UNIVERSITY OF CALIFORNIA, SAN DIEGO

Delta-Sigma FDC Based Fractional-*N* PLLs
with
Multi-Rate Quantizing Dynamic Element Matching

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

in

Electrical Engineering (Electronic Circuits and Systems)

by

Christian Venerus

Committee in charge:

Professor Ian A. Galton, Chair
Professor Peter M. Asbeck
Professor James F. Buckwalter
Professor Bruce K. Driver
Professor Thomas T. Liu

2013

The dissertation of Christian Venerus is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2013

DEDICATION

To my parents, Graziella and Nevio.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Ian Galton, for his constant support and encouragement through the course of the past several years. His passion, enthusiasm, mastery and commitment to this field were fundamental to my choice of starting my research endeavors. His guidance, daily advice and faith in this work made him a mentor and friend.

I want to thank Kevin Wang for his technical and personal advice, the very many hours he dedicated to mentor me and his constant encouragement.

I would like to thank all my lab colleagues and friends for their friendship and support. They were my companions in many technical conversations and enjoyable moments of distraction.

Finally, I would like thank Greg Harrison and Ginger Weavil for their CAD support.

Chapter 1, in full, has been published in the IEEE Transactions on Circuits and Systems I: Regular Papers, volume 60, number 5, pages 1274-1285, May 2013. 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 for this paper.

VITA

2013	Doctor of Philosophy in Electrical Engineering (Electronic Circuits and Systems), University of California, San Diego
2007	Laurea specialistica degree in Electrical Engineering, University of Padova, Italy
2006 – 2007	EAP Exchange Student, University of California, San Diego
2005	Laurea degree in Information Engineering, University of Padova, Italy

ABSTRACT OF THE DISSERTATION

Delta-Sigma FDC Based Fractional-*N* PLLs
with
Multi-Rate Quantizing Dynamic Element Matching

by

Christian Venerus

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

University of California, San Diego, 2013

Professor Ian A. Galton, Chair

Fractional-*N* phase-locked loop (PLL) frequency synthesizers are ubiquitous in modern communication systems, where they are used to synthesize a signal of high spectral purity from a reference signal of much lower frequency. In order to meet the requirements of wireless communication standard, strict limitation are placed on the

spectral content of the synthesized signal.

In recent years, PLL based on time-to-digital converters (TDC-PLLs) have been proposed that aim at moving the complexity of the design from the analog section to the digital section of the synthesizer: the advantages are a reduction in area, cost and power consumption over competing architectures based on delta-sigma modulation and charge pumps (ΔΣ-PLLs). Although TDC-PLLs with good performance have been demonstrated, TDC quantization noise has so far kept their phase noise and spurious tone performance below that of the best comparable ΔΣ-PLLs. An alternative approach is to use a delta-sigma frequency-to-digital converter (ΔΣ FDC) in place of a TDC to retain the benefits of TDC-PLLs and ΔΣ-PLLs.

Chapter 1 describes a practical ΔΣ FDC based PLL in which the quantization noise is equivalent to that of a ΔΣ-PLL. It presents a linearized model of the PLL, design criteria to avoid spurious tones in the ΔΣ FDC quantization noise, and a design methodology for choosing the loop parameters in terms of standard PLL target specifications.

Chapter 2 presents a multi-rate quantizing dynamic element matching (DEM) encoder for digital to analog converters (DACs) that allows a significant reduction in the encoder power consumption with respect to a conventional encoder for oversampling DEM DACs, at the expense of a minimal signal-to-noise ratio reduction.

In Chapter 3, the implementation details of a ΔΣ FDC based fractional-*N* phase-locked loop prototype are shown. The PLL was built to showcase the capability of the architecture analyzed in Chapter 1 to comply with the most stringent wireless

communication standards. The prototype extends the architecture described in Chapter 1 by including an FDC quantization noise cancelling algorithm, and an hardware efficient implementation of a multi-rate quantizing DEM encoder for digital to frequency conversion.

Chapter 1

Delta-Sigma FDC Based Fractional-*N* PLLs

Abstract—Fractional-*N* phase-locked loop frequency synthesizers based on time-to-digital converters (TDC-PLLs) have been proposed to reduce the area and linearity requirements of conventional PLLs based on delta-sigma modulation and charge pumps (ΔΣ-PLLs). Although TDC-PLLs with good performance have been demonstrated, TDC quantization noise has so far kept their phase noise and spurious tone performance below that of the best comparable ΔΣ-PLLs. An alternative approach is to use a delta-sigma frequency-to-digital converter (ΔΣ FDC) in place of a TDC to retain the benefits of TDC-PLLs and ΔΣ-PLLs. This paper proposes a practical ΔΣ FDC based PLL in which the quantization noise is equivalent to that of a ΔΣ-PLL. It presents a linearized model of the PLL, design criteria to avoid spurious tones in the ΔΣ FDC quantization noise, and a design methodology for choosing the loop parameters in terms of standard PLL target specifications.

I. INTRODUCTION

Delta-sigma modulator based fractional-*N* phase-locked loops (ΔΣ-PLLs) of

Manuscript received February 24, 2012; revised August 01, 2012; accepted August 24, 2012. Date of publication November 16, 2012. date of current version April 24, 2013. This work was supported by the National Science Foundation under Award 0914748. This paper was recommended by Associate Editor J. Kim.

The authors are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail:galton@ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org

Digital Object Identifier 10.1109/TCSI.2012.2221197

the type shown in Figure 1 are widely used as local oscillator frequency synthesizers in wireless communication systems because they offer excellent spectral purity with virtually unlimited frequency tuning resolution [1-4]. Unfortunately, to achieve the performance necessary for most wireless applications a ΔΣ-PLL requires a highly-linear charge pump and large loop filter capacitance, often on the order of hundreds of pico-Farads. Typically, this necessitates an off-chip loop filter, which increases the pin count, circuit footprint, and overall system cost. Furthermore, in highly-scaled CMOS technology, low voltage headroom on the input node of the voltage controlled oscillator (VCO) necessitates tradeoffs that limit performance. Reducing the voltage swing requires an increase in the VCO gain which tends to increase the phase noise, yet increasing the voltage swing for a given supply voltage reduces charge pump linearity which increases spurious tones.

Recently, fractional-N PLLs have been proposed that exploit digital signal processing to avoid these problems [5, 6, 7, 8, 9, -10]. They use a time-to-digital converter (TDC), a digital loop filter, and a digitally controlled oscillator (DCO) in place of a divider, phase-frequency detector (PFD), charge pump, analog loop filter, and VCO. The TDC generates a quantized estimate of the instantaneous phase of the DCO at each positive edge of the reference oscillator. The difference between the TDC output and the calculated instantaneous phase of an ideal oscillator running at the desired output frequency is digitally lowpass filtered and the resulting digital sequence controls the DCO.

Although such TDC-based PLLs (TDC-PLLs) have been demonstrated to have very good performance, TDC quantization noise has so far kept their phase noise and spurious tone performance below those of the best comparable ΔΣ-PLLs. TDC quantization noise is relatively coarse and, unlike quantization noise in ΔΣ-PLLs, it is not highpass shaped so it is not as well suppressed by the PLL.

An alternative approach that offers the advantages of both ΔΣ-PLLs and TDC-PLLs is to use a delta-sigma frequency-to-digital converter (ΔΣ FDC) in place of a TDC [11, 12, 13, -14]. Such ΔΣ FDC based fractional-*N* PLLs (FDC-PLLs) have been proposed in which the ΔΣ FDC performs 1-bit quantization and the DCO is implemented as a DAC followed by a VCO [15, 16]. It is likely that improved performance can be achieved in future FDC-PLLs by using the type of high-performance DCOs developed for TDC-PLLs and, as quantified in this paper, by avoiding 1-bit quantization in the ΔΣ FDC.

This paper proposes a practical FDC-PLL architecture and proves that its quantization noise performance is equivalent to that of a ΔΣ-PLL with a second-order delta-sigma modulator. It shows that 5-level quantization in the ΔΣ FDC is both necessary and sufficient to avoid spurious tones that would otherwise be caused by quantizer overloading. It derives a linearized model that accurately predicts the transfer functions imposed by the FDC-PLL on its component noise sources, and provides a design methodology based on the model for choosing the loop parameters in terms of standard PLL target specifications.

II. Background Information

A. Phase Noise in Fractional-*N* PLLs

The instantaneous frequency of the reference oscillator in Hz is

$$f_{ref}+\psi_{ref}(t) \tag{1}$$

where f_{ref} is the nominal reference frequency, and $\psi_{ref}(t)$ is the reference oscillator's instantaneous frequency error. The reference oscillator's instantaneous phase in cycles relative to an initial time, t_0, is the integral of (1) from time t_0 to time t:

$$p_{ref}(t)=\left(t-t_0\right)f_{ref}+\theta_{ref}(t) \tag{2}$$

where

$$\theta_{ref}(t)=\int_{t_0}^{t}\psi_{ref}(u)du \tag{3}$$

is the reference oscillator's instantaneous phase noise in cycles.

A fractional-*N* PLL generates a periodic output signal with an average frequency of $(N+\alpha)f_{ref}$, where N is an integer and α is a fractional value with a magnitude less than 1. Therefore, its instantaneous output frequency in Hz can be written as

$$f_{PLL}(t)=\left(N+\alpha\right)f_{ref}+\psi_{PLL}(t) \tag{4}$$

where $\psi_{PLL}(t)$ is its instantaneous frequency error. The PLL's instantaneous output phase in cycles relative to time t_0 is the integral of (4) from time t_0 to time t:

$$p_{PLL}(t)=\left(t-t_0\right)\left(N+\alpha\right)f_{ref}+\theta_{PLL}(t) \tag{5}$$

where

$$\theta_{PLL}(t)=\int_{t_0}^{t}\psi_{PLL}(u)du \tag{6}$$

is the PLL's instantaneous phase noise in cycles.

A fractional-*N* PLL must control its output frequency such that $\psi_{PLL}(t)$ has zero mean and the power spectral density (PSD) of $\theta_{PLL}(t)$ is within acceptable limits for the desired application. As described below, the ΔΣ-PLL and the FDC-PLL each do this by estimating a phase error sequence proportional to

$$-\theta_{PLL}(t)+\left(N+\alpha\right)\theta_{ref}(t) \tag{7}$$

sampled at the reference frequency, passing the estimated phase error sequence through their loop filter, and using the output of the loop filter to control the output frequency of their VCO or DCO. The feedback ensures that $\psi_{PLL}(t)$ has zero mean (provided that $\psi_{ref}(t)$ has zero mean), and the characteristics of the loop filter, the DC loop gain, and the accuracy with which (7) is estimated determine the spectral properties of $\theta_{PLL}(t)$.

B. ΔΣ Fractional-*N* PLLs

A typical ΔΣ-PLL is shown in Figure 1. It consists of a PFD, a charge pump, an analog loop filter, a VCO, a frequency divider, and a second-order digital delta-sigma modulator clocked by the divider output.

The divider output is a two-level signal in which the *n*th and (*n*+1)th rising edges are separated by $N + y[n]$ VCO periods, where $y[n]$ is an integer-valued sequence from the delta-sigma modulator. As indicated in the figure for the case where the PLL is locked, if the *n*th rising edge of the reference signal, $v_{ref}(t)$, occurs before that of the divider output, $v_{div}(t)$, the charge pump generates a current pulse of nominal amplitude I_{CP} and duration equal to the time difference between the two edges. Otherwise, the pulse has the same magnitude and duration, but its polarity is reversed.

The input to the delta-sigma modulator is α plus pseudo-random least significant bit dither, $d[n]$, so its output has the form $y[n] = \alpha + d[n] + e_{\Delta\Sigma}[n]$, where $e_{\Delta\Sigma}[n]$ is second-order highpass shaped delta-sigma quantization noise. As proven in [17], the dither prevents $e_{\Delta\Sigma}[n]$ from containing spurious tones that would otherwise show up as spurious tones in the ΔΣ-PLL's output.

As shown in [18], the net charge delivered to the loop filter by the charge pump's current pulse each reference period is proportional to the sum of a phase error term and first-order highpass shaped delta-sigma quantization noise. The phase error term is

$$-\theta_{PLL}(\tau_n) + \left(N + \alpha\right)\theta_{ref}(t_n) \tag{8}$$

where t_n and τ_n, are the times of the positive-going zero-crossings of $v_{ref}(t)$ and $v_{div}(t)$, respectively, corresponding to the nth charge pump pulse.

The loop bandwidth of the ΔΣ-PLL is designed to be low enough that the delta-sigma quantization noise is largely suppressed by the lowpass filtering operation of the loop. Hence, the average output frequency settles to $(N + \alpha)f_{ref}$, as desired, with the delta-sigma quantization noise contributing only a small amount of phase noise.

III. Overview of The FDC-PLL

A. System Description

The proposed FDC-PLL is shown in Figure 2. It consists of three main components: a ΔΣ FDC, a digital loop controller, and a DCO. The digital loop controller is clocked and the output of the digital loop controller is latched into the DCO on each rising edge of the reference signal.

The ΔΣ FDC consists of a PFD, charge pump, integrating capacitor, 5-level ADC, $2 - z^{-1}$ digital block, and multi-modulus divider. The PFD and charge pump are the same as those in a ΔΣ-PLL. As in a ΔΣ-PLL, when the FDC-PLL is locked the magnitude of the difference between the time of each rising edge of $v_{div}(t)$ and the time of the corresponding rising edge $v_{ref}(t)$ is a small fraction of the reference oscillator period, T_{ref}. Therefore, the charge pump generates a relatively narrow (compared to T_{ref}) positive or negative pulse of current around the time of each rising edge of $v_{ref}(t)$. The 5-level ADC is clocked with a delayed version of the reset signal within the PFD, such that it samples the capacitor voltage shortly after each charge pump current pulse settles to zero. The divider in the FDC-PLL is identical to that in a ΔΣ-PLL, but its modulus is varied by $v[n] = 2y[n] - y[n - 1]$ instead of $y[n]$.

By design, α is restricted to the range

$$-\frac{1}{2} \leq \alpha \leq \frac{1}{2} \tag{9}$$

and the charge pump current, I_{CP}, ideally satisfies

$$I_{CP} = \left(N + \alpha\right) f_{ref} C\Delta \tag{10}$$

where Δ is the step-size of the 5-level ADC.[†]

As shown in the Appendix, the $\Delta\Sigma$ FDC implicitly implements second-order delta-sigma modulation. In particular, $y[n] + \alpha$ is a measure of the PLL's frequency error plus second-order highpass shaped ADC quantization noise, so it averages to zero when the average DCO frequency is $(N+\alpha)f_{ref}$.

The accumulator in the digital loop controller converts the PLL's frequency error to phase error and reduces the second-order highpass shaped ADC quantization noise to first-order highpass shaped ADC quantization noise. Specifically, as shown in Section IV the output of the accumulator, $p[n]$, consists of the phase error term given by (8) plus first-order highpass shaped ADC quantization noise. Therefore, the properties of $p[n]$ are very similar to those of the sequence of charge pulses delivered by the charge pump to the analog loop filter in the $\Delta\Sigma$-PLL of Figure 1. Accordingly, the digital loop filter in the FDC-PLL performs the same function as the analog loop filter in the $\Delta\Sigma$-PLL. It suppresses out-of-band quantization noise and circuit error, and sets the loop dynamics.

The DCO is an analog oscillator with a means for the frequency to be controlled by a digital sequence, in this case the output of the digital loop filter, $d[n]$. Depending on the transfer function of the digital loop filter, the required DCO frequency change corresponding to the minimum step-size of $d[n]$ can be very small. A common method of implementing a DCO with a very small minimum frequency step is to

[†] As demonstrated in Section IV, deviations of the charge pump current sources on the order of several percent do not significantly degrade the performance of the FDC-PLL. Therefore, since α typically has a magnitude much smaller than N, usually it is reasonable to set $I_{CP} = Nf_{ref}C\Delta$.

quantize *d*[*n*] with a digital delta-sigma modulator clocked at a rate much higher than the reference frequency, where the clock signal is obtained by dividing the PLL output signal by a small integer [19]. For each value of *d*[*n*], the delta-sigma modulator generates multiple output values with a minimum step-size greater than that of *d*[*n*] which are used to modulate the frequency of the DCO. The natural lowpass filtering imposed by the DCO suppresses much of the quantization noise introduced by the delta-sigma modulator, so the effective minimum frequency step of the DCO is that of *d*[*n*] at the cost of additive phase noise.

In this paper, any quantization of *d*[*n*] performed by the DCO as described above is considered to happen within the DCO, so it is not shown explicitly in Figure 2. Accordingly, the *DCO phase noise* is defined to be the sum of the phase noise caused by analog oscillator noise and any phase noise caused by quantizing *d*[*n*] within the DCO.

B. Digital Loop Filter

Given that the digital loop filter in the FDC-PLL plays the role of the analog loop filter in the ΔΣ-PLL, it is reasonable to design the digital loop filter such that it has comparable filtering characteristics to the analog loop filter shown in Figure 1. This can be achieved with a digital loop filter transfer function of

$$L(z) = L_{PI}(z)L_{LPF}(z) \tag{11}$$

where

$$L_{PI}(z) = K_P + K_I \frac{1}{1-z^{-1}}, \tag{12}$$

K_P and K_I are constants called the *proportional path gain* and *integral path gain*, respectively, and $L_{LPF}(z)$ is an all-pole lowpass filter section described shortly [20]. The $L_{PI}(z)$ portion of the filter is often called a proportional-integral filter and is sufficient to obtain a stable feedback system. The $L_{LPF}(z)$ portion of the filter provides attenuation above the PLL bandwidth to reduce phase noise.

Such a digital loop filter has comparable filtering characteristics to the analog loop filter shown in Figure 1 if $L_{LPF}(z)$ contains a single pole. Unfortunately, neither filter rolls off very sharply with frequency.

In the ΔΣ-PLL this problem is often addressed by adding an extra pole outside the PLL bandwidth. Usually, no more than one extra pole is added, though, because of the increased area and power consumption associated with adding multiple extra poles.

In contrast, the incremental area and power consumption associated with adding multiple extra poles to a digital filter tend to be modest. Therefore, the loop filter used in the FDC-PLL analyzed in this paper has a transfer function given by (11) with

$$L_{LPF}(z) = \prod_{i=0}^{3} \frac{\lambda_i}{1-(1-\lambda_i)z^{-1}} \tag{13}$$

where $1 - \lambda_i$ for $i = 0, 1, 2,$ and 3 are real poles. A design procedure for selecting K_P, K_I, and λ_i for $i = 0, 1, 2,$ and 3 in terms of the desired loop bandwidth and phase margin is presented in Section IV.

C. ΔΣ-PLL and FDC-PLL Capacitance Comparison

It is mentioned in the Introduction that the capacitance in the loop filter of the

ΔΣ-PLL tends to be large, often on the order of hundreds of pico-Farads. To sufficiently suppress the delta-sigma quantization error, the loop bandwidth of a fractional-N PLL is usually a small fraction (e.g., several hundredths) of the reference frequency. With the analog loop filter shown in Figure 1, the loop bandwidth is proportional to RI_{CP}, and the total capacitance is approximately inversely proportional to R. Therefore, for any given loop bandwidth, C_1 and C_2 can be reduced by simultaneously increasing R and decreasing I_{CP}. Unfortunately, decreasing I_{CP} tends to increase the PLL's phase noise because the loop gain of the ΔΣ-PLL's linearized model is proportional to I_{CP} [18]. This places a lower bound on I_{CP} for any given application, which, in turn, typically dictates a large total capacitance when the loop bandwidth is small.

In contrast, as shown in Section IV the loop bandwidth of the FDC-PLL is independent of I_{CP} and C, and the overall phase noise is not a strong function of either I_{CP} or C, so C can be much smaller than the loop filter capacitance in a comparable ΔΣ-PLL. For example, C = 1.25 pF in the FDC-PLL design example presented in Section IV.

IV. The FDC-PLL Linearized Model

A. Model Derivation

It is proven in the Appendix that the ΔΣ FDC behaves as the second-order delta-sigma modulator shown in Figure 3 along with the α adder and accumulator of the digital loop controller. It is further shown that the output of the accumulator can be written as

$$p[n] = -\theta_{PLL}\left(\tau_n\right) + \left(N + \alpha\right)\theta_{ref}\left(t_n\right) + \frac{e_p[n]}{\Delta} + e_{\Delta\Sigma 1}[n] \tag{14}$$

neglecting a possible constant offset, where $e_p[n]$ represents the combined error from noise and other non-ideal circuit behavior in the charge pump, PFD, and divider,

$$e_{\Delta\Sigma 1}[n] = e_{ADC}[n] - e_{ADC}[n-1], \tag{15}$$

and $e_{ADC}[n]$ is the sum of quantization noise and any additional error from non-ideal circuit behavior in the ADC. As explained in the Appendix, a five-level ADC is necessary and sufficient to ensure that the delta-sigma modulator does not overload when the PLL is locked, which would introduce spurious tones.

The output of the loop filter, $d[n]$, is latched into the DCO on each positive-going zero-crossing of $v_{ref}(t)$, so $d[n-1]$ is applied to the DCO during the time interval $t_n < t \leq t_{n+1}$ for each positive integer n. It is assumed that the DCO's control word latency is negligible, so its instantaneous frequency during each time interval $t_n < t \leq t_{n+1}$ is

$$f_{PLL}(t) = f_c + K_{DCO}d[n-1] + \psi_{DCO}(t) \tag{16}$$

where f_c is the nominal center frequency of the DCO in Hz, K_{DCO} is the DCO gain in Hz, and $\psi_{DCO}(t)$ is the DCO's instantaneous frequency error.[†] It follows from (4) and (16) that during the time interval $t_n < t \leq t_{n+1}$ the FDC-PLL's instantaneous frequency error can be written as

$$\psi_{PLL}(t) = f_c - \left(N + \alpha\right)f_{ref} + K_{DCO}d[n-1] + \psi_{DCO}(t) \tag{17}$$

[†] The *DCO Gain* is defined as the amount by which the DCO frequency changes when $d[n]$ changes by unity.

The ideal output frequency when the FDC-PLL is locked is $(N + \alpha)f_{ref}$, so $d[n]$ can be written as

$$d[n] = \frac{1}{K_{DCO}}\left[\left(N+\alpha\right)f_{ref} - f_c\right] + f[n] \tag{18}$$

where $f[n]$ is the zero-mean component of $d[n]$. It follows from (17) and (18) that

$$\psi_{PLL}(t) = K_{DCO} f[n-1] + \psi_{DCO}(t) \tag{19}$$

during the time interval $t_n < t \leq t_{n+1}$ for each positive integer n.

Integrating (19) from time t_0 to t where $t_n < t \leq t_{n+1}$ gives

$$\begin{aligned}\theta_{PLL}\left(t\right) = K_{DCO}\sum_{k=1}^{n-1}\left(t_{k+1}-t_k\right)f[k-1] \\ +K_{DCO}f[n-1]\left(t-t_n\right)+\theta_{DCO}\left(t\right)\end{aligned} \tag{20}$$

where

$$\theta_{DCO}(t) = \int_{t_0}^{t}\psi_{DCO}(u)du \tag{21}$$

is the instantaneous phase noise introduced by the DCO. Typical reference oscillators have high spectral purity, so

$$t_{k+1} - t_k \cong T_{ref} \tag{22}$$

holds to a high degree of accuracy. Hence, (20) implies that $\theta_{PLL}(t)$ can be written as

$$\begin{aligned}\theta_{PLL}\left(t\right) = K_{DCO}T_{ref}\sum_{k=1}^{n-1}f[k-1] \\ +K_{DCO}f[n-1]\left(t-t_n\right)+\theta_{DCO}\left(t\right)\end{aligned} \tag{23}$$

for $t_n < t \leq t_{n+1}$ which can be rewritten as

$$\begin{aligned}\theta_{PLL}\left(t\right) = \theta_{DCO}\left(t\right) \\ +\left(\theta_{loop}[n+1]-\theta_{loop}[n]\right)\frac{\left(t-t_n\right)}{T_{ref}}+\theta_{loop}[n]\end{aligned} \tag{24}$$

where

$$\theta_{loop}[n] = K_{DCO}T_{ref}\sum_{k=1}^{n-1} f[k-1]\,. \tag{25}$$

The second and third term in (24) represent a linear interpolation between the nth and (n+1)th samples of $\theta_{loop}[n]$. This type of interpolation is called *first-order hold* interpolation [21]. To extend (24) to hold for any $t > t_0$, the first-order hold component can be written as a sequence of triangular time pulses with amplitudes $\theta_{loop}[n]$, i.e.,

$$\theta_{PLL}\left(t\right) = \theta_{DCO}\left(t\right) + \sum_{n=0}^{\infty}\theta_{loop}[n]h_{tri}\left(t-nT_{ref}-t_0\right) \tag{26}$$

for arbitrary $t > t_0$, where

$$h_{tri}(t) = \begin{cases} 1-\dfrac{|t|}{T_{ref}} & \text{if } |t| < T_{ref}\,, \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

The bandwidth of a practical PLL is much smaller than the reciprocal of the maximum magnitude of the difference between τ_n and t_n , so

$$\theta_{PLL}\left(\tau_n\right) \cong \theta_{PLL}\left(t_n\right) \tag{28}$$

holds to a high degree of accuracy. Hence (24) yields

$$\theta_{PLL}\left(\tau_n\right) \cong \theta_{DCO}\left(t_n\right) + \theta_{loop}[n]\,. \tag{29}$$

Combining (14), (15), (25), (26), and (29) results in the linearized model shown in Figure 4 where the sample-rate of the discrete-time blocks and the first-order hold interpolator is f_{ref}. The discrete-time portion of the model implements the FDC-PLL's feedback system and generates $\theta_{loop}[n]$, which is linearly interpolated by the first-order hold block as described above.

It follows from Figure 4 that the discrete-time loop gain of the FDC-PLL is

$$T(z) = K_{DCO} T_{ref} L(z) \frac{z^{-2}}{1 - z^{-1}} \tag{30}$$

and the various FDC-PLL discrete-time transfer functions are

$$\frac{\theta_{loop}}{\theta_{ref}}(z) = (N + \alpha) \frac{T(z)}{1 + T(z)} \tag{31}$$

$$\frac{\theta_{loop}}{\theta_{DCO}}(z) = -\frac{T(z)}{1 + T(z)} \tag{32}$$

$$\frac{\theta_{loop}}{e_{ADC}}(z) = \left(1 - z^{-1}\right) \frac{T(z)}{1 + T(z)} \tag{33}$$

and

$$\frac{\theta_{loop}}{e_p}(z) = \frac{1}{\Delta} \left(\frac{T(z)}{1 + T(z)} \right). \tag{34}$$

These equations describe the loop dynamics of the FDC-PLL.

B. Phase Noise PSD Calculation

It is assumed that the noise signals $\theta_{ref}(t_n)$, $e_{ADC}[n]$, $e_p[n]$, and $\theta_{DCO}(t)$ can be modeled as uncorrelated, zero-mean, wide-sense stationary random processes, so the PSD of $\theta_{PLL}(t)$ is the sum of PSD components that each correspond to one of the noise signals. Likewise, the discrete-time PSD of $\theta_{loop}[n]$ is the sum of the discrete-time

PSD components that each correspond to one of the noise signals.

It follows from (31) that the component of the discrete-time PSD of $\theta_{loop}[n]$ corresponding to $\theta_{ref}(t_n)$ is

$$S_{\theta_{ref}}\left(e^{j2\pi T_{ref}f}\right)\left|\left(N+\alpha\right)\frac{T\left(e^{j2\pi T_{ref}f}\right)}{1+T\left(e^{j2\pi T_{ref}f}\right)}\right|^2 \tag{35}$$

where $S_{\theta_{ref}}\left(e^{j2\pi T_{ref}f}\right)$ is the discrete-time PSD of $\theta_{ref}(t_n)$. The continuous-time Fourier transform of the output of a first-order hold interpolator with input $u[n]$ and sample-rate f_{ref} is

$$U\left(e^{j2\pi T_{ref}f}\right)T_{ref}\left[\frac{\sin\left(\pi T_{ref}f\right)}{\pi T_{ref}f}\right]^2 \tag{36}$$

where $U\left(e^{j2\pi T_{ref}f}\right)$ is the discrete-time Fourier transform of $u[n]$ [21]. Therefore, the component of the PSD of $\theta_{PLL}(t)$ corresponding to $\theta_{ref}(t_n)$ is

$$S_{\theta_{ref}}\left(e^{j2\pi T_{ref}f}\right)T_{ref}\left|\frac{\left(N+\alpha\right)T\left(e^{j2\pi T_{ref}f}\right)}{1+T\left(e^{j2\pi T_{ref}f}\right)}\right|^2\left[\frac{\sin\left(\pi T_{ref}f\right)}{\pi T_{ref}f}\right]^4 \tag*{.(37)}$$

By similar reasoning, the component of the PSD of $\theta_{PLL}(t)$ corresponding to $e_p[n]$ is

$$S_{e_p}\left(e^{j2\pi T_{ref}f}\right)\frac{T_{ref}}{\Delta^2}\left|\frac{T\left(e^{j2\pi T_{ref}f}\right)}{1+T\left(e^{j2\pi T_{ref}f}\right)}\right|^2\left[\frac{\sin\left(\pi T_{ref}f\right)}{\pi T_{ref}f}\right]^4 \tag{38}$$

where $S_{e_p}\left(e^{j2\pi T_{ref}f}\right)$ is the discrete-time PSD of $e_p[n]$.

