional-N Frequency Synthesizers," *IEEE Journal of Solid-State Circuits* , vol. 41, no. 11, pp. 2500-2511, November 2006.

[24] B. De Muer, M. S. J. Steyaert, "On the analysis of ΔΣ fractional-N frequency synthesizers for high-spectral purity," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 50, no.11, pp. 784-793, November 2003.

[25] B. De Muer, M. Steyaert, *CMOS Fractional-N Synthesizers, Design for High Spectral Purity and Monolithic Integration*, Kluwer, Norwell, MA, 2003.

Chapter II:

A Discrete-Time Model For the Design of Type-II PLLs with Passive Sampled Loop Filters

Abstract—Type-II charge pump PLLs are used extensively in electronic systems for frequency synthesis. Recently, a passive sampled loop filter (SLF) was shown to offer major benefits over the conventional continuous-time loop filter (CLF) traditionally used in such PLLs. These benefits include greatly enhanced reference spur suppression, elimination of charge pump pulse-position modulation nonlinearity, and, in the case of phase noise cancelling fractional-*N* PLLs, improved phase noise cancellation. The main disadvantage of the SLF to date has been the lack of a linear time-invariant (LTI) model with which to perform the system-level design of SLF-based PLLs. Without such a model, designers are forced to rely on trial and error iteration supported by lengthy transient simulations. This paper presents an accurate LTI model of SLF-based Type-II PLLs that eliminates this disadvantage.

INTRODUCTION

Integer-*N* and fractional-*N* phase locked loops (PLLs) are used extensively in electronic systems to synthesize higher frequency signals from lower-frequency references. The majority of these PLLs are charge pump based Type-II PLLs [1].

Recently, sampled loop filters (SLFs) have been shown to offer advantages over continuous-time loop filters (CLFs) in PLLs. SLFs can greatly reduce reference spurs in

both integer-*N* and fractional-*N* PLLs, [2, 3]. They eliminate charge pump pulse-position modulation distortion in fractional-*N* PLLs [4, 5], and they improve phase noise cancellation in phase noise cancelling fractional-*N* PLLs [5, 6]. Moreover, SLFs eliminate the large reference spur that would otherwise arise as a side effect of the charge pump offset current method for reducing fractional spurs in fractional-*N* PLLs [3, 7].

Several different types of SLFs for PLLs have been published. In [4] an active SLF is implemented by preceding a CLF with an op-amp based sample-and-hold circuit. In [2] a passive switched-capacitor SLF is implemented for a type-I PLL. In [3], a passive SLF is implemented with the addition of a transistor switch within an otherwise conventional CLF.

The SLF presented in [3] offers a major benefit over the other SLFs: it is the only published passive SLF applicable to Type-II PLLs. The sampling operation involves only a single switch, so it consumes very little power and circuit area beyond those of a comparable CLF. Its applicability to Type-II PLLs is important because such PLLs are by far the most widely used PLLs at present. Furthermore, the SLF has been demonstrated in a fractional-*N* PLL with record-setting reference and fractional spur performance.

The main drawback to date of the SLF presented in [3] has been the lack of a linear, time-invariant (LTI) model with which to perform the system-level design of PLLs based on the SLF. Without such a model, designers are forced to rely on trial and error iteration and lengthy transient simulations as their primary design tools.

Despite its implementation simplicity, the SLF presented in [3] is more difficult to analyze than the other published SLFs because it behaves as a time-varying continuous-time filter. Therefore, it cannot be well-approximated as a continuous-time LTI system.

Nevertheless, as proven in this paper, PLLs based on the SLF can be modeled accurately as discrete-time LTI systems. The paper derives such an LTI model, and demonstrates how it enables the system-level design of PLLs without the need to resort to computer simulation. Hence, the results of the paper eliminate the drawback described above.

The model yields equations which accurately predict the transfer functions, bandwidth, and phase margin of the PLL in terms of its component values. While the equations are not simple, they each have closed form. They can be implemented easily in a tool such as Matlab and used to rapidly generate results that heretofore required lengthy transient simulations. The PLL design process is inherently iterative, so not having to simulate the PLL at each iteration step significantly speeds up the design process.

The paper is organized such that all the information required to use the model to design PLLs is presented separately from the derivation of the model. This allows readers to use the model prior to understanding its derivation. The information required to use the model is presented in Sections II-III and Appendix A, and the detailed mathematical derivation of the model is presented in Section IV and Appendix B.

OVERVIEW OF THE SAMPLED LOOP FILTER PLL

The block diagram of a typical charge-pump based integer-N PLL is shown in Fig. 14a [1]. Its purpose is to generate a spectrally pure periodic output signal with a frequency of Nf_{ref}, where N is a positive integer, and f_{ref} is the frequency of the reference signal, $V_{ref}(t)$. It consists of a phase-frequency detector (PFD), a charge pump (CP), a lowpass loop filter (LF), a voltage controlled oscillator (VCO), and a digital divider.

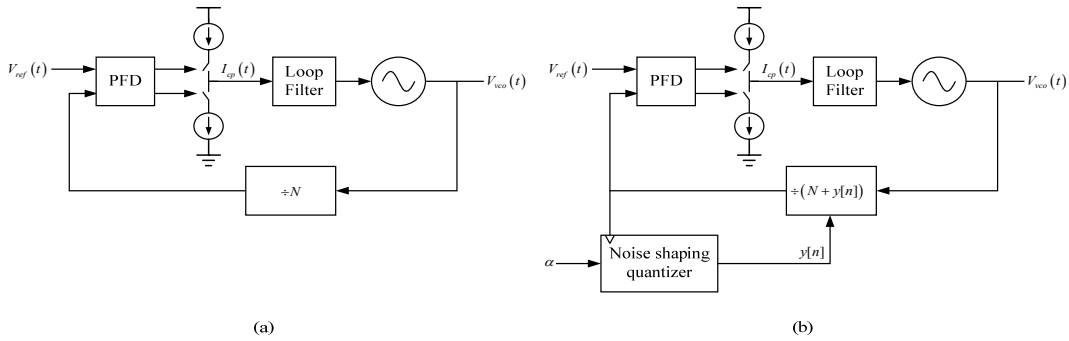


Fig. 14: Block diagram of a typical (a) integer-N PLL and (b) fractional-N PLL.

The divider output is a two-level signal in which the nth and (n+1)th rising edges, for n = 0, 1, 2, …, are separated by N periods of the VCO output. The PFD compares the positive-going edges of the reference signal to those of the divider's output signal and causes the charge pump to drive the loop filter with current pulses whose widths are proportional to the phase difference between the two signals. The pulses are lowpass filtered by the loop filter and the resulting waveform drives the VCO.

Fig. 15a shows a continuous-time loop filter, and Fig. 15b shows the sampled loop filter addressed in this paper. The SLF differs from the continuous-time LF only in that it includes a switch which splits C_p into λC_p and $(1-\lambda)C_p$, where $0 < \lambda < 1$. For example, in [3], $\lambda = 0.5$. The switch is opened and closed once per reference period such that when the PLL is locked, λC_p and $(1-\lambda)C_p$ are disconnected whenever $I_{cp}(t) \neq 0$. As explained and experimentally demonstrated in [3], this significantly reduces the reference spur compared to the conventional LF.

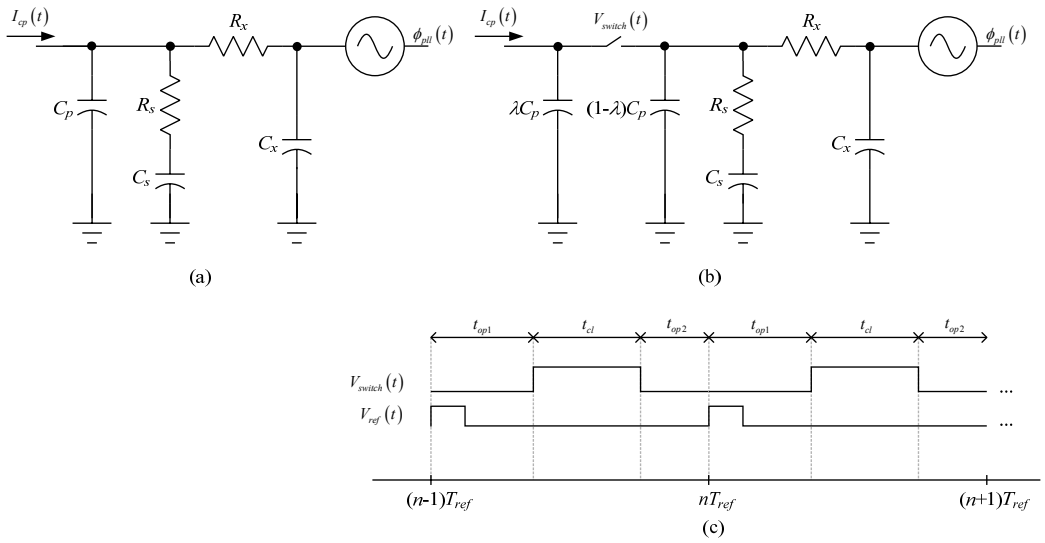


Fig. 15: Circuit diagram of (a) a continuous time loop filter with the VCO and (b) a sampled loop filter with VCO and (c) the timing of $V_{switch}(t)$.

The switch is controlled by the two-level signal $V_{switch}(t)$; it is closed when $V_{switch}(t)$ is high, and open when $V_{switch}(t)$ is low. A typical waveform for $V_{switch}(t)$ is shown in Fig. 15c. The nth reference period is defined as the time interval between the nth and $(n + 1)$th rising edges of the reference signal. In the case of a noise-free reference signal, these edges occur at times nT_{ref} and $(n + 1)T_{ref}$, respectively, where $T_{ref} = 1/f_{ref}$. As indicated in Fig. 15c, during each reference period, the switch is first open for a duration of t_{op1}, then closed for a duration of t_{cl}, and then open for a duration of t_{op2}, where t_{op1}, t_{cl}, and t_{op2} are constants chosen by the designer. Together with the loop filter components, these constants define the behavior of the SLF.

As described in Section III and suggested by the model equations in Appendix A, decreasing t_{cl} has the effect of decreasing the phase margin of the PLL whereas the values of t_{op1} and t_{op2} for any given value of t_{cl} have little effect on the dynamics of the PLL. Therefore, t_{op1}, t_{cl}, and t_{op2} should be chosen such that t_{cl} is as large as possible subject to

the requirement that the switch be open whenever $I_{cp}(t) \neq 0$ once the PLL is locked.

The block diagram of a typical charge-pump based fractional-N PLL is shown in Fig. 14b [1]. Its purpose is to generate a spectrally pure periodic output signal with a frequency of $(N + \alpha)f_{ref}$, where N is again a positive integer and α is a fractional value between 0 and 1. The fractional-N PLL differs from the integer-N PLL only in that the nth and $(n+1)$th rising edges of the divider output, for $n = 0, 1, 2, \ldots$, are separated by $N + y[n]$ periods of the VCO output, where $y[n]$ is the integer-valued output sequence from a noise-shaping quantizer with input α. Typically, the noise-shaping quantizer is a digital delta-sigma modulator, but other types of quantizers such as a successive requantizer can also be used [3].

Description and Application of the PLL Model

This section describes the proposed model of the PLLs shown in Fig. 14 with the SLF of Fig. 15b, and explains how the model can be used to analyze and design such PLLs. The mathematical derivations that underlie the models are deferred to Section IV and Appendix B.

Model Description

The phase of the fractional-N PLL's output signal at time t can be written as

$$2\pi\left(N+\alpha\right)f_{ref}t+\phi_{pll}\left(t\right) \tag{12}$$

where $\phi_{pll}(t)$ represents the PLL's phase error, i.e., the difference between the actual phase and ideal phase of the PLL output signal at time t.

The purpose of a PLL model is to provide a simple means of evaluating $\phi_{pll}(t)$ in terms of the PLL's design parameters and error signals such as circuit noise, assuming

the PLL is already locked. PLLs are neither linear nor time-invariant, but when locked they can be approximated as linear time-invariant (LTI) systems. For example, the most commonly used model for PLLs with conventional loop filters is a continuous-time LTI system that accurately models the locked behavior of such PLLs [8, 9, 12]. Discrete-time LTI models have also been developed for such PLLs [8, 10, 11].

The model presented in this section is a discrete-time LTI system applicable to the SLF-PLL. As described in the next section, the sampling operation of the SLF would result in a time-varying continuous-time model which would be difficult to analyze, and this problem is avoided by using a discrete-time model.

Two versions of the model are presented: a single-rate version and a multi-rate version. The single-rate version provides samples of $\phi_{pll}(t)$ at a sample-rate of f_{ref}. The multi-rate version provides samples of $\phi_{pll}(t)$ at a sample-rate of Lf_{ref}, where $L \geq 2$ is a positive integer.

The two versions of the model are identical in terms of how they represent the PLL's feedback behavior, but the latter performs interpolation to obtain an extra $L - 1$ output samples per reference period. When $\phi_{pll}(t)$ has most of its power concentrated at frequencies with magnitudes less than $f_{ref}/2$, the single-rate version of the model is sufficient. The multi-rate version, although more complicated than the single-rate version, is useful in cases where $\phi_{pll}(t)$ has enough power at frequencies with magnitudes above $f_{ref}/2$ that it is necessary to sample $\phi_{pll}(t)$ at a higher sample-rate than f_{ref}.

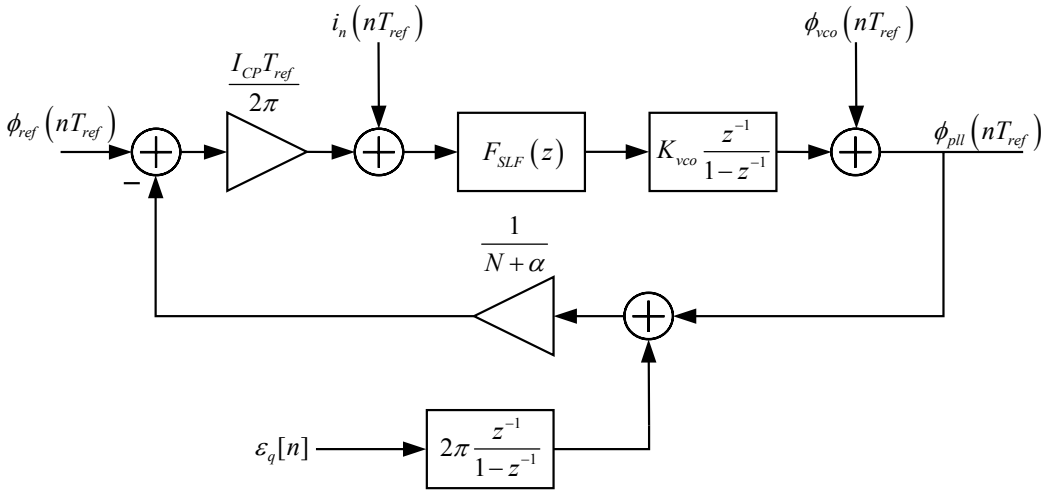


Fig. 16: Single rate discrete-time, linearized model of a SLF-PLL with noise sources.

The single-rate version of the model is shown in Fig. 16, where I_{CP} is the magnitude of current pulses sourced and sunk by the charge pump, $i_n(nT_{ref})$ is charge pump noise sampled at nT_{ref}, $\phi_{ref}(nT_{ref})$ is the reference signal's phase noise sampled at nT_{ref}, $\phi_{vco}(nT_{ref})$ is the open-loop VCO phase noise sampled at nT_{ref}, $\varepsilon_q[n] = y[n] - \alpha$ is the quantization noise from the noise-shaping quantizer,

$$F_{SLF}(z) = K\frac{\left(1-\gamma_1 z^{-1}\right)\left(1-\gamma_2 z^{-1}\right)\left(1-\gamma_3 z^{-1}\right)}{\left(1-z^{-1}\right)\left(1-\beta_2 z^{-1}\right)\left(1-\beta_3 z^{-1}\right)} \tag{13}$$

and β_i, γ_j, and K are constants. Appendix A provides equations that yield the values of β_i, γ_j, and K given the loop filter design values, i.e., the values of C_p, C_s, C_x, R_s, R_x, λ, t_{op1}, t_{cl}, and t_{op2}. The model as drawn in Fig. 16 applies to the fractional-N PLL, but when modified to have $\alpha = 0$ and $\varepsilon_q[n] = 0$ it also applies to the integer-N PLL.

The PLL's locked behavior can be analyzed by applying well-known LTI system techniques to the model of Fig. 16. Specifically, the model indicates that the loop gain is

$$T(z) = -\frac{I_{CP}T_{ref}K_{vco}}{2\pi(N+\alpha)}F_{SLF}(z)\frac{z^{-1}}{1-z^{-1}}. \tag{14}$$

Therefore, the PLL's phase margin (PM) is

$$PM = \angle T\left(e^{j\omega_u T_{ref}}\right) \tag{15}$$

where ω_u is the unity-gain frequency of $T(e^{j\omega Tref})$, and the loop-bandwidth of the PLL is approximately equal to ω_u. The noise transfer functions from $\phi_{ref}(nT_{ref})$, $\phi_{vco}(nT_{ref})$, $\varepsilon_q[n]$, and $i_n(nT_{ref})$ to $\phi_{pll}(nT_{ref})$, respectively, are

$$\frac{\phi_{pll}}{\phi_{ref}}(z) = (N+\alpha)\frac{T(z)}{1+T(z)} \tag{16}$$

$$\frac{\phi_{pll}}{\phi_{vco}}(z) = \frac{1}{1+T(z)} \tag{17}$$

$$\frac{\phi_{pll}}{\varepsilon_q}(z) = -2\pi\frac{z^{-1}}{1-z^{-1}}\frac{T(z)}{1+T(z)} \tag{18}$$

and

$$\frac{\phi_{pll}}{i_{cp}}(z) = \frac{2\pi(N+\alpha)}{I_{CP}T_{ref}}\frac{T(z)}{1+T(z)}. \tag{19}$$

The multi-rate version of the PLL model differs from the single-rate version shown in Fig. 16 only in its representation of the SLF and VCO. The components of the single-rate model that represent the SLF and VCO are drawn separately in Fig. 17a. The multi-rate version is obtained by removing these components in the single-rate model of Fig. 16 and replacing them with the components shown in Fig. 17b. The resulting multi-rate model is shown in Fig. 18.

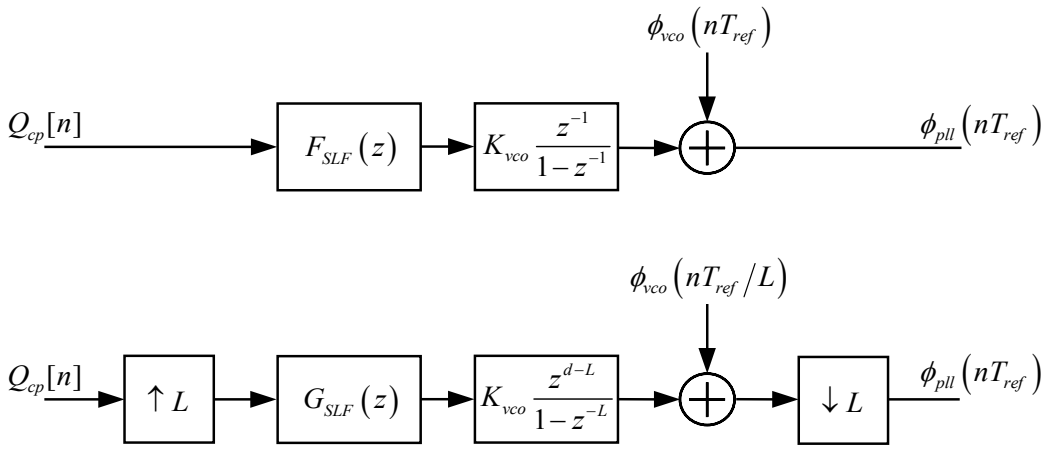


Fig. 17: Model of the SLF and VCO for the (a) single rate and (b) multi-rate cases.

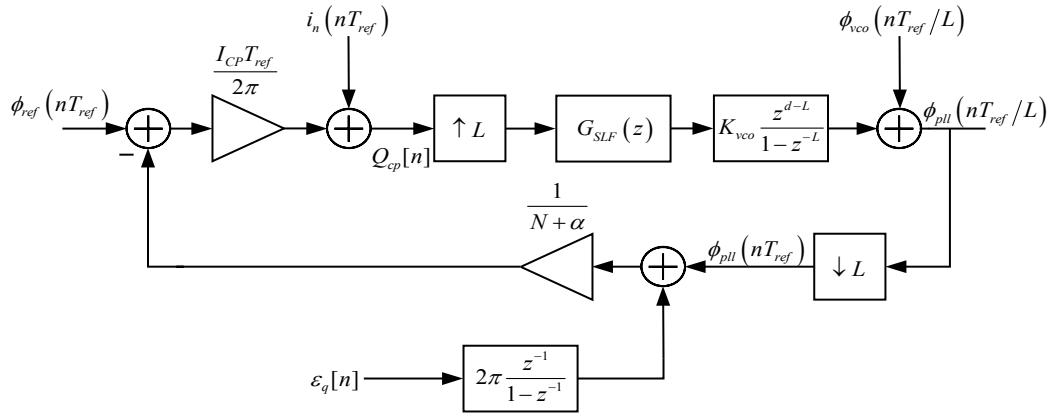


Fig. 18: Multi-rate discrete-time, linearized model of a SLF-PLL with noise sources.

Therefore the SLF and VCO in the multi-rate model are represented by the components shown in Fig. 17b: an L-fold up-sampler, a discrete-time filter with sample-rate Lf_{ref} and transfer function

$$K_{vco}G_{SLF}(z)\frac{z^{d-L}}{1-z^{L}}, \tag{20}$$

the addition of the VCO phase noise sampled at a rate of Lf_{ref}, and an L-fold down-sampler. The integer d is defined as

$$d = \left\lfloor \frac{t_{cl} + t_{op2}}{T_{ref}} L \right\rfloor \tag{21}$$

where $\lfloor y \rfloor$ is the largest integer less than or equal to y. The output of the L-fold up-sampler is given by

$$R_{cp}[n] = \begin{cases} Q_{cp}[n/L], & \text{if } n \text{ is an integer multiple of } L \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

and the L-fold down-sampler discards all but every Lth sample of $\phi_{pll}(nT_{ref}/L)$ to obtain $\phi_{pll}(nT_{ref})$. The transfer function $G_{SLF}(z)$ has the form

$$G_{SLF}(z) = \frac{1}{L} \sum_{i=0}^{L-1} z^{-i} F_{SLF,i}\left(z^{L}\right) \tag{23}$$

where each $F_{SLF,i}(z)$ has the same three poles as (13), and can have either two or three zeros. Appendix A provides equations that yield the full transfer function of each $F_{SLF,i}(z)$ given L and the loop filter design values, i.e., the values of C_p, C_s, C_x, R_s, R_x, λ, t_{op1}, t_{cl}, and t_{op2}.

Analysis Example

The parameters that specify the system-level design of an SLF-PLL are N, f_{ref}, I_{CP}, K_{VCO}, and the loop filter design values, i.e., the values of C_p, C_s, C_x, R_s, R_x, λ, t_{op1}, t_{cl}, and t_{op2}. Both versions of the model described in Section III-A describe the locked behavior of the PLL in terms of these parameters. An example is presented below for the case of an SLF-PLL with $N = 200$, $f_{ref} = 10$ MHz, $I_{CP} = 2$ mA, $K_{vco} = 2\pi \cdot 120 \cdot 10^6$ rad/(s·V), $C_p = 2.53$ pF, $C_s = 328$ pF, $R_s = 5408\ \Omega$, $C_x = 795$ fF, $R_x = 20\ \text{K}\Omega$, $\lambda = 0.5$, $t_{op1} = 50$ ns, $t_{op2} = 10$ ns, and $t_{cl} = 40$ ns.

To apply the single-rate version of the model, it is first necessary to calculate

$F_{SLF}(z)$, and to apply the multi-rate version of the model it is first necessary to calculate $G_{SLF}(z)$. Appendix A provides the equations required to calculate $F_{SLF}(z)$ and $G_{SLF}(z)$ starting from the loop filter design values. Executing Steps 1-6 in Appendix A with the loop filter design values listed above yields

$$F_{SLF}(z) = 225\frac{\left(1.06 - z^{-1}\right)\left(-4.74 - z^{-1}\right)\left(-67749 - z^{-1}\right)}{\left(1 - z^{-1}\right)\left(22.9 - z^{-1}\right)\left(775 - z^{-1}\right)}. \tag{24}$$

Additionally executing Steps 7-8 in Appendix A for $L = 2$ yields

$$F_{SLF,0}(z) = F_{SLF}(z) \tag{25}$$

where $F_{SLF}(z)$ is given by (24) and

$$F_{SLF,1}(z) = -637335\frac{\left(1.06 - z^{-1}\right)\left(-139 - z^{-1}\right)}{\left(1 - z^{-1}\right)\left(22.9 - z^{-1}\right)\left(775 - z^{-1}\right)}. \tag{26}$$

Substituting (25) and (26) into (23) yields $G_{SLF}(z)$ for $L = 2$. The same procedure can be used to obtain $G_{SLF}(z)$ for any positive integer L.

These $F_{SLF}(z)$ and $G_{SLF}(z)$ functions can be used in the versions of the model shown in Fig. 16 and Fig. 18, respectively, to analyze the locked behavior of the SLF-PLL. As described above the model implies that the loop gain of the PLL is given by (14) . Substituting (24) into (14) and solving for the unity gain frequency indicates that the loop bandwidth of the PLL is 1 MHz and it follows from (15) that the phase margin of the PLL is 60 degrees. Figures 19-21 show various additional aspects of the behavior of the SLF-PLL as predicted by the two versions of the model.

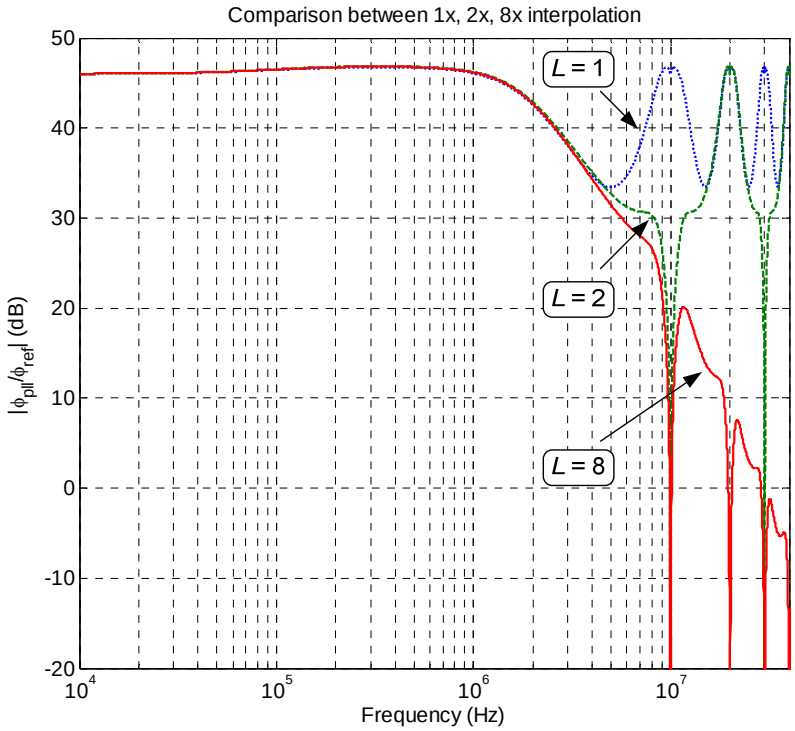


Fig. 19: Comparison between the transfer function for the single-rate (L=1) and the transfer functions for the multi-rate (L=2, L=8) cases.

Fig. 19 shows the phase noise transfer function from the reference signal input, i.e., the squared magnitude of (16) in dB with $z = \exp(j2\pi f/f_{ref})$, as predicted by the single-rate version of the model (L = 1), and the multi-rate version of the model for L = 2 and L = 8. As expected, there is little deviation among the predicted transfer functions for frequencies below $f_{ref}/4$, and each transfer function is periodic with a period of Lf_{ref}. In general, the larger the value of L, the higher the maximum frequency at which the transfer function predicted by the model accurately represents that of the actual SLF-PLL. The PLL bandwidth is relatively wide in this example, so the transfer function is not highly attenuated at $f_{ref}/4$. In such cases the multi-rate version of the model provides useful information.