As described in the Appendix, $e_{ADC}[n]$ is asymptotically white and uniformly distributed between −0.5 and 0.5, so the discrete-time PSD of $e_{ADC}[n]$ is 1/12. It follows from reasoning similar to that which led to (37) and (38) that the component of the PSD of $\theta_{PLL}(t)$ corresponding to $e_{ADC}[n]$ is

$$\frac{T_{ref}}{3}\sin^2\left(\pi T_{ref}f\right)\left|\frac{T\left(e^{j2\pi T_{ref}f}\right)}{1+T\left(e^{j2\pi T_{ref}f}\right)}\right|^2\left[\frac{\sin\left(\pi T_{ref}f\right)}{\pi T_{ref}f}\right]^4 \tag{39}$$

in units of cycles squared per Hz. If the desired units of the PSD are radians squared per Hz, then (39) must be scaled by $4\pi^2$.

The component of the PSD of $\theta_{PLL}(t)$ corresponding to DCO phase noise depends on both $\theta_{DCO}(t)$ and $\theta_{DCO}(t_n)$, which are obviously correlated. Consequently, the effects of $\theta_{DCO}(t)$ and $\theta_{DCO}(t_n)$ must be considered together when calculating the component of the PSD of $\theta_{PLL}(t)$ corresponding to DCO noise.

The component of $\theta_{loop}[n]$ corresponding to DCO noise is $\theta_{DCO}(t_n)$ filtered by the discrete-time lowpass transfer function (32). As implied by (36), the first-order hold interpolator imposes a continuous-time lowpass filtering operation on this signal component that rolls off in frequency at 40 dB per decade. As shown in Figure 4, $\theta_{DCO}(t)$ is added to the output of the first-order hold interpolator and for a typical DCO the PSD of $\theta_{DCO}(t)$ rolls off in frequency by no more than 20 dB per decade (except at low frequencies where $1/f$ noise is significant). Therefore, in calculating the component of the PSD of $\theta_{PLL}(t)$ corresponding to DCO noise, the output of the first-order hold interpolator can be neglected for frequencies above $f_{ref}/2$ with a high degree of accuracy.

It follows that the effect of adding $\theta_{DCO}(t_n)$ in the feedback loop of Figure 4 is practically equivalent to adding $\theta_{DCO}(t)$ filtered by

$$-\frac{T\left(e^{j2\pi T_{ref}f}\right)}{1+T\left(e^{j2\pi T_{ref}f}\right)}\left[\frac{\sin\left(\pi T_{ref}f\right)}{\pi T_{ref}f}\right]^2 \tag{40}$$

to the output of the first-order hold interpolator. This result relies on the reasonable assumption that aliasing error in $\theta_{DCO}(t_n)$ within the passband of (32) is negligible. It follows that the component of the PSD of $\theta_{PLL}(t)$ corresponding to DCO noise is

$$S_{\theta_{DCO}}\left(f\right)\left|1-\frac{T\left(e^{j2\pi T_{ref}f}\right)}{1+T\left(e^{j2\pi T_{ref}f}\right)}\left[\frac{\sin\left(\pi T_{ref}f\right)}{\pi T_{ref}f}\right]^2\right|^2 \tag{41}$$

where $S_{\theta_{DCO}}\left(f\right)$ is the continuous-time PSD of $\theta_{DCO}(t)$.

The PSD of $\theta_{PLL}(t)$ from all of the FDC-PLL noise sources is the sum of (37), (38), (39), and (41). Typically, estimates of $S_{\theta_{ref}}\left(e^{j2\pi T_{ref}f}\right)$, $S_{e_p}\left(e^{j2\pi T_{ref}f}\right)$, and $S_{\theta_{DCO}}\left(f\right)$ are obtained via circuit simulation. As described in Section II, the DCO phase noise is the combination of phase noise introduced by the underlying analog oscillator and any quantization of $d[n]$, so circuit simulations used to estimate the DCO phase noise PSD must include any such quantization noise.

C. Loop Filter Design

The loop filter transfer function, $L(z)$, determines the FDC-PLL's loop bandwidth, phase margin, and noise filtering characteristics. In analogy to a conventional ΔΣ-PLL, the FDC-PLL's phase noise consists of highpass filtered DCO noise, i.e., (41), and lowpass filtered noise from the reference oscillator, ADC, divider, PFD, and charge pump, i.e., (37)-(39). The design objective for $L(z)$ is to strike a compromise among these noise filtering operations appropriate to the application's requirements while maintaining a given desired loop bandwidth and phase margin.

In a ΔΣ-PLL, having a zero-frequency pole in the loop filter ensures the charge pump output current pulse sequence has zero mean, which simplifies the design of both the charge pump and PFD [4]. In contrast, the delta-sigma modulator relationship derived in the Appendix implies that in the FDC-PLL the charge pump output current pulse sequence has zero mean regardless of whether the loop filter has a zero-frequency pole. Therefore, a major reason for having a zero-frequency loop filter pole in ΔΣ-PLLs does not apply to the FDC-PLL.

Nevertheless, there are still advantages to having a zero-frequency pole in an FDC-PLL's loop filter. One advantage is that it causes the transfer function portion of (41) to have a second zero-frequency zero. DCO phase noise typically has a PSD proportional to $1/f^3$ for $0 < f < f_c$ where f_c is the frequency below which $1/f$ noise is significant. Having two zero-frequency zeros in the transfer function portion of (41) prevents the portion of the PSD proportional to $1/f^3$ from contributing significantly to the overall FDC-PLL phase noise. Another advantage is that the zero-frequency pole eliminates the dependence of $\theta_{PLL}(t)$ on the DCO's center frequency and gain, which both vary with process, supply voltage, and temperature.

The primary disadvantage of having a zero-frequency pole in the loop filter is that it introduces negative phase into the loop gain which limits the achievable sharpness of the filter's transition band for a given phase margin. Therefore, in some applications not having a zero-frequency pole in the loop filter may offer an advantage with respect to minimizing phase noise.

Unlike the case of an analog PLL, there is a great deal of flexibility in the choice of $L(z)$, regardless of whether it has a zero-frequency pole. The remainder of this section evaluates the practical choice of $L(z)$ given by (11) with (12) and (13), which includes a zero-frequency pole.

A reasonable design procedure is to first choose values of K_P and K_I via the equations derived below that result in the desired loop bandwidth and phase margin to the extent that

$$\left| L_{LPF}\left(e^{j2\pi T_{ref} f_u}\right)\right|^2 \cong 1 \quad \text{and} \quad 2\pi T_{ref} f_u \ll 1 \tag{42}$$

where f_u is the unity-gain frequency of the FDC-PLL's loop gain. This requires that the poles of $L_{LPF}(z)$ be initially chosen such that

$$2\pi T_{ref} f_u \ll \lambda_i < 1. \tag{43}$$

Then a trial and error procedure can be used in which the λ_i values are reduced to improve noise suppression while the K_P and K_I values are adjusted to maintain the desired loop bandwidth and phase margin.

By definition, the unity gain frequency of the FDC-PLL's loop gain, f_u, satisfies

$$\left| T\left(e^{j2\pi T_{ref} f_u}\right)\right|^2 = 1. \tag{44}$$

It can be verified from (12), (30), and (42) that

$$f_u \cong \frac{K_{DCO}}{2\pi}\sqrt{\frac{K}{2}}\sqrt{1+\sqrt{1+\frac{4K_I^2}{\left(K_{DCO}T_{ref}\right)^2 K^2}}} \tag{45}$$

where

$$K \triangleq K_P\left(K_P + K_I\right). \tag{46}$$

Typically, $K_P \gg K_I$ in which case (45) reduces to

$$f_u \cong \frac{K_{DCO}K_P}{2\pi} \tag{47}$$

Furthermore,

$$\left|\frac{T\left(e^{j2\pi T_{ref}f_u}\right)}{1+T\left(e^{j2\pi T_{ref}f_u}\right)}\right|^2\left[\frac{\sin\left(\pi T_{ref}f_u\right)}{\pi T_{ref}f_u}\right]^4 \cong \frac{1}{2} \tag{48}$$

given $2\pi T_{ref}f_u \ll 1$, so the FDC-PLL's loop bandwidth, f_{BW}, is approximately given by

$$f_{BW} \cong f_u\,. \tag{49}$$

The FDC-PLL's phase margin in radians is

$$PM = \pi + \measuredangle T\left(e^{j2\pi T_{ref}f_u}\right). \tag{50}$$

With (12), (11), (30), (42), and (45), this can be written as

$$\begin{aligned} PM = \pi &+ \tan^{-1}\left(\frac{K_P\sin\left(2\pi T_{ref}f_u\right)}{K_I + K_P\left[1-\cos\left(2\pi T_{ref}f_u\right)\right]}\right) \\ &-2\tan^{-1}\left(\frac{\sin\left(2\pi T_{ref}f_u\right)}{1-\cos\left(2\pi T_{ref}f_u\right)}\right) - 4\pi T_{ref}f_u. \end{aligned} \tag{51}$$

It follows from the above analysis that for fixed K_{DCO} and T_{ref} the loop bandwidth depends primarily on K_P and for fixed K_{DCO}, T_{ref}, and loop bandwidth the phase margin depends primarily on K_I. Therefore, it is straightforward to choose K_I and K_P using (47), (49), and (51) to achieve a desired loop bandwidth and phase margin provided (42) holds. Then, a trial and error process can be applied in which the λ_i values are reduced to improve phase noise suppression and K_I and K_P are increased to maintain the desired loop bandwidth and phase margin. The trial and error process is guided by plotting (37), (38), (39), and (41) at each iteration.

V. Design Example

The design methodology described above has been applied to select the example FDC-PLL design parameters presented in Table I. This section applies the linearized model to calculate the example FDC-PLL's expected performance with realistic input noise levels, and compares the calculated performance to the performance predicated by computer simulation. The example was chosen because it is suitable for use as a carrier synthesizer for the widely-used GSM mobile handset standard and facilitates comparison with previously published TDC-based PLLs [7, 10].

To apply the linearized model to calculate the FDC-PLL's output phase noise PSD, i.e., the PSD of $\theta_{PLL}(t)$, the PSDs of the input noise sources $\theta_{DCO}(t)$, $\theta_{ref}(t_n)$, and $e_P[n]$ must be known or estimated. In this example, the input noise sources are estimated to be in line with what can be achieved in a 65 nm CMOS process with a 1 V power supply. The simulated DCO is identical to that presented in [19], so $\theta_{DCO}(t)$, which includes both DCO quantization noise and analog noise, is taken to have a PSD consistent with the results presented in [19] as shown in Figure 5. The $\theta_{ref}(t_n)$ and $e_p[n]$ input noise source levels were estimated via periodic steady-state (PSS) circuit simulations of transistor-level reference buffer, divider, PFD, and charge pump circuits.

PSS simulation of the reference buffer indicates that $\theta_{ref}(t)$ can be modeled as white noise with a PSD level of −150 dBc/Hz. Therefore, the discrete-time PSD level of $\theta_{ref}(t_n)$ is $-150 - 10\log_{10}(T_{ref}) = -76$ dBc.

Simulations indicate that $e_p[n]$ is dominated by the charge pump, which has the form of the single-ended design presented in [22]. The quantization step-size of the ADC is 80 mV. Its input voltage, and, therefore, the output voltage of the charge pump, ranges from 0.3 V to 0.7 V. The choices of C and I_{CP} are related via (10), and for this example design they are 1.25 pF and 359 μA, respectively. Additionally, $I_{OC} = -I_{CP}$ (I_{OC} is defined in the Appendix) and $T_{OC} = 2$ ns. PSS simulations of the charge pump and offset current circuitry indicate that $e_p[n]$ can be modeled as white noise with a discrete-time PSD level of −64 dBV.

All the PSD plots in Figures 6 through 8 were obtained with the input noise source levels described above. The calculated PSD plots shown in the figures where obtained via (37), (38), (39), and (41). The simulated PSD plots shown in the figures where obtained via an event-driven C-language simulator. The simulator calculates the times of successive events, which include the positive-going zero crossings of $v_{ref}(t)$, $v_{div}(t)$, and $v_{out}(t)$, the sample times of the 5-level ADC, and the desired output sample times of the $\theta_{PLL}(t)$. Each event time is calculated as a function of the FDC-

PLL's state variables, and the state variables are updated at each event time.

Figure 6 shows the simulated and calculated PSD of $\theta_{PLL}(t)$ for several cases.[†] In one of the cases all of the noise sources presented above are considered together. In each of the other cases, only one of the noise sources is considered with all the other noise sources set to zero. Therefore, the figure shows how each noise source contributes to the total FDC-PLL output phase noise.

Figure 7 shows the simulated and calculated PSD of $\theta_{PLL}(t)$ for two cases to demonstrate the effect of the $L_{LPF}(z)$ portion of the FDC-PLL's loop filter. One case is that shown in Figure 6 for all the noise sources acting together. The other case differs only in that the simulation and calculations were made with $L_{LPF}(z)$ effectively disabled by setting its λ_i coefficients to 1, and K_I and K_P adjusted to maintain approximately the same phase margin and bandwidth as the first case.

Figure 8 shows the effect of typical non-ideal circuit behavior. The smooth curve is the same calculated PSD of $\theta_{PLL}(t)$ shown in Figure 6 for all the noise sources acting together. The jagged curve is the corresponding simulated PSD but with several non-ideal circuit effects taken into account in addition to noise. The non-ideal circuit effects involve the charge pump, offset current, sampling capacitor, and 5-level ADC. The magnitudes of the positive and negative charge pump current sources were increased and decreased, respectively, by 5%, and the offset current magnitude was decreased by 5%. A capacitor leakage current of −200 nA per reference period was introduced. Randomly chosen errors of 10 mV, −5 mV, 6 mV, and −8 mV, respec-

[†] In each plot, the smooth curves represent the calculated PSDs, and the jagged curves represent simulated PSDs.

tively, were introduced into the ADC threshold voltages. The errors were made larger than would be expected in practice to demonstrate the robustness of the FDC-PLL architecture.

The simulated and calculated results presented in Figures 6 and 7 demonstrate that the linearized model accurately predicts the expected phase noise performance of the example FDC-PLL for the considered evaluation settings. Furthermore, the simulation results presented in Figure 8 suggest that the FDC-PLL is robust with respect to non-ideal circuit behavior. Numerous additional FDC-PLL design parameters and evaluation cases considered by the authors have yielded consistently positive results.

VI. Appendix

This Appendix proves that the ΔΣ FDC followed by the α adder and accumulator in the digital loop controller perform the signal processing operations shown in Figure 3. It also applies known delta-sigma modulator results to draw various conclusions about the quantization noise introduced by the 5-level ADC.

The derivation consists of four parts. The first two parts derive expressions for the positive-going zero-crossing times of the reference oscillator and the divider output, respectively. The third part derives an expression for the voltage across the capacitor at the output of the charge pump. The fourth part combines the results of the previous parts to show that the ADC's quantization noise is that of a second-order delta-sigma modulator.

A. Reference Oscillator Zero-Crossing Time Derivation

Recall that t_k, for $k = 0, 1, 2, \ldots$, are the times of consecutive positive-going zero-crossings of the reference oscillator signal, $v_{ref}(t)$. The phase in cycles of an oscillator at each of its positive-going zero crossings is integer-valued, so the definition of t_k implies that the phase of the reference oscillator at time t_k is

$$p_{ref}\left(t_k\right) = k \tag{52}$$

for all non-negative integers k.

Exactly one reference oscillator cycle occurs during the time interval $t_{k-1} < t \leq t_k$, so it follows from (2) and (52) that

$$t_k - t_{k-1} = \frac{1 - \psi_{ref}[k]}{f_{ref}} \tag{53}$$

where

$$\psi_{ref}[k] = \theta_{ref}\left(t_k\right) - \theta_{ref}\left(t_{k-1}\right) \tag{54}$$

is the change in the reference oscillator's instantaneous phase noise in cycles between times t_{k-1} and t_k.[†] Summing (53) from $k = 1$ through any positive integer n yields

$$t_n = t_0 + \frac{1}{f_{ref}} \sum_{k=1}^{n} \left(1 - \psi_{ref}[k]\right). \tag{55}$$

B. Divider Output Zero-Crossing Time Derivation

Recall that τ_k, for each $k = 0, 1, 2, \ldots$, is the time of the positive-going zero-

[†] Note that $\psi_{ref}[n]$ is a different function than $\psi_{ref}(t)$, but they are related in that $\psi_{ref}[n]$ is proportional to the average of $\psi_{ref}(t)$ over the nth reference period. The functions $\psi_{PLL}[n]$ and $\psi_{PLL}(t)$ are similarly distinct.

crossing of the FDC-PLL's output signal, $v_{out}(t)$, that triggers the kth rising edge of the divider output, $v_{div}(t)$. Without loss of generality, assume that τ_k is indexed such that

$$p_{PLL}\left(\tau_0\right) = 0\,. \tag{56}$$

The kth output value of the 5-level ADC, $y[k]$, is a digitized sample of the charge pump capacitor voltage sampled after time τ_k, but well before time τ_{k+1}, and it follows from Figure 2 that

$$v[k] = 2y[k] - y[k-1]\,. \tag{57}$$

Therefore, the kth sample of $v[k]$ is available prior to time τ_{k+1}. The divider modulus is immediately updated when the sample is available such that exactly $N-v[k]$ DCO cycles occur during the time interval $\tau_k < t \leq \tau_{k+1}$. The definition of τ_k implies that

$$p_{PLL}\left(\tau_k\right) - p_{PLL}\left(\tau_{k-1}\right) = N - v[k-1]\,. \tag{58}$$

It follows from (5) and (58) that

$$\tau_k - \tau_{k-1} = \frac{N - v[k-1] - \psi_{PLL}[k]}{\left(N+\alpha\right) f_{ref}} \tag{59}$$

where

$$\psi_{PLL}[k] = \theta_{PLL}\left(\tau_k\right) - \theta_{PLL}\left(\tau_{k-1}\right) \tag{60}$$

is the change in the FDC-PLL's instantaneous output phase noise in cycles over the interval $\tau_{k-1} < t \leq \tau_k$. Summing (59) from $k = 1$ through any positive integer n yields

$$\tau_n = \tau_0 + \frac{1}{\left(N+\alpha\right) f_{ref}} \sum_{k=1}^{n} \left(N - v[k-1] - \psi_{PLL}[k]\right). \tag{61}$$

C. Charge Pump Output Derivation

Subtracting (55) from (61) gives

$$\begin{aligned}\tau_n - t_n &= \tau_0 - t_0 \\ &\quad + \frac{1}{\left(N+\alpha\right) f_{ref}} \sum_{k=1}^{n} \left(x[k] - v[k-1] \right)\end{aligned} \tag{62}$$

where

$$x[k] = -\alpha - \psi_{PLL}[k] + \left(N+\alpha\right)\psi_{ref}[k]. \tag{63}$$

The $-\alpha$ term in $x[n]$ can be interpreted as the phase change in cycles over one reference period of an ideal oscillator of frequency Nf_{ref} minus that of the ideal output of the FDC-PLL. The definitions of $\psi_{ref}[k]$ and $\psi_{PLL}[k]$ imply that the average value of $x[k]+\alpha$ is zero when the FDC-PLL is locked, and that $(x[k]+\alpha)/T_{ref}$ is a measure of the average over the kth reference period of the difference between $(N+\alpha)$ times the instantaneous frequency of the reference oscillator and the instantaneous frequency of the output signal.

It follows from (5) and (56) that

$$\tau_0 - t_0 = -\frac{\theta_{PLL}\left(\tau_0\right)}{\left(N+\alpha\right) f_{ref}}. \tag{64}$$

Substituting (57) and (64) into (62) gives

$$\begin{aligned}\left(\tau_n - t_n\right)\left(N+\alpha\right) f_{ref} &= -y[n-1] + \sum_{k=1}^{n} \left(x[k] - y[k-1] \right) \\ &\quad + y[-1] - \theta_{PLL}\left(\tau_0\right).\end{aligned} \tag{65}$$

Suppose the FDC-PLL is locked for all $t \geq t_0$ so that

$$\left|\tau_n - t_n\right| < T_{ref} \tag{66}$$

for all $n \geq 0$, where $T_{ref} = 1/f_{ref}$ is the nominal period of the reference oscillator. If the PFD and charge pump are as shown in Figure 1, then in the absence of non-ideal circuit behavior the output of the charge pump is a sequence of current pulses given by

$$i_{cp}(t) = \begin{cases} I_{CP} & \text{when } t_n \leq t \leq \tau_n \\ -I_{CP} & \text{when } \tau_n \leq t \leq t_n \\ 0 & \text{otherwise} \end{cases} \tag{67}$$

for all positive integers n. An additional current pulse of fixed duration and fixed (positive or negative) amplitude may also be included in $i_{cp}(t)$ each reference period to reduce nonlinear distortion introduced by the PFD and charge pump [23], [24].

The ADC samples the capacitor voltage each reference period shortly after the charge pump current sources settle to zero. Let $V_c[n]$ be the voltage sampled by the ADC during the nth reference period minus the midscale voltage of the ADC (i.e., $V_c[n] = 0$ corresponds to the middle of the ADC's input range). The operation of the charge pump implies that

$$V_c[n] = V_c[n-1] + \left(\tau_n - t_n\right)\frac{I_{CP}}{C} + T_{OC}\frac{I_{OC}}{C} + e_p[n] \tag{68}$$

where T_{OC} and I_{OC} are the duration and amplitude, respectively, of the additional current pulse if it is used (otherwise $I_{OC} = 0$), and $e_p[n]$ represents the combined error from noise and other non-ideal circuit behavior in the charge pump, PFD, and divider. Each sample of $e_p[n]$ is the result of error in the amount of charge in the current pulses integrated onto the capacitor during the nth reference period.

Substituting (10) and (65) into (68) results in

$$\begin{aligned}\frac{V_c[n]}{\Delta} = \frac{V_c[n-1]}{\Delta} - y[n-1] + \sum_{k=1}^{n}\left(x[k]-y[k-1]\right) \\ +y[-1]-\theta_{PLL}\left(\tau_0\right)+\theta_{offset}+\frac{e_p[n]}{\Delta}\end{aligned} \tag{69}$$

where

$$\theta_{offset} = T_{OC}\frac{I_{OC}}{C\Delta}. \tag{70}$$

D. Delta-Sigma Modulator Equivalence and Implications

The output of the ADC, $y[n]$, can take on values from the set {−2, −1, 0, 1, 2} and can be written as

$$y[n] = \frac{1}{\Delta}V_c[n] + e_{ADC}[n] \tag{71}$$

where $e_{ADC}[n]$ is the sum of quantization noise and any additional error from non-ideal circuit behavior in the ADC. The ADC has only five levels, so its quantization is very coarse. Therefore, it is assumed that the only non-negligible component of $e_{ADC}[n]$ is quantization noise, so the *n*th output sample of the ADC is taken to be $V_c[n]/\Delta$ rounded to the nearest integer when

$$-2.5\Delta \le V_c[n] \le 2.5\Delta, \tag{72}$$

and −2 or 2, respectively, when $V_c[n]$ is less than -2.5Δ or greater than 2.5Δ.

Equations (69) and (71) are equivalent to the block diagram shown in Figure 3 to the left of the α adder for n = 1, 2, 3, …, where

$$u_2[n] = \frac{V_c[n]}{\Delta} \tag{73}$$

and the initial condition on $u_1[n]$ is

$$u_1[0] = y[-1] - \theta_{PLL}\left(\tau_0\right) + \theta_{offset} . \tag{74}$$

The block diagram has the well-known form of a second-order delta-sigma modulator, so its output can be written as

$$y[n] = x[n] + \frac{e_p[n] - e_p[n-1]}{\Delta} + e_{\Delta\Sigma 2}[n] \tag{75}$$

where

$$e_{\Delta\Sigma 2}[n] = e_{ADC}[n] - 2e_{ADC}[n-1] + e_{ADC}[n-2] \tag{76}$$

is second-order highpass shaped quantization noise [25, 26].

If (72) is satisfied for a given integer n, then $e_{ADC}[n]$ is the quantization noise caused by rounding $u_2[n]$ to the nearest integer. In this case the delta-sigma modulator is said to be non-overloading at time n. Otherwise, the delta-sigma modulator is said to be overloaded at time n. If the delta-sigma modulator is non-overloading for all n = 1, 2, 3, …, then $e_{ADC}[n]$ is asymptotically white and uniformly distributed between −0.5 and 0.5 under the realistic assumption that $x[n]$ contains a small amount of independent random noise [27]. In contrast, if the delta-sigma modulator becomes overloaded, then $e_{ADC}[n]$ becomes correlated with $x[n]$, its variance increases, and it often contains spurious tones. Hence, for best phase noise performance it is desirable to keep the delta-sigma modulator non-overloading once the FDC-PLL is locked.

Sufficient conditions for the delta-sigma modulator to be non-overloading for n = 1, 2, 3, … are that it is non-overloading for n = 1 and n = 2, and

$$\left| x[n] + \frac{e_p[n] - e_p[n-1]}{\Delta} \right| \le 1 \tag{77}$$

for n = 3, 4, 5, …. The proof of this result is as follows. It can be verified from Figure 3 that

$$\begin{aligned} u_2[k] = x[k] + \frac{e_p[k] - e_p[k-1]}{\Delta} \\ -2e_{ADC}[k-1] + e_{ADC}[k-2] \end{aligned} \tag{78}$$

If the delta-sigma modulator is non-overloading for $n = k - 1$ and $n = k - 2$, then $e_{ADC}[k{-}1]$ and $e_{ADC}[k{-}2]$ are each bounded in magnitude by 0.5, so (73) and (78) imply that

$$\left|V_c[k]\right| \le \left(\left| x[k] + \frac{e_p[k] - e_p[k-1]}{\Delta} \right| + 1.5 \right) \Delta\,. \tag{79}$$

This implies that (72) is satisfied and therefore that the delta-sigma modulator is non-overloading for $n = k$ provided (77) holds for $n = k$. The result follows from induction.

It follows from (63) that (77) is satisfied for any α in the range given by (9) if

$$\left| \frac{e_p[n] - e_p[n-1]}{\Delta} - \psi_{PLL}[n] + \left(N + \alpha\right)\psi_{ref}[n] \right| \le \frac{1}{2}\,. \tag{80}$$

Frequency synthesizers usually are designed to have low phase noise, so the left side of (80) is expected to be far less than ½ in practice. Furthermore, in most practical cases the magnitude of α is much larger than the left side of (80). In such cases it can be verified that only four of the five ADC levels are exercised once the FDC-PLL is locked. Thus, the five ADC levels are easily sufficient to ensure that the delta-sigma modulator remains non-overloading once the FDC-PLL is locked, which also ensures that $e_{ADC}[n]$ does not contain spurious tones induced by quantizer overloading.

Nevertheless, it can be verified from well-known delta-sigma modulator properties, that when the magnitude of α is 0.5, four ADC levels would only be sufficient to ensure that the delta-sigma modulator remains non-overloading in the absence of any noise other than quantization noise. Therefore, in practice five ADC levels are necessary to avoid overloading for values of α with magnitudes close to 0.5.

Substituting (63) into (75) gives

$$\begin{aligned} y[n]+\alpha = -\psi_{PLL}[n]+\left(N+\alpha\right)\psi_{ref}[n] \\ +\frac{e_p[n]-e_p[n-1]}{\Delta}+e_{\Delta\Sigma 2}[n]. \end{aligned} \tag{81}$$

This sequence is accumulated prior to the loop filter, so the input to the loop filter can be written as (14) neglecting a possible offset that depends on the initial value of the accumulator output.