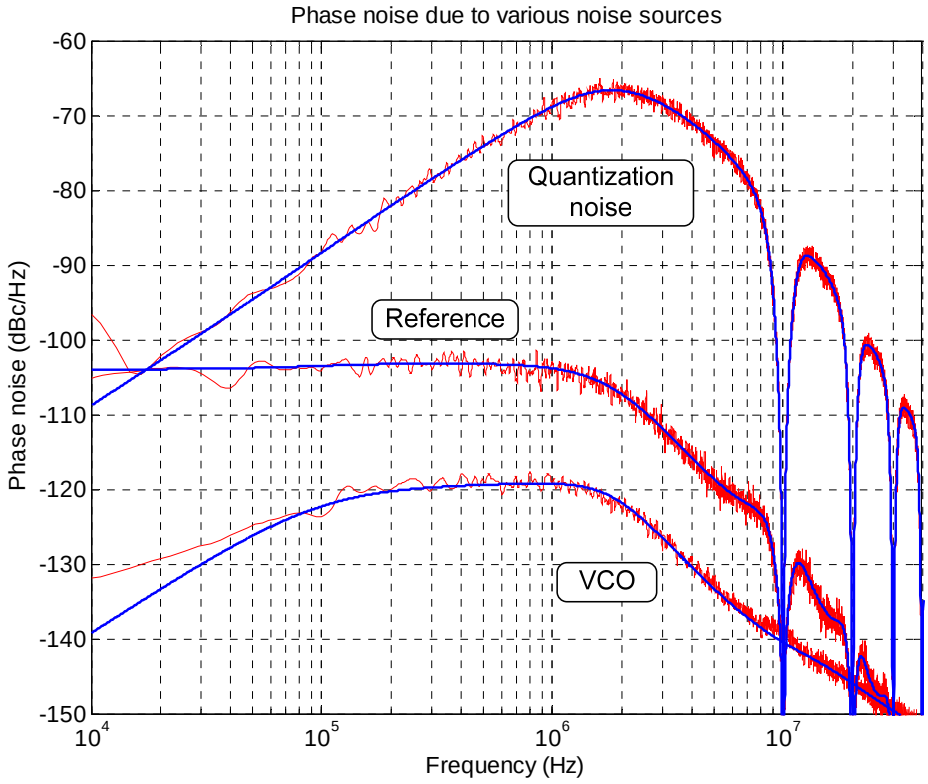


Fig. 20: Comparison between the model and simulated results for various noise sources for a fractional-*N* PLL. (*L*=8)

Fig. 20 shows plots of the squared magnitudes of (16), (17), and (18) in dB with $z = \exp(j2\pi f/f_{ref})$ as predicted by the model with $L = 8$ and the corresponding transfer functions as predicted by computer simulation. The plots suggest that model agrees well with the simulation. The one exception is that the transfer functions corresponding to (17) deviate somewhat at low frequencies, but this has been traced to limitations of the simulator.

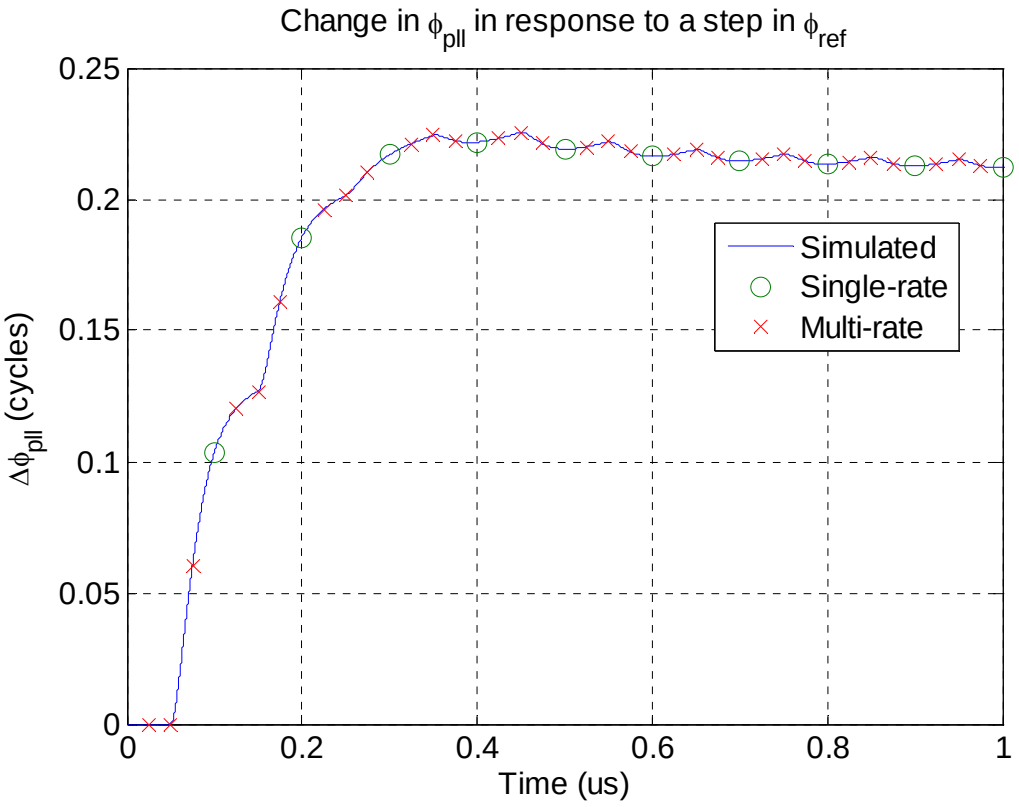


Fig. 21: Simulated VCO output phase of the SLF-PLL corresponding to a reference signal phase step, and the corresponding sample values predicted by the model.

Fig. 21 shows a time-domain plot of the simulated VCO output phase corresponding to a reference signal phase step, and the corresponding sample values predicted by the model. As expected, the sample values predicted by the model fall precisely on the simulated curve.

The Synthesis Problem

As shown above, the proposed model allows for straightforward analysis of an SLF-PLL given the PLL design parameters, i.e. given N, f_{ref}, I_{CP}, K_{VCO}, C_p, C_s, C_x, R_s, R_x, λ, t_{op1}, t_{cl}, and t_{op2}. However, designers are often faced with the *synthesis problem* of choosing the SLF component values, i.e., C_p, C_s, C_x, R_s, and R_x, such that the PLL has a

desired loop bandwidth and phase margin. Typically, N, f_{ref}, I_{CP}, K_{VCO}, λ, t_{op1}, t_{cl}, and t_{op2} are known prior to choosing the SLF component values because they depend on circuit-level considerations and application requirements.

The model equations could be solved numerically to provide the SLF component values in terms of the other PLL design parameters and the desired loop bandwidth and phase margin, but it is simpler to use the following iterative approach. The first step is to choose the loop filter component values for a conventional CLF-PLL that approximately achieves the desired loop bandwidth and phase margin. Approximate equations that provide the values of C_p, C_s, and R_s for a conventional CLF-PLL in the absence of C_x and R_x are well-known [12]. Typically, designers use these equations to find C_p, C_s, and R_s and then choose C_x and R_x such that the extra pole they introduce has a high-enough frequency that it negligibly affects the loop bandwidth and phase margin. The second step is to iteratively adjust the values of C_p, C_s and R_s, to compensate for the sampling operation in the SLF using the proposed SLF-PLL model to guide the iteration process.

As observed in [4], the sampling operation in a sampled loop filter decreases the phase margin of the PLL by approximately the product of the loop bandwidth and the duration over which the switch is open each reference period, i.e.,

$$PM_{SLF} = PM_{CLF} - \omega_{LBW}\left(t_{op1} + t_{op2}\right), \tag{27}$$

where ω_{LBW} is the loop bandwidth. This loss in phase margin can be addressed by increasing the ratio, C_s/C_p. Increasing C_s moves one of the filter zeros to a lower frequency, but C_s typically is large, so moving the zero significantly requires a significant increase in circuit area. Decreasing C_p moves one of the filter poles to a higher frequency, but this has the disadvantage of reducing the high frequency attenuation of the loop. Experimen-

tally, a combination of these two adjustments yields the best tradeoff between area and high frequency attenuation. The approach is to iteratively adjust C_s and C_p and at each iteration step use the proposed SLF-PLL model to evaluate whether further adjustment of C_s and C_p is necessary. A similar iterative process can be used to optimize the choices of C_x and R_x if necessary.

The amount by which the sampling operation affects the behavior of the SLF-PLL depends to a large extent on the loop bandwidth. If the loop bandwidth is sufficiently low, the loop filter components obtained in Step 1 above for the CLF-PLL can be used in the SLF-PLL with only a minor degradation of the phase margin. Nevertheless, in such cases the proposed SLF-PLL model is useful to verify that no further adjustment is necessary.

Synthesis Example

Consider an integer-N SLF-PLL for which $N = 200$, $f_{ref} = 10$ MHz, $I_{CP} = 2$ mA, $K_{vco} = 2\pi \cdot 120 \cdot 10^6$ rad/(s·V), $\lambda = 0.5$, $t_{op1} = 50$ ns, $t_{cl} = 40$ ns and $t_{op2} = 10$ ns. Suppose it is desired to choose C_p, C_s, C_x, R_s, and R_x, such that the loop bandwidth is 1 MHz and the phase margin is 60 degrees.

The first step of the procedure described above is to choose C_p, C_s, C_x, R_s, and R_x for a corresponding conventional CLF-PLL. Applying the equations in [12] with C_x = 795 fF, $R_x = 20$ kΩ, a loop bandwidth of 1.015 MHz, and a phase margin of 67 degrees yields

$$C_p = 6 \text{ pF}, C_s = 139 \text{ pF}, R_s = 5543\ \Omega. \tag{28}$$

Note that the loop bandwidth and phase margin have both been increased relative to the

target values of 1 MHz and 60 degrees, respectively, to approximately account for the effects of C_x and R_x which are neglected by the equations in [12].

If the values in (28) are used without modification in the SLF-PLL, the resulting loop bandwidth and phase margin are 960kHz and 44 degrees, respectively. Iteratively adjusting C_p and C_s and, to a lesser extent, R_s, as described above indicates that the SLF-PLL achieves the target loop bandwidth and phase margin with

$$C_p = 2.53 \text{ pF}, C_s = 328 \text{ pF}, R_s = 5408 \ \Omega. \quad (29)$$

Fig. 22 shows the phase noise transfer functions from the reference signal and the VCO for both the CLF-PLL and the SLF-PLL in the above design example. Although the corresponding transfer functions of the two PLLs are similar, some differences are evident. The difference between the transfer functions from the reference signal occur because C_p in (29) is less than half of C_p in (28). The difference between the transfer functions from the VCO occur because C_p+C_s in (29) is greater than that of (28). These differences are exaggerated because of the high loop bandwidth in this example. A lower loop bandwidth would result in less significant differences between the two sets of curves.

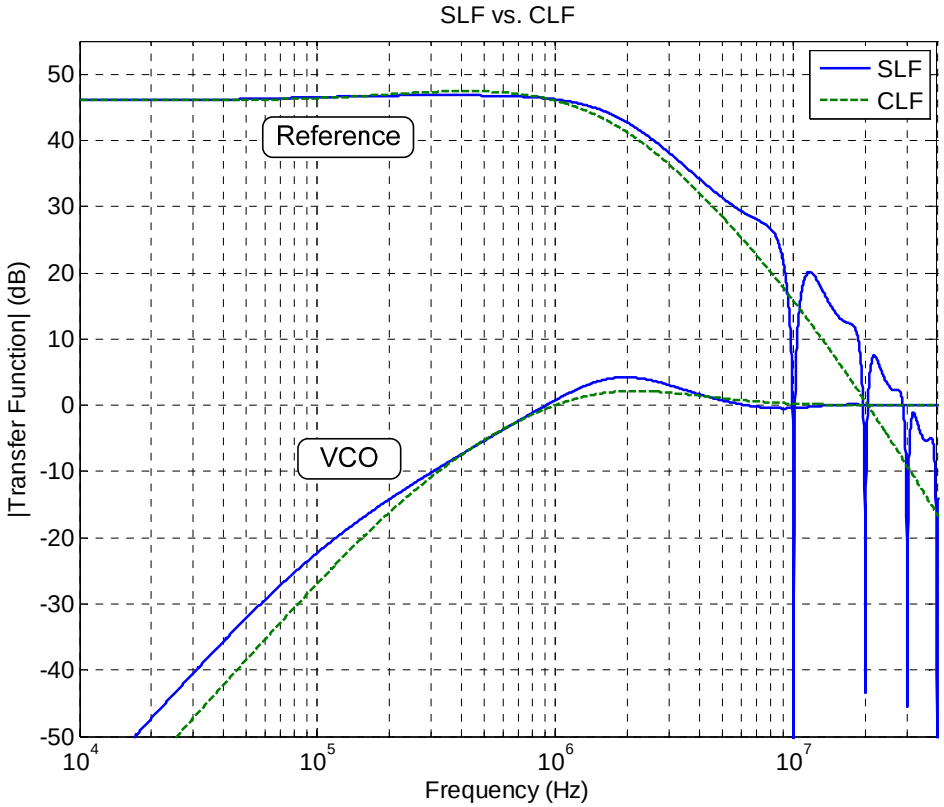


Fig. 22: Reference and VCO phase noise transfer functions of the original CLF-PLL and the SLF-PLL obtained via the synthesis procedure.

Fig. 23 shows a comparison of SLF-PLLs and CLF-PLLs using the same loop filter components. Two cases are examined: a low loop bandwidth (200kHz) design and a high loop bandwidth (1MHz) design. In each case, f_{ref} = 10Mhz, and for the SLF-PLL, $\{t_{op1}, t_{op2}, t_{cl}\}$ = {50 ns, 10 ns, 40 ns }. The results demonstrate that it is reasonable to use the component values derived for a CLF-PLL in a SLF-PLL when $\omega_{LBW}(t_{op1} + t_{op2})$ is small.

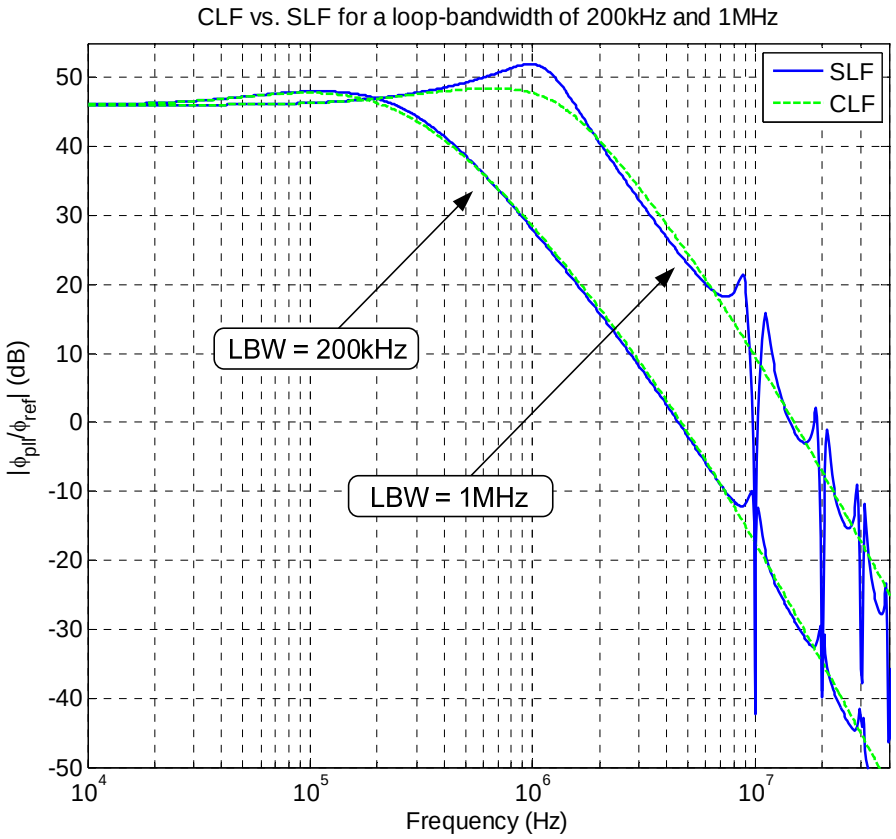


Fig. 23: Reference phase noise transfer functions of the CLF-PLL and SLF-PLL for low and high loop-bandwidth (LBW).