ACKNOWLEDGEMENTS

Chapter 1, in full, has been published in the IEEE Transactions on Circuits and Systems I: Regular Papers, volume 60, number 5, pages 1274-1285, May 2013. 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.

FIGURES

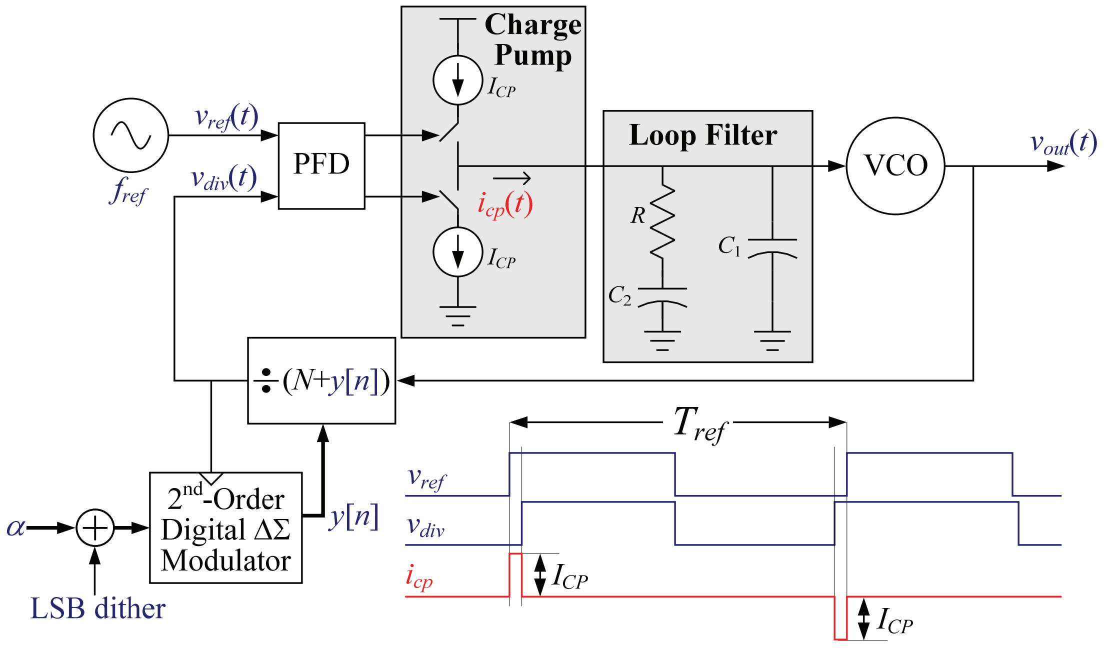


Figure 1: A delta-sigma modulator based fractional-*N* PLL (ΔΣ-PLL)

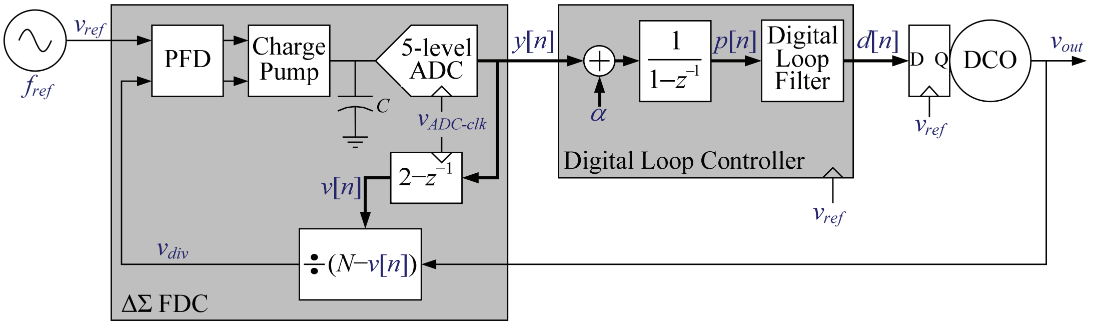


Figure 2: A delta-sigma FDC based fractional-*N* PLL (FDC-PLL)

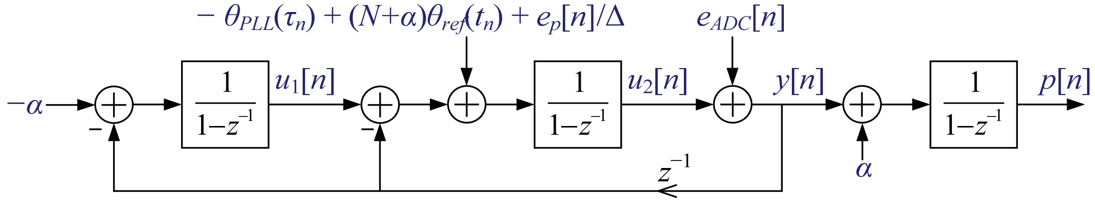


Figure 3: The implicit second-order delta-sigma modulator implemented by the ΔΣ FDC followed by the *α* adder and accumulator of the digital loop controller

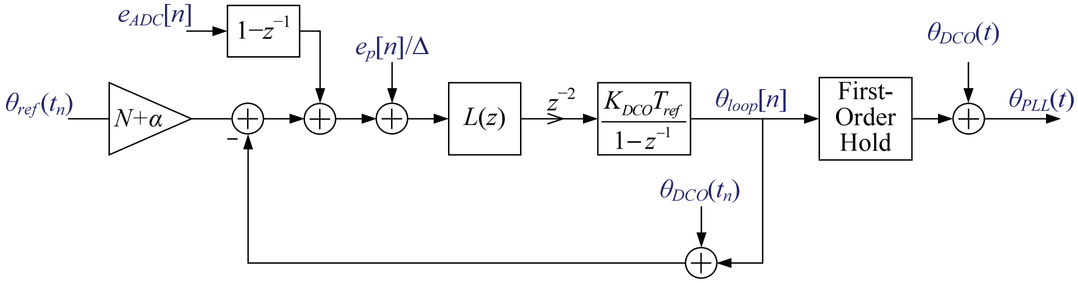


Figure 4: Phase noise model of the FDC-PLL

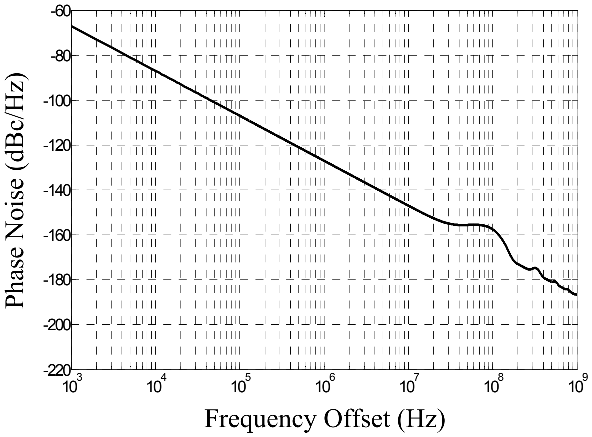


Figure 5: PSD of $\theta_{DCO}(t)$ used in the design and simulation of the FDC-PLL example

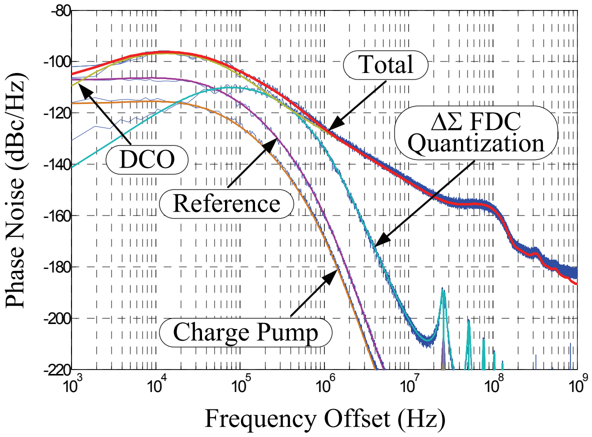


Figure 6: Calculated PSD plots (smooth curves) and simulated PSD plots (jagged curves) of the FDC-PLL output phase noise resulting from each of the noise sources individually and all together

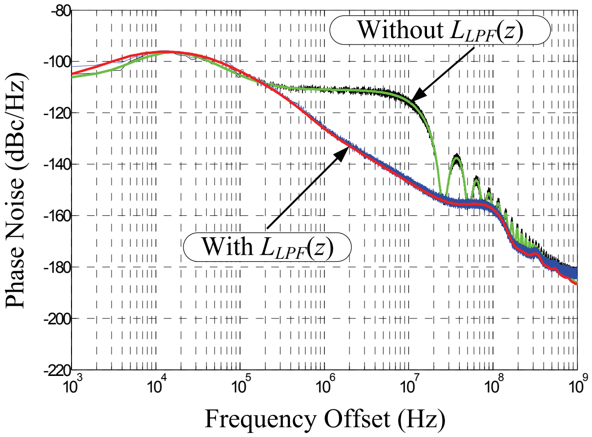


Figure 7: Calculated and simulated PSD plots of the FDC-PLL output phase noise with all noise sources with and without the $L_{LPF}(z)$ portion of the loop filter

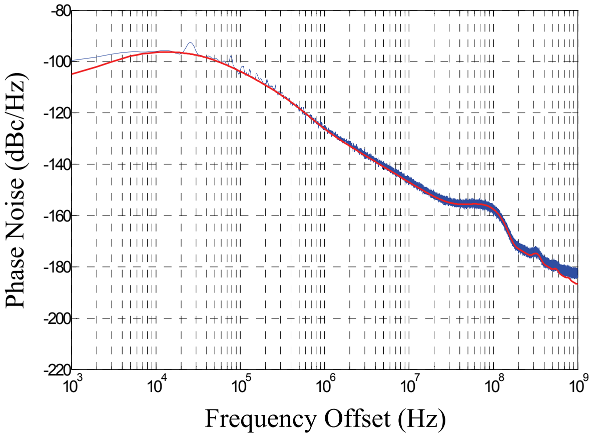


Figure 8: Calculated PSD plot of the FDC-PLL output phase noise with all noise sources, and the corresponding simulated PSD with several non-ideal circuit effects taken into account in addition to noise

TABLES

Table 1: Parameters and evaluation settings of the example FDC-PLL design

Design Parameters and Evaluation Settings	Value
f_{ref}	26 MHz
f_{PLL}	3588.026 MHz
N	138
α	0.001
K_{DCO}	24 kHz
V_{DD}	1.0 V
Δ	80 mV
ADC Input Range	0.3-0.7 V
C	1.25 pF
I_{CP}	359 μA
I_{OC}	$-I_{CP}$
T_{OC}	2 ns
T_{DZ}	1 ns
K_P	$2^{-7} f_{ref} / K_{DCO}$
K_I	$2^{-17} f_{ref} / K_{DCO}$
λ_0, λ_1, λ_2, λ_3	2^{-2}, 2^{-2}, 2^{-3}, 2^{-4}
Loop-filter Word Width	32 bits
DCO Input Word Width	14 bits
DCO's ΔΣ Modulator Input Word Width	8 bits
DCO's ΔΣ Modulator Update Rate	$f_{PLL}/16$

References

1. B. Miller, B. Conley, “A multiple modulator fractional divider,” *Annual IEEE Symposium on Frequency Control*, vol. 44, pp. 559-568, March 1990.

2. B. Miller, B. Conley, “A multiple modulator fractional divider,” *IEEE Transactions on Instrumentation and Measurement*, vol. 40, no. 3, pp. 578-583, June 1991.

3. 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.

4. R. Best, *Phase-Locked Loops: Design, Simulation, and Applications*. Sixth Edition, New York, NY: McGraw-Hill, 2007.

5. Kajiwara, M, Nakagawa, “A new PLL frequency synthesizer with high switching speed”, *IEEE Transactions on Vehicular Technology*, vol. 41, pp. 407-413, November 1992.

6. R. B. Staszewski et al., “All-Digital TX Frequency Synthesizer and Discrete-Time Receiver for Bluetooth Radio in 130-nm CMOS”, *IEEE J. Solid-State Circuits*, vol. 39, pp. 2278-2291, December 2004.

7. R. B. Staszewski, et. al., “All-Digital PLL and Transmitter for Mobile Phones,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2469-2482, December, 2005.

8. K. Muhammad, et. al., “The first fully integrated quad-band GSM/GPRS receiver in a 90-nm digital CMOS process,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp.1772-1783, August 2006

9. C. Weltin-Wu, E. Temporiti, D. Baldi, F. Svelto, “A 3GHz Fractional-N All-Digital PLL with Precise Time-to-Digital Converter Calibration and Mismatch Correction”, *IEEE International Solid-State Circuits Conference*, pp. 344-345, February 2008.

10. C. 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, pp. 340-341, Feb. 2008.

11. I. Galton, G. Zimmerman, "Combined RF phase extraction and digitization," *IEEE International Symposium on Circuits and Systems*, vol. 2, pp.1104-1107, May 1993.

12. R. D. Beards, M. A. Copeland, "An Oversampling Delta-Sigma Frequency Discriminator," *IEEE Transactions on Circuits and Systems II: Analog and Digital Processing*, vol. 41, no. 1, pp. 26-32, January 1994.

13. I. Galton, "Analog-input digital phase-locked loops for precise frequency and phase demodulation," *IEEE Transactions on Circuits and Systems II: Analog and Digital Processing*, vol. 42, no. 10, pp. 621-630, November 1995.

14. I. Galton, W. Huff, P. Carbone, E. Siragusa, "A delta-sigma PLL for 14b 50kSample/s frequency-to-digital conversion of a 10 MHz FM signal," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 2042-2053, December 1998.

15. W. T. Bax, T. A. D. Riley, C. Plett, M. A. Copeland, "A Σ-Δ Frequency Discriminator Based Synthesizer," *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 1-4, May 1995.

16. W. T. Bax, M. A. Copeland, "A GMSK Modulator Using a ΔΣ Frequency Discriminator-Based Synthesizer," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 8, pp. 1218-1227, August 2001.

17. 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.

18. 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.

19. R. B. Staszewski et al., "A First RF Digitally-Controlled Oscillator for Mobile Phones," *IEEE Radio Frequency integrated Circuits (RFIC) Symposium*, pp. 119-122, June 2005.

20. F. Gardner, *Phaselock Techniques*, Third Edition, John Wiley, Hoboken, 2005.

21. J. Proakis, D. Manolakis, *Digital Signal Processing*, Third Edition, Upper Saddle River, Prentice Hall, 1996.

22. B. Razavi, C. Lam, "A 2.6-GHz/5.2-GHz frequency synthesizer in 0.4-μm CMOS technology," *Symposium on VLSI Circuits*, pp.117-120, 1999.

23. E. Temporiti, G. Albasini, I. Bietti, R. Castello, and M. Colombo, "A 700 kHz bandwidth SD fractional sythesizer with spurs compensation and linearization tecniques for WCDMA applications," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1446–1454, Sep. 2004.

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

25. G. Lainey, R. Saintlaurens, P. Senn, "Switched-capacitor second-order noise-shaping coder," *Electronics Letters*, vol. 19, pp. 149-150, February 1983.

26. R. Schreier, G. C. Temes, *Understanding Delta-Sigma Data Converters*, John Wiley and Sons, 2005.

27. I. Galton, "Granular quantization noise in a class of delta-sigma modulators," *IEEE Transactions on Information Theory*, vol. 40, no.3, pp.848-859, May 1994.

Chapter 2

Multi-Rate Quantizing Dynamic Element Matching For Oversampling Digital-to-Analog Conversion

Abstract—Mismatch-shaping dynamic element matching (DEM) is widely used in high-performance oversampling delta-sigma (ΔΣ) DACs because it prevents component mismatches from limiting performance. In such DACs, both the DAC mismatch noise and ΔΣ quantization noise are shaped such that most of their power lies outside the signal band. Typically, the power of the ΔΣ quantization noise is higher than that of the DAC mismatch noise, so a higher oversampling ratio is used than would be necessary in the absence of ΔΣ quantization noise. This paper presents the first DEM technique that allows different oversampling ratios to be applied to the two noise sources. The technique allows much of the DEM circuitry and all but a subset of the lowest-weighted DAC elements to run at a lower rate than would be necessary for comparable performance in a conventional oversampling DAC. This reduces power dissipation and DAC element switching noise.

I. Introduction

Multi-bit quantization has all but supplanted single-bit quantization in high-performance oversampling delta-sigma (ΔΣ) DACs, resulting in significant DAC per-

This work was supported by the National Science Foundation under Award 0914748.

formance improvements over the last 15 years. Dynamic element matching (DEM) has enabled this transition by preventing component mismatches and layout asymmetries from limiting performance in multi-bit designs [28-45].

The 1-bit DACs that make up the coarse multi-bit DAC within a ΔΣ DAC are subject to pulse shape, timing, and amplitude errors as a result of component mismatches that inevitably arise during the manufacturing process and layout asymmetries. The resulting error component in the ΔΣ DAC output is called *DAC mismatch noise*. Without DEM, the DAC mismatch noise would be a nonlinear function of the coarse multi-bit DAC's input sequence, so even very small mismatches and asymmetries would be problematic in high-performance applications. DEM addresses this problem. It scrambles the usage pattern of the 1-bit DACs such that the DAC mismatch noise is uncorrelated with the coarse multi-bit DAC's input sequence, is free of spurious tones and nonlinear distortion, and, in the case of mismatch-shaping DEM, is spectrally shaped such that most of its power lies outside of the ΔΣ DAC's signal band [28,46].

In previously published ΔΣ DACs all the 1-bit DACs in the coarse multi-bit DAC participate in the digital-to-analog conversion of the ΔΣ quantization noise. Therefore, their oversampling ratio must be high enough that the ΔΣ quantization noise is sufficiently attenuated within the ΔΣ DAC's signal band. However, the power of the ΔΣ quantization noise typically is much higher than that of the DAC mismatch noise. Therefore, if it were not necessary to suppress the signal-band portion of the ΔΣ quantization noise, the 1-bit DAC update-rate could be reduced because the oversampling ratio would only need to be high enough that the signal-band portion of the DAC mismatch noise is sufficiently attenuated.

This paper presents a new DEM technique that incorporates ΔΣ modulation such that only a subset of the lowest-weighted 1-bit DACs participate in the digital-to-analog conversion of the ΔΣ quantization noise. Therefore, only these 1-bit DACs need to run at the full sample-rate. The other 1-bit DACs do not carry information about the ΔΣ quantization noise, so their update rate and the clock-rate of the DEM logic which drives them can be reduced. This reduces power dissipation. Furthermore, reducing the update-rate of most of the 1-bit DACs reduces the number of 1-bit DAC transitions per unit time which reduces the power of the DAC mismatch noise component caused by 1-bit DAC transient mismatches [47].

The remainder of the paper consists of three sections. Section II presents an overview of continuous-time segmented DEM DACs on which the results of this paper build. Section III presents the new DEM technique in a general form, and Section IV presents a specific oversampling DAC example enabled by the new technique.

II. Continuous-Time DEM DAC Overview

A. Ideal Continuous-Time DAC Behavior

A continuous-time DAC converts a sequence of digital codewords into a continuous-time analog waveform, $y(t)$. The digital codewords represent a sequence of numbers, $x[n]$, for n = 0, 1, 2, …, each of which has a value in the set

$$\left\{-\frac{M}{2}\Delta,\ -\left(\frac{M}{2}-1\right)\Delta,\ -\left(\frac{M}{2}-2\right)\Delta, \ldots,\ \frac{M}{2}\Delta\right\} \tag{82}$$

where M is a positive integer and Δ is the minimum *step-size* of $x[n]$. The nth sample-interval of the DAC, for each n = 0, 1, 2, …, is defined as the time interval $nT \leq t < (n + 1)T$, where T is the sample-interval duration. Ideally, during the nth sample-interval, the output of the DAC is given by

$$y(t) = x[n]a(t - nT)\,. \tag{83}$$

where $a(t)$ is a *unit analog pulse* that is zero outside of $0 \leq t < T$.

B. Ideal Continuous-Time DAC Behavior

A general DEM DAC architecture is shown in Figure 1 [39]. It consists of an all-digital block called a *DEM encoder*, followed by N 1-bit DACs. During the nth sample-interval, the output of the ith 1-bit DAC is

$$y_i(t) = \begin{cases} K_i \dfrac{\Delta}{2} a(t-nT) + e_{hi}(t-nT), & \text{if } c_i[n] = 1, \\ -K_i \dfrac{\Delta}{2} a(t-nT) + e_{li}(t-nT), & \text{if } c_i[n] = 0, \end{cases} \tag{84}$$

where $c_i[n]$ is the input to the 1-bit DAC, K_i is the *weight* of the 1-bit DAC, and $e_{hi}(t)$ and $e_{li}(t)$ are the 1-bit DAC’s *mismatch error pulses*. The mismatch error pulses are unique to each 1-bit DAC and represent the 1-bit DAC’s pulse shape, timing, and amplitude errors caused by non-ideal circuit behavior such as component mismatches and layout asymmetries. The only assumption made about $e_{hi}(t)$ and $e_{li}(t)$ is that they are zero outside of $0 \leq t < T$. By definition, $K_1 = 1$ and each K_i for i = 2, 3, …, N, is a positive integer multiple of K_{i-1} with

$$\sum_{i=1}^{N} K_i = M \ . \tag{85}$$

The DEM encoder sets each of its 1-bit output sequences at time n to 0 or 1 such that

$$\Delta \sum_{i=1}^{N} K_i \left(c_i[n] - \tfrac{1}{2} \right) = x[n] \tag{86}$$

which ensures that the DAC output during the nth sample-interval would be given by (83) in the absence of mismatch error pulses. As shown in [39], this implies that the DEM encoder's 1-bit output sequences can be written as

$$c_i[n] = \left(m_i x[n] + \lambda_i[n] \right) / \Delta + \tfrac{1}{2} \tag{87}$$

for $i = 1, 2, \ldots, N$, where the m_i are constants and $\lambda_i[n]$ are sequences which satisfy

$$\sum_{i=1}^{N} K_i m_i = 1, \quad \text{and} \quad \sum_{i=1}^{N} K_i \lambda_i[n] = 0, \tag{88}$$

respectively. As also shown in [39], this ensures that during the nth sample-interval the DAC output is

$$y(t) = \alpha(t - nT)x[n] + \beta(t - nT) + e_{DAC}(t) \tag{89}$$

where $\alpha(t)$ and $\beta(t)$ are pulses that are zero outside of $0 \le t < T$, and

$$e_{DAC}(t) = \sum_{i=1}^{N} K_i \lambda_i[n] \alpha_i(t - nT) \, , \tag{90}$$

where

$$\alpha_i(t) = a(t) + \frac{e_{hi}(t) - e_{li}(t)}{K_i \Delta} \tag{91}$$

In the ideal case of zero mismatch error pulses, $\alpha(t) = a(t)$, $\beta(t) = 0$, and

$e_{DAC}(t) = 0$, so (89) reduces to (83). Although non-zero mismatch error pulses cause $\alpha(t)$ to deviate from its ideal pulse shape of $a(t)$ and give rise to a train of *offset pulses*, each with a shape given by $\beta(t)$, these effects do not introduce nonlinear distortion and therefore are tolerable in most applications. However, $e_{DAC}(t)$, which is the *DAC mismatch noise* mentioned in the introduction, is related to the input sequence and would introduce nonlinear distortion if it were not for the DEM encoder.

In DEM DACs, the 1-bit DAC weights are such that for most values that $x[n]$ can take on there are multiple sets of DEM encoder output bit values that satisfy (86). The DEM encoder exploits this flexibility to ensure that each of the $\lambda_i[n]$ sequences in (87) is a noise-like sequence that is zero mean, free of spurious tones, and uncorrelated with $x[n]$. It follows from (90) that this ensures $e_{DAC}(t)$ has a noise-like structure, is free of spurious tones, and satisfies

$$\mathrm{E}\left\{e_{DAC}\left(t\right)\right\} = 0 \quad \text{and} \quad \mathrm{E}\left\{e_{DAC}\left(t\right)x\left[n\right]\right\} = 0 \tag{92}$$

regardless of the mismatch error pulses. Achieving these objectives ensures that the DAC does not introduce non-linear distortion. Depending on the application, the DEM encoder algorithm may be designed to impart additional properties to the $\lambda_i[n]$ sequences, such as spectral shaping to suppress the power of $e_{DAC}(t)$ within one or more frequency bands.

As shown in [48] it is possible to achieve these objectives only if $x[n]$ is restricted so that it never takes on the smallest $K_N - 1$ values or the largest $K_N - 1$ values in (82). As explained in [48], this range restriction requirement represents a fundamental tradeoff between power consumption and complexity in DEM DACs. In the special case of DEM DACs with unity-weighted 1-bit DACs (i.e., $K_i = 1$ for $i = 1, 2, \ldots, N$), the objectives can be achieved for the full range specified by (82) because $K_N - 1 = 0$.

C. A General Oversampling DEM DAC

The 1-bit DACs in the DEM DAC shown in Figure 9 are said to have a *weight spread* of K_N, because the weight of the largest 1-bit DAC is K_N times that of the smallest 1-bit DAC. Unfortunately, it is not always practical to design 1-bit DACs with sufficiently large weight spreads to achieve the desired resolution. Oversampling ΔΣ DACs can be used to circumvent this problem in applications where it is practical for the sample rate to be many times higher than the Nyquist rate of the desired output signal [28].

A general conventional oversampling ΔΣ DAC architecture is shown in Figure 10. It consists of a digital ΔΣ modulator and a DEM DAC. The input sequence, $x[n]$, represents a bandlimited continuous-time desired output signal with a Nyquist rate of f_N sampled at a rate of Rf_N, where R is greater than 1 and is called the *oversampling ratio*. The digital ΔΣ modulator quantizes $x[n]$ such that each quantized sample is a multiple of $K_{J+1}\Delta$ and the quantization noise sequence, i.e., the component of the ΔΣ modulator output arising from quantization error, is spectrally shaped so as to reside mostly above the $f_N/2$ bandwidth of the desired signal. The DEM DAC has the form shown in Figure 9 but its smallest 1-bit DAC has a weight of K_{J+1} instead of unity.

In such oversampling ΔΣ DACs $y(t)$ is subjected to some form of analog filtering (not shown in Figure 10) to remove most of the quantization noise and out-of-band DAC mismatch noise. The resulting waveform is a high-resolution representation of $x[n]$ despite the relatively large step-size of the smallest 1-bit DAC.

III. Multi-Rate Quantizing DEM

A. Problem Statement

Oversampling ΔΣ DACs of the type shown in Figure 10 are ubiquitous in modern communications and consumer electronics applications. However, they are subject to inefficiency if the range spanned by $x[n]$ is greater than twice the quantization step-size of the digital ΔΣ modulator, i.e., if

$$\max_{n}\left\{x[n]\right\} - \min_{n}\left\{x[n]\right\} > 2K_{J+1}\Delta\,. \tag{93}$$

In an oversampling ΔΣ DAC with a given quantization step-size, increasing the range spanned by $x[n]$ increases the signal-to-quantization noise ratio so it is desirable to make this span as large as is practical. Accordingly, (93) is usually satisfied in practice.

As demonstrated below, in such cases it is possible to design an oversampling ΔΣ DAC that achieves comparable performance to that shown in Figure 10, but wherein the update-rate of all but a subset of its lowest-weighted 1-bit DACs and the clock-rate of the DEM encoder logic that drives them is $(R/Q)f_N$ instead of Rf_N, where Q is an integer in the range $1 < Q \le R$. Each time Q is doubled, the power dissipation of this DEM encoder logic and the switch driver circuitry within these 1-bit DACs is halved, so a value of Q as low as 2 can significantly reduce power dissipation.

Furthermore, having the update-rate of all but a subset of the lowest-weighted 1-bit DACs be $(R/Q)f_N$ instead of Rf_N reduces the number of 1-bit DAC transitions per unit time compared to that in a comparable conventional oversampling ΔΣ DAC. It also reduces the average magnitude of the transitions relative to that in a conventional oversampling ΔΣ DAC because only lowest-weight 1-bit DACs are updated at the higher frequency.

Although DEM prevents error associated with mismatches among the transitions from introducing nonlinear distortion, it does not prevent them from introducing noise. Therefore, reducing the number of 1-bit DAC transitions per unit time and their average magnitude tends to decrease the oversampling DAC's noise power component arising from 1-bit DAC transition mismatches. In general, each time Q is doubled the power of the noise resulting from transient mismatches is reduced by up to 3 dB, because for most of the 1-bit DACs the transients occur half as often. However, doubling Q also halves the oversampling ratio for this noise component, so whether or not the power of the signal band portion of the noise arising from transient mismatches is reduced relative to that in a comparable conventional ΔΣ DAC depends on the extent of the transient mismatches among the 1-bit DACs.