Derivation of the PLL Model

Background Results

Once the PLL is locked, the output of the VCO can be modeled as $v_{vco}(t) = A(t)\sin(\omega_{pll}t + \phi_{pll}(t))$, where $A(t)$ is some non-zero positive waveform, and $\omega_{pll} = 2\pi(N+\alpha)f_{ref}$ is the ideal output frequency of the PLL. The PLL's total phase noise is given by

$$\phi_{pll}(t) = \phi_{ctrl}(t) + \phi_{vco}(t) \tag{30}$$

where $\phi_{ctrl}(t)$ is the phase noise caused by deviations of the VCO control voltage from its mean value, and $\phi_{vco}(t)$ is the open-loop VCO phase noise, i.e., the phase noise that would remain if the VCO control voltage were held constant. Therefore,

$$\phi_{ctrl}(t) = K_{vco}\int_0^t \left(v_{ctrl}(\tau) - \bar{v}_{ctrl}\right) d\tau, \tag{31}$$

where K_{vco} is the VCO gain, and $\bar{v}_{ctrl}$ is the voltage for which the free-running frequency of the VCO would be exactly ω_{pll} in the absence of $\phi_{vco}(t)$.

Suppose the PLL is already locked at time $t = 0$. Let τ_n be the time of the nth rising edge of the reference signal, and let t_n be the corresponding rising edge of the divider output for $n = 0, 1, 2, \ldots$. As shown in [8], the net charge delivered to or removed from the loop filter by the charge pump during the nth reference period is

$$Q_{cp}[n] = \frac{I_{CP}T_{ref}}{2\pi}\left[\frac{2\pi\sum_{k=0}^{n}\varepsilon_q[k] - \phi_{ctrl}(t_n) - \phi_{vco}(t_n)}{N+\alpha} + \phi_{ref}(\tau_n)\right] \tag{32}$$

where I_{CP} is the magnitude of current pulses sourced and sunk by the charge pump, and $\varepsilon_q[k] = y[k] - \alpha$ is the quantization noise from the noise-shaping quantizer, and $\phi_{ref}(t)$ is the phase noise of the reference signal.

Derivation of the Single-Rate Version of the Model

The state of the SLF and $\phi_{ctrl}(t)$ at time t can be represented together as a vector, $\mathbf{x}(t)$, given by

$$\mathbf{x}(t) = \begin{bmatrix} q_T(t) & q_s(t) & q_x(t) & \phi_{ctrl}(t) \end{bmatrix}^T \tag{33}$$

where $q_T(t)$ is the total charge on all the loop filter capacitors, $q_s(t)$ is the charge on C_s, and $q_x(t)$ is the charge on C_x, all at time t. Let $\mathbf{x}[n]$, for $n = 0, 1, 2, \ldots$, be a sampled ver-

sion of $\mathbf{x}(t)$, defined as

$$\mathbf{x}[n] = \mathbf{x}\left(nT_{ref} + t_{op1}\right). \tag{34}$$

As proven in the next sub-section,

$$\mathbf{x}[n] = \mathbf{A}\mathbf{x}[n-1] + \mathbf{B}Q_{cp}[n] \tag{35}$$

and

$$\phi_{ctrl}\left(nT_{ref}\right) = \mathbf{C}\mathbf{x}[n-1] \tag{36}$$

where $\mathbf{B} = [1\ 0\ 0\ 0]^{T}$, and $\mathbf{A}$ and $\mathbf{C}$ are a 4×4 matrix and a 1×4 vector, respectively. The elements of $\mathbf{A}$ and $\mathbf{C}$ are fixed numbers that depend only on the loop filter component values, t_{op1}, t_{cl}, and t_{op2}.

In a practical PLL $\phi_{ctrl}(t)$ has a bandwidth that is less than a tenth of the reference frequency and $t_n \approx nT_{ref}$ for $n = 0, 1, 2, \ldots$, so it follows that

$$\phi_{ctrl}\left(t_n\right) \approx \phi_{ctrl}\left(nT_{ref}\right) \tag{37}$$

to a good approximation [8]. Consequently, (35) and (36) provide an expression for $\phi_{ctrl}(t_n)$ in terms $Q_{cp}[n]$.

Equations (35) and (36) are called state-space equations [13]. They indicate that $\phi_{ctrl}(nT_{ref})$ is the output of a linear time-invariant (LTI) discrete-time system with input $Q_{cp}[n]$. Appendix A provides equations with which to obtain **A**, **B**, and **C**, and well-known techniques are available to calculate the transfer function once **A**, **B**, and **C** are known. For example, the built-in Matlab command: ss2zp(**A**, **B**, **C**, 0, 1) can be used. The result is a four pole, three zero function which can be written as

$$K_{vco}F_{SLF}\left(z\right)\frac{z^{-1}}{1-z^{-1}} \tag{38}$$

where $F_{SLF}(z)$ has the form given by (13).

The single-rate model shown in Fig. 17 follows directly from and is a graphical representation of (30), (32), (37), and (38). The derivation above applies to fractional-N PLLs. However, by setting $\alpha = 0$ and $\varepsilon_q[n] = 0$, it also applies to integer-N PLLs.

The only approximation made in the model's derivation is (37). The standard model for conventional PLLs also relies on this approximation. However, in contrast to the model presented in this paper, the standard model for conventional PLLs relies on several additional approximations.

Derivation of the Single-Rate State-Space Equations

Without loss of generality $v_{ctrl}(t)$ can be taken to have zero mean so that (31) reduces to

$$\phi_{ctrl}(t) = K_{vco}\int_0^t v_{ctrl}(\tau)d\tau. \tag{39}$$

This simplifies the notation of the following derivation, yet it can be verified that it does not change the results of the derivation.

The SLF is a time-varying circuit, but during any time interval over which the switch either remains closed or remains open, it reduces to an LTI system. Furthermore, it follows from (39) that $\phi_{ctrl}(t)$ is an LTI function of the loop filter output.

Suppose that the switch is closed for the time interval from t_0 to $t_0+\Delta t$. As described in Section II, $I_{cp}(t)$ is zero when the switch is closed, so the total charge in the SLF remains unchanged during this time interval, i.e.,

$$q_T(t_0 + \Delta t) = q_T(t_0). \tag{40}$$

Given that the system is linear and time-invariant over the interval, well-known

linear systems theory results can be invoked (e.g., See Appendix B) to write the other elements of (33) at time $t_0+\Delta t$ in terms of their values at time t_0 as

$$\begin{aligned} q_s(t_0+\Delta t) &= r_{2,1}(\Delta t)\,q_T(t_0)+r_{2,2}(\Delta t)\,q_s(t_0) \\ &\quad +r_{2,3}(\Delta t)\,q_x(t_0), \end{aligned} \tag{41}$$

$$\begin{aligned} q_x(t_0+\Delta t) &= r_{3,1}(\Delta t)\,q_T(t_0)+r_{3,2}(\Delta t)\,q_s(t_0) \\ &\quad +r_{3,3}(\Delta t)\,q_x(t_0), \end{aligned} \tag{42}$$

$$\begin{aligned} \phi_{ctrl}(t_0+\Delta t) &= r_{3,1}(\Delta t)\,q_T(t_0)+r_{3,2}(\Delta t)\,q_s(t_0) \\ &\quad +r_{3,3}(\Delta t)\,q_x(t_0)+\phi_{ctrl}(t_0), \end{aligned} \tag{43}$$

where each $r_{i,j}(t)$ is an LTI system impulse response. For instance, $r_{3,2}(t)$ is the charge on C_x as a function of t in response to a Dirac delta function current impulse, $I(t) = \delta(t)$, injected across the terminals of C_s for the case in which the charge on each capacitor is zero for $t < 0$. In particular, the time-invariance property of (39) and the SLF over the time interval implies that the $r_{i,j}(\Delta t)$ factors in (41)-(43) depend only on the duration of the interval but not on the start time, t_0, of the interval.

Equations (40)-(43) can be written more compactly as

$$\mathbf{x}(t_0+\Delta t) = \mathbf{H}_{\mathbf{cl}}(\Delta t)\,\mathbf{x}(t_0), \tag{44}$$

where $\mathbf{x}(t)$ is given by (33), and

$$\mathbf{H}_{\mathbf{cl}}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ r_{2,1}(t) & r_{2,2}(t) & r_{2,3}(t) & 0 \\ r_{3,1}(t) & r_{3,2}(t) & r_{3,3}(t) & 0 \\ r_{4,1}(t) & r_{4,2}(t) & r_{4,3}(t) & 1 \end{bmatrix}. \tag{45}$$

Now suppose that the switch is open for the time interval from t_1 to $t_1+\Delta\tau$. As described in Section II, $I_{cp}(t)$ is not necessarily zero when the switch is open, so

$$q_T(t_1+\Delta\tau) = q_T(t_1)+\int_{t_1}^{t_1+\Delta\tau} I_{cp}(t)\,dt\,. \tag{46}$$

However, it follows from Fig. 15b that when the switch is open, $I_{cp}(t)$ does not af-

fect the other elements of (33). Therefore, equations for the other elements of (33) that apply to the case in which the switch is open can be obtained by exactly the same reasoning that led to Equations (41)-(43). These equations, along with (46) can be written as

$$\mathbf{x}(t_1+\Delta\tau)=\mathbf{H_{op}}(\Delta\tau)\mathbf{x}(t_1)+\mathbf{B}\int_{t_1}^{t_1+\Delta\tau} I_{cp}(t)\,dt\,, \tag{47}$$

where $\mathbf{B} = [1\ 0\ 0\ 0]^{\mathrm{T}}$,

$$\mathbf{H_{op}}(t)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ s_{2,1}(t) & s_{2,2}(t) & s_{2,3}(t) & 0 \\ s_{3,1}(t) & s_{3,2}(t) & s_{3,3}(t) & 0 \\ s_{4,1}(t) & s_{4,2}(t) & s_{4,3}(t) & 1 \end{bmatrix}, \tag{48}$$

and each $s_{i,j}(t)$ is an LTI system impulse response. Specifically, the expression for each $s_{i,j}(t)$ is identical to that of the corresponding $r_{i,j}(t)$ except with C_p replaced by $(1-\lambda)C_p$.

These results can be combined to prove (35) and (36). It follows from Fig. 15c, (34), (44) and (47) that

$$\mathbf{x}\left(nT_{ref}+t_{op1}+t_{cl}\right)=\mathbf{H_{cl}}(t_{cl})\mathbf{x}[n] \tag{49}$$

and

$$\begin{aligned}\mathbf{x}[n+1]=&\mathbf{H_{op}}\left(t_{op1}+t_{op2}\right)\mathbf{x}\left(nT_{ref}+t_{op1}+t_{cl}\right)\\ &+\mathbf{B}Q_{cp}[n].\end{aligned} \tag{50}$$

Substituting (49) into (50) yields (35) with

$$\mathbf{A}\triangleq\mathbf{H_{op}}\left(t_{op1}+t_{op2}\right)\mathbf{H_{cl}}(t_{cl}) \tag{51}$$

Similar reasoning leads to (36) with

$$\mathbf{C}\triangleq[0\ 0\ 0\ 1]\mathbf{H_{op}}\left(t_{op2}\right)\mathbf{H_{cl}}(t_{cl}) \tag{52}$$

Extension to the Multi-Rate Version of the Model

Nearly identical reasoning to that presented above which led to (36) and (52) also

implies that

$$\phi_{ctrl}\left((n-1)T_{ref}+t_{op1}+\Delta t\right)=\begin{bmatrix}0&0&0&1\end{bmatrix}\mathbf{D}(\Delta t)\,\mathbf{x}[n-1] \tag{53}$$

for Δt in the range $0 < \Delta t \le T_{ref}$., where

$$\mathbf{D}(\Delta t)=\begin{cases}\mathbf{H}_{\mathbf{cl}}(\Delta t), & \text{if } 0<\Delta t\le t_{cl}\\ \mathbf{H}_{\mathbf{op}}(\Delta t-t_{cl})\,\mathbf{H}_{\mathbf{cl}}(t_{cl}), & \text{if } t_{cl}<\Delta t\le T_{ref}\end{cases} \tag{54}$$

Therefore, (53) and (54) can be used to obtain any sample of $\phi_{ctrl}(t)$ in the range $(n-1)T_{ref}+t_{op1} < t \le nT_{ref}+t_{op1}$.

In particular,

$$\phi_{ctrl}\left(\left(n+\frac{i-d}{L}\right)T_{ref}\right)=\mathbf{C}_i\,\mathbf{x}[n-1], \tag{55}$$

for $i = 0, 2, \ldots L-1$, where

$$\mathbf{C}_i=\mathbf{D}\left(t_{cl}+t_{op2}+\frac{i-d}{L}T_{ref}\right), \tag{56}$$

and d is given by (21). For each value of i, (55) defines an LTI filter with input $Q_{cp}[n]$ and output samples given by (55) for $n = 0, 1, 2, \ldots$. The transfer function of the ith such filter has the form

$$K_{vco}F_{SLF,i}(z)\frac{z^{-1}}{1-z^{-1}} \tag{57}$$

where $F_{SLF,i}(z)$, is obtained in the same way that $F_{SLF}(z)$ is obtained from **A**, **B**, and **C**, as described in Section III-B, except with **C** replaced by $\mathbf{C}_i$. Note, in particular, that $\mathbf{C}_d = \mathbf{C}$ by definition, so $F_{SLF,d}(z) = F_{SLF}(z)$.

It follows that the SLF and VCO can be modeled as shown Fig. 24a. With the Noble identity for up-sampling, this can be redrawn as shown in Fig. 24b which is equivalent to the system shown in Fig. 17b, with $G_{SLF}(z)$ as given by (23).

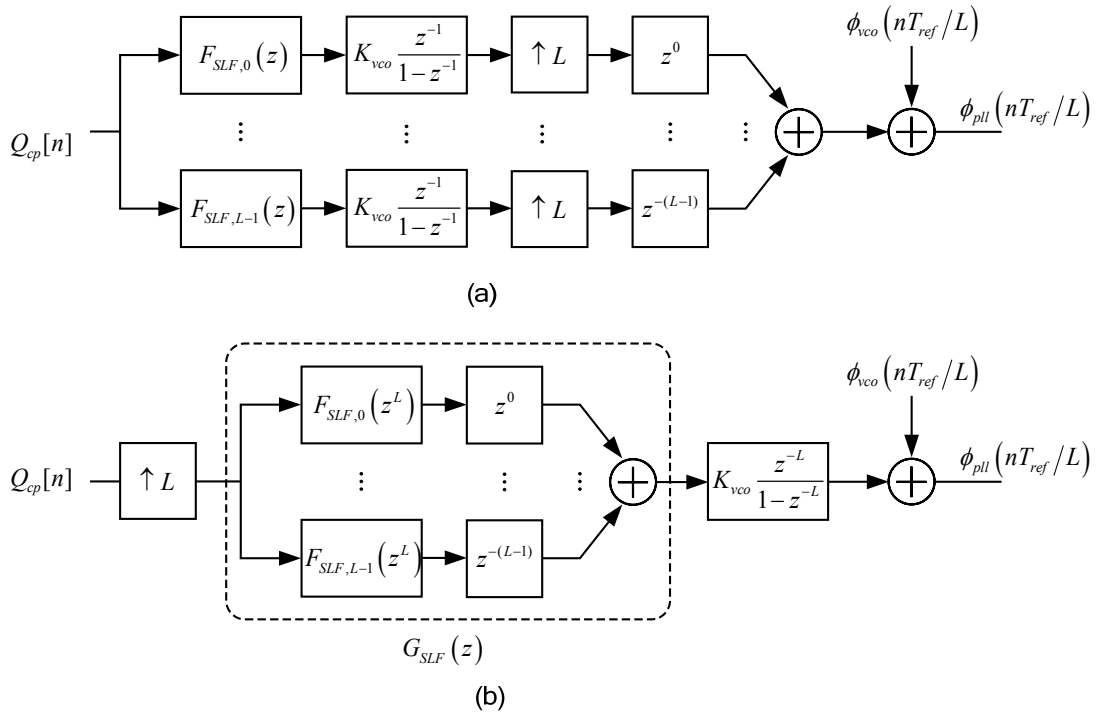


Fig. 24: Model for the multi-rate SLF and transformation to $G_{SLF}(z)$.

APPENDIX A

This appendix describes all the calculations necessary to obtain $F_{SLF}(z)$ and $G_{SLF}(z)$ starting from the values of T_{ref}, C_p, C_s, C_x, R_s, R_x, λ, t_{op1}, t_{cl}, and t_{op2}. The calculations are most easily implemented via a computer calculation script executed by a software tool such as Matlab. Therefore, the calculations are listed below in the form of specific steps that must be executed by such a calculation script. Steps 1-6 below specify the calculation details of $F_{SLF}(z)$ in (13). Steps 1-5 followed by Steps 7-8 below specify calculation details of $G_{SLF}(z)$ in (23).

1. Define the following functions of the variables C_p, C_s, R_s, C_x, R_x, and Γ:

$$p_{1,\Gamma} = \frac{-b_\Gamma + \sqrt{{b_\Gamma}^2 - 4a_\Gamma c_\Gamma}}{2a_\Gamma}, \quad p_{1,\Gamma} = \frac{-b_\Gamma - \sqrt{{b_\Gamma}^2 - 4a_\Gamma c_\Gamma}}{2a_\Gamma}$$

where

$$\begin{aligned} a_\Gamma &= C_p(1-\Gamma)R_s R_x C_s C_x \\ b_\Gamma &= \left(C_p(1-\Gamma)+C_x\right)R_s C_s + \left(C_p(1-\Gamma)+C_s\right)R_x C_x \\ c_\Gamma &= C_p(1-\Gamma)+C_s+C_x. \end{aligned}$$

2. Define the following functions of the variables C_u, C_v, R_u, R_v, Γ, and the functions defined in Step 1:

$$\begin{aligned} E_{u,v,\Gamma} &= (1-\Gamma)C_u\left(p_{1,\Gamma} - p_{2,\Gamma}\right) \\ F_{u,v,\Gamma} &= (1-\Gamma)C_u p_{2,\Gamma}\left(1 + p_{1,\Gamma}R_v C_v\right) \\ G_{u,v,\Gamma} &= -(1-\Gamma)C_u p_{1,\Gamma}\left(1 + p_{2,\Gamma}R_v C_v\right) \\ I_{u,v,\Gamma} &= \Gamma C_u\left(p_{1,\Gamma} - p_{2,\Gamma}\right) \\ J_{u,v,\Gamma} &= C_u p_{2,\Gamma}\left(\left(1 + p_{1,\Gamma}R_v C_v\right)\left(\Gamma + p_{1,\Gamma}R_u C_p(1-\Gamma)\right) + p_{1,\Gamma}R_u C_v\right) \\ L_{u,v,\Gamma} &= -C_u p_{1,\Gamma}\left(\left(1 + p_{2,\Gamma}R_v C_v\right)\left(\Gamma + p_{2,\Gamma}R_u C_p(1-\Gamma)\right) + p_{2,\Gamma}R_u C_v\right) \\ P_{u,v,\Gamma} &= C_u p_{2,\Gamma}\left(\Gamma\left(1 + p_{1,\Gamma}R_v C_v\right) - p_{1,\Gamma}R_v C_v\right) \end{aligned}$$

and

$$Q_{u,v,\Gamma} = -C_u p_{1,\Gamma}\left(\Gamma\left(1 + p_{2,\Gamma}R_v C_v\right) - p_{2,\Gamma}R_v C_v\right).$$

3. Define the following 4×4 matrix function of the variable t and the functions defined in Steps 1 and 2:

$$\mathbf{H}_\Gamma(t) = K_\Gamma \begin{bmatrix} 1/K_\Gamma & 0 & 0 & 0 \\ h_{2,1,\Gamma}(t) & h_{2,2,\Gamma}(t) & h_{2,3,\Gamma}(t) & 0 \\ h_{3,1,\Gamma}(t) & h_{3,2,\Gamma}(t) & h_{3,3,\Gamma}(t) & 0 \\ h_{4,1,\Gamma}(t) & h_{4,2,\Gamma}(t) & h_{4,3,\Gamma}(t) & 1/K_\Gamma \end{bmatrix} \tag{58}$$

where

$$\begin{aligned} K_\Gamma &= -\frac{1}{\left(C_p(1-\Gamma)+C_s+C_x\right)\left(p_{1,\Gamma} - p_{2,\Gamma}\right)} \\ h_{2,1,\Gamma}(t) &= E_{s,x,\Gamma} + F_{s,x,\Gamma}e^{p_{1,\Gamma}t} + G_{s,x,\Gamma}e^{p_{2,\Gamma}t} \end{aligned}$$

$$h_{2,2,\Gamma}(t) = I_{s,x,\Gamma} + J_{s,x,\Gamma}e^{p_{1,\Gamma}t} + L_{s,x,\Gamma}e^{p_{2,\Gamma}t}$$
$$h_{2,3,\Gamma}(t) = I_{s,x,\Gamma} + P_{s,x,\Gamma}e^{p_{1,\Gamma}t} + Q_{s,x,\Gamma}e^{p_{2,\Gamma}t}$$
$$h_{3,1,\Gamma}(t) = E_{x,s,\Gamma} + F_{x,s,\Gamma}e^{p_{1,\Gamma}t} + G_{x,s,\Gamma}e^{p_{2,\Gamma}t}$$
$$h_{3,2,\Gamma}(t) = I_{x,s,\Gamma} + P_{x,s,\Gamma}e^{p_{1,\Gamma}t} + Q_{x,s,\Gamma}e^{p_{2,\Gamma}t}$$
$$h_{3,3,\Gamma}(t) = I_{x,s,\Gamma} + J_{x,s,\Gamma}e^{p_{1,\Gamma}t} + L_{x,s,\Gamma}e^{p_{2,\Gamma}t}$$
$$h_{4,1,\Gamma}(t) = \frac{1}{C_x}\left(E_{x,s,\Gamma}t + \frac{F_{x,s,\Gamma}}{p_{1,\Gamma}}\left(e^{p_{1,\Gamma}t} - 1\right) + \frac{G_{x,s,\Gamma}}{p_{2,\Gamma}}\left(e^{p_{2,\Gamma}t} - 1\right)\right)$$
$$h_{4,2,\Gamma}(t) = \frac{1}{C_x}\left(I_{x,s,\Gamma}t + \frac{P_{x,s,\Gamma}}{p_{1,\Gamma}}\left(e^{p_{1,\Gamma}t} - 1\right) + \frac{Q_{x,s,\Gamma}}{p_{2,\Gamma}}\left(e^{p_{2,\Gamma}t} - 1\right)\right)$$

and

$$h_{4,3,\Gamma}(t) = \frac{1}{C_x}\left(I_{x,s,\Gamma}t + \frac{J_{x,s,\Gamma}}{p_{1,\Gamma}}\left(e^{p_{1,\Gamma}t} - 1\right) + \frac{L_{x,s,\Gamma}}{p_{2,\Gamma}}\left(e^{p_{2,\Gamma}t} - 1\right)\right).$$

Therefore, with the functions defined in Steps 1 and 2 substituted into the functions defined in Step 3, $\mathbf{H}_\Gamma(t)$ is a matrix function of the variables C_p, C_s, R_s, C_x, R_x, Γ, and t.

4. Define the following 4×4 matrix functions of C_p, C_s, R_s, C_x, R_x, and t using the matrix function defined in Step 4:

$$\mathbf{H_{op}}(t) = \mathbf{H}_\Gamma(t)\big|_{\Gamma=\lambda} \quad \text{and} \quad \mathbf{H_{cl}}(t) = \mathbf{H}_\Gamma(t)\big|_{\Gamma=0}$$

5. With numerical values for C_p, C_s, C_x, R_s, R_x, λ, t_{op1}, t_{cl}, and t_{op2} and the matrix functions from Step 4, calculate the following 4×4 matrix of numbers:

$$\mathbf{A} = \mathbf{H_{op}}\left(t_{op1} + t_{op2}\right)\mathbf{H_{cl}}\left(t_{cl}\right)$$

and the following 1×4 vector of numbers:

$$\mathbf{C} = \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}\mathbf{H_{op}}\left(t_{op2}\right)\mathbf{H_{cl}}\left(t_{cl}\right).$$

6. Calculate the poles, zeros, and scale factor of $F_{SLF}(z)$ in (13) from its state-space repre-

sentation as specified by the matrix **A** and vector **C** from Step 5 and **B** = $[1\ 0\ 0\ 0]^{\mathrm{T}}$. For example, this can be done using the Matlab function:

ss2zp(**A**, **B**, **C**, 0, 1)

7. With numerical values for C_p, C_s, C_x, R_s, R_x, λ, T_{ref}, t_{op2}, and t_{cl}, and the matrix functions from Step 4, calculate the following 1×4 vectors of numbers for $i = 0, 1, \ldots L-1$:

$$\mathbf{C}_i = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{D}\left(t_{cl} + t_{op2} + \frac{i-d}{L} T_{ref} \right)$$

where

$$\mathbf{D}(\Delta t) = \begin{cases} \mathbf{H_{cl}}(\Delta t), & \text{if } 0 < \Delta t \le t_{cl} \\ \mathbf{H_{op}}(\Delta t - t_{cl})\mathbf{H_{cl}}(t_{cl}), & \text{if } t_{cl} < \Delta t \le T_{ref} \end{cases}$$

and

$$d = \left\lfloor \frac{t_{cl} + t_{op2}}{T_{ref}} L \right\rfloor.$$

8. For each $i = 0, 1, \ldots L-1$, calculate the poles, zeros, and scale factor of each function $F_{SLF,i}(z)$ (which has the same form as (13)) from its state-space representation as specified by the matrix **A** from Step 5, the vector and vector $\mathbf{C}_i$ from Step 7, and **B** = $[1\ 0\ 0\ 0]^{\mathrm{T}}$. For example, this can be done using the Matlab function:

ss2zp(**A**, **B**, $\mathbf{C}_i$, 0, 1)

Substitute the L resulting $F_{SLF,i}(z)$ functions into (23) to obtain $G_{SLF}(z)$.

APPENDIX B

This appendix derives (42) and (43) to find expressions for the third and fourth

rows elements of $\mathbf{H_{cl}}(t)$ in (45). The derivation of (41) is not presented because it is almost identical to that of (42).

Let C_{pA} denote the capacitor to the left of the switch, and C_{pB} denote the capacitor to the immediate right of the switch in Fig. 15b, and let their respective charges at t_0 be $q_{pA}(t_0)$ and $q_{pB}(t_0)$. Then the charge on $q_x(t_0+\Delta t)$ can be written as

$$\begin{aligned} q_x(t_0+\Delta t) &= h_1(\Delta t)\,q_{p_A}(t_0) + h_2(\Delta t)\,q_{p_B}(t_0) \\ &\quad + h_3(\Delta t)\,q_s(t_0) + h_4(\Delta t)\,q_x(t_0) \end{aligned} \tag{59}$$

where $h_1(\Delta t)$ is the charge transfer function from capacitor C_{pA} to capacitor C_x, $h_2(\Delta t)$ is the charge transfer function capacitor C_{pB} to capacitor C_x, $h_3(\Delta t)$ is the charge transfer function from capacitor C_s to capacitor C_x, and $h_4(\Delta t)$ is the charge transfer function from capacitor C_x to capacitor C_x, all over a time interval of Δt. These charges can be expressed in terms of the elements of $\mathbf{x}(t_0)$ as

$$\begin{aligned} q_{p_A}(t_0) &= \left[q_T(t_0) - q_s(t_0) - q_x(t_0)\right]\lambda \\ q_{p_B}(t_0) &= \left[q_T(t_0) - q_s(t_0) - q_x(t_0)\right](1-\lambda) \end{aligned} \tag{60}$$

Substituting (60) into (59) leads to (42) with

$$\begin{aligned} r_{3,1}(\Delta t) &= \lambda h_1(\Delta t) + (1-\lambda)h_2(\Delta t) \\ r_{3,2}(\Delta t) &= h_3(\Delta t) - \lambda h_1(\Delta t) - (1-\lambda)h_2(\Delta t). \\ r_{3,3}(\Delta t) &= h_4(\Delta t) - \lambda h_1(\Delta t) - (1-\lambda)h_2(\Delta t) \end{aligned} \tag{61}$$

The $h_i(\Delta t)$ functions can be found by computing the inverse Laplace transform of the s-domain charge transfer function from any one of the capacitors to any other and then evaluating the result at $t = \Delta t$. For example, suppose the switch is closed (λ=0) and consider $h_3(\Delta t)$. In the s-domain, the charge on capacitor C_x due to charge on capacitor C_s is given by

$$\begin{aligned} Q_x(s) &= C_x V_x(s) = C_x \frac{V_x(s)}{I_s(s)} I_s(s) \\ &= \frac{C_x}{\left(C_p + C_s + C_x\right) s \left(1 - \frac{s}{p_1}\right)\left(1 - \frac{s}{p_2}\right)} q_s(0) \end{aligned} \tag{62}$$

where V_x and I_x are the s-domain voltage and current associated with capacitor C_x, and V_s and I_s are those for capacitor C_s, p_1 and p_2 are the two non-DC poles of $V_x(s)/I_s(s)$, and $q_s(0) = C_s v_s(0)$ represents the initial charge on capacitor C_s.