B. A General MRQ-DEM DAC

A general version of the proposed oversampling DAC which achieves the objectives outlined above is shown in Figure 11. It is called a multi-rate quantizing DEM (MRQ-DEM) DAC. It consists of an MRQ-DEM encoder followed by a bank of 1-bit DACs, the details of which are described below. Its input sequence, $x[n]$, represents a bandlimited continuous-time desired output signal with a Nyquist rate of f_N sampled at a rate of $(R/Q)f_N$. Therefore, the oversampling ratio of the input sequence is R/Q.

There are two types of 1-bit DACs in the MRQ-DEM DAC. The 1-bit DACs that appear in the figure without shading are identical to the corresponding 1-bit DACs in the DEM DAC shown in Figure 9. Their input sequences are updated at a rate of $(R/Q)f_N$ and their behavior is described by (84) with $T = \mathrm{Q}/(Rf_N)$. The 1-bit DACs that are shaded in the figure are driven by input sequences that are updated at a rate of Rf_N. During the time interval $mT/Q \le t < (m + 1)T/Q$ for each $m = 0, 1, \ldots$, the output of the ith of these 1-bit DACs for each $i = 1, 2, \ldots, L$ is

$$y'_i(t) = \begin{cases} K_{J+1}\dfrac{\Delta}{2}a'\left(t - m\dfrac{T}{Q}\right) + e'_{hi}\left(t - m\dfrac{T}{Q}\right), & \text{if } c'_i[m] = 1, \\ -K_{J+1}\dfrac{\Delta}{2}a'\left(t - m\dfrac{T}{Q}\right) + e'_{li}\left(t - m\dfrac{T}{Q}\right), & \text{if } c'_i[m] = 0, \end{cases} \tag{94}$$

where $c'_i[m]$ is the input to the 1-bit DAC, $a'(t)$ is the unit analog output pulse, and $e'_{hi}(t)$ and $e'_{li}(t)$ are the 1-bit DAC's *mismatch error pulses*. It is assumed that $a'(t)$, $e'_{hi}(t)$, and $e'_{li}(t)$ are all zero outside of $0 \le t < T/Q$.

The MRQ-DEM encoder consists of a modified version of the DEM encoder in Figure 9, a digital ΔΣ modulator, and a local DEM encoder (so-named because it drives only a subset of the 1-bit DAC bank). The clock-rate of the modified DEM encoder is $(R/Q)f_N$, and that of the digital ΔΣ modulator and local DEM encoder is Rf_N. The modified DEM encoder's $c_{J+1}[n]$, $c_{J+2}[n]$, …, $c_N[n]$ output bits are identical to those of the DEM DAC shown in Figure 9. Its $x_f[n]$ output sequence is given by

$$x_f[n] = \Delta\sum_{i=1}^{J} K_i\left(c_i[n] - \tfrac{1}{2}\right) \tag{95}$$

where Δ, M, K_i and $c_i[n]$ for i = 1, 2, …, J, are the same as those of the DEM DAC shown in Figure 9 and described in Section II-B. The digital ΔΣ modulator is clocked Q times faster than the modified DEM encoder, so it samples each value of $x_f[n]$ Q times. Hence, its Rf_N -rate output sequence is

$$x_{\Delta\Sigma}[m] = x_f\left[\left\lfloor m/Q \right\rfloor\right] + d_{\Delta\Sigma}[m] + e_{\Delta\Sigma}[m] \tag{96}$$

where $\lfloor m/Q \rfloor$ denotes the largest integer less than or equal to m/Q, $d_{\Delta\Sigma}[m]$ represents the component of the digital ΔΣ modulator's output sequence arising from dither, and $e_{\Delta\Sigma}[m]$ is the digital ΔΣ modulator's quantization noise.[2]

It follows from Figure 11 that the output of the MRQ-DEM DAC can be written as

$$y(t) = y_{\Delta\Sigma}(t) + \sum_{i=J+1}^{N} y_i(t) \tag{97}$$

where each $y_i(t)$ is the output of the 1-bit DAC driven by $c_i[n]$, and $y_{\Delta\Sigma}(t)$ is the sum of the outputs of the L 1-bit DACs driven by the local DEM encoder, i.e.,

$$y_{\Delta\Sigma}(t) = \sum_{i=1}^{L} y'_i(t)\,. \tag{98}$$

Thus, $y_{\Delta\Sigma}(t)$ can be viewed as the output of a *local* DEM DAC which consists of the local DEM encoder and the subsequent L 1-bit DACs. The local DEM DAC has the form of that shown in Figure 9, except that all of its 1-bit DACs have a weight

[2] Although not shown explicitly in Figures 2 and 3, some form of dither typically is applied to digital ΔΣ modulators to ensure that their quantization noise is well-behaved [28, 50, 51].

of K_{J+1}. Its update-rate is Rf_N, so its mth sample-interval, for each $m = 0, 1, 2, \ldots$, is defined as the time interval $mT/Q \le t < (m + 1)T/Q$.

By the same arguments outlined in Section II and proven in [39], the local DEM DAC's output during its mth sample-interval can be written as

$$y_{\Delta\Sigma}(t) = \alpha'\left(t - m\frac{T}{Q}\right)x_{\Delta\Sigma}[m] + \beta'\left(t - m\frac{T}{Q}\right) + e'_{DAC}(t) \tag{99}$$

with

$$\alpha'(t) = K_{J+1}\sum_{i=1}^{L} m'_i\,\alpha'_i(t), \qquad \beta'(t) = \sum_{i=1}^{L}\beta'_i(t), \tag{100}$$

and

$$e'_{DAC}(t) = K_{J+1}\sum_{i=1}^{L}\lambda'_i[m]\alpha'_i\left(t - m\frac{T}{Q}\right), \tag{101}$$

where

$$\beta'_i(t) = \frac{e'_{hi}(t) + e'_{li}(t)}{2}, \tag{102}$$

$$\alpha'_i(t) = a'(t) + \frac{e'_{hi}(t) - e'_{li}(t)}{K_{J+1}\Delta} \tag{103}$$

and the constants, m'_i, and sequences, $\lambda'_i[m]$, satisfy

$$K_{J+1}\sum_{i=1}^{L} m'_i = 1 \quad \text{and} \quad \sum_{i=1}^{L}\lambda'_i[m] = 0, \tag{104}$$

respectively. In analogy to the DEM encoder in the DEM DAC of Figure 9, the purpose of the local DEM encoder is to ensure that $e'_{DAC}(t)$ has a noise-like structure, is free of spurious tones, and satisfies

$$\mathrm{E}\left\{e'_{DAC}\left(t\right)\right\}=0 \quad \text{and} \quad \mathrm{E}\left\{e'_{DAC}\left(t\right)x_{\Delta\Sigma}\left[m\right]\right\}=0 \tag{105}$$

regardless of the mismatch error pulses of the shaded 1-bit DACs, $e'_{hi}(t)$ and $e'_{li}(t)$. Achieving these objectives ensures that the local DEM DAC does not introduce nonlinear distortion, which is a necessary condition for the MRQ-DEM DAC not to introduce nonlinear distortion. As in the case of the DEM encoder shown in Figure 9, the local DEM encoder algorithm may be designed to impart additional properties to the $\lambda'_i[m]$ sequences, such as spectral shaping to suppress the power of $e'_{DAC}(t)$ within one or more frequency bands.

It follows from (96) and (99) that during each time interval $mT/Q \leq t < (m + 1)T/Q$

$$\begin{aligned} y_{\Delta\Sigma}(t) = {} & \alpha'\left(t-m\frac{T}{Q}\right)x_f\left[\lfloor m/Q \rfloor\right] \\ & +\beta'\left(t-m\frac{T}{Q}\right)+e_{\Delta\Sigma DAC}(t) \end{aligned} \tag{106}$$

where

$$e_{\Delta\Sigma DAC}(t)=\alpha'\left(t-m\frac{T}{Q}\right)\left(d_{\Delta\Sigma}[m]+e_{\Delta\Sigma}[m]\right)+e'_{DAC}\left(t\right) \tag{107}$$

During each sample-interval $nT \leq t < (n + 1)T$, this can be written as

$$y_{\Delta\Sigma}(t)=\alpha''\left(t-nT\right)x_f[n]+\beta''\left(t-nT\right)+e_{\Delta\Sigma DAC}(t) \tag{108}$$

where

$$\alpha''(t)=\sum_{k=0}^{Q-1}\alpha'\left(t-\frac{k}{Q}T\right), \tag{109}$$

$$\beta''(t) = \sum_{k=0}^{Q-1} \beta'\left(t - \frac{k}{Q}T\right), \tag{110}$$

and $e_{\Delta\Sigma\mathrm{DAC}}(t)$ is given by (107) and (101) with

$$m = nQ + \left\lfloor Q\frac{t-nT}{T} \right\rfloor . \tag{111}$$

With (100), this implies that during the nth sample-interval, $nT \le t < (n + 1)T$, $e_{\Delta\Sigma DAC}(t)$ has the form

$$\begin{aligned} e_{\Delta\Sigma DAC}(t) = K_{J+1}\sum_{i=1}^{L}\Bigg\{ & \left[\left(d_{\Delta\Sigma}[m] + e_{\Delta\Sigma}[m]\right)m'_i + \lambda'_i[m]\right] \\ & \cdot \alpha'_i\left(t - nT - \left\lfloor Q\frac{t-nT}{T} \right\rfloor \frac{T}{Q}\right) \Bigg\} \end{aligned} \tag{112}$$

The output of each non-shaded 1-bit DAC in Figure 11 during the nth sample-interval is given by (94), which is equivalent to

$$y_i(t) = \left(c_i[n] - \tfrac{1}{2}\right)\alpha_i\left(t - nT\right)K_i\Delta + \beta_i\left(t - nT\right) \tag{113}$$

with

$$\alpha_i(t) = a(t) + \frac{e_{hi}(t) - e_{li}(t)}{K_i\Delta} \quad \text{and} \quad \beta_i(t) = \frac{e_{hi}(t) + e_{li}(t)}{2}. \tag{114}$$

Substituting (95) and (108) into (97), and substituting (113) and then (87) into the result leads to the conclusion that during the nth sample-interval, $nT \le t < (n + 1)T$,

$$y(t) = \alpha'''(t - nT)x[n] + \beta'''(t - nT) + e'''_{DAC}(t) + e_{\Delta\Sigma DAC}(t) \tag{115}$$

with

$$\alpha'''(t) = \alpha''(t)\sum_{i=1}^{J} K_i m_i + \sum_{i=J+1}^{N} K_i m_i \alpha_i(t), \tag{116}$$

$$\beta'''(t) = \beta''(t) + \sum_{i=J+1}^{N} \beta_i(t), \tag{117}$$

and

$$e'''_{DAC}(t) = \alpha''(t-nT)\sum_{i=1}^{J} K_i\lambda_i[n] + \sum_{i=J+1}^{N} K_i\lambda_i[n]\alpha_i(t-nT) \tag{118}$$

where $e_{\Delta\Sigma DAC}(t)$ is given by (112), $\alpha_i(t)$ and $\beta_i(t)$ are given by (114), and $\alpha''(t)$, $\beta''(t)$, $\alpha'_i(t)$ and $\beta'_i(t)$ are given by (109), (110), (102) and (103), respectively.

It follows from (88) and (118) that if $\alpha''(t)$ and $\alpha_i(t)$ for each $i = J+1, \ldots, N$ were identical, then $e'''_{DAC}(t)$ would be zero. Of course, given that $\alpha''(t)$ is a concatenation of Q pulses each with a maximum duration of T/Q and each $\alpha_i(t)$ is a single pulse of duration T, this is not the case in general. However, as explained above, the modified DEM encoder ensures that each of the $\lambda_i[n]$ sequences is a noise-like sequence that is zero mean, free of spurious tones, and uncorrelated with $x[n]$. Therefore, (118) implies that $e'''_{DAC}(t)$ also has a noise-like structure, is free of spurious tones, and satisfies

$$\mathrm{E}\left\{e'''_{DAC}(t)\right\} = 0 \quad \text{and} \quad \mathrm{E}\left\{e'''_{DAC}(t)x[n]\right\} = 0 \tag{119}$$

regardless of both the 1-bit DAC mismatch error pulses and any deviations between $\alpha''(t)$ and $\alpha_i(t)$ for $i = J+1, \ldots, N$.

Furthermore, in practice K_N tends to be much larger than K_{J+1}, so the term in $e'''_{DAC}(t)$ corresponding to mismatches between $\alpha''(t)$ and the $\alpha_i(t)$ pulses, i.e., the first summation term in (118), tends to have a much lower weight than the sum of the rest of the terms in $e'''_{DAC}(t)$. This mitigates the effect of the inherent mismatch between

$\alpha''(t)$ and the $\alpha_i(t)$ pulses. The effect can be mitigated further over the signal bandwidth if $x[n]$ represents an interpolated desired signal and the MRQ-DEM encoder is such that $e'''_{DAC}(t)$ has a highpass spectral shape as in the example MRQ-DEM DAC presented in the next section.

IV. A Multi-Rate Quantizing DEM DAC Example

This section presents the details of an example MRQ-DEM DAC which adheres to the general structure shown in Figure 11. As described in Section III, each MRQ-DEM DAC can be viewed as a modified version of a particular DEM DAC that adheres to the general structure shown in Figure 9. The details of the particular underlying DEM DAC upon which the MRQ-DEM DAC example is based are presented below. Then a functional description of the MRQ-DEM DAC example is presented followed by that of a comparable conventional ΔΣ DAC. Finally, tradeoffs between the two example oversampling DACs are described.

A. The Underlying DEM DAC

The underlying DEM DAC upon which the MRQ-DEM DAC example is based is shown in Figure 12. The weights of its 1-bit DACs are

$$\begin{aligned} &K_{2i-1} = K_{2i} = 2^{i-1} && \text{for } i = 1,...,11, \text{ and} \\ &K_i = 2048 && \text{for } i = 23,\ldots,30. \end{aligned} \tag{120}$$

Thus, the first two 1-bit DACs each have a weight of unity, the next two each have a weight of 2, the next two each have a weight of 4, and so on, up to the 22nd 1-

bit DAC which has a weight of 1024. The 23rd through 30th 1-bit DACs each have a weight of 2048.

In principle, the DAC can accommodate an input sequence that takes on the range of values given by (82), where it follows from (85) and (120) that $M = 20478$. However, as described in Section II, a necessary condition for any DEM encoder to ensure that the DAC mismatch noise satisfies (92) and is free of nonlinear distortion regardless of the mismatch error pulses is that $x[n]$ be restricted to avoid the smallest $K_N - 1$ values and the largest $K_N - 1$ values of (82). The DEM encoder described below is optimal in the sense that its DAC mismatch noise has these properties without any additional restrictions on $x[n]$.

In terms of analyzing a DAC's performance, it is convenient to interpret the sequence of input codewords as a sequence of numerical values, $x[n]$, each in the set (82) as described in Section II. However, when implementing the DEM encoder, it is often convenient to interpret the sequence of input codewords as a sequence of nonnegative integers, $c[n]$, related to $x[n]$ by

$$c[n] = \frac{x[n]}{\Delta} + \frac{M}{2}. \tag{121}$$

Therefore, without loss of generality, $c[n]$ can be considered to be the DEM encoder's input sequence. Given that $x[n]$ is restricted to avoid the smallest $K_N - 1$ values and largest $K_N - 1$ values of (82) as described above, it follows that

$$c[n] \in \{2047, 2048, ..., 18430, 18431\} \tag{122}$$

for each n.

As shown in Figure 14, the DEM encoder of the underlying DEM DAC consists of 29 digital *switching blocks*, labeled $S_{k,r}$ for k = 1, 2…, 14, and r = 1, 2, …, 15, configured in a tree structure [48]. The 11 switching blocks that are shaded in the figure are called *segmenting switching blocks* and the other 18 switching blocks are called *non-segmenting switching blocks*.

The functional details of the switching blocks are shown in Figure 15. The top and bottom outputs of each segmenting switching block, $S_{k,1}$, are

$$\frac{1}{2}\left(c_{k,1}[n]-1-s_{k,1}[n]\right), \qquad \text{and} \qquad 1+s_{k,1}[n], \tag{123}$$

respectively, where $c_{k,1}[n]$ is the switching block input sequence, and $s_{k,1}[n]$, called a *switching sequence*, is 0 when $c_{k,1}[n]$ is odd and 1 or −1 otherwise. Similarly, the top and bottom outputs of each non-segmenting switching block, $S_{k,r}$, are

$$\frac{1}{2}\left(c_{k,r}[n]-s_{k,r}[n]\right) \qquad \text{and} \qquad \frac{1}{2}\left(c_{k,r}[n]+s_{k,r}[n]\right), \tag{124}$$

respectively, where $c_{k,r}[n]$ is the switching block input sequence, and the switching sequence, $s_{k,r}[n]$, in this case is 0 when $c_{k,r}[n]$ is even and 1 or −1 otherwise. Regardless of the switching block type, each switching sequence is generated in two's complement format by the logic shown in Figure 15c, wherein $d_k[n]$ for k = 1, 2, …, 14 are independent random sequences that each take on values of 0 and 1 with equal probability.

A nearly identical analysis to that presented in [39] shows that (87) through (91) hold with

$$m_i = \begin{cases} 0, & \text{for } 1 \le i \le 22, \\ 2^{-14}, & \text{for } 23 \le i \le 30, \end{cases} \tag{125}$$

and each $\lambda_i[n]$ sequence is a linear combination of the switching sequences. Therefore, provided the switching sequences are noise-like sequences that are zero mean, free of spurious tones, and uncorrelated with $x[n]$, then $e_{DAC}(t)$ has a noise-like structure, is free of spurious tones, and satisfies (92).

It can be verified from Figure 15c that

$$s_{k,r}[n] = t_{k,r}[n] - t_{k,r}[n-1] \tag{126}$$

with

$$t_{k,r}[n] = \begin{cases} t_{k,r}[n-1], & \text{if } c_{k,r}[n] \text{ is odd,} \\ 0, & \text{if } c_{k,r}[n] \text{ is even and } t_{k,r}[n-1] \ne 0, \\ 2d_k[n]-1, & \text{if } c_{k,r}[n] \text{ is even and } t_{k,r}[n-1] = 0, \end{cases} \tag{127}$$

when k = 4, 5, …, or 14, and

$$t_{k,r}[n] = \begin{cases} t_{k,r}[n-1], & \text{if } c_{k,r}[n] \text{ is even,} \\ 0, & \text{if } c_{k,r}[n] \text{ is odd and } t_{k,r}[n-1] \ne 0, \\ 2d_k[n]-1, & \text{if } c_{k,r}[n] \text{ is odd and } t_{k,r}[n-1] = 0, \end{cases} \tag{128}$$

when k = 1, 2, or 3. The results presented in [49] imply that the $s_{k,r}[n]$ sequences for k = 1, 2, …, 14 and r = 1, 2, …, 15, are zero-mean sequences that are free of spurious tones, have a highpass spectral shape with a zero-frequency null, and are uncorrelated with each other and $x[n]$. Given that the $\lambda_i[n]$ sequences are each a linear combination of the switching sequences, it follows from (90) that $e_{DAC}(t)$ also has these properties.

B. MRQ-DEM DAC Functional Description

A high-level diagram of the example MRQ-DEM DAC is shown in Figure 16. The top 14 1-bit DACs and their input sequences, $c_{17}[n]$, $c_{18}[n]$, …, $c_{30}[n]$, are identical to those in the underlying DEM DAC shown in Figure 12 with an update rate of $(R/Q)f_N$. The bottom 4 1-bit DACs each have the minimum weight of K_{17} = 256 and an update-rate of Rf_N.

The $x_f[n]$ output of the modified DEM encoder is given by (95) with J = 16. Therefore, the $c_i[n]$ sequences that determine $x_f[n]$ are the 16 outputs of the bottom 8 non-segmenting switching blocks in the DEM encoder shown in Figure 13. As implied by (124), the sum of the output sequences from each of these switching blocks is equal to the switching block's input sequence. Therefore, (95) can be rewritten as

$$x_f[n] = \Delta \sum_{r=1}^{8} 2^{r-1} \left(c_{1,r}[n] - 1 \right) \tag{129}$$

where $c_{1,r}[n]$ is the bottom output of the $S_{15-r,1}$ switching block for r = 1, 2, …, 8. It follows that the modified DEM encoder in Figure 15 can be obtained from the DEM encoder in Figure 13 by replacing the $S_{1,r}$ switching blocks for r = 1, 2, …, 8 with a power-of-two summing network described by (129).

The resulting modified DEM encoder is shown in Figure 16. Each $c_{1,r}[n]$ sequence in (129) is equal to the right-most expression in (123) with k = 15−r and each $s_{k,1}[n]$ sequence can only take on values of −1, 0, and 1, so it follows from (129) that

$$x_f[n] \in \{-255\Delta, -254\Delta, -253\Delta, \ \ldots \ , 255\Delta\} \tag{130}$$

for each n (the LSB of the codeword sequence that represents $x_f[n]$ in the modified

DEM encoder shown in Figure 16 is defined to have a weight of Δ).

A functional diagram of the dithered second-order digital ΔΣ modulator in the MRQ-DEM encoder is shown in Figure 17. Its input-output relationship and quantization noise are identical to those of a comparably configured second-order digital ΔΣ modulator with two accumulators in the forward path and feedback from just the output of the quantizer, but it offers the advantage of a slightly more efficient digital implementation [28].

The dither sequence, $d_{\Delta\Sigma}[m]$, is a white random sequence that takes on values of 0 and Δ with equal probability. Given that the minimum step-size of $x_f[n]$ is Δ, the dither is called *least-significant-bit (LSB) dither*.

The output of the digital ΔΣ modulator, $x_{\Delta\Sigma}[m]$, satisfies (96) and the input range indicated by (130) implies that

$$x_{\Delta\Sigma}[m] \in \{-512\Delta, -256\Delta, 0, 256\Delta, 512\Delta\} \tag{131}$$

for each m [28]. The LSB dither ensures that $e_{\Delta\Sigma}[m]$ is asymptotically independent of $x_f[n]$ and $d_{\Delta\Sigma}[m]$, and has a power spectrum equal to that of the output of a filter with transfer function $(1-z^{-1})^2$ driven by white noise with a variance of $2^{15}\Delta^2$ [50, 51].

Given that $x_{\Delta\Sigma}[m]$ takes on only five values, only three switching blocks are necessary in the local DEM encoder: $S_{2,1}$, $S_{1,1}$, and $S_{1,2}$. These switching blocks have the same properties as those in the DEM encoder described above, except that they are clocked at a rate of Rf_N instead of $(R/Q)f_N$. The input to the local DEM encoder's $S_{2,1}$ switching block is

$$c_{\Delta\Sigma}[m] = \frac{x_{\Delta\Sigma}[m]}{256\Delta} + 2\,. \tag{132}$$

The inputs to the local DEM encoder's $S_{1,1}$ and $S_{1,2}$ switching blocks are the bottom and top outputs, respectively, of its $S_{2,1}$ switching block. The local DEM encoder's outputs are the outputs of its $S_{1,1}$ and $S_{1,2}$ switching blocks.

C. Signal Path Bypasses the Fast 1-bit DACs

In both conventional ΔΣ DACs and MRQ-DEM DACs, DEM eliminates harmonic distortion that would otherwise arise from 1-bit DAC pulse shape, amplitude, and timing errors, but it does not prevent harmonic distortion caused by nonlinearity introduced by the individual 1-bit DACs. For example, a major cause of such nonlinearity is intersymbol interference [52]. In most cases, 1-bit DAC nonlinearity arises from parasitic circuit elements, so for a given circuit topology it tends to increase with the 1-bit DAC update-rate.

An important feature of the MRQ-DEM DAC example is that nonlinearity introduced by the high update-rate 1-bit DACs does not introduce harmonic distortion. This is because each $c_{1,r}[n]$ term in (129) is equal to the right-most expression in (123), so $x_f[n]$ is a pseudo-random sequence that is uncorrelated with the MRQ-DEM DAC's input sequence. Consequently, nonlinearity introduced by the 1-bit DACs driven by the digital ΔΣ modulator does not cause harmonic distortion. This is an advantage of the MRQ-DEM DAC over a comparable conventional ΔΣ DAC wherein all the 1-bit DACs (all of which operate at the full update rate) convert sequences that are correlated with the DAC's input sequence, so any nonlinearity they introduce harmonically distorts the input sequence.

D. A Comparable Conventional ΔΣ DAC Example

A conventional ΔΣ DAC with an input dynamic range that is comparable to that of the example MRQ-DEM DAC described in Section IV-C above is shown in Figure 18. The dithered second-order digital ΔΣ modulator is the same as that shown in Figure 17, except it is driven by the ΔΣ DAC's input sequence instead of $x_f\left[\lfloor m/Q \rfloor\right]$. The switching blocks and 1-bit DACs are identical to the corresponding components of the DEM DAC shown in Figures 13-5 and described above with $T = 1/(Rf_N)$.

It can be verified that the DEM DAC's input range restriction described in Section II-B and the no-overload range of the ΔΣ modulator described in [28] imply that the ΔΣ DAC's input sequence must be restricted to values in the range

$$\{-7808\Delta, -7807\Delta, -7806\Delta, \ldots 7808\Delta\} \tag{133}$$

In contrast, it follows from (121) and (122) that the example MRQ-DEM DAC's input sequence must be restricted to values in the range

$$\{-8192\Delta, -8191\Delta, -8190\Delta, \ldots 8192\Delta\} \tag{134}$$

Therefore, the MRQ-DEM DAC and the ΔΣ DAC examples have input dynamic ranges that are within half a dB of each other, so it is reasonable to compare their signal to quantization and mismatch noise ratio performances.

The input sequence to the ΔΣ DAC example has an update-rate of Rf_N whereas that of the MRQ-DEM DAC has an update-rate of $(R/Q)f_N$. Consequently, the signal band replicas in the output signals from the two DAC examples occur at multiples Rf_N and $(R/Q)f_N$, respectively. This is a potential disadvantage of the MRQ-DEM DAC relative to the ΔΣ DAC in applications where Q is large and it is necessary to filter out the signal band replicas.

E. Performance Comparison

Results from simulations of the two oversampling DAC examples that support the theoretical results presented above are presented in this section. The simulations use dual return-to-zero (RTZ) 1-bit DACs, each of which is implemented as the sum of two half-period RTZ 1-bit DACs offset in time by half a period [52]. Ideally, this achieves the effect of having non-RTZ 1-bit DACs without incurring as much nonlinear distortion from inter-symbol interference as tends to occur with conventional non-RTZ 1-bit DACs.

The output pulses from actual 1-bit DAC circuits depend on many non-ideal effects and are highly dependent on circuit topology and layout. The purpose of the simulations described below is to qualitatively demonstrate the results described above, so a first-order pulse shape model with exponential settling has been adopted for simplicity and ease of explanation. Specifically, during the nth sample-interval the outputs of the two RTZ 1-bit DACs in each dual-RTZ 1-bit DAC with an update rate of $(R/Q)f_N$ are given by (84) with $a(t)$ equal to

$$p\left(t, \frac{Q}{Rf_N}, \tau_r, \tau_f\right), \quad \text{and} \quad p\left(t - \frac{T}{2}, \frac{Q}{Rf_N}, \tau_r, \tau_f\right), \tag{135}$$

respectively, where

$$p\left(t,T,\tau_r,\tau_f\right)=\begin{cases}1-e^{-t/\tau_r}, & \text{if } 0\le t<\dfrac{T}{2},\\ \left(1-e^{-T/(2\tau_r)}\right)e^{-(t-T/2)/\tau_f}, & \text{if } \dfrac{T}{2}\le t<T,\\ 0, & \text{otherwise,}\end{cases} \tag{136}$$

and τ_r and τ_f are the rising and falling edge time constants. Similarly, the outputs of the two RTZ 1-bit DACs in each dual-RTZ 1-bit DAC with an update rate of Rf_N are given by (94) with $a'(t)$ equal to

$$p\left(t,\frac{1}{Rf_N},\tau_r,\tau_f\right), \quad \text{and} \quad p\left(t-\frac{T}{2},\frac{1}{Rf_N},\tau_r,\tau_f\right), \tag{137}$$

respectively.

The simulations model the mismatch error pulses by applying randomly chosen static errors to the amplitude and time constants for each RTZ 1-bit DAC that depend upon whether the 1-bit DAC's input bit is high or low. For example, the resulting mismatch error pulses for the first RTZ 1-bit DAC in the ith dual-RTZ 1-bit DAC with an update rate of $(R/Q)f_N$ are

$$e_{hi}(t)=K_i\frac{\Delta}{2}\left[\left(1+\varepsilon_{ahi}\right)p\left(t,\frac{Q}{Rf_N},\tau_r+\varepsilon_{rhi},\tau_f+\varepsilon_{fhi}\right)-a(t)\right] \tag{138}$$

and

$$e_{li}(t)=K_i\frac{\Delta}{2}\left[a(t)-\left(1+\varepsilon_{ali}\right)p\left(t,\frac{Q}{Rf_N},\tau_r+\varepsilon_{rli},\tau_f+\varepsilon_{fli}\right)\right] \tag{139}$$

where ε_{ahi} and ε_{ali} are amplitude errors and ε_{rhi}, ε_{fhi}, ε_{rli}, and ε_{fli}, are time constant errors, all of which are randomly chosen fixed values.

Figure 19 shows output power spectra from simulations of the two DAC examples, each with a full-scale sinusoidal input sequence, R = 40, 0.1% 1-bit DAC amplitude deviations, 0.5% 1-bit DAC time constant deviations, and the same $1/(20Rf_N)$ time constant for all the 1-bit DACs. For the minimum-weight 1-bit DACs the amplitude deviations were randomly chosen with a standard deviation of 0.1%, and the time constant deviations were randomly chosen with a standard deviation of 0.5% (e.g., for the minimum-weight 1-bit DACs ε_{ahi} and ε_{ali} in (138) and (139) are samples of a random variable with a standard deviation of 10^{-3}, whereas ε_{rhi}, ε_{fhi}, ε_{rli}, and ε_{fli} are samples of a random variable with a standard deviation of $5 \cdot 10^{-3}$). For each of the other 1-bit DACs the standard deviations of the error are scaled by $(K_i \cdot /256)^{0.5}$ to account for the $2^{0.5}$ factor increase in mismatch standard deviation that typically occurs in CMOS integrated circuits when the size of matched components is doubled [53]. Identical mismatches were used for the corresponding 1-bit DACs in the two DAC examples. The MRQ-DEM DAC was simulated with Q = 4, so the clock-rate of the modified DEM encoder and the update-rate of all but four of the minimum-weight 1-bit DACs is a quarter of those of the DEM encoder and 1-bit DACs in the ΔΣ DAC.

The simulation results indicate that the peak SNR over the signal band is 86.2 dB for the ΔΣ DAC and 84.9 dB for the MRQ-DEM encoder. As expected, no harmonic distortion was detectable in either case. The simulations were run for several different random number seeds, and the results indicate that on average the peak SNR of the MRQ-DEM DAC example is 1.3 dB lower than that of the conventional ΔΣ DAC example. Therefore, relative to the ΔΣ DAC example the MRQ-DEM DAC example trades an average of 1.3 dB of peak SNR for a four-fold reduction in the update-rate of all but four of its lowest-weighted 1-bit DACs and the clock rate of the logic that drives them.

Figure 20 shows the output spectra from simulations of the MRQ-DEM DAC example configured as described above but with DEM disabled. This is achieved by effectively replacing both the underlying DEM encoder used to generate the modified DEM encoder and the local DEM encoder with thermometer encoders. As expected, the 1-bit DAC mismatches give rise to significant harmonic distortion across the spectrum in the absence of DEM. When DEM was enabled for this case as well as all of numerous other cases simulated by the authors, no harmonic distortion was detectable.

ACKNOWLEDGEMENTS

Chapter 2, in full, has been submitted for publication 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 for this paper.

FIGURES

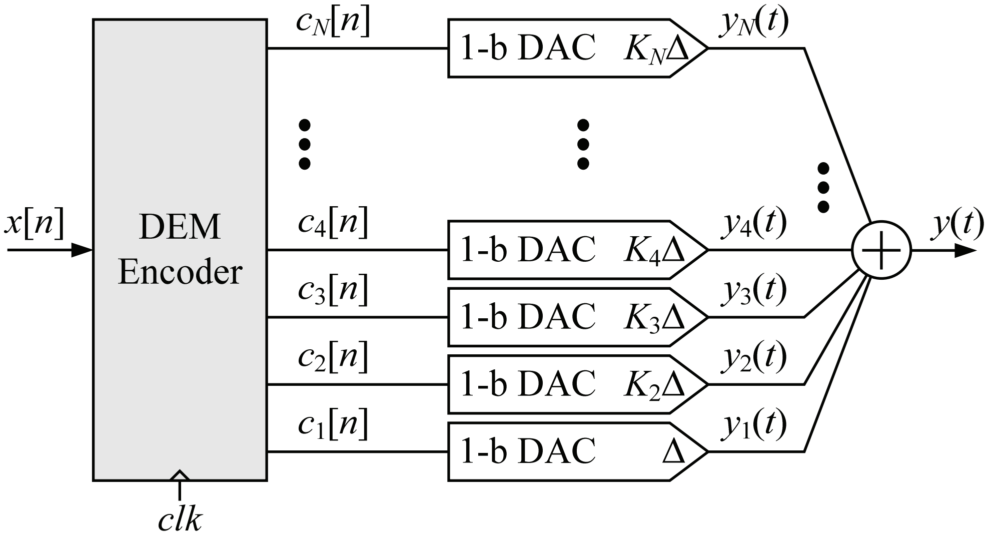


Figure 9: A general DEM DAC architecture.

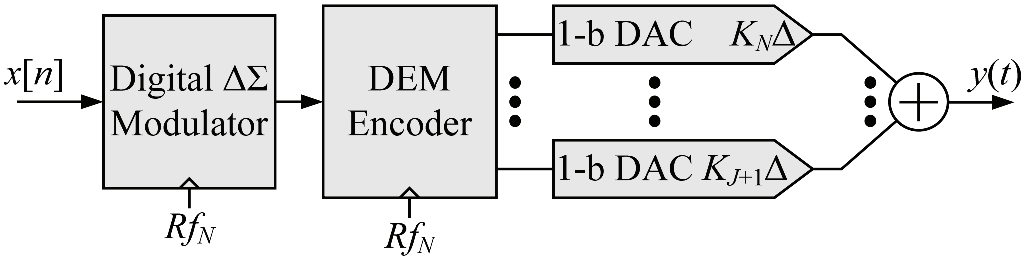


Figure 10: A general conventional oversampling DEM DAC architecture.

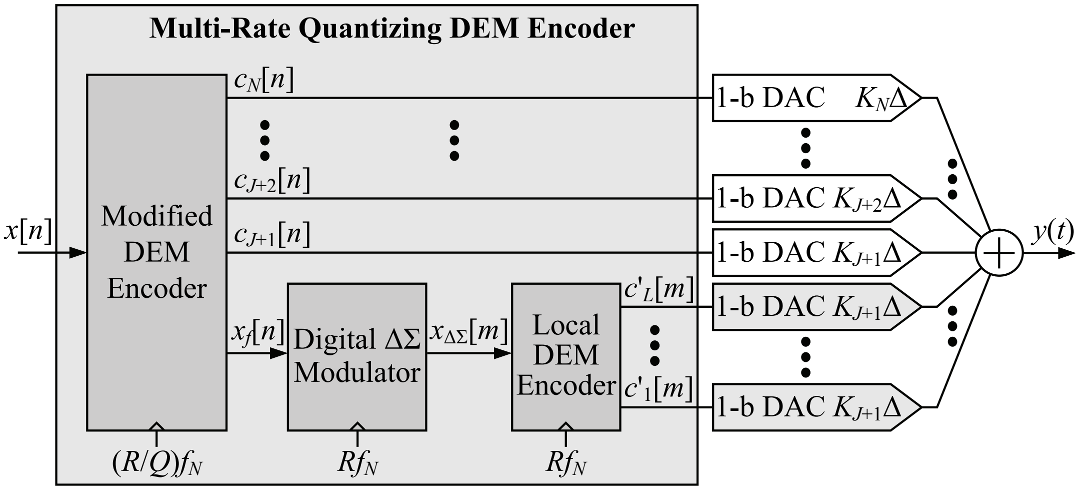


Figure 11: A general oversampling DEM DAC based on the proposed multi-rate quantizing DEM encoder.

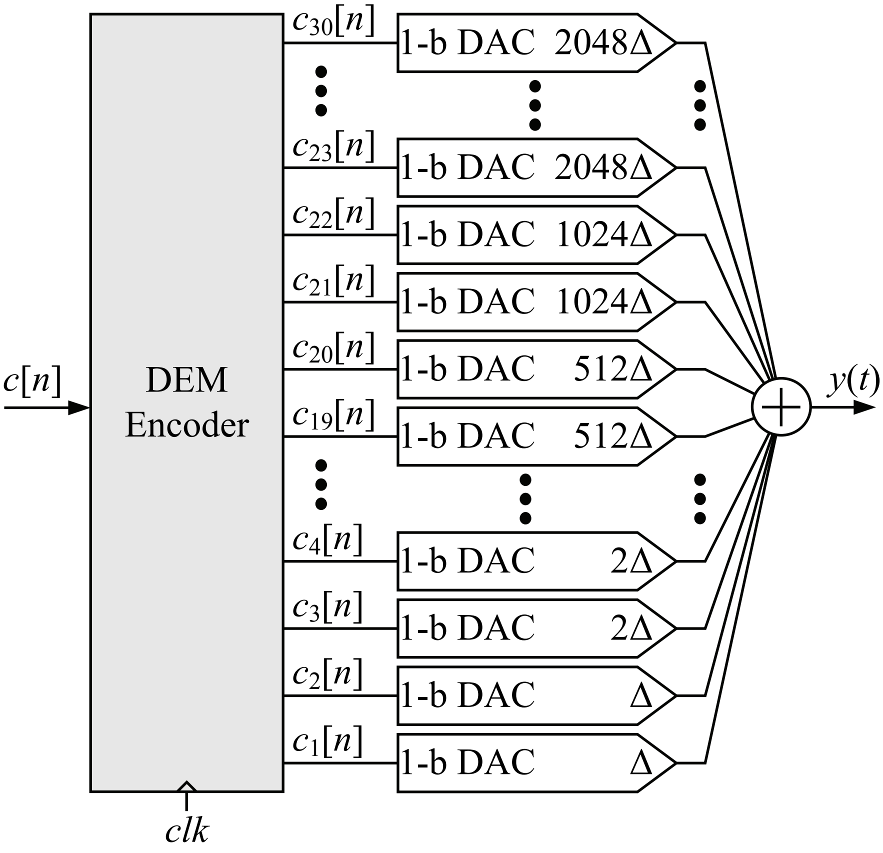


Figure 12: The underlying DEM DAC from which the example MRQ-DEM DAC is derived.

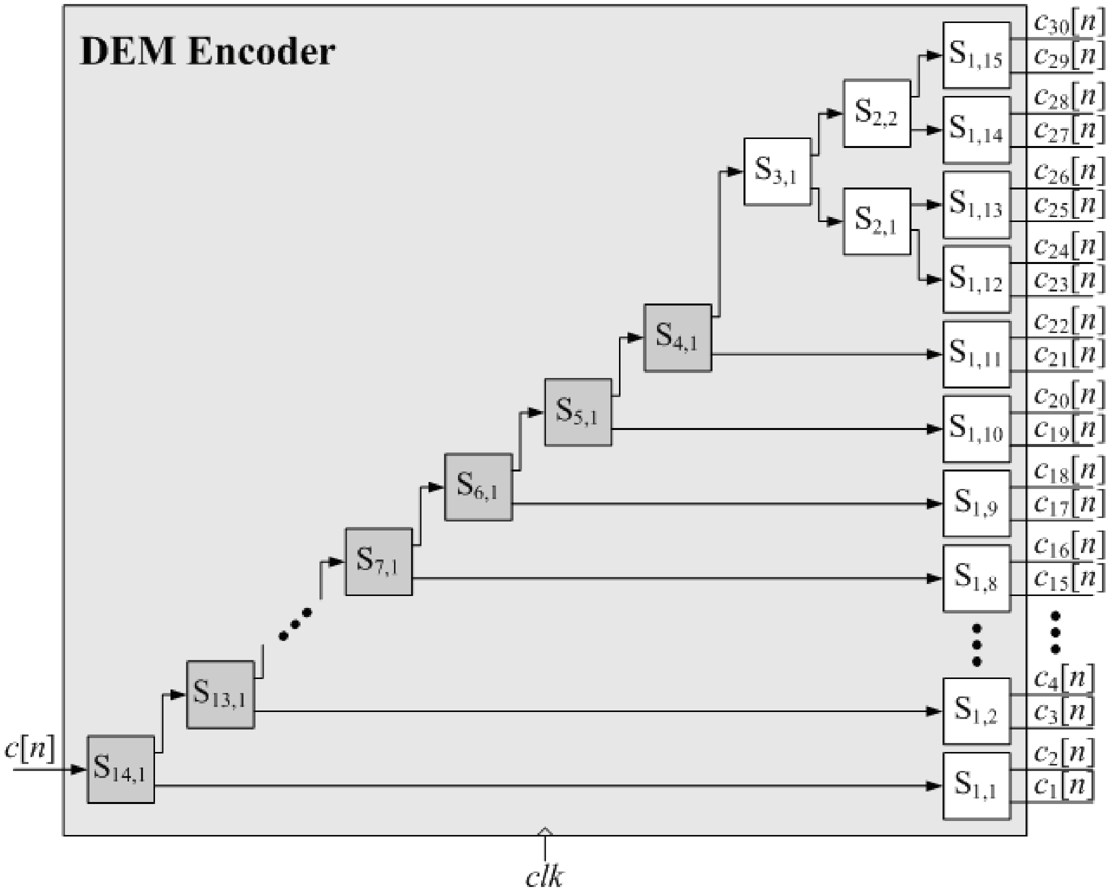


Figure 13: Structure of the DEM encoder in the underlying DEM DAC from which the example MRQ-DEM DAC is derived.

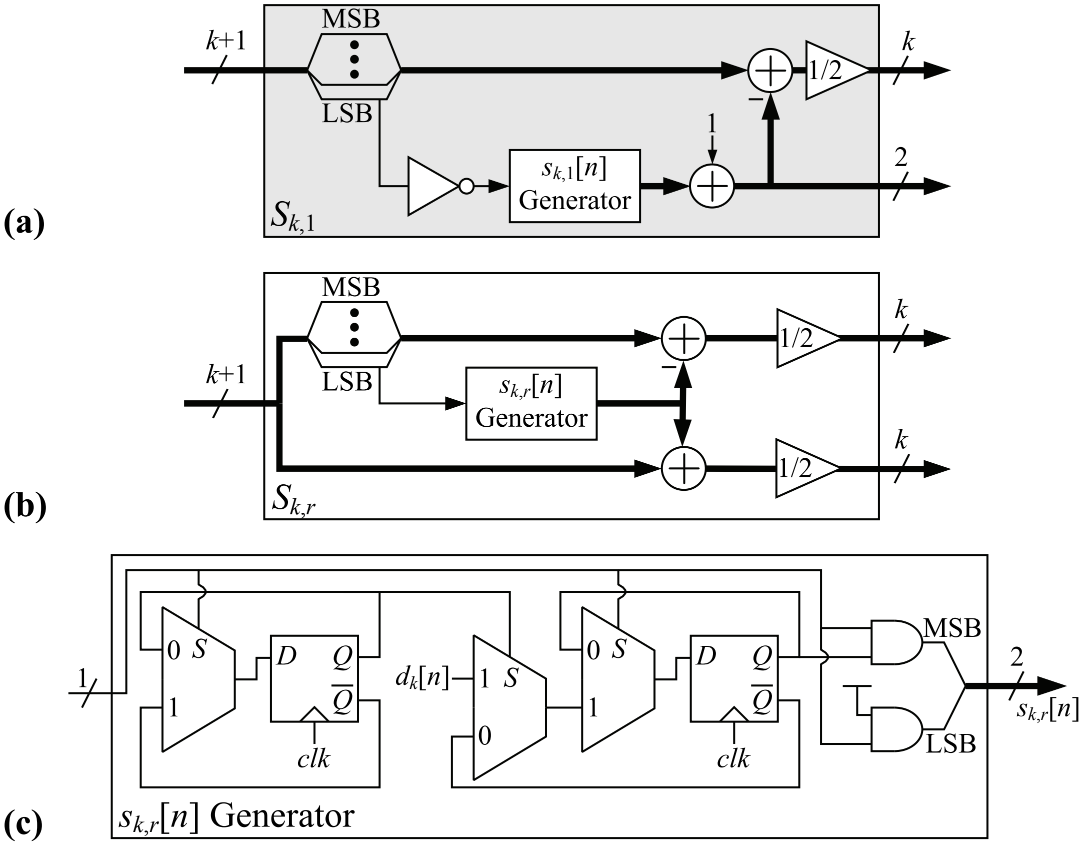


Figure 14: Functional diagrams of (a) each segmenting switching block (b) each non-segmenting switching block, and c) the switching sequence generator within each switching block.

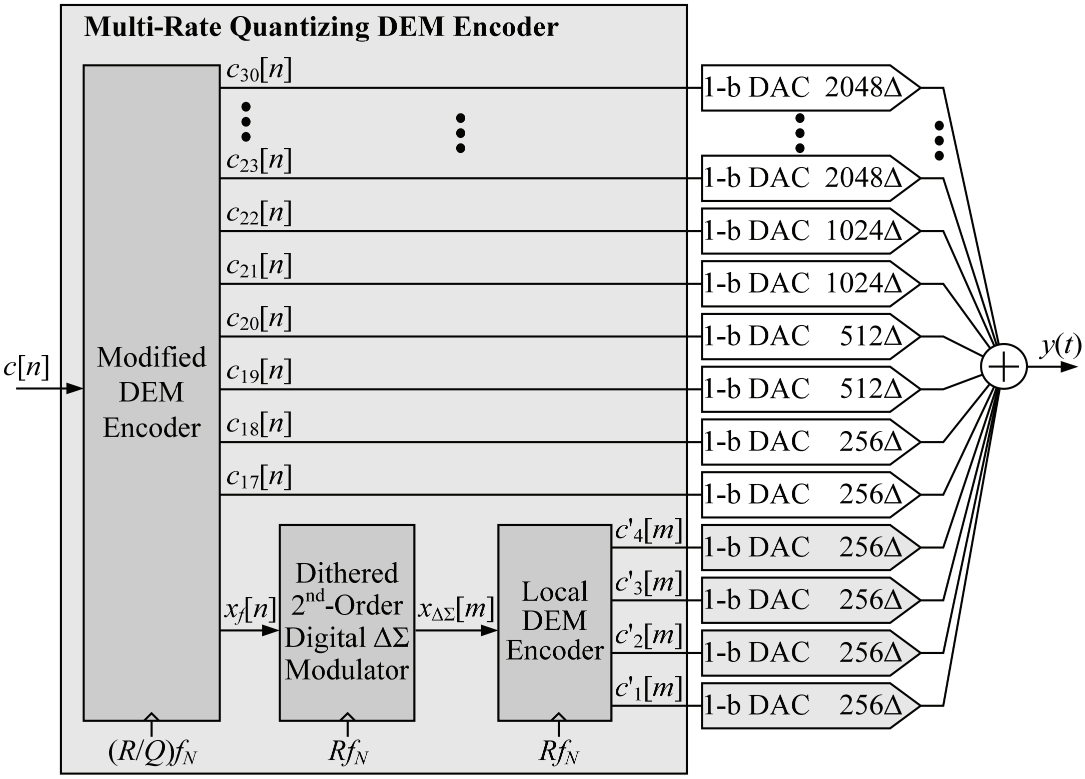


Figure 15: High-level diagram of the example MRQ-DEM DAC.

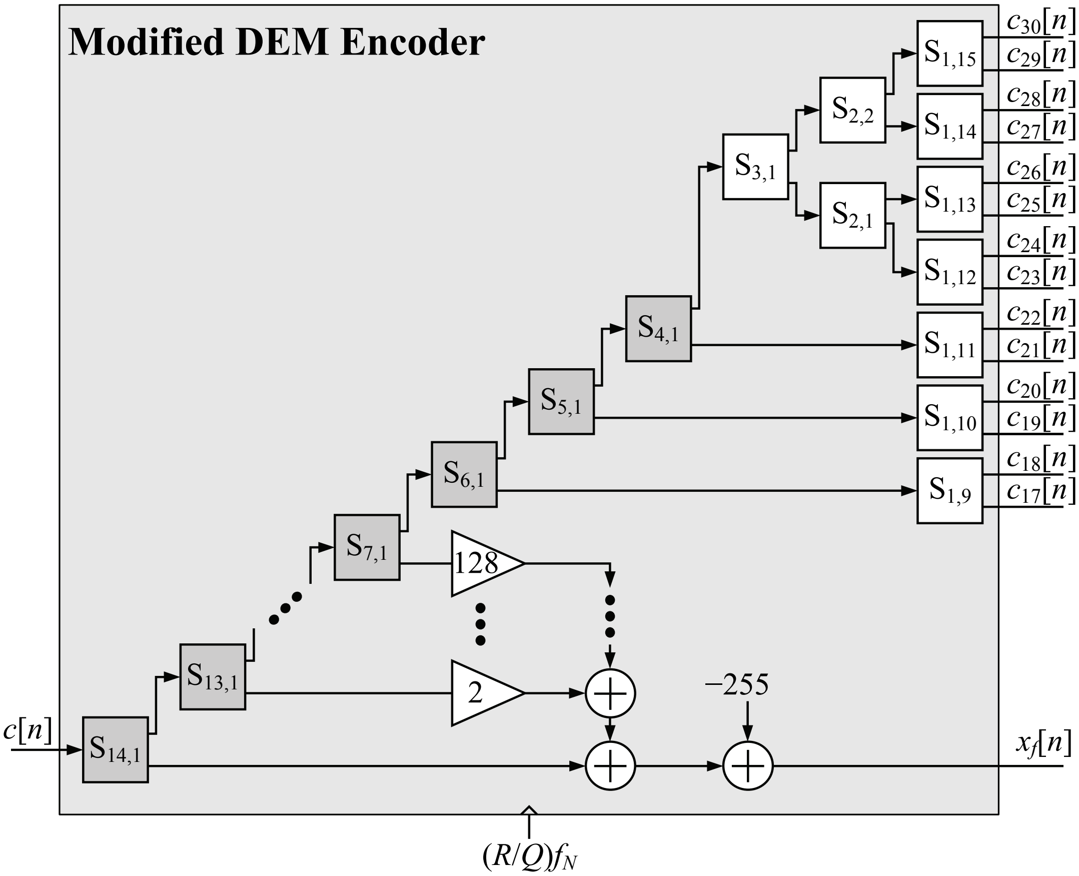


Figure 16: Top-level functional diagram of the modified DEM encoder in the MRQ-DEM DAC.

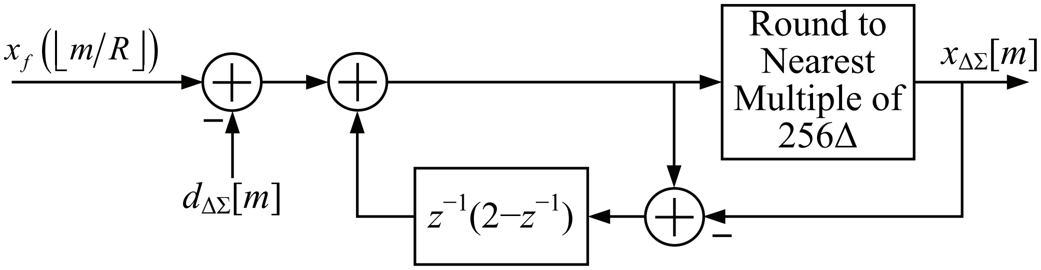

$x_f\left(\lfloor m/R \rfloor\right)$
$-$
$d_{\Delta\Sigma}[m]$
Round to Nearest Multiple of 256Δ
$x_{\Delta\Sigma}[m]$
$z^{-1}(2-z^{-1})$
$-$

Figure 17: Functional diagram of the dithered second-order digital ΔΣ modulator.

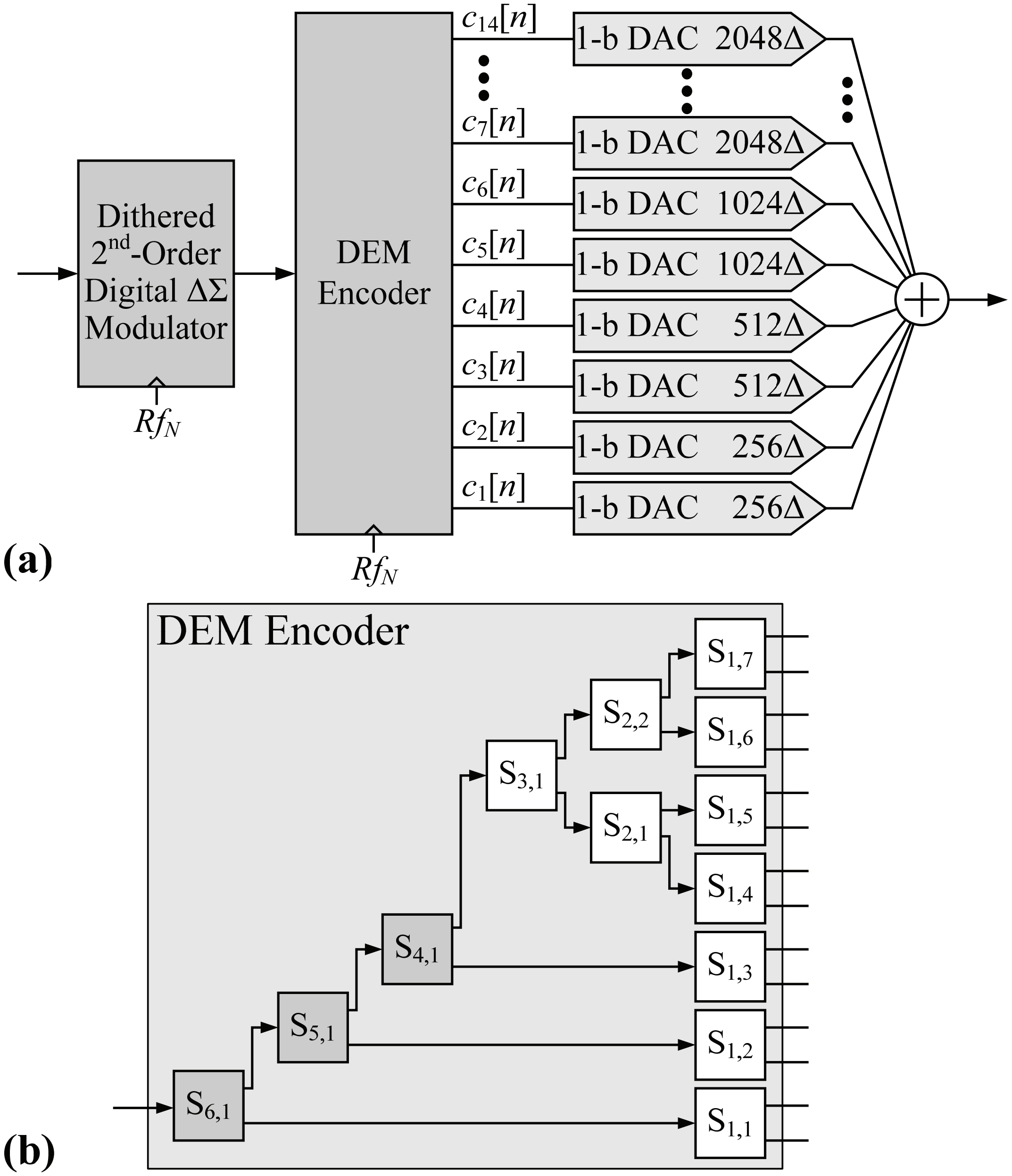


Figure 18: A conventional ΔΣ DAC with an input dynamic range comparable to that of the example MRQ-DEM DAC: (a) high-level diagram of the ΔΣ DAC, (b) top-level functional diagram of the DEM encoder.