Taking the inverse Laplace transform of (62) yields

$$\begin{aligned} q_x(t) &= L^{-1}\left\{Q_x(s)\right\} \\ &= h_3(t) q_s(0) \end{aligned} \tag{63}$$

where

$$K = \frac{1}{\left(C_p + C_s + C_x\right)\left(p_1 - p_2\right)}. \tag{64}$$

and

$$h_3(t) = K C_x \left[\left(p_1 - p_2\right) + p_2 e^{p_1 t} + p_1 e^{p_2 t}\right] \tag{65}$$

Repeating this calculation for all the $h_i(\Delta t)$ functions and substituting the results into (61) leads to the third row of (45).

Now consider the transfer functions associated with the state variable, $\phi_{ctrl}(t)$. The VCO integrates the voltage on capacitor C_x. Thus,

$$r_{4,i}(\Delta t) = \frac{1}{C_x} \int_0^{\Delta t} r_{3,i}(\tau)\, d\tau. \tag{66}$$

ACKNOWLEDGEMENT

Chapter 2, in full, has been submitted for review to the IEEE Transactions on Circuits and Systems I: Regular papers. The dissertation author is the primary investigator

and author of this paper. Professor Ian Galton supervised the research which forms the basis of this paper.

REFERENCES

[1] B. Razavi, *Phase-Locking in High-Performance Systems: From Devices to Architectures,* Wiley-IEEE Press, 2003.

[2] B. Zhang, P. Allen, and J. Huard, "A fast switching PLL frequency synthesizer with an on-chip passive discrete-time loop filter in 0.25-μm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 38, pp. 855-865, Jun. 2003.

[3] K. Wang, A. Swaminathan, and I. Galton, "Spurious Tone Suppression Techniques Applied to a Wide-Bandwidth 2.4 GHz Fractional-N PLL," *Solid-State Circuits, IEEE Journal of*, vol. 43, pp. 2787-2797, Dec. 2008.

[4] M. Cassia, P. Shah, and E. Bruun, "Analytical model and behavioral simulation approach for a ΣΔ fractional-N synthesizer employing a sample-hold element," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 50, pp. 850-859, Nov. 2003.

[5] S. E. Meninger, 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.

[6] L. Liu and B. Li, "Phase noise cancellation for a ΣΔ fractional-N PLL employing a sample-and-hold element," *Microwave Conference Proceedings, 2005. APMC 2005. Asia-Pacific Conference Proceedings*, 2005.

[7] 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," *Solid-State Circuits, IEEE Journal of*, vol. 39, pp. 1446-1454, Sep. 2004.

[8] F. M. Gardner, "Charge-Pump Phase-Lock Loops," *IEEE Transactions on Communications*, vol. COM-28, no. 11, pp. 1849-1858., Nov. 1980.

[9] I. Galton, "Delta-Sigma Fractional-*N* Phase-Locked Loops," in Phase-Locking in High-Performance Systems: From Devices to Architectures, Behzad Razavi, Ed., John Wiley & Sons, 2003.

[10] J. P. Hein, J. W. Scott, "z-Domain Model for Discrete-Time PLL's," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 11, pp l393-1400, November 1988.

[11] J. A. Crawford, *Frequency Synthesizer Handbook*, Artech House, Inc., 1994.

[12] J. Craninckx and M.S. Steyaert, "A fully integrated CMOS DCS-1800 frequency synthesizer," *Solid-State Circuits, IEEE Journal of*, vol. 33, pp. 2054 -2065, Dec. 1998.

[13] J.G. Proakis and D.K. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, Prentice Hall, 2006.

Chapter III:

A 2.5GHz Realigned PLL with Spur Calibration

Abstract— This chapter discusses a 2.5GHz realigned PLL (RAPLL) in 65nm CMOS. The RAPLL utilizes a relaxation oscillator to avoid the timing issues associated with ring oscillator based RAPLLs. In addition, the RAPLL presented here includes a calibration loop to suppress the realignment spur. Simulations indicate that the spur can be reduced to -60dBc. The IC draws 6mA out of a 1V supply and has an active area of 0.2mm^2.

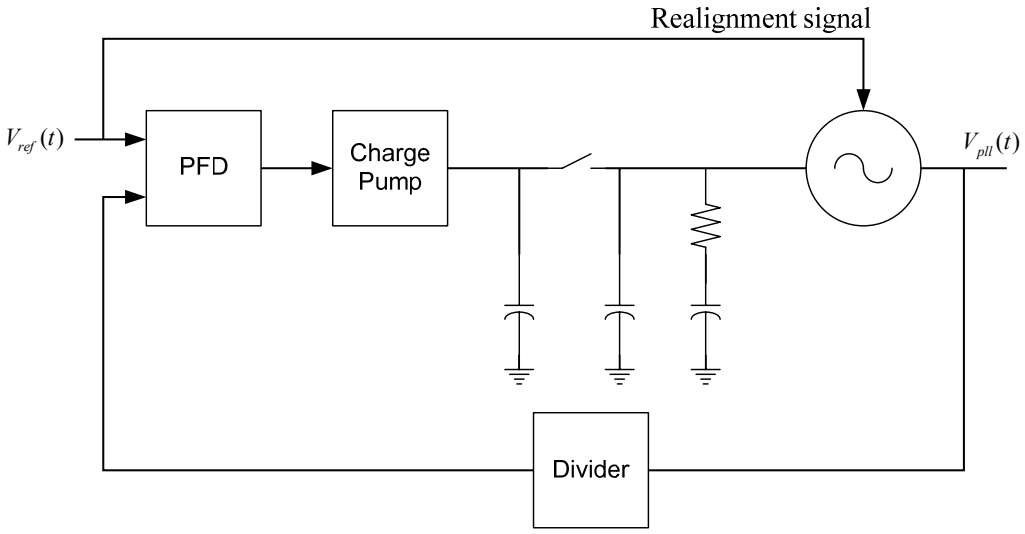


Fig. 25: Block diagram of a conventional RAPLL.

A block diagram of a conventional realigning PLL (RAPLL) is shown in Fig. 25. It consists of a phase-frequency detector (PFD), a charge pump (CP), a lowpass loop filter (LF), a voltage controlled oscillator (VCO), and an integer divider. $V_{ref}(t)$ is a refer-

ence signal with a frequency of f_{ref} while $V_{pll}(t)$ is an output signal with a frequency of f_{pll}. The PFD compares the rising edges of the reference with that of the divider and together with the charge pump, injects a charge proportional to the time difference between these edges into the loop filter. This causes the VCO to oscillator either faster or slower. When locked, the net charge injected is zero, and $f_{pll} = Nf_{ref}$. Finally, a feed-forward path allows the reference to periodically reset or realign the VCO. Since a reference edge is available at a rate of f_{ref}, the realignment frequency, f_{RA}, is f_{ref}.

The VCO integrates noise generated by its own circuitry and noise appearing at $V_{tune}(t)$ and hence, its noise has a power spectral density (PSD) with a *1/f* characteristic. Realigning resets the accumulated noise thereby flattening the PSD up to approximately f_{ref} [2]. One drawback is that while the noise of the VCO is suppressed, it is replaced with the noise in the realignment signal, namely reference noise. Moreover, since the realignment path bypasses the loop filter, this noise is not filtered.

Non-*LC* based VCOs such as ring oscillators typically exhibit very high phase noise relative to *LC* based VCOs. As a result, the suppression of the VCO noise, in exchange for unfiltered reference noise, is often a good tradeoff. Thus, the majority of the published RAPLLs utilize ring oscillators [1]-[4], though there is one published work utilizing an *LC* based VCO [5].

Fig. 26 shows the phase noise of a RAPLL compared with a non-realigning PLL (NRAPLL). Notice that the phase noise is suppressed for frequencies between 100KHz and 1MHz when compared to the non-realigning design. For

Fig. 26, the jitter is $1.4ps_{rms}$ for the RAPLL and $5ps_{rms}$ for an equivalent NRAPLL.

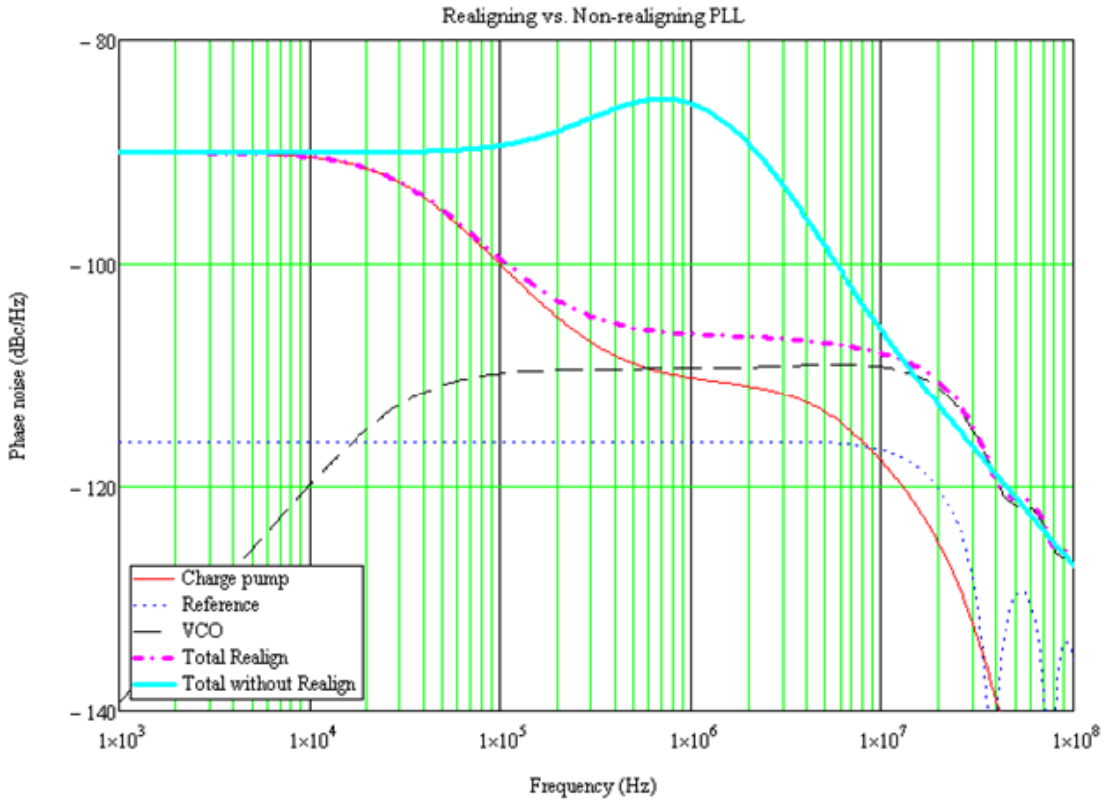


Fig. 26: Comparison between the phase noise of a realigning PLL and a traditional non-realigning PLL.

In [1]-[4], realignment is accomplished by periodically stopping the VCO and then using the reference edge to trigger its restart. The divider, which is predictive of the reference edge when the RAPLL is locked, triggers the stopping. The time marked by the stopping of the ring to the restarting of ring represents a window during which the realignment must take place. The timing of this window is critical. The window must be opened no sooner than half a VCO cycle before the realignment edge and closed no later than half a VCO cycle after; otherwise, the VCO will miss one oscillation cycle. For high frequency VCOs, this requires careful design [3][4]. Furthermore, the presence of the

window complicates the initial locking of the RAPLL during which time the divider is not predictive of the reference, and hence, realignment edge [3].

RAPLL's also exhibit a realignment spur at f_{RA}. Ideally, the realignment edge occurs at the exact time at which the VCO edge would have occurred in the absence of noise. However, in the presence of circuit mismatches, this alignment is difficult to achieve. Since $f_{RA} = f_{ref}$, the realignment spur occurs at the f_{ref}. Typically, the realignment spur is between -25 to -35dBc [2][3]. A method is given in [1] to address some, but not all of these issues.

In this work, two techniques are introduced. One addresses the challenge of the realignment window by proposing an oscillator without a window; while the other is a calibration scheme to properly align the phase of the realignment edge with that of the VCO edge it is meant to replace.

PROPOSED DESIGN

Fig. 27 shows a block diagram of the proposed RAPLL. It differs from Fig. 25 in two ways. First, a calibration loop is added. The calibration path monitors the output of the PFD and adjusts a variable delay element inserted prior to the reference input of the PFD. The variable delay is adjusted so that the realignment edge is aligned with that of the VCO edge it is to replace. This loop is discussed later. Secondly, the VCO is controlled by two paths rather than one. This is a consequence of modifying the VCO to allow for windowless realignment and will be discussed below.

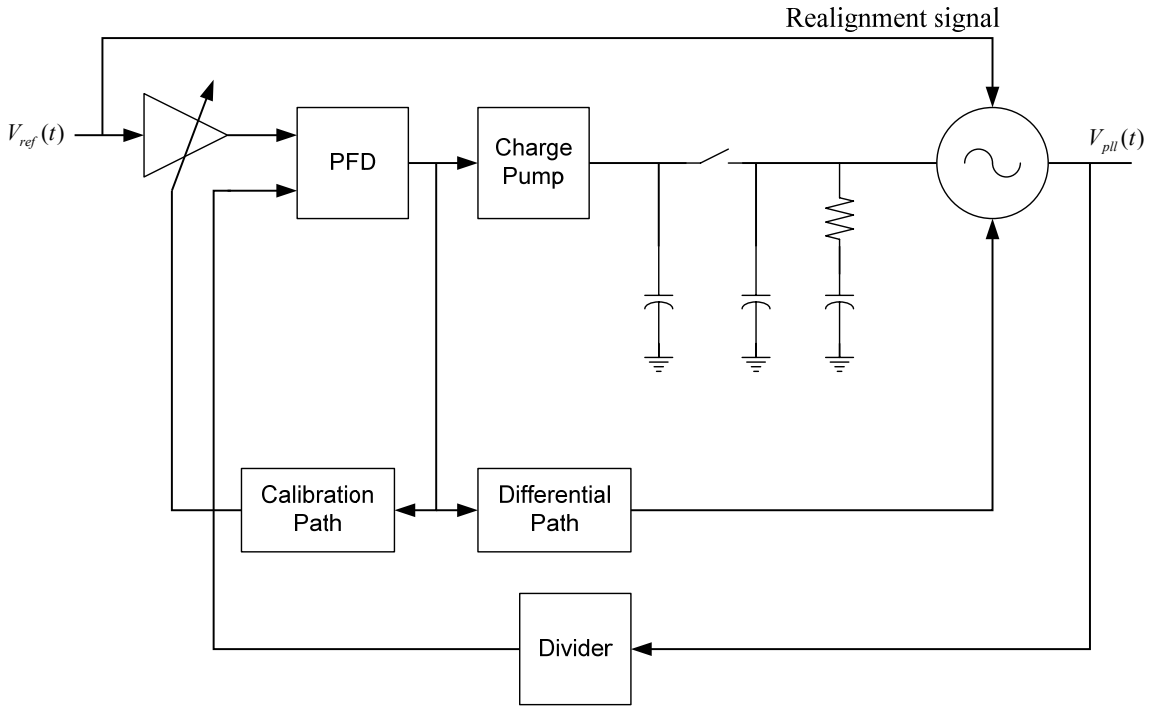


Fig. 27: Block diagram of the proposed RAPLL.