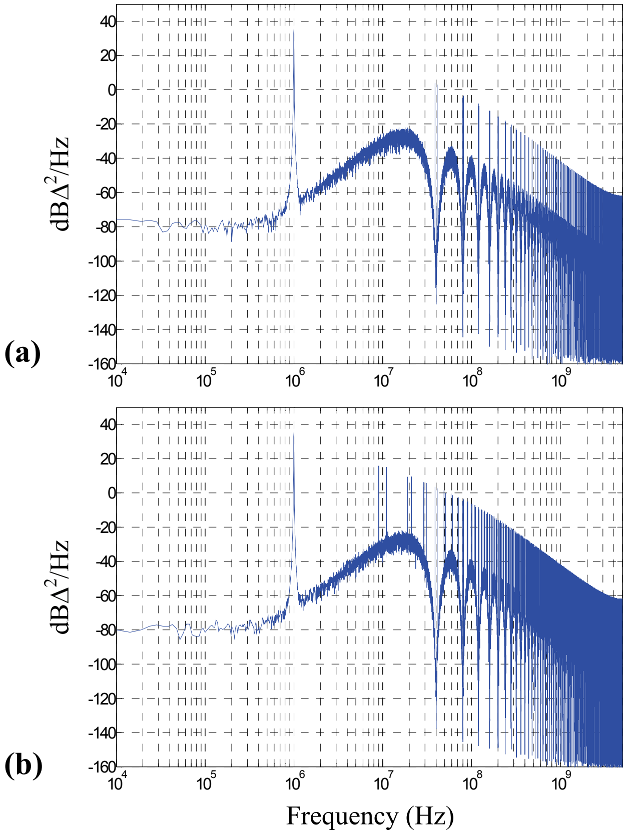


Figure 19: Representative output power spectra from simulations of (a) the ΔΣ DAC and (b) the MRQ-DEM DAC with amplitude and time constant mismatches among the 1-bit DACs.

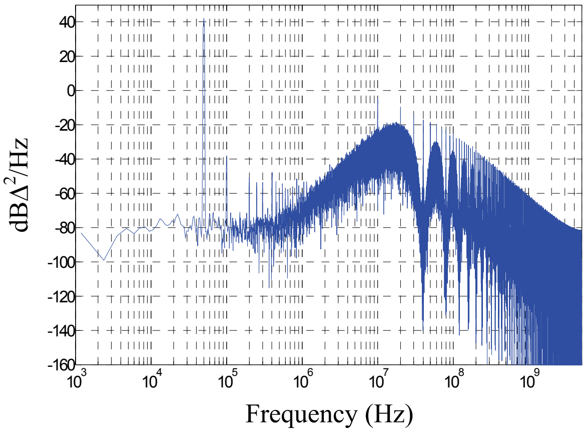


Figure 20: Representative output power spectrum from simulation of the MRQ-DEM DAC with DEM disabled.

References

28. R. Schreier, G. C. Temes, *Understanding Delta-Sigma Data Converters*, John Wiley and Sons, 2005.

29. M. J. Story, “Digital to analogue converter adapted to select input sources based on a preselected algorithm once per cycle of a sampling signal,” U.S. Patent 5 138 317, August 11, 1992.

30. H. S. Jackson, “Circuit and method of cancelling nonlinearity error associated with component mismatches in a data converter,” U.S. Patent 5 221 926, June 22, 1993.

31. R. Schreier, B. Zhang, “Noise-shaped multibit D/A converter employing unit elements,” *Electronics Letters*, vol. 31, pp. 1712-1713, September, 1995.

32. R. T. Baird, T. S. Fiez, “Linearity enhancement of ΔΣ A/D and D/A converters using data weighted averaging,” *IEEE Transactions of Circuits and Systems II*, *Analog and Digital Signal Processing,* vol. 42, pp. 753-762, December, 1995.

33. T. W. Kwan, R. W. Adams, R. Libert, “A stereo multibit Sigma Delta DAC with asynchronous master-clock interface,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 12, pp. 1881-1887, Dec. 1996.

34. I. Galton, “Spectral shaping of circuit errors in digital-to-analog converters,” *IEEE Transactions of Circuits and Systems II*, *Analog and Digital Signal Processing,* vol. 44, pp. 808-817, October, 1997.

35. T. Shui, R. Schreier, F. Hudson, “Mismatch shaping for a current-mode multibit delta-sigma DAC,” *IEEE Journal of Solid State Circuits*, vol. 34, no. 3, pp. 331-338, March, 1999.

36. I. Fujimori, A. Nogi, T. Sugimoto, “A Multibit Delta-Sigma Audio DAC with 120-dB Dynamic Range,” *IEEE Journal of Solid State Circuits*, vol. 35, no. 8, pp.1066-1073, August, 2000.

37. S. Luschas, R. Schreier, H. S. Lee, “Radio Frequency Digital-to-Analog Converter,” *IEEE Journal of Solid State Circuits*, vol. 39, no. 9, pp.1462-1467, September, 2004.

38. B. Jewett, J. Liu, K. Poulton, “A 1.2Gs/s 15b DAC for precision signal genera-

tion," *IEEE International Solid-State Circuits Conference*, pp. 110-111, February, 2005.

39. T. S. Kaplan, J. F. Jensen, C. H. Fields, M. F. Chang, "A 2-GS/s 3-bit ΔΣ-Modulated DAC With Tunable Bandpass Mismatch Shaping," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 3, pp. 603-610, March 2005.

40. K. L. Chan, J. Zhu, I. Galton, "Dynamic Element Matching to Prevent Nonlinear Distortion From Pulse-Shape Mismatches in High-Resolution DACs," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 9, pp. 2067-2078, September 2008.

41. K. Nguyen, A. Bandyopadhyay, R. Adams, K. Sweetland, P. Baginski, "A 108 dB NSR 1.1 mW Oversampling Audio DAC With A Three-level DEM Technique," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2592-2600, December, 2008.

42. K. Lee, Q. Meng, T. Sugimoto, K. Hamashita, K. Takasuka, S. Takeuchi. U. Moon, G. C. Temes, "A 0.8 V, 2.6 mW, 88 dB Dual-Channel Audio Delta-Sigma D/A Converter With Headphone Driver," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 3, pp. 916-927, March, 2009.

43. A. Bandyopadhyay, M. Determan, S. Kim, K. Nguyen, "A 120dB-SNR 100dB-THD+N 21.5mW/Channel Multibit CT ΔΣ DAC," *IEEE International Solid-State Circuits Conference*, pp. 482-483, February, 2011.

44. L. Risbo, R. Hezar, B. Kelleci, H. Kiper, M. Fares, "Digital Approaches to ISI-Mitigation in High-Resolution Oversampled Multi-Level D/A Converters," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 12, pp. 2892-2903, December 2011.

45. F. Van de Sande, et. al. "A 7.2 GSa/s, 14 Bit or 12 GSa/s, 12 Bit Signal Generator on a Chip in a 165 GHz f_T BiCMOS Process," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 4, pp. 1003-1010, April 2012.

46. I. Galton, "Why Dynamic-Element-Matching DACs Work", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 2, pp. 69-74, February 2010.

47. T. Shui, R. Schreier, F. Hudson, "Mismatch Shaping for a Current-Mode Multibit Delta-Sigma DAC," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 3, pp. 331-338, March 1999.

48. K. L. Chan, N. Rakuljic, I. Galton, “Segmented Dynamic Element Matching for High-Resolution Digital-to-Analog Conversion,” *IEEE Transactions On Circuits* and *Systems I: Regular Papers*, vol. 55, pp. 3383-3392, December, 2008.

49. J. Welz, I. Galton, “A Tight Signal-Band Power Bound on Mismatch Noise in a Mismatch-Shaping Digital-to-Analog Converter,” *IEEE Transactions on Information Theory*, vol. 50, no. 4, pp. 593-607, April, 2004.

50. I. Galton, “One-bit dithering in delta-sigma modulator-based D/A conversion,” *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 1310-1313, May 1993.

51. 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.

52. R. Adams, K. Q. Nguyen, K. Sweetland, “A 113-dB SNR Oversampling DAC with Segmented Noise-Shaped Scrambling,” *IEEE Journal of Solid State Circuits*, vol. 33, no. 12, pp. 1871-1878, December 1998.

53. M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, “Matching Properties of MOS Transistors,” *IEEE Journal of Solid State Circuits*, vol. 24, no. 5, pp. 1433-1440, October 1989.

Chapter 3

A 3-4 GHz GSM-Compliant 1.0/1.2V ΔΣ FDCBased Fractional *N* PLL in 65 nm CMOS Technology

Abstract—This Chapter presents a ΔΣ FDC based fractional-*N* phase-locked loop for frequency synthesis. The PLL was built to showcase the capability of the architecture analyzed in Chapter 1 to comply with the most stringent wireless communication standards. Several enhancements were added to the base architecture of Chapter 1, including an FDC quantization noise cancelling algorithm and an hardware efficient multi-rate quantizing DEM encoder for digital to frequency conversion. The frequency synthesizer was fabricated in a 65 nm CMOS process and draws 20.4 mA of current from 1.2 V and 1.0 V power supplies. Preliminary measurements results for the PLL are available at the end of the Chapter.

I. Architecture Overview

The implemented synthesizer represents a proof of concept for the architecture presented in Chapter 1. As such, the fundamental building blocks of the phase-locked loop are the ones shown in Figure 2. The amount of noise introduced by the circuits that constitute the synthesizer is comparable to what assumed in the design example of Chapter 1. Hence, the performance of the ΔΣ FDC based fractional-*N* PLL prototype can be expected to be comparable to the performance obtained in Chapter 1 for the design example provided.

In order to optimize the architecture for the available 65 nm CMOS process, some design parameters were varied with respect to Table 1. The parameters for the prototype are summarized in Table 2. The circuit implementation of the main FDC components is described in detail in Section II.

A few additions have been made to the standard topology shown in Figure 2, resulting in the architecture shown in Figure 21.

A quantization noise cancelling algorithm has been implemented to reduce the impact of the FDC quantization noise to the overall synthesizer output phase noise. A general description of the algorithm and details of its implementation are provided in Section III.

The logic necessary to implement the quantization noise cancelling algorithm is merged with the phase-to-frequency accumulator and the digital loop filter in the digital loop controller, described in Section IV.

In order to meet the stringent phase noise specifications of the GSM standard the digitally controlled oscillator minimum frequency step needs to be about 200 Hz. Unfortunately, the minimum capacitance step achievable in the technology node used does not allow for such a fine frequency resolution. Hence the digital loop controller output need to be requantized by an oversampling quantizer. To perform the mentioned requantization, while preventing mismatches among the varactors from introducing distortion that will translate in spurious tones at the output of the PLL, the MRQ-DEM encoder of Figure 15 has been interposed between the digital loop controller output and the DCO fine tuning varactor bank. In order to reduce the latency, area, and power consumption of the encoder, the adder-free simplified logic implementation described in Section V has been devised. Details of the digitally controlled oscillator are presented in Section VI.

The timing of the FDC-PLL has been modified with respect with what described in Chapter 1. When the PLL is locked, the presence of an offset current source, described in Chapter 1 and Section II, causes the *n*th divider edge to always occur after the *n*th reference edge. The ADC latches are then turned on following the phase-frequency detector reset pulse, as described in Chapter 1, by a rising edge of v_{ADC_clk}. The ADC output is passed to the $2-z^{-1}$ block and the digital loop controller, both of which are clocked on the reference edge. When the divider is ready to accept a new modulus, a signal named *load* notifies the $2-z^{-1}$ block to update its output $v[n]$.

Finally, for reasons described at length in Section V, the MRQ-DEM encoder output is applied to the DCO on a PLL output edge instead than on a reference edge. The output of the local DEM encoder is applied to the oscillator on the rising edge of a divided down version of the synthesizer output, called *clkP.* The modified DEM encoder outputs are instead applied to the DCO on the divider rising edge delayed by one cycle of *clkP*. The corresponding clock signal is named v_{divP}.

A simplified timing diagram for the synthesizer when locked is shown in Figure 22, where $x[n]$ represents the input sequence to the MRQ-DEM encoder, and $x_f[n]$ was defined in Chapter 2.

II. ΔΣ FDC Circuit Implementation Details

A. Charge Pump

To minimize the impact of finite turn on time of the charge pump current sources, a delay T_{DZ} = 1ns was added in the tri-state phase-frequency detector reset path to ensure that all charge pump current sources fully settle each reference period, as explained in Chapter 1. Furthermore, the addition of an offset current of nominal value $I_{OC} = -I_{CP}$ and duration T_{OC} = 2ns, effectively allows the phase error between the *n*th reference rising edge and the *n*th divider edge to modulate only the duration for which current is sourced to the integrating capacitor. As a result, static mismatches between the current sources in the single-ended charge pump do not significantly affect the performance of the ΔΣ FDC [14, 23, 24].

Although the integrator represented by the combination of the charge pump and capacitor in the FDC-PLL is immune to the level of static mismatches expected for current sources in a 65nm CMOS process, the transient behavior of the charge pump does impact the quantization noise shaping properties of the phase-locked loop. To minimize the phase noise contributed by the charge pump current sources to the overall PLL output, it is desirable to minimize their on time. Similarly, reducing the time the charge pump is active allows to relax the timing requirements of the ADC and the digital loop controller.

Operating the current sources for a small amount of time though requires fast settling of the current sources, otherwise the signal dependency of the net current magnitude will hinter the charge pump linearity. A modified cascode current steering charge pump was implemented in this prototype [14]. The charge pump current sources are shown in Figure 23**,** and a replica of this architecture, in which the positive current source is always off, is used to implement the offset current source.

This circuit permits very fast settling with negligible turn-on or turn-off overshoot. To minimize the current sources switching time, the parasitic capacitance at the drains of M_{n1} and M_{p1}, and the voltage swing at such nodes both need to be kept at a minimum. However, the use of small feature devices and the drain low voltage swing adversely affects both the on and off output impedance of the cascode current sources, which in turn increases the charge pump current dependency on its output voltage.

Extensive simulations show that signal dependency due to a low charge pump output impedance alters the ΔΣ FDC quantization noise spectrum and negatively affects the PLL performance by increasing its output spectrum in-band noise and spurious tone content.

In order to keep the output impedance to an acceptable level when the current sources are on, the channel lengths of M_{n1} and M_{p1} have been set to several multiples of the process minimum feature, while the intrinsic gain of M_{n2} and M_{p2} has been maximized, given the expected output voltage swing. To boost the charge pump output impedance when all current sources are off, a switch is inserted between the charge pump output and the integrating capacitor. The switch uses two sets of dummy devices to minimize charge injection at turn off [24]. Its closed or open state is controlled by the PFD in such a way that the switch closes right before either of the charge pump current sources turn on, and opens right after all of the current sources are turned off.

B. Analog to Digital Converter

The 10-level ADC of this prototype was implemented as a flash ADC. Figure 24 shows the comparator architecture, which consists of a low-gain pre-amplifier followed by a dynamic comparator. The cross-coupled transistors M_{n3} and M_{n4} allow to increase the pre-amplifier gain while maintaining a tight control on it, and a sufficiently low impedance load for the differential pair. The low impedance load, together with two sets of switches, prevents the kick-back from the dynamic comparator from injecting charge into the ADC input capacitor. In order to meet the offset requirements for the comparators, the length of all transistors in the pre-amplifier is at least three times the technology minimum feature. A resistor ladder and an error-correcting encode [54] complete the ADC.

C. Divider

The chosen divider topology, shown in Figure 25, is a modified version of the architecture presented in [55]. The structure consists of a chain of seven programmable divide-by-2/3 cells connected like a ripple counter. If the divider modulus is greater than 127, the NAND gate between cell 5 and cell 6 can be replaced by a wire connecting its negated input to its output. In such a case, the programmable divider operates as follows: once in a division period, the last cell in the chain generates the signal mod_6. This signal then propagates “down” the chain, i.e. from cell 6 to cell 0, being re-clocked by each cell along the way on the rising edge of its input clock. An active mod_6 signal enables the ith cell to divide by 3 once in a division cycle, provided that its programming input p_i is set to 1. Division by 3 adds one extra period of each cell’s input signal to the period of the output signal div_{out}. Hence, a chain of seven divide-by-2/3 cells provides an output signal with period:

$$\begin{aligned} T_{div_{out}} &= 2^7 T_{div_{in}} + 2^6 T_{div_{in}} \cdot p_6 + \ldots + 2T_{div_{in}} \cdot p_1 + T_{div_{in}} \cdot p_0 \\ T_{div_{out}} &= \left(2^7 + 2^6 p_6 + \ldots + 2p_1 + p_0\right) T_{div_{in}} \end{aligned} \tag{140}$$

The purpose of the additional logic between cell 5 and cell 6 is to allow a divider modulus lower than 128, necessary to tune the PLL to the lower output frequency range. This is accomplished by setting mod_6 high whenever the divider modulus is less than 128, thus effectively eliminating the influence of cell 6 from the period of the divider output and shorting the divider chain length to six cells [55].

As the capability of achieving a divisor ratio greater than 127 has to be retained, the divider chain should not be shortened to a length less than 7 when $p_6 = 0$ but $p_7 = 1$, hence the NAND gate [55]. The frequency of the output signal is:

$$f_{div_{out}} = \frac{f_{div_{in}}}{2^7 p_7 + 2^6 p_6 + \ldots + 2p_1 + p_0} \tag{141}$$

while the allowed division range is:

$$2^6 \leq 2^7 p_7 + 2^6 p_6 + \ldots + 2p_1 + p_0 \leq 2^8 - 1 \tag{142}$$

Each divide-by-2/3 cell is implemented following [55] as shown in Figure 26. For the divider to work properly, the following necessary condition needs to be satisfied for the *i*th divider cell:

$$\tau_{clk_i \to Q} + \tau_{clk_{i+1} \to \overline{Q}} + \tau_{clk_i \to Q} + \tau_{NAND_i} < \frac{3}{2} T_{clk_i} \tag{143}$$

Sufficient conditions for correct operation of the *i*th divider cell are instead:

$$\tau_{clk_i \to Q} + \tau_{clk_{i+1} \to \overline{Q}} < T_{clk_i} \tag{144}$$

$$\tau_{clk_i \to Q} + \tau_{NAND_i} < \frac{1}{2} T_{clk_i} \tag{145}$$

For the desired PLL output frequency range these conditions are easily met using CMOS logic in the available 65 nm technology.

Correct operation of the divider in the contest of the FDC-PLL requires the divider modulus loaded during *n*th divider period, and set by the output of the $2{-}z^{-1}$ block, to affect the duration of the *n*th divider period itself. This requirement cannot be met using one of the mod_i signals as divider output as in [55].

However, the loading of a new modulus during *n*th divider period will affect the duration of the *n*th divider period itself, provided that clk_7 or clk_6 are used as divider output signals when $p_7 = 1$ or 0 respectively, and that the new divider modulus is loaded after all of the $(n{-}1)$th mod_i falling edges have occurred, and before $\exists\, i, j$ such that *n*th mod_i falling edge occurs while mod_j is high or has yet to become high during the current division cycle.

In order to satisfy the last condition, mod_5 is used to signal the $2{-}z^{-1}$ block that the divider is ready to accept a new modulus. Notice that the falling edge of the divider output might have been used for this purpose as well. The signal mod_5 was here preferred as in the worst case the falling edge of the divider output occurs 32 input cycles after its rising edge, which was deemed too close to the divider rising edge for one of the multiple configurations the prototype was designed to work in, in which a significant amount of time is given to both the charge pump and ADC to settle.

The divider output is instead set to be clk_7 or clk_6 depending on whether $p_7 = 1$ or 0 respectively, by a multiplexer. The status of p_7 is sampled by mod_4 and used as select signal for such multiplexer. A representative timing diagram for the divider is shown in Figure 27.

III. Quantization Noise Cancellation

In Chapter 1, the contribution of the ADC quantization noise $e_{\text{ADC}}[n]$ to the total FDC-PLL phase noise has been quantified in equation (39). If the conditions outlined in Chapter 1 are satisfied so that $e_{\text{ADC}}[n]$ can be assumed to be asymptotically white and uniformly distributed, the phase-locked loop suppresses the spectral components of $e_{\text{ADC}}[n]$ well within the PLL bandwidth. Outside the PLL bandwidth, $e_{\text{ADC}}[n]$ can become the dominant phase noise source, unless it is heavily filtered by the $L_{LPF}(z)$ portion of the loop filter.

As the phase-locked loop bandwidth is widen, the need to suppress out-of-band quantization noise increases, which in turn increases the filtering requirements of the first-order IIR filters. Unfortunately though, reducing the bandwidth of such filters to improve quantization noise suppression significantly impacts the stability of the synthesizer. The use of IIR or FIR filters with steeper roll-off introduces similar concerns due to their phase response. Hence, a fundamental tradeoff exists between FDC-PLL bandwidth and suppression of FDC quantization noise.

A. General Quantization Noise Cancelling Architecture

A wide PLL bandwidth is however necessary to fight the corruption of the DCO output by nearby circuitry, and in applications that require direct synthesis of phase modulated signals by in-loop modulation. In such scenarios, the degradation in performance associated with a lower suppression of the phase noise due to $e_{\text{ADC}}[n]$ might not be acceptable. Moreover, the analysis presented in [27] and relied upon in Chapter 1 to show that $e_{\text{ADC}}[n]$ is asymptotically white and uniformly distributed, assumes that the feedback path of the ΔΣ FDC does not affect the statistical properties of $e_{\text{ADC}}[n]$. The phase-locked loop however introduces an additional feedback from the ΔΣ FDC output $y[n]$ to its input $x[n]$ in Figure 3. Extensive simulations suggest that this path does alter the statistical properties of $e_{\text{ADC}}[n]$ as the amount of suppression of quantization noise by the loop is decreased, i.e. as the PLL bandwidth is widened, leading to the presence of spurious tones at multiples of αf_{ref} in the synthesizer output spectrum. For the reasons outlined above, further suppression of the quantization noise than what achievable with the $L_{LPF}(z)$ portion of the loop filter implementation given in Chapter 1 is desirable.

The problem of reducing the phase noise that would otherwise arise due to the ADC quantization noise can be addressed by using an N_{ADC}-level ADC, for $N_{\text{ADC}} > 5$, to obtain an estimate, $\hat{e}_{\text{ADC}}[n]$, of $e_{\text{ADC}}[n]$. The phase-to-frequency accumulator output

$p[n]$, as given by equation (14), is available in digital form. Therefore the sequence:

$$-\widehat{e}_{\Delta\Sigma 1}[n] = -\left(\widehat{e}_{ADC}[n] - \widehat{e}_{ADC}[n-1]\right), \tag{146}$$

can be computed and added to $p[n]$ to from the new input to the digital loop filter:

$$\widehat{p}[n] = -\theta_{PLL}\left(\tau_n\right) + \left(N+\alpha\right)\theta_{ref}\left(t_n\right) + \frac{e_p[n]}{\Delta} + e_{\Delta\Sigma 1}[n] - \widehat{e}_{\Delta\Sigma 1}[n] \tag{147}$$

The functional diagram of the general quantization noise cancelling implementation described is shown in Figure 28. To the extent that $\hat{e}_{\mathrm{ADC}}[n]$ is a faithful representation of $e_{\mathrm{ADC}}[n]$, the contribution of $e_{\mathrm{ADC}}[n]$ to the PSD of $\theta_{PLL}(t)$ can be cancelled. Hence, increasing the precision of the analog-to-digital converter allows to suppress to a greater extent the ADC quantization noise impact on the synthesizer phase noise. Consequently, this technique allows to decouple the PLL bandwidth from the $e_{\mathrm{ADC}}[n]$ induced phase noise reduction requirements.

In Figure 29 the contribution of the ADC quantization noise to the output of the PLL is shown before and after the quantization noise cancelling algorithm is applied. If $e_{\mathrm{ADC}}[n]$ were the dominant noise source out of band, the PLL bandwidth could be increased by a factor of ten with respect to the design example of Chapter 1 by using an ADC with 80 levels. A 10-level ADC will instead allow to reduce the contribution of $e_{\mathrm{ADC}}[n]$ to the synthesizer phase noise by 6 dB.

B. Practical Quantization Noise Cancelling Algorithm Implementation

The quantization noise cancelling technique can be efficiently implemented by starting with a 5 level ADC that implements a mid-tread quantizer as stated in Chapter 1. If then every level is bisected, the resulting quantization characteristic will be that of a 10-level mid-rise quantizer. Repeating the bisecting procedure on this mid-rise quantizer doubles again the number of levels while preserving the mid-rise characteristic.

Bisecting the initial mid-treat quantizer characteristic Q times, the total number of levels becomes $N_{\mathrm{ADC}} = 5\cdot 2^{Q}$. If the resulting levels are assigned a binary unsigned or two's complement representation, the $\lceil \log_2(N_{ADC}) \rceil - 3$ LSBs of such representation provide a biased estimate $-\hat{e}_{\mathrm{ADC}}[n]$ of the opposite of the ADC quantization error $e_{\mathrm{ADC}}[n]$ equal to:

$$-\hat{e}_{ADC}[n] = -e_{ADC}[n] + \frac{2^{Q}-1}{2^{Q+1}} + e_{FADC}[n] \tag{148}$$

where $e_{\mathrm{FADC}}[n]$ is the quantization error of the N_{ADC}-level ADC. Notice that the bias of $-\hat{e}_{\mathrm{ADC}}[n]$ is irrelevant for the purpose of the quantization noise algorithm as per equation (146) $\hat{e}_{\Delta\Sigma}[n]$ is zero-mean regardless of the average value of $-\hat{e}_{\mathrm{ADC}}[n]$

The quantization noise cancelling algorithm was implemented in this prototype as shown in Figure 21. The digital loop controller logic was designed to implement the digital portion of the algorithm for a value of $Q = 2$, i.e. for a 20-level ADC. During the design of the ADC comparator though, the power and area consumption required to achieve the necessary comparator offset was deemed excessive. Hence a 10-level ADC was chosen instead. A representative plot of the measured effect of the quantization noise algorithm on the synthesizer output spectrum is shown in Figure 30.

IV. Digital Loop Controller

The digital loop controller consists of three main parts, as shown in Figure 31: the phase-to-frequency accumulator and quantization noise cancelling unit, the digital loop filter and the multiply and requantizer stage. Every part operates in saturation arithmetic, as a two's complement roll-over will cause a phase-locked loop polarity inversion.

The input to the digital loop controller consists of the ADC output and the fractional divisor modulus α. In this prototype, the ADC output sequence $\hat{y}[n] = y[n] - \hat{e}_{\mathrm{ADC}}[n]$ is represented by a 4-bit two's complement binary number. Its 3 MSBs are interpreted as the ADC output integer part $y[n]$, while the LSB represent its fractional part $-\hat{e}_{\mathrm{ADC}}[n]$. The fractional part of the divisor modulus, α is instead represented by a 27-bit number, of which the first bit is interpreted as the sign of α (1 meaning a negative α) and the remaining 26-bit are interpreted as the fractional part of α. The sum of the ADC output integer part and α is accumulated, while the ADC output fractional part is passed through a $1-z^{-1}$ filter and the result is added to the accumulator output to implement the quantization noise cancelling algorithm described in Section III. The output of the accumulator and quantization noise cancelling stage is a 32-bit two's complement number, of which 6 MSBs are interpreted as the number integer part. Only a 7-bit accumulator is required to perform the addition at the output of the accumulator, necessary for the quantization noise cancelling algorithm

The digital loop filter consists of four single-pole infinite impulse response

(IIR) filters followed by a proportional-integral (PI) stage. With the notation used in Chapter 1, assuming none of the internal nodes saturates, the cascade of IIR filters implements the transfer function in (13) scaled by a factor of 1/768, while the PI stage implements the transfer function in equation (12), scaled by a factor of 1/768.

The loop filter coefficients are restricted to powers of two scaled by 768, a factor roughly equal to f_{ref}/K_{DCO}. Doing so allows to implement all the divisions inside the IIR and PI filters as right shifts. As a result, no multipliers and dividers are used in the digital loop controller. All stages operate on a 32-bit input to generate a 32-bit output. The internal signals are also represented using 32-bit.

Finally, the multiply and requantizer stage multiplies the 32-bit digital loop filter output by 768 and requantizes the results to 14-bit. The multiplication is performed by adding the results of two left-shift operations so no multipliers are used here either. A 31-bit LFSR, not shown, provides the set of 18 pseudo-random bits used to perform the requantization step, in order to avoid truncation artifacts.

V. Multi-rate Quantizing DEM Encoder Hardware Efficient Implementation

The multi-rate quantizing DEM encoder presented in Chapter 2 represents a natural choice when deciding which type of dynamic element matching encoder to adopt in order to prevent mismatches among DCO varactor units from introducing non-linear distortion artifacts in the synthesizer output spectrum. By translating the varactor mismatches into highpass shaped frequency noise for the oscillator, the multi-rate quantizing DEM encoder prevents them from introducing spurious tones at the PLL output, while minimally affecting the synthesizer phase noise.

As only the digital ΔΣ modulator and local DEM encoder are operated at a high rate, while most of the logic is operated at the reference oscillator frequency, the multi-rate quantizing DEM encoder provides a significant power reduction over a conventional oversampled DEM encoder of the type shown in Figure 11. Moreover, most varactor elements are operated at the reference frequency so the switching noise is lower than what generated using an conventional oversampled DEM encoder.

Finally, when the PLL is locked, the input to the DEM encoder has a noise-like structure, so no strong signal replicas at multiples of the multi-rate DEM encoder sample rate are present, and the replicas are filtered by the natural frequency-to-phase integration performed by the DCO.

However, due to the presence of multi-bit adders, the power consumption, required area and propagation delay of both segmenting and non-segmenting switching blocks in Figure 15 increase as the layer index k increases. Switching blocks in each layer quantize their input sequences by 1-bit so the power consumption, required area and latency of the whole multi-rate quantizing DEM encoder increases with the number of bits used to represent the input $c[n]$. Furthermore, the implementation details of each switching block are specific to the layer the switching block belongs to.

In the following paragraphs, implementations for non-segmenting and segmenting switching blocks whose power and area requirements as well as propagation delay are independent on the number of bits of their input sequences will be given.

Building upon these results, the MRQ-DEM encoder in Figure 15 will be shown to be implementable without the need for any adders except for the ones present in the ΔΣ modulator of Figure 17.

A. Adder-free Non-segmenting Switching Blocks

An hardware efficient implementation of non-segmenting switching blocks $S_{k,r}$ for $k = 2, 3$, $r = 1, 2$, and $k = 1$, $r = 9, \ldots, 15$, was presented in [56] and is shown with notation compliant to [17] in Figure 32a and Figure 32c. By using an alternative coding convention for the input and output sequences of each switching block, the adders implied by equation (124) can be eliminated. The coding scheme is called *extra-LSB encoding* [57, 58] and it represents each sequence $c_{k,r}[n]$ with k+1 bits denoted $c_{k,r}^{(i)}[n]$, for $i = 0, \ldots, k$. Each bit can have a value of one or zero and the numerical value associated with $c_{k,r}[n]$ is:

$$c_{k,r}[n] = \sum_{i=1}^{k} 2^{i-1} c_{k,r}^{(i)}[n] + c_{k,r}^{(0)}[n] \tag{149}$$

From equations (124), (126) and (128), non-segmenting switching blocks add or subtract a 1 from their input only when their input is odd. For odd inputs, only one between $c_{k,r}^{(0)}[n]$ and $c_{k,r}^{(1)}[n]$ can be set at 1, hence the extra-LSB encoding allows each $S_{k,r}$ switching block to perform the additions and subtractions in (124) without affecting the $k-1$ MSBs of $c_{k,r}[n]$, which can be simply routed to the next block as in Figure 32a.

The sequence generators implement equations (126) and (128), therefore the

$S_{k,r}$ blocks perform spectral shaping of the errors due to mismatch among 1-bit DACs [56].

B. Adder-free Local DEM Encoder

A similar topology is used for non-segmenting switching blocks $S_{k,r}$ for $k = 1, 2$, $r = 1, 2$, in the local DEM encoder of Figure 15. As no spectral shaping of 1-bit DAC mismatches is performed by these switching blocks, their implementation, in Figure 33a, is relatively simpler than what shown in Figure 32a.

The final architecture for the local DEM encoder is shown in Figure 33b. The additional logic shown at the input of the encoder is used to map the ΔΣ modulator output sequence $x_{\Delta\Sigma}[m]$ into the non-negative sequence $c_{\Delta\Sigma}[m]$ for values of $x_{\Delta\Sigma}[m]$ between −2 and +2.

C. Adder-free Segmenting Switching Blocks

For segmenting switching blocks $S_{k,1}$ for $k = 4, \ldots, 14$, the bottom output is a 3-level sequence $c_{15-k,1}[n]$ restricted to the set of values: {0, 1, 2} according to equations (123), (126) and (127). The bottom output can then be represented by a 2-bit sequence in which each bit is unity weighted so it can be coded according to the extra-LSB encoding scheme presented above and interpreted as:

$$c_{15-k,1}[n] = 1 + s_{k,1}[n] = c^{(0)}_{15-k,1}[n] + c^{(1)}_{15-k,1}[n] \qquad (150)$$

Whenever $c_{k,1}[n]$ is even, (150) is either 0 or 2. If $c_{15-k,1}[n] = 0$, the $c^{(0)}_{15-k,1}[n]$ and $c^{(1)}_{15-k,1}[n]$ are both 0. If $c_{15-k,1}[n] = 2$, $c^{(0)}_{15-k,1}[n]$ and $c^{(1)}_{15-k,1}[n]$ are both set to 1. Whenever $c_{k,1}[n]$ is odd, $c_{15-k,1}[n]$ is 1 and the bottom output bits $c_{15-k,1}^{(0)}[n]$ and $c_{15-k,1}^{(1)}[n]$ are set to 1 and 0 respectively.

The extra-LSB encoding used for non-segmenting switching blocks can't instead be used for the input $c_{k,1}[n]$ and top output $c_{k\text{-}1,1}[n]$ sequences. A slightly different encoding scheme called *negative-extra-LSB encoding* has been devised, in which the negative-extra-LSB encoding of $c_{k,1}[n]$ consists of k+1 bits that are denoted $c_{k,1}^{(i)}[n]$ (i = 1,…, k) and $c_{k,1}^{(-)}[n]$, each of which take on a value of one or zero. The numerical value of $c_{k,1}[n]$ interpreted as:

$$c_{k,1}[n] = \sum_{i=1}^{k} 2^{i-1} c_{k,1}^{(i)}[n] - c_{k,1}^{(-)}[n] \tag{151}$$

so that the extra bit $c_{k,1}^{(-)}[n]$ has an effective weight of −1. A conventional unsigned binary encoded number can be converted to a negative-extra-bit encoded number by appending an extra 0th bit and setting it low.

Whenever $c_{k,1}[n]$ is odd, $s_{k,1}[n]$ = 0 and the top output has to be set to:

$$\frac{c_{k,1}[n]-1}{2} = \frac{c_{k,1}[n]}{2} - \left\{\frac{c_{k,1}[n]}{2}\right\} = \left\lfloor \frac{c_{k,1}[n]}{2} \right\rfloor \tag{152}$$

which is implemented by right shifting by 1 bit the k−1 MSBs of $c_{k,1}[n]$ and setting $c_{k-1,1}^{(-)}[n]$ = 0.

Whenever $c_{k,1}[n]$ is even, its $c_{k,1}^{(1)}[n]$ bit is zero. If $s_{k,1}[n]$ = −1, the top output is:

$$\frac{1}{2}\left(c_{k,1}[n]-1-s_{k,1}[n]\right) = \frac{c_{k,1}[n]}{2} \tag{153}$$

which is simply implemented by right shifting by 1 bit the k–1 MSBs of $c_{k,1}[n]$ and setting $c_{k-1,1}^{(-)}[n] = 0$. If instead $s_{k,1}[n] = +1$, $c_{k\text{-}1,1}[n]$ has to be set to:

$$\frac{1}{2}\left(c_{k,1}[n]-1-s_{k,1}[n]\right)=\frac{c_{k,1}[n]-2}{2}=\frac{c_{k,1}[n]}{2}-1 \qquad (154)$$

which is implemented by right shifting by 1 bit the k–1 MSBs of $c_{k,1}[n]$ and setting $c_{k-1,1}^{(-)}[n] = 1$. The RTL view for adder-free segmenting switching blocks is shown in Figure 32b and Figure 32c.

D. Adder-free Multi-rate Quantizing DEM Encoder

Switching blocks $S_{1,r}$ for r = 1, …, 15, $S_{k,r}$ for k = 2, 3, r = 1, 2, $S_{k,1}$ for k = 4, …, 14 in the modified DEM encoder of Figure 15, and $S_{k,r}$ for k = 1, 2, r = 1, 2, in the local DEM encoder of Figure 33b can be implemented as shown in Figure 32a, Figure 32b, Figure 32c and Figure 33a to eliminate the need for adders and therefore the dependence between the hardware complexity and the number of bits used to represent the input sequence. However, the negative-extra-LSB representation of the top output of $S_{4,1}$ still needs to be mapped to the extra-LSB representation adopted for the input of switching block $S_{3,1}$. Moreover, the adders and scaling factors interposed between the bottom outputs of switching blocks $S_{k,1}$ for k = 7, …, 14 and the ΔΣ modulator input have to be removed as well.

Assume a 14-bit two’s complement representation for the input sequence $x[n]$, where the bits are denoted $x^{(i)}[n]$ (i = 1,…, 14), so that its numerical value is interpreted to be:

$$x[n] = \Delta\left(\sum_{i=1}^{13} 2^{i-1} x^{(i)}[n] - 2^{13} x^{(14)}[n]\right) \tag{155}$$

The sequence $x[n]$ has to be mapped to the integer sequence $c[n]$ of Figure 15, whose range is restricted to the set specified in (122). From (155), $x[n]$ can take on any value in the set $\{-2^{13}\Delta, \ldots, (2^{13}-1)\Delta\}$ and therefore it can be mapped to a non-negative integer sequence $c[n]$ within the valid range of (122) by adding an offset of either $2^{13}+2047$ or $2^{13}+2048$ to its two's complement representation $x[n]/\Delta$.

An offset of 2^{13} can be simply added by inverting the MSB of $x[n]/\Delta$, $x^{(14)}[n]$. Accordingly we define a new sequence $c_c[n] = x[n]/\Delta + 2^{13}$, whose encoding consists of 14-bit denoted $c_c^{(i)}[n]$ $(i = 1, \ldots, 14)$ assigned as:

$$\begin{aligned} c_c^{(14)}[n] &= 1 - x^{(14)}[n] \\ c_c^{(i)}[n] &= x^{(i)}[n] \quad \text{for } i = 1, 2, \ldots, 13 \end{aligned} \tag{156}$$

and interpreted as:

$$c_c[n] = \frac{x[n]}{\Delta} + 2^{13} = \sum_{i=1}^{14} 2^{i-1} c_c^{(i)}[n] \tag{157}$$

By recursively applying equation (123) for $k = 4, 5, \ldots, 14$ it can be shown that an offset of 2048 added to the MRQ-DEM encoder input sequence $c[n]$ won't affect the operation of switching blocks $S_{k,1}$ for $k = 4, 5, \ldots, 14$. However, the offset will affect the operation of $S_{3,1}$ by adding a one to its input sequence. Therefore an offset of 2048 added to the encoder input sequence $c[n]$ is equivalent to direct addition of a one at the input of switching block $S_{3,1}$. By design, the input to $S_{3,1}$ plus one is equal to the top output of $S_{4,1}$ plus one. In negative-extra-LSB notation, this is:

$$c_{3,1}[n]+1=\sum_{i=1}^{3}2^{i-1}c_{3,1}^{(i)}[n]-c_{3,1}^{(-)}[n]+1 \tag{158}$$

The sequence specified by (158) can be represented in extra-LSB notation as:

$$c_{3,1}[n]+1=\sum_{i=1}^{3}2^{i-1}c_{3,1}^{(i)}[n]+c_{3,1}^{(0)}[n] \tag{159}$$

by assigning:

$$c_{3,1}^{(0)}[n]=1-c_{3,1}^{(-)}[n] \tag{160}$$

The implementation of (159) and (160) consists simply in assigning as input to $S_{3,1}$ the 4-bit sequence composed by bits $c_{3,1}^{(i)}[n]$ for i = 0, …, 3, where $c_{3,1}^{(3)}[n]$, $c_{3,1}^{(2)}[n]$, $c_{3,1}^{(1)}[n]$ are provided as top output by switching block $S_{4,1}$, and $c_{3,1}^{(0)}[n]$ is obtained by inverting $c_{3,1}^{(-)}[n]$ as generated by $S_{4,1}$.

The sequence of operations outlined so far allows to map the multi-bit MRQ-DEM DAC input sequence $x[n]$ into a sequence $c[n] = c_c[n]+2048 = x[n]/\Delta+2^{13}+2048$ which takes on values within the range of equation (122). The mapping is performed without requiring any extra hardware except for two inverters.

Finally, the adders and scaling factors at the input of the second-order digital ΔΣ modulator are removed. Recursively applying (123) for k = 7, 8, …, 14, the input sequence to switching block $S_{6,1}$ can be written as:

$$c_{6,1}[n]=\frac{c[n]}{2^8}-\sum_{k=7}^{14}\frac{1+s_{k,1}[n]}{2^{k-(14-8)}} \tag{161}$$

From equation (129):

$$\begin{aligned} c_f[n] &= \frac{x_f[n]}{\Delta} + 255 \\ &= \sum_{k=7}^{14} 2^{14-k}\left(1 + s_{k,1}[n]\right) = 2^8 \sum_{k=7}^{14} \frac{1 + s_{k,1}[n]}{2^{k-(14-8)}} \\ &= 2^8 \left[\sum_{k=7}^{14} \frac{1 + s_{k,1}[n]}{2^{k-(14-8)}} - \frac{c[n]}{2^8}\right] + c[n] \end{aligned} \tag{162}$$

which can be rewritten using equation (161) as:

$$c_f[n] = c[n] - 2^8 c_{6,1}[n] \tag{163}$$

From (151), the input sequence to switching block $S_{6,1}$ is represented in negative-extra-LSB notation as:

$$c_{6,1}[n] = \sum_{i=1}^{6} 2^{i-1} c_{6,1}^{(i)}[n] - c_{6,1}^{(-)}[n] \tag{164}$$

Furthermore, bits $c_{6,1}^{(i)}[n]$ (i = 1, …, 6) are equal to the 6 MSBs of $c_c[n]$, hence:

$$\sum_{i=1}^{6} 2^{i-1} c_{6,1}^{(i)}[n] = \frac{1}{2^8}\left[c[n] - \sum_{i=1}^{8} 2^{i-1} c_c^{(i)}[n]\right] \tag{165}$$

Plugging equation (165) into (164) and the result into (163) leads to:

$$c_f[n] = 2^8 c_{6,1}^{(-)}[n] + \sum_{i=1}^{8} 2^{i-1} c_c^{(i)}[n] \tag{166}$$

The input to the second-order digital ΔΣ modulator in Figure 15 is by equation (162): $x_f[n] = \Delta(c_f[n] - 255)$. In this adder-free implementation, it will instead be set to:

$$x_f[n] = \Delta\left(c_f[n] - 2^8\right) = \Delta\left[2^8\left(c_{6,1}^{(-)}[n] - 1\right) + \sum_{i=1}^{8} 2^{i-1} c_c^{(i)}[n]\right] \tag{167}$$

The additional offset of −Δ is added in order to simplify the logic needed to generate the ΔΣ modulator input sequence, as it will be evident in the following paragraph.

The second term in parenthesis in equation (167) is such that:

$$0 \le \sum_{i=1}^{8} 2^{i-1} c^{(i)}[n] \le 255 \tag{168}$$

while the first term in parenthesis is either 0 or −256. Hence, using a 9-bit two's complement representation for $x_f[n]/\Delta$ where the bits are denoted $x_f^{(i)}[n]$ (i = 1,…, 9), equation (168) is satisfied by setting:

$$\begin{aligned} x_f^{(9)}[n] &= c_{6,1}^{(-)}[n] - 1 \\ x_f^{(i)}[n] &= c_c^{(i)}[n] \quad \text{ for } i = 1, 2, \ldots, 8 \end{aligned} \tag{169}$$

which can be implemented by simply routing the 8 LSBs of $c_c[n]$ to form the 8 LSBs of $x_f[n]/\Delta$, inverting $c_{6,1}^{(-)}[n]$ and appending it as MSB of $x_f[n]/\Delta$. Notice that the possibility of using the inverted version of $c_{6,1}^{(-)}[n]$ as MSB of $x_f[n]/\Delta$ comes from using an offset of -2^8 in equation (167), instead of −255 as required by equations (129) and (162).

To summarize, it is possible to remove all adders from the multi-rate quantizing DEM encoder of Figure 15, except the ones used to implement the digital ΔΣ modulator, by:

- implementing the local DEM encoder in Figure 15 as shown in Figure 33b,
- replacing all segmenting and non-segmenting switching blocks in Figure 15 and Figure 33b with their adder-free versions in Figure 32a, Figure 32b, Figure 32c and Figure 33a,
- using as input to $S_{14,1}$ the sequence $c_c[n]$, obtained by tying $c_{14,1}^{(-)}[n]$ to 0, routing

bits 1 through 13 of $x[n]/\Delta$ to be bits 1 through 13 of $c_c[n]$, and assigning the inverted version of $x^{(14)}[n]$ to be $c_c^{(14)}[n]$,

- assigning the inverted version of $c_{4,1}^{(-)}[n]$ to $c_{3,1}^{(0)}[n]$,
- routing bits 1 through 8 of $c_c[n]$, i.e. bits 1 through 8 of $x[n]/\Delta$, to be bits 1 through 8 of $x_f[n]/\Delta$ and assign the inverted version of $c_{6,1}^{(-)}[n]$ to $x_f^{(9)}[n]$.

The final adder-free MRQ-DEM encoder is shown in Figure 34. Observe that, although to be consistent with previous figures and notation the inputs to each switching blocks consist of multiple bits, the digital logic of each switching block operates only on two bits, so all switching blocks of the same type share the same implementation and routing complexity regardless of the layer they belong to.

E. MRQ-DEM ΔΣ Modulator Timing

In Chapter 1, the input sequence to the digitally controlled oscillator is applied to the varactor bank on a rising edge of the reference oscillator signal. Unfortunately, this choice would require the generation of a clock signal synchronous with the reference oscillator output, yet of frequency several times that of the reference, in order to be able to clock the second-order ΔΣ modulator in the MRQ-DEM encoder of Figure 15.

A practical alternative is instead to use a *P*-times divided down version of the synthesizer output as clock for the ΔΣ modulator [6]. Increasing the ΔΣ modulator clock frequency trades power consumption for higher frequency resolution and lower phase-noise at the PLL output. A major drawback of this approach though is that the digital ΔΣ modulator clock phase is free to rotate with respect to the phase of the reference. Hence, the input to the ΔΣ modulator needs to be re-sampled across different frequency domains introducing the potential for metastability [6].

In the prototype here presented, a different approach was taken to avoid the metastability issue. The modified DEM encoder outputs are made available on the rising edge of the reference, and the encoder itself is clocked once per reference period. The sequence $x_f[n]$ is then re-sampled on the rising edge of the divider output and provided as input to the ΔΣ modulator. The presence of an offset current in the PLL architecture ensures that, once the PLL is locked, the *n*th divider rising edge always occurs sufficiently after the *n*th reference edge. Hence, the input to the second-order ΔΣ modulator is always sampled correctly without potential for metastability.

In order to avoid metastability inside the ΔΣ modulator itself, the clock signal for its feedback path is derived from the output of the divider divide-by-2/3 cells. Restricting P to be a power of 2, and using one of the divider's internal signals, $clk_{\log_2 P}$, the number of delta-sigma modulator clock periods per divider period is constant, removing any risk of metastability. The clock signal used for the ΔΣ modulator is dubbed *clkP*.

The output of the local DEM encoder is then applied to the DCO on the next rising edge of *clkP.* The modified DEM encoder outputs are instead applied to the DCO on the divider rising edge delayed by one cycle of *clkP*. The corresponding clock signal is named v_{divP} and its purpose is to prevent a systematic timing mismatch between the time the *n*th output word of the modified DEM encoder is applied to the oscillator and time the local DEM encoder output representing the first sample generated by the ΔΣ modulator with input $x_f[n]$ is applied to the oscillator. Such a mismatch would be translated into shaped white frequency noise by the MRQ-DEM encoder, but it would still unnecessarily degrade the synthesizer phase noise performance.

F. Period Distortion Compensating ΔΣ Modulator

Using the output of one of the divider's divide-by-2/3 cells as clock for the digital ΔΣ modulator in the MRQ-DEM encoder eliminates the risk of metastability as mentioned above. However, if the during the *n*th divider period, the divider modulus $N-v[n]$, is not an integer multiple of P, the duration of one of the *clkP* periods is extended by up to $P-1$ DCO periods. As a result, due to this clock period stretching, once per divider period, the synthesizer frequency variation induced by one of the ΔΣ modulator output samples will be effectively multiplied by $\gamma[n] = 1 + \left\langle \left(N - v[n] \right) / P \right\rangle$.

The ΔΣ modulator output sequence is equal to its input plus quantization noise. The input signal $x_f[n]$, being a constant over each divider period, doesn't undergo any distortion due to the period stretching: it's corresponding average frequency variation over a divider period is the same regardless of the duration of each ΔΣ clock period.

On the other hand, the ΔΣ modulator quantization noise $e_{\Delta\Sigma}[m]$ and the input dither $d_{\Delta\Sigma}[m]$ are sequences of non-constant values over the *n*th divider period. Only the frequency variation induced by one these values per each sequence is effectively multiplied by $\gamma[n]$. As a result of this multiplication, some of the desirable properties of the quantization noise are lost. For example, there is no guarantee that the quantization noise power spectral density has a double zero at zero frequency, as the sum of its running sum is not bounded anymore.

In order to maintain the desirable noise shaping characteristic of the digital ΔΣ modulator, the period distortion compensating ΔΣ modulator of Figure 35 can be implemented in place of the one shown in Figure 17. Here, in each divider period, a multiplier at the input of the $2z^{-1}-z^{-2}$ feedback filter multiplies the quantization error sample generated during the stretched ΔΣ clock period by $\gamma[n]$. In doing so, it ensures that the ΔΣ modulator keeps track of the quantization error as it appears in terms of frequency variation at the output of the PLL. A division of $1/\gamma[n]$ is then performed at the output of the feedback filters to keep the ΔΣ modulator signal and noise transfer functions constant over all clock cycles.

As a result of the multiplication by $\gamma[n]$, the number of quantization levels necessary to prevent the modulator from overloading at any given time is increased from five to seven. In order to represent seven thermometer coded output values, the number of varactor units of weight $256\Delta_F$ driven by the local DEM encoder needs to be six, instead of four. However, using a non-power of two number of varactors increases the complexity of the local DEM encoder, hence it might be preferable to use a total number of eight varactors.

The period compensating ΔΣ modulator with $P = 16$ allows to reduce the synthesizer output phase noise by 8.6 dB at an offset of 10 MHz, in absence of DCO noise. Once the DCO phase noise is taken into account, the period compensating ΔΣ modulator allows to reduce the PLL phase noise at an offset frequency of 10 MHz from the carrier by about 3 dB. For smaller values of *P*, the performance improvement is more modest. The reason is that in this case the relative amount of distortion per divider period is less, hence the noise due to ΔΣ period stretching is lower than the noise introduced by the oscillator itself. The hardware overhead required to implement the period compensating ΔΣ modulator was deemed excessive and removed from the prototype described in this chapter before fabrication. The ΔΣ modulator of Figure 17 was used instead, in the form shown in Figure 36. However, for small ratios of *N*/*P*, or better performing oscillators, the advantage of using the ΔΣ modulator in Figure 35 is significant.

VI. DIGITALLY CONTROLLED OSCILLATOR

The digitally controlled oscillator implemented in this prototype is built starting from a standard LC oscillator. The final design is similar to what presented in [7], except for the capacitor banks, which are modified with respect to [7] to reduce the supply sensitivity of nMOS based varactors found in a previous prototype of this DCO. A diagram of the oscillator, together with the implementation of the coarse, intermediate and fine varactor banks is shown in Figure 37, Figure 38 and Figure 39 respectively.

A single switch topology was chosen to maximize the oscillator output amplitude, in order to minimize its phase noise [59].

A custom three-turn center-tapped inductor was originally designed to have a differential inductance of 1.856 nH with a differential quality factor *Q* of 17 at 3.6 GHz. The custom inductor employs the techonology aluminum redistribution layer stacked with metal 7 for the main spiral, with underpass in metal 6. Due to the necessity to account for the self-inductance of the inductor feeding lines, after post-layout simulation, the custom inductor was replaced with a two-turn 1.5 nH inductor of *Q* = 16.5 available in the design kit library. A patterned ground ring is used to attenuate substrate losses.

The inductance tunes with three banks of capacitors to provide an output frequency range between 3.1 GHz and 4 GHz, which was found suitable to cover the range of GSM bands targeted. The three capacitor banks are designed to provide coarse, intermediate and fine frequency tuning respectively. Due to unavailability of MIM capacitors in the process used, all capacitor banks employ interdigitized MOM capacitors.

The coarse bank consists of seven binary-weighted varactors, each of which has the architecture shown in Figure 37. A set of nMOS switches is programmed via serial port to switch in and out of the tank each varactor unit. The input coming from the serial port is re-buffered on the DCO supply to avoid unintended frequency modulation of the oscillator. Two *R* = 40 kΩ high resistivity polysilicon resistors are used to maintain the switch voltages so that none of the nMOSes is forward biased under any condition, a situation that would cause additional loading of the LC tank with consequent degradation of the oscillator phase noise. The large value of *R* helps reducing the impact of the buffers non-linear output capacitance. The unit frequency step corresponding to a variation of 1 LSB of the coarse tuning bank input word is designed to be about $\Delta_C = 8$ MHz.

The intermediate bank consists of six binary-weighted varactors, each of which has the architecture shown in Figure 38. The intermediate bank topology is similar to the one used for the coarse varactor unit, except for one extra capacitor placed across the nMOS switches in order to reduce the frequency step achievable with the available MOM capacitors. The unit frequency step corresponding to a variation of 1-LSB of the intermediate tuning bank input word is designed to be $\Delta_I = 400$ kHz, so that the intermediate bank full range is approximately equal to three times the coarse bank minimum frequency step.

The fine tuning bank is shown in Figure 39. The minimum fine bank varactor frequency step is set to be $256\Delta_F = 34$ kHz in order to meet the stringent phase noise requirements of the GSM standard. The bank consists of eighteen varactor units, six of size $256\Delta_F$, two of size $512\Delta_F$, two of size $1024\Delta_F$, and eight of size $2048\Delta_F$. The input to the fine varactor bank is provided by the MRQ-DEM encoder as described in Section V. Hence the input to four of the $256\Delta_F$ sized varactors is updated at a rate of f_{DCO}/P, while the input to the remaining fourteen varactors is updated once per divider period. The fine bank topology consists of a capacitor bridge which can be balanced by an nMOS switch. Each arm of the bridge has two capacitor whose capacitance differs by $\Delta_{FC} = 0.7$ fF, the minimum amount allowed in the technology used. A set of nMOSes whose sources and drains are shorted is driven by a signal of opposite polarity with respect to the varactor input signal in order to prevent the main switch charge injection from altering the amount of charge in the MOM capacitors. This architecture provides the desired frequency step of 34 kHz while being very insensitive to supply variations. Each varactor unit switch input is set by a fully differential D flip-flop, which consists of two copies of the flip-flop presented in [60]. Both the clocks and the inputs to the flip-flops are provided differentially to minimize the amount of current coming from the digital supply and returning through the dedicated fine bank varactor supply.

In order to withstand a peak-to-peak voltage swing of almost twice the supply value of 1.0 V, the cross-coupled devices of the DCO were chosen to be thick-oxide devices available for IO purposes in the process used.

The oscillator bias current can be adjusted using a bank of triode nMOSes whose gate is controlled via serial port [7]. A tail LC filter tuned around the second harmonic of the nominal output frequency is used to suppress the noise coming from the triode nMOSes and reduce the oscillator frequency pushing due to power supply variations [59].

VII. Measurement Results

The FDC based phase-locked loop described in this Chapter was fabricated in a 65nm 7 metal layer CMOS process provided by ST Microelectronics. The prototype occupies an area of 1.3 mm^2 with an active area of about 0.5 mm^2 including all input and output buffers. The total current drawn is 20.4 mA. The DCO, digital loop controller, MRQ-DEM encoder, divider, and output buffers are powered from a 1.0V supply, while the phase-frequency detector, charge-pump, ADC and reference buffer supply voltage is set to 1.2 V. A photograph of a packaged die is shown in Figure 40.

At the time this dissertation was completed the prototype testing was still in progress. A representative set of measurements of the PLL phase noise across the five boards tested is shown in Table 3**.** A plot of the measured PLL output phase noise is presented in Figure 41. The power of in band fractional spurious tones is reported in Table 4 for several values of the fractional divisor modulus and a 40 kHz bandwidth.