Relaxation oscillator as a VCO

The relaxation oscillator shown in Fig. 28 is used as the VCO.

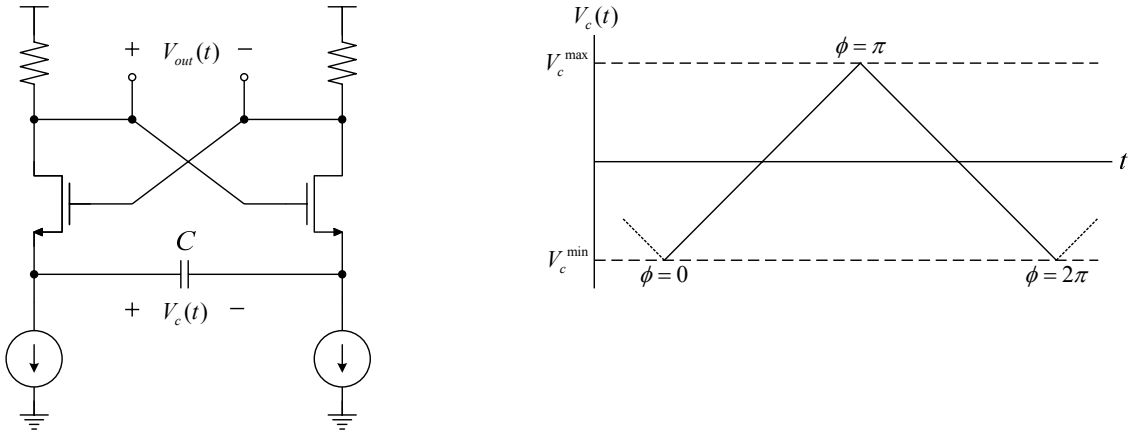


Fig. 28: Schematic of a relaxation oscillator and the voltage across the timing capacitor.

Unlike a traditional ring oscillator, a relaxation oscillator has only one timing

element, capacitor C, whose voltage, V_c, encodes the phase of the oscillator. Windowless realignment is achieved if V_c (or phase of the oscillator) can be reset to V_{RA} regardless of what the voltage across C (or the phase of the oscillator) is prior to realignment. As illustrated in Fig. 29, the phase is either delayed or advanced depending on whether the realignment occurs when $V_c < V_{RA}$ or when $V_c > V_{RA}$. The phase of the oscillator is not affected if the realignment edge occurs when $V_c = V_{RA}$. After resetting, the oscillation proceeds as usual.

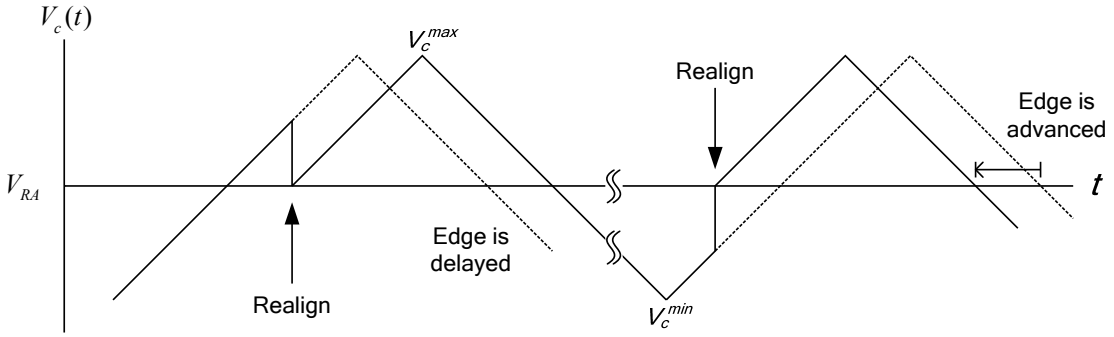


Fig. 29: Illustration of realignment in a relaxation oscillator.

Optimally, V_{RA} is chosen to be halfway between V_c^{min} and V_c^{max}, which in the design of Fig. 28 is zero. Given this choice, two stable phases exist within a single cycle, $\pi/2$ and $3\pi/2$ where $V_c = V_{RA} = 0$. Both are acceptable realignment points and under lock, the oscillator will settle to one of them. While it is possible for a locked oscillator to be kicked from one stable point to the other, it is unlikely, as it would require a disturbance in the oscillator greater than $T_{vco}/4$.

The resetting is accomplished by slightly modifying the relaxation oscillator (Fig. 30a). Here, the timing capacitor is duplicated and a shorting switch is added across each

capacitor. During normal operation, only one of the switches is closed. Realignment is achieved by toggling which switch is closed. When a switch is closed, its accompanying capacitor is shorted and hence the voltage across it is zero. Furthermore, it does not contribute to the oscillation of the VCO. When the switch is then toggled, the previously shorted capacitor now becomes part of the oscillator; moreover, since this capacitor was previously shorted, the oscillation always proceeds from a specific phase, namely the one corresponding to $V_c = 0$. Hence, realignment is achieved. Note that realignment can, at most, change the VCO phase by $\pi/2$ or $T_{vco}/4$. As a result, realignment does not significantly affect the locking dynamics of the RAPLL.

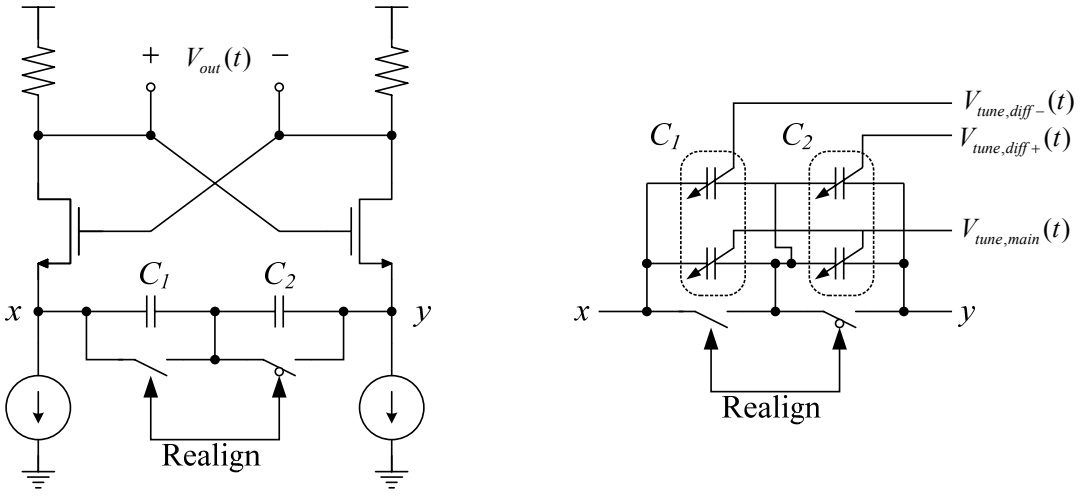


Fig. 30: Modified relaxation oscillator to support realignment.

Using a relaxation oscillator in this manner, however, introduces a new problem, as there are now two oscillating modes. Due to mismatches between C_1 and C_2, for a given tuning voltage, there are now two oscillating frequencies, $f_{osc,1}$ and $f_{osc,2}$. The first oc-

curs when capacitor C_1 is used, while the other is when C_2 is used.

This is addressed by having two tuning voltages (Fig. 30b), one to control the common mode and one to control the differential mode values of C_1 and C_2. The main path provides a tuning voltage that adjusts a pair of varactors to $(C_1+C_2)/2$. The differential path provides a voltage that tunes another pair of varactors; in this case, one of them is tuned to $(C_1-C_2)/2$ while the other is tuned to $-(C_1-C_2)/2$. As a result, the oscillating frequency of the two modes, $f_{osc,1}$ and $f_{osc,2}$ can be set independently.

The main path, controlling the common mode, is a traditional design consisting of a charge pump and a loop filter; however, the differential path is new. A block diagram of this path is shown in Fig. 31.

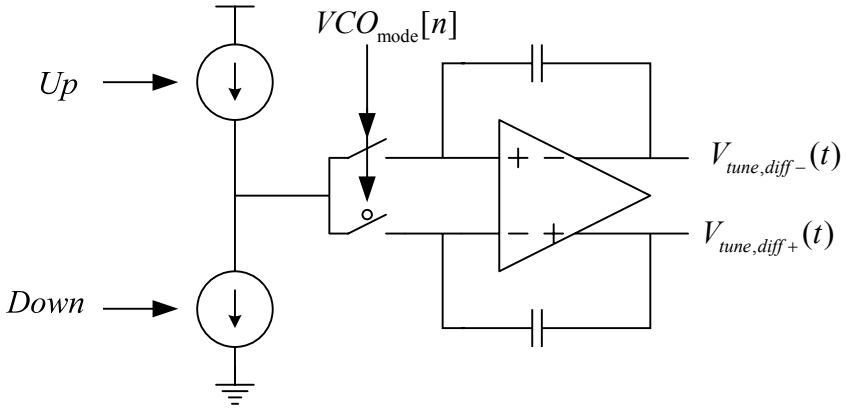


Fig. 31: Block diagram of the differential control path for the VCO.

The differential path consists of a local charge pump, a set of switches and an integrator; a common mode feedback circuit to stabilize the common mode input voltage of the amplifier is not shown. The switches direct the output of the local charge pump either to the positive or negative input terminal of the integrator, depending on whether the

VCO is oscillating with capacitor C_1 or with C_2. This information is conveyed in the digital signal, $VCO_{mode}[n]$. Note that $VCO_{mode}[n]$ toggles whenever there is a realignment.

If the differential loop is converged, then $f_{osc,1} = f_{osc,2}$. As a result, the net charge delivered by the charge pump to the amplifier is the same regardless of which mode the VCO is oscillating in. Since the amplifier rejects common mode signals, the output of the amplifier will not be affected. However, if $f_{osc,1} \neq f_{osc,2}$, then a different amount of charge will be applied to the positive input terminal of the amplifier versus the negative, thereby causing an adjustment of $Vtune_{diff+}$ and $Vtune_{diff-}$. This adjustment continues until $f_{osc,1} = f_{osc,2}$.

Calibration loop for the realignment spur

The second problem in conventional realigned PLLs is the phase alignment between the realignment edge and the VCO edge it is supposed to replace. A calibration loop is added to measure and adjust the alignment of these two edges. The details of this path are shown in Fig. 32; notice that this is identical to the differential path mentioned above, except that a new digital signal, $s_{RA}[n]$ is used rather than $VCO_{mode}[n]$. Typically, realignment occurs once every reference period. In this design, occasionally, but randomly, one of these realignments is dropped. This occurs very infrequently so that the realignment frequency is still approximately f_{ref}. $s_{RA}[n]$ is high when a realignment has occurred in the previous reference period and low when it has not.

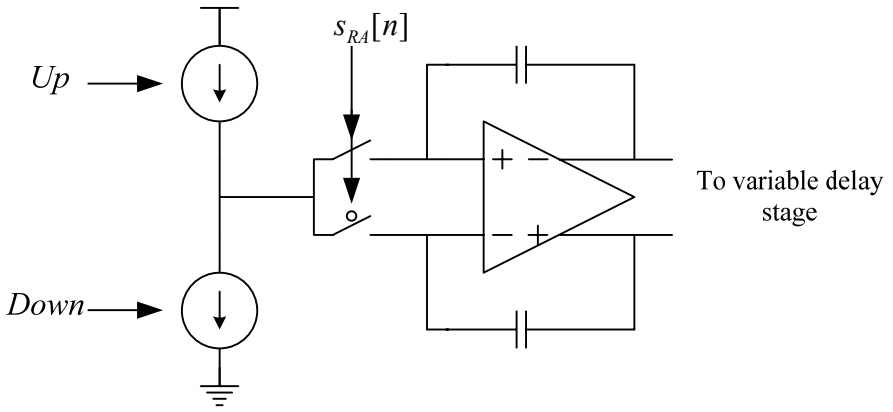


Fig. 32: Block diagram of the calibration path.

The calibration loop measures the differences in timing between reference periods that have experienced realignment and those that have not. A variable delay in the path of the reference signal is adjusted until the timing difference is driven to zero.

IMPLEMENTATION DETAILS

Calibration using a charge pump and integrator

A detailed diagram of the CP and integrator described above is shown in Fig. 33. It consists of a charge pump, a set of switches, and an offset cancelling integrator; a common mode feedback circuit to set the common mode input voltage of the amplifier is not shown. The switches are controlled by a two-level logic signal wherein the switch is closed when the signal is one and open when the signal is zero. These switches can be divided into two groups. The first group steers the output of the charge pump either to the positive or negative input terminals of the integrator depending on a sequence $s[n]$. The second group control whether the amplifier is integrating or measuring its offset. Finally,

a signal, *skip*[*n*], is used to enable or disable the charge pump.

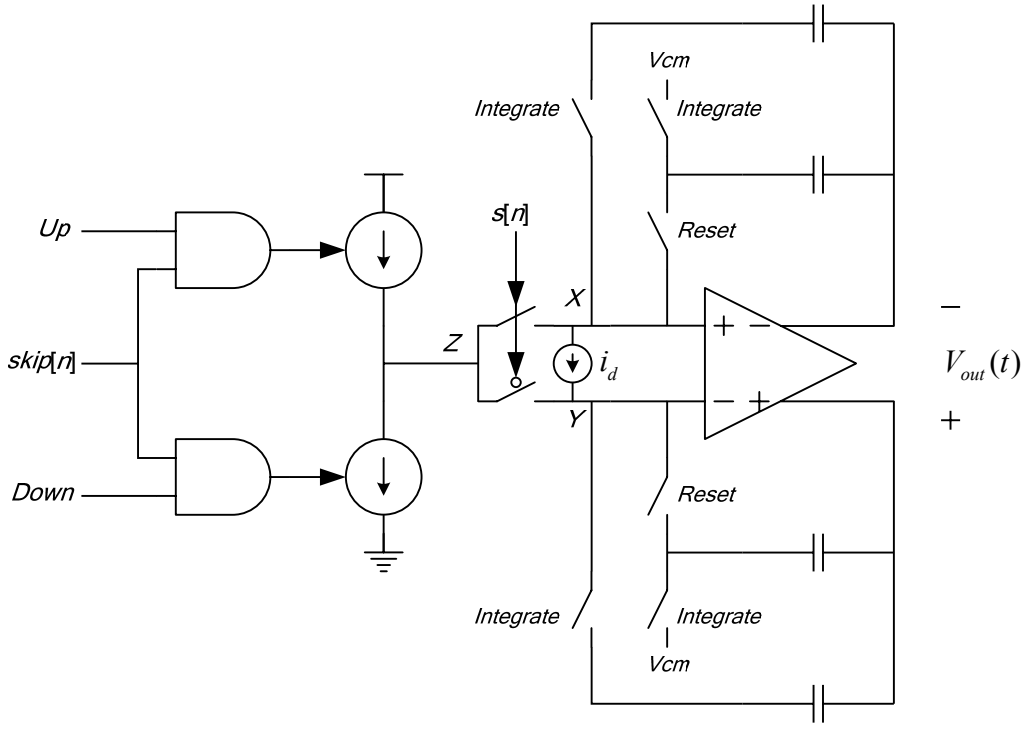


Fig. 33: Detailed diagram of the calibration circuit.

During normal operation, the PLL will be configured in one of two states, denoted here as state “A” and state “B”. For example, state A could represent a configuration where realignment is occurring, while state B could represent a configuration where it is not. The combination of the charge pump and integrator measure the phase difference arising from the PLL being configured in state A and the phase difference arising from being in state B, and attempts to drive that difference to zero. When this is achieved, the system behaves identically regardless of the state.

The sequence, *s*[*n*], is 0 when the PLL is in state A, and 1 when it is in state B. It does not need to be random; however, it is beneficial it is as it ensures that disturbances

in the system do not result in a spur. Furthermore, to ensure that the output of the charge pump is not steered preferentially either to the positive or negative terminals of the integrator, a sequence *skip*[*n*] is used. This sequence selectively suppresses the charge pump such that its output is steered either to the positive or negative input terminal of the op-amp with equal probability and in a first-order shaped fashion.

Without offset-cancellation in the opamp, parasitic capacitances at nodes X, Y, and Z would cause charge to be transferred from one input terminal of the integrator to the other if $s[n] \neq s[n+1]$ and $V_x \neq V_y$. The former condition is satisfied since $s[n]$ is not constant valued, and the latter condition is satisfied if the op-amp has a non-zero, input referred voltage offset. The transfer of charge can be modeled by a differential current, i_d, as shown in Fig. 33. Notice that i_d is unipolar, going from the larger of V_x and V_y to the smaller. As a result, the calibration loop will misalign in the presence of i_d. To mitigate this, the op-amp shown in Fig. 33 employs offset cancellation.

Calibration applied to the differential path and realignment delay path

In the case of the differential path, the two states are the frequency of the VCO when C_1 is used and the frequency of the VCO when C_2 is used. Here, the output of the integrator is $V_{tune,diff+}$ and $V_{tune,diff-}$ and, $s[n] = VCO_{mode}[n]$. When the calibration loop is settled, the PLL oscillates at the same frequency regardless of whether C_1 or C_2 is used.

In the case of the realignment calibration path, the two states are whether or not realignment has occurred. Here the output of the integrator adjusts a continuously tunable delay stage in series with the reference input of the PFD and $s[n] = s_{RA}[n]$. When the calibration loop is settled, the PLL does not see a difference between reference periods where

realignment has occurred and those where it has not.

SYSTEM DIAGARM AND RESULTS

The block diagram of the RAPLL is given in Fig. 34.

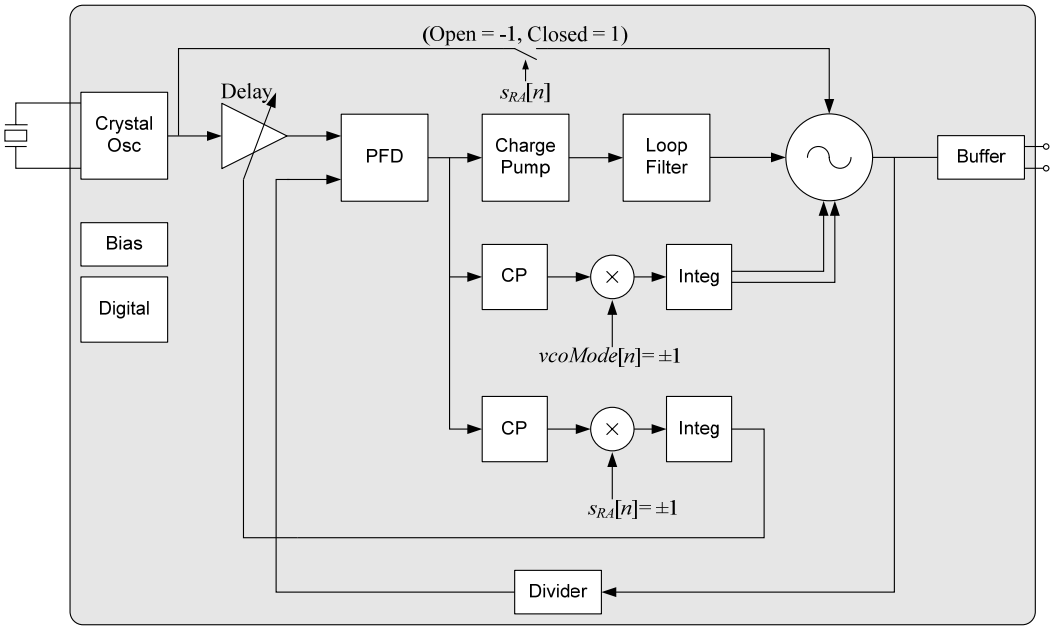


Fig. 34: Block diagram of the IC

The realigned PLL is designed in the TSMC 65nm/GP technology with a target output frequency of 2.5GHz. Here, it is simulated in a corner condition of Slow/125C/V_{dd}=1V. In this simulation, the charge pumps and calibration paths are modelled at the transistor level while the other blocks, such as the divider and PFD, are behavioral. Additionaly, to speed up the simulation, the VCO is also behavioral. Separate transistor level simulations have been done on the VCO to verify the range and realignment mechanism.

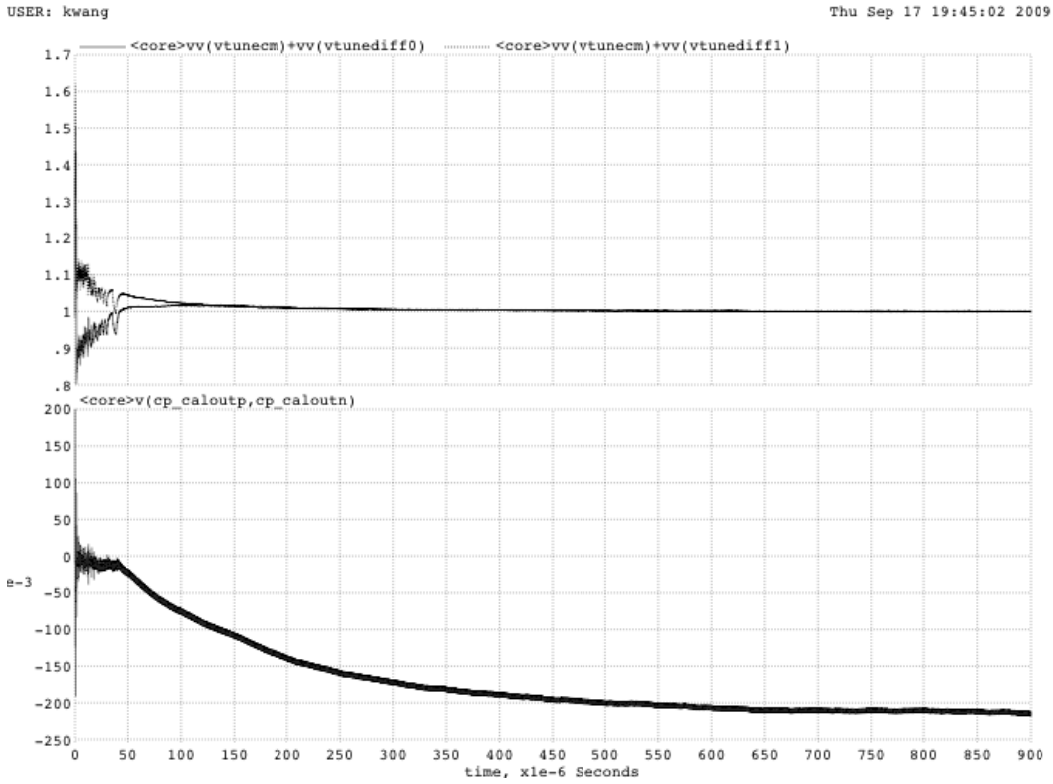


Fig. 35: Transient response showing the locking of the RAPLL. The top curve shows the loop filter voltages while the bottom shows the output of the calibration loop.

Fig. 35 shows the PLL locking from an off state, a process that takes approximately 700us in this corner. More typically, the loops are fully settled in 500us. The top curve shows the loop filter voltages, while the bottom show the output of the calibration loop. Due to the sparsity of the non-RA case, the calibration loop is the slowest to settle. This can be addressed by initially having equal probabilities for RA and not RA. Fig. 36 show the output spectrum of the PLL. Notice that the reference spur is at approximately -60dBc.

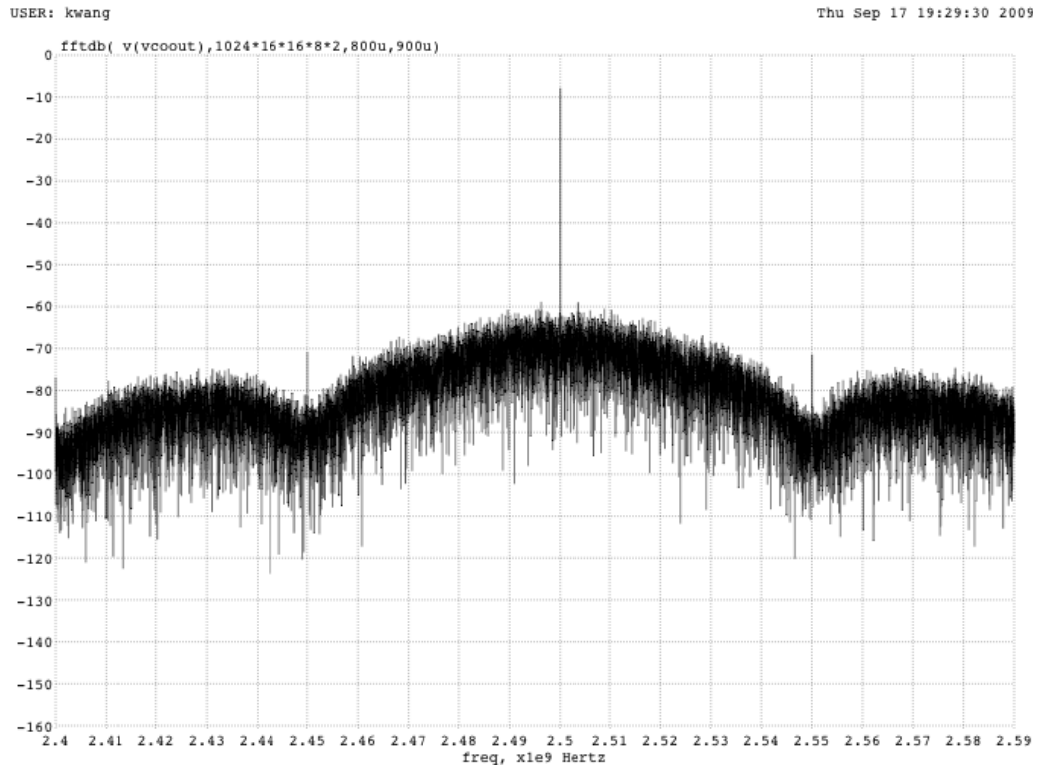


Fig. 36: FFT of the VCO output.

Table 4 compares this design with other published works.

Table 4: Comparison of this RAPLL with other works.

Specification	**[2]**	**[3]**	**[1]**	**[5]**	**This work**	**Units**
f_{out}	96	2000	176	1600	2500	MHz
f_{ref}	0.48	250	8	50	50	MHz
Random jitter	4.65	1.64	5	0.68	1.3	ps_{rms}
RA spur	-34	-32	-70	-58	-60	dBc
Deterministic jitter	293	12	1.8	0.76	0.4	$ps_{peak\text{-}peak}$
Current	2.2	6.6	8.8	6.1 (est)	5.3	mA
Active area	0.22	0.05	0.5	Discrete	0.21	mm^2
Technology	0.35um	0.18um	0.18um	0.13um	65nm	CMOS
Year of publication	2002	2002	2007	2008	2010	

REFERENCES

[1] P. Maulik and D. Mercer, “A DLL-Based Programmable Clock Multiplier in 0.18-μ m CMOS With −70 dBc Reference Spur,” *Solid-State Circuits, IEEE Journal of*, vol. 42, Aug. 2007, pp. 1642-1648.

[2] S. Ye, L. Jansson, and I. Galton, “A multiple-crystal interface PLL with VCO realignment to reduce phase noise,” *Solid-State Circuits, IEEE Journal of*, vol. 37, Dec. 2002, pp. 1795-1803.

[3] R. Farjad-Rad, W. Dally, H. Ng, R. Senthinathan, M. Lee, R. Rathi, and J. Poulton, “A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips,” *Solid-State Circuits, IEEE Journal of*, vol. 37, Dec. 2002, pp. 1804-1812.

[4] B. Helal, M. Straayer, G. 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,” *Solid-State Circuits, IEEE Journal of*, vol. 43, Apr. 2008, pp. 855-863.

[5] B. Helal, C. Hsu, K. Johnson, and M. Perrott, “A Low Jitter Programmable Clock Multiplier Based on a Pulse Injection-Locked Oscillator With a Highly-Digital Tuning Loop,” *Solid-State Circuits, IEEE Journal of*, vol. 44, May. 2009, pp. 1391-1400.