The DCO measured $1/f^3$ phase noise is significantly higher than what obtained by simulation. For the measured amount of DCO noise, the FDC PLL model developed in Chapter 1 predicts the phase noise to be −80 dBc/Hz at a 10 kHz offset for a frequency synthesizer with the settings listed in Table 2. The measured in-band phase noise though is several dB higher, and the fractional spurious tone power is sensitive to both the charge pump current and ADC step size. For some values of α, the fractional spurious tone power is lower than the noise floor of the spectrum analyzer used, as shown in Figure 42, while for others values of α it can be as high as -36 dBc. The worst case measured reference spur across all five boards is -82 dBc, for a 40 kHz bandwidth.

All the plots shown were obtained from the part in board number 4, for which a FIB edit was performed in order to correct a wiring mistake in the MRQ-DEM encoder $\Delta\Sigma$ modulator input MSB. The MRQ-DEM encoder mismatch shaping algorithm is disabled for the all the evaluations reported. The reason for the increased in-band phase noise and spurious tone degradation is currently being investigated.

ACKNOWLEDGEMENTS

The dissertation author is the primary investigator and author of this chapter. Professor Ian Galton supervised the research which forms the basis for this chapter. The author would like to thank Colin Weltin-Wu, for place and route of the digital loop controller and the modified DEM encoder, Jason Remple for contributing to the development of part of the simplified logic for the MRQ-DEM encoder, Kevin Wang and Gerry Taylor for helpful discussions.

FIGURES

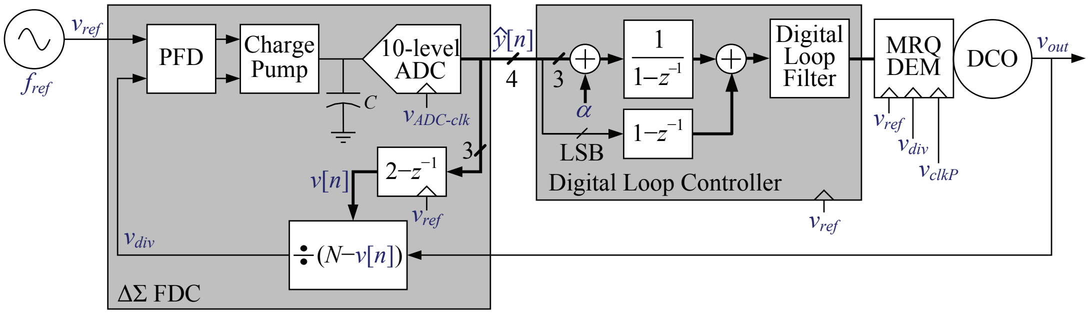


Figure 21: Delta-sigma FDC-PLL prototype architecture

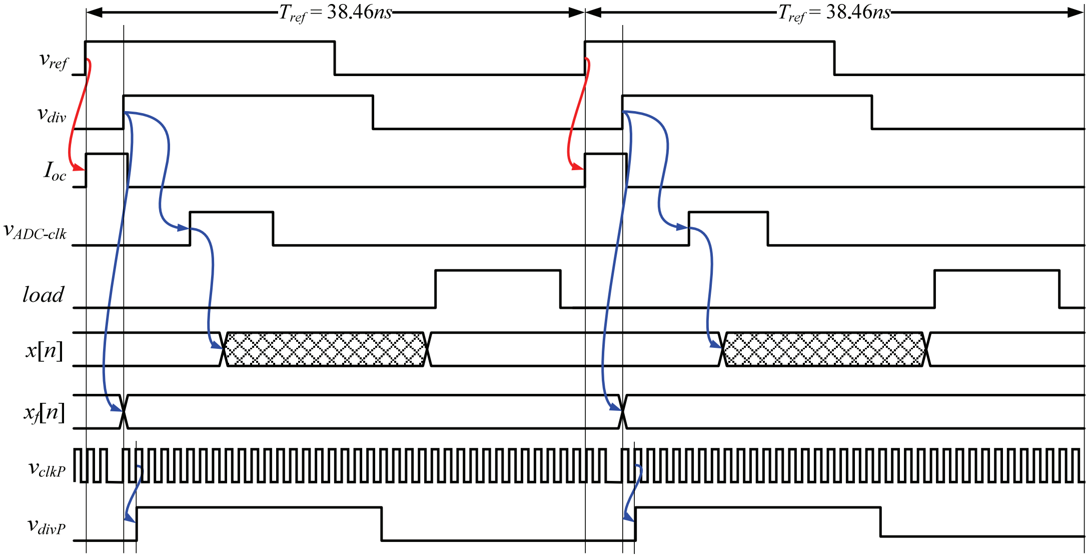


Figure 22: Simplified timing diagram of the FDC-PLL synthesizer prototype

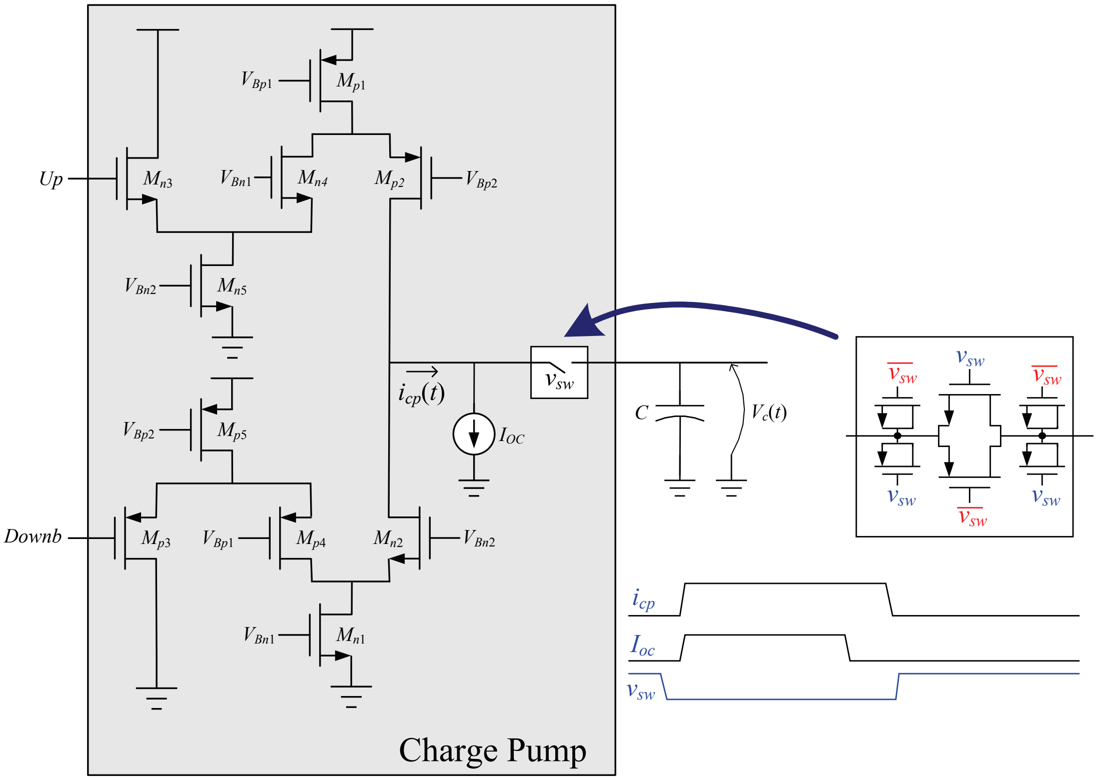


Figure 23: Charge pump circuit implementation

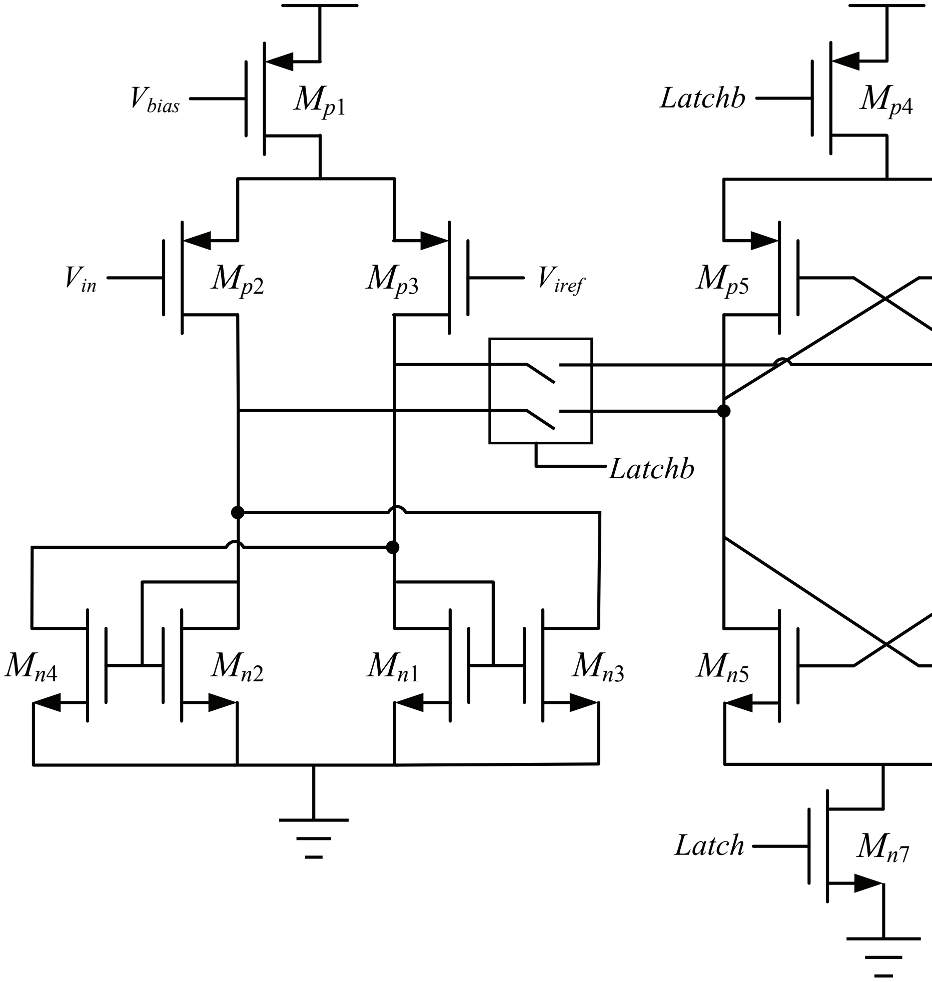


Figure 24: Comparator for the 10-level flash ADC

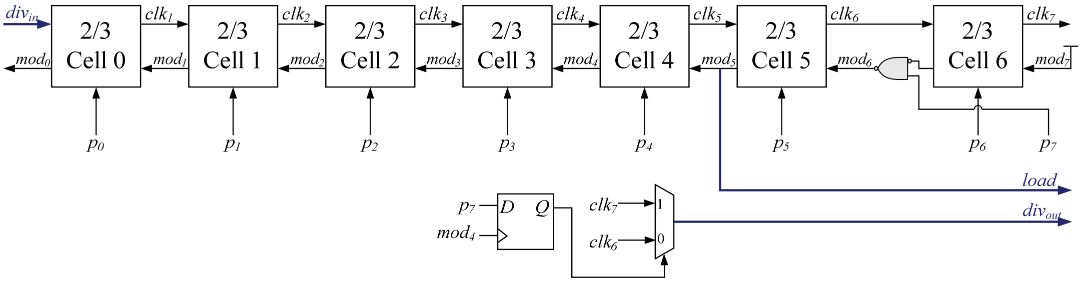


Figure 25: Divider architecture

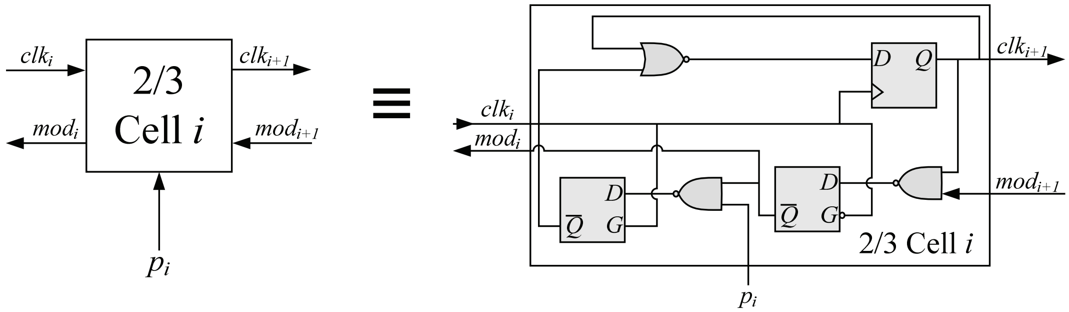


Figure 26: Divide-by-2/3 cell implementation

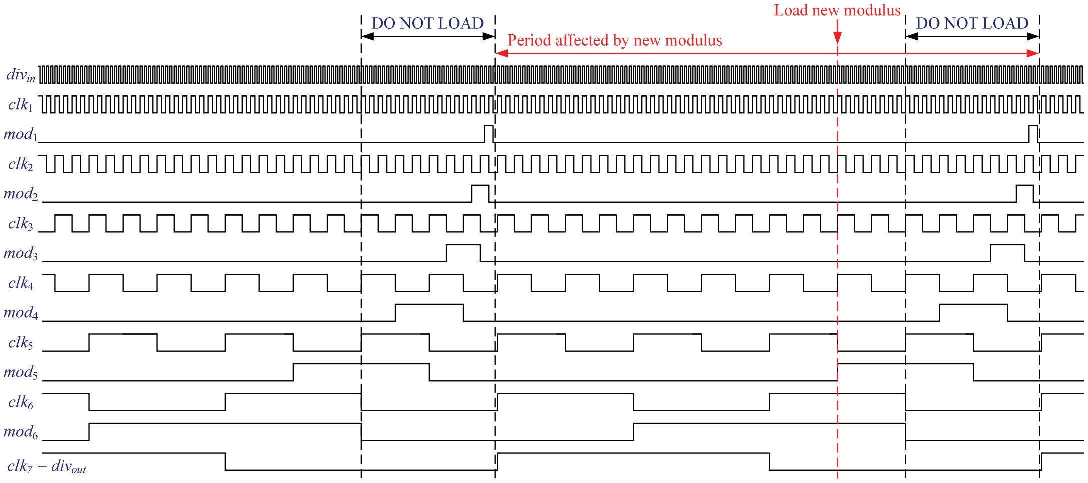


Figure 27: Representative divider time diagram for the case of a divider modulus equal to 128

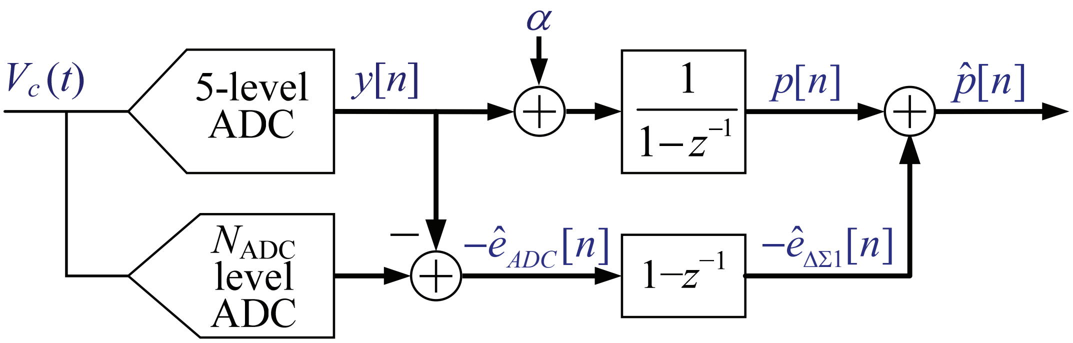


Figure 28: Functional diagram of the general quantization noise cancelling algorithm implementation

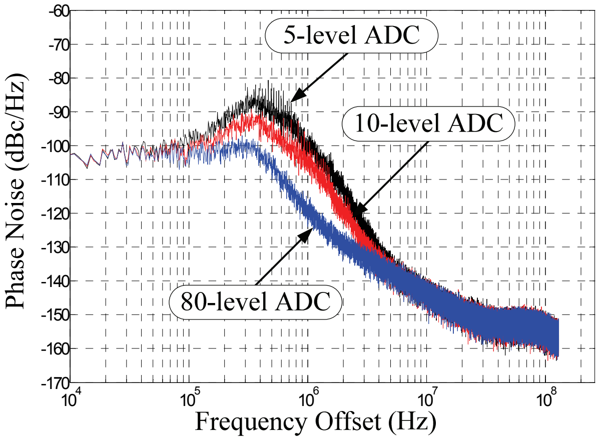
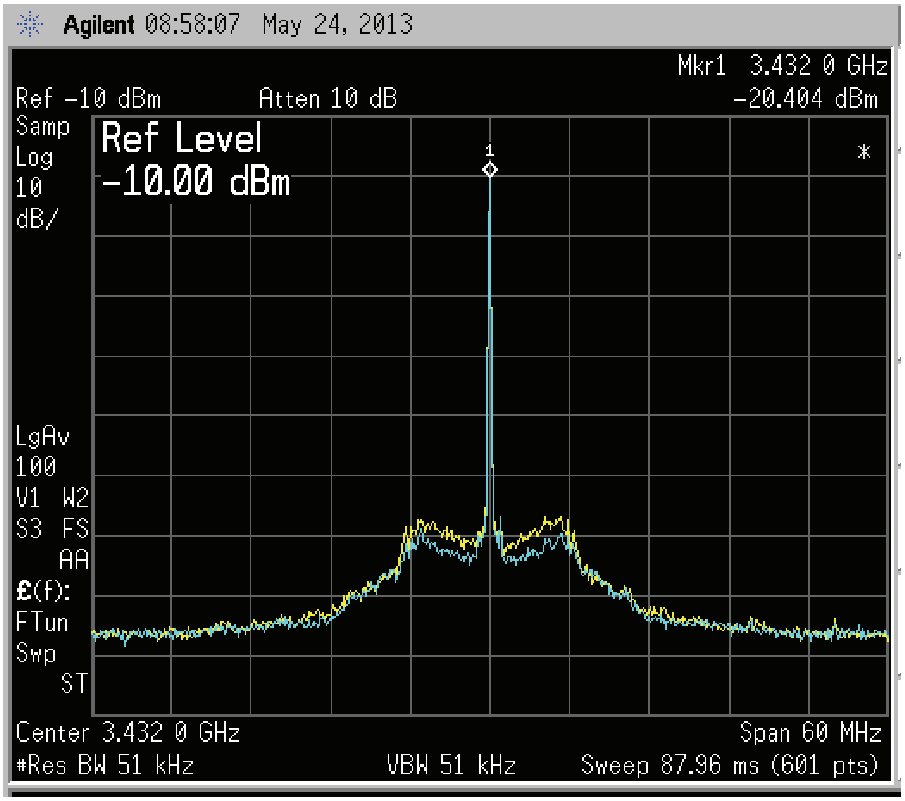


Figure 29: Effect of quantization noise cancellation on the synthesizer output phase noise. The black curve represents the PLL output phase noise when the quantization noise algorithm is disabled. When quantization noise cancellation is enabled with a 10-level ADC, a 6 dB reduction in the PLL output phase noise is achievable at frequencies where the FDC quantization noise dominates (red curve). A 80-level ADC allows to suppress the quantization noise to a point where it is no more a dominant source of phase noise for the synthesizer (blue curve).



Agilent 08:58:07 May 24, 2013

Mkr1 3.432 0 GHz
Ref -10 dBm Atten 10 dB -20.404 dBm
Samp
Log
10
dB/

Ref Level
-10.00 dBm

LgAv
100
V1 W2
S3 FS
AA
£(f):
FTun
Swp
ST

Center 3.432 0 GHz Span 60 MHz
#Res BW 51 kHz VBW 51 kHz Sweep 87.96 ms (601 pts)

Figure 30: Measured effect of the quantization noise cancelling algorithm. The yellow curve shows the synthesizer output spectrum with both the IIR filters and the quantization noise cancelling algorithm disabled. The blue curve shows the synthesizer output spectrum with the IIR filters disabled and the quantization noise cancelling algorithm enabled

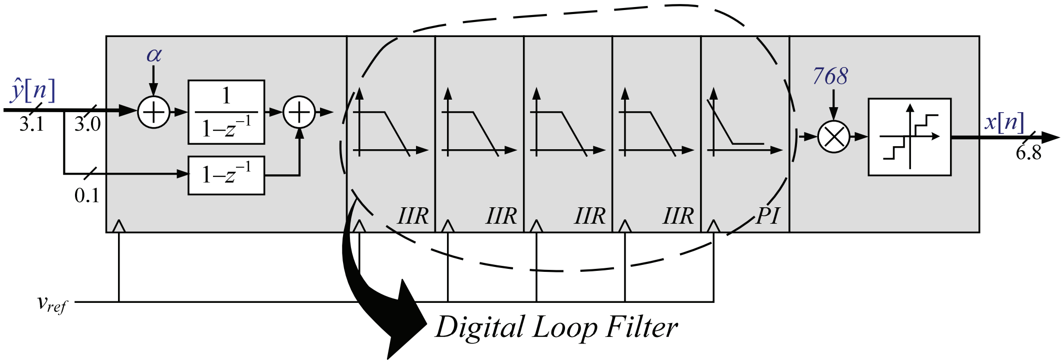


Figure 31: Digital loop controller functional diagram

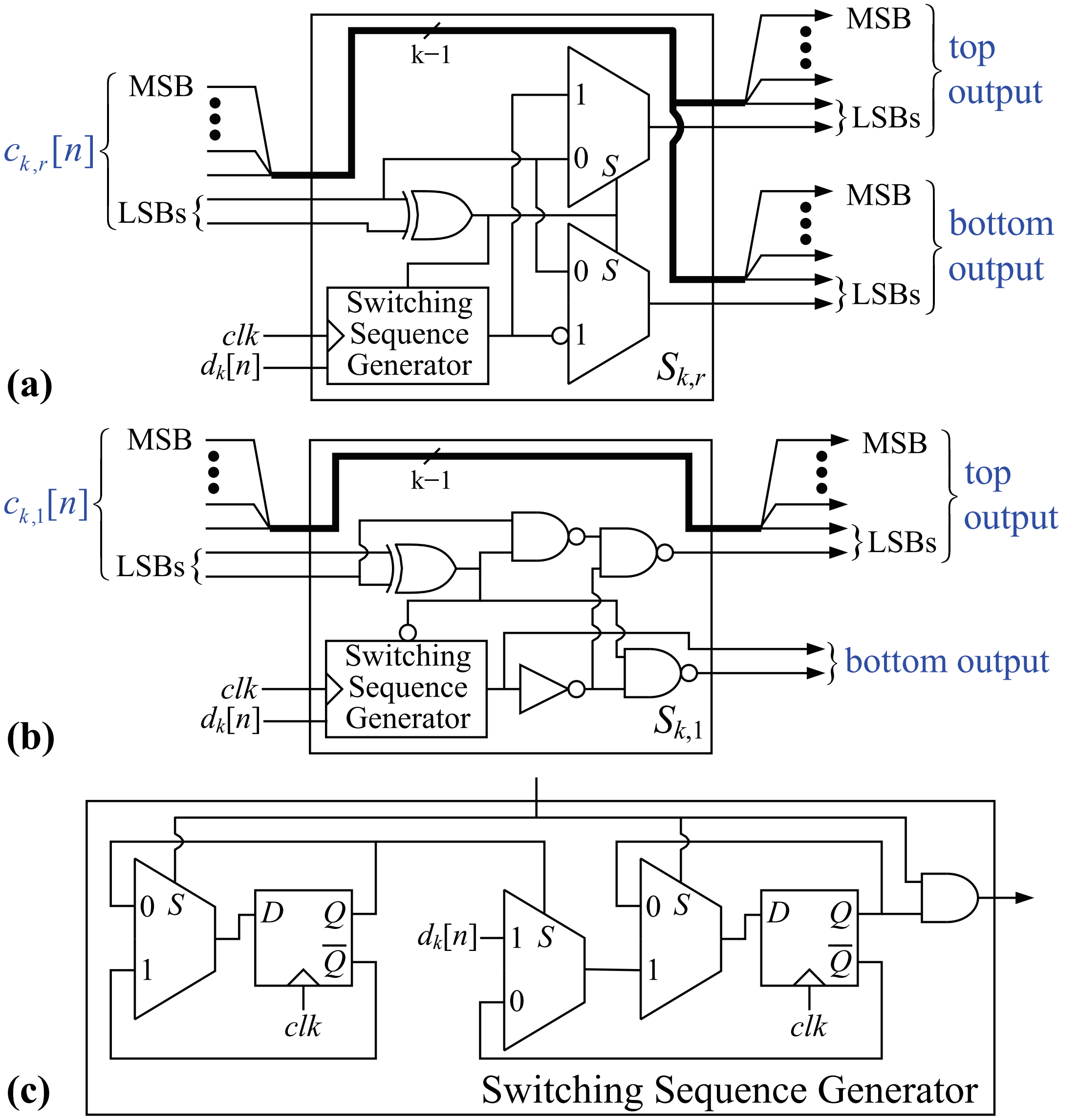


Figure 32: Adder-free implementation of a) non-segmenting and b) segmenting switching blocks and c) the switching sequence generator

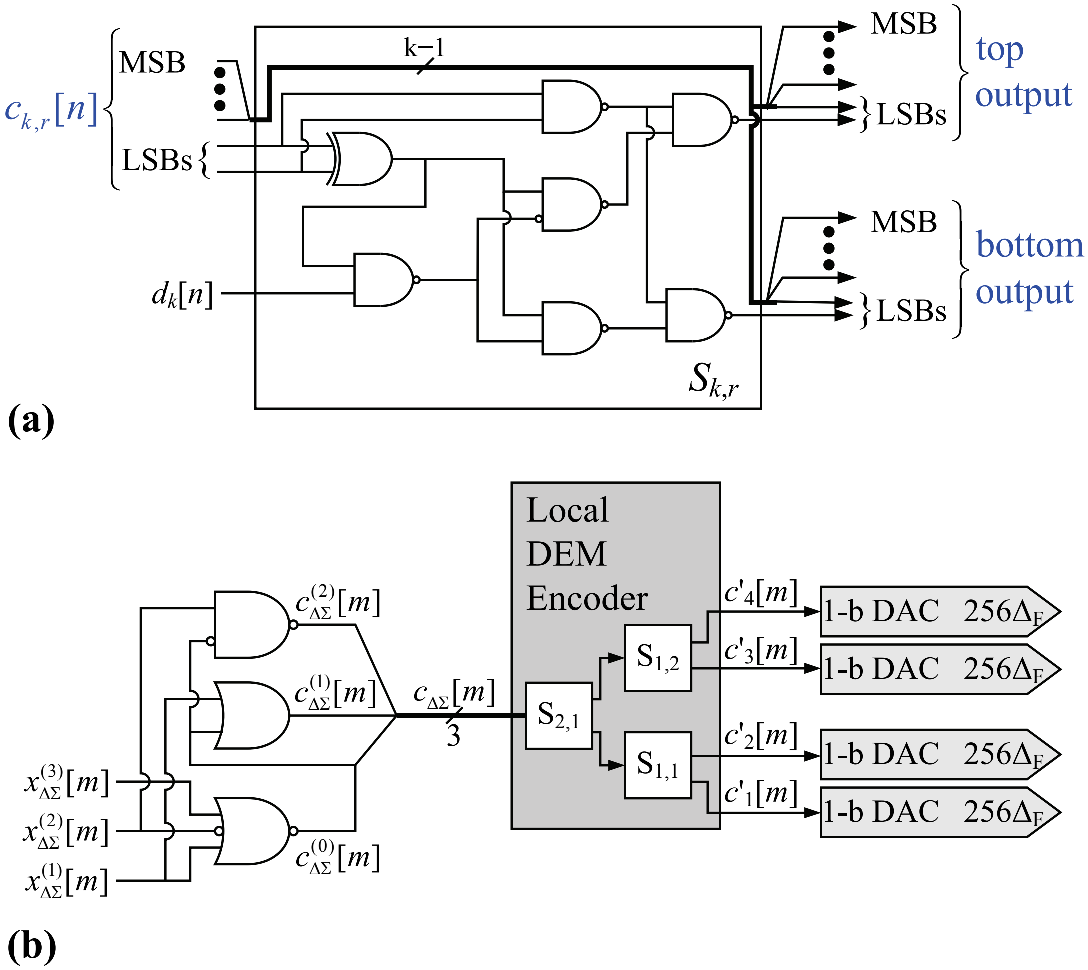


Figure 33: Implementation of a) non-segmenting switching block in the local DEM encoder and b) the local DEM encoder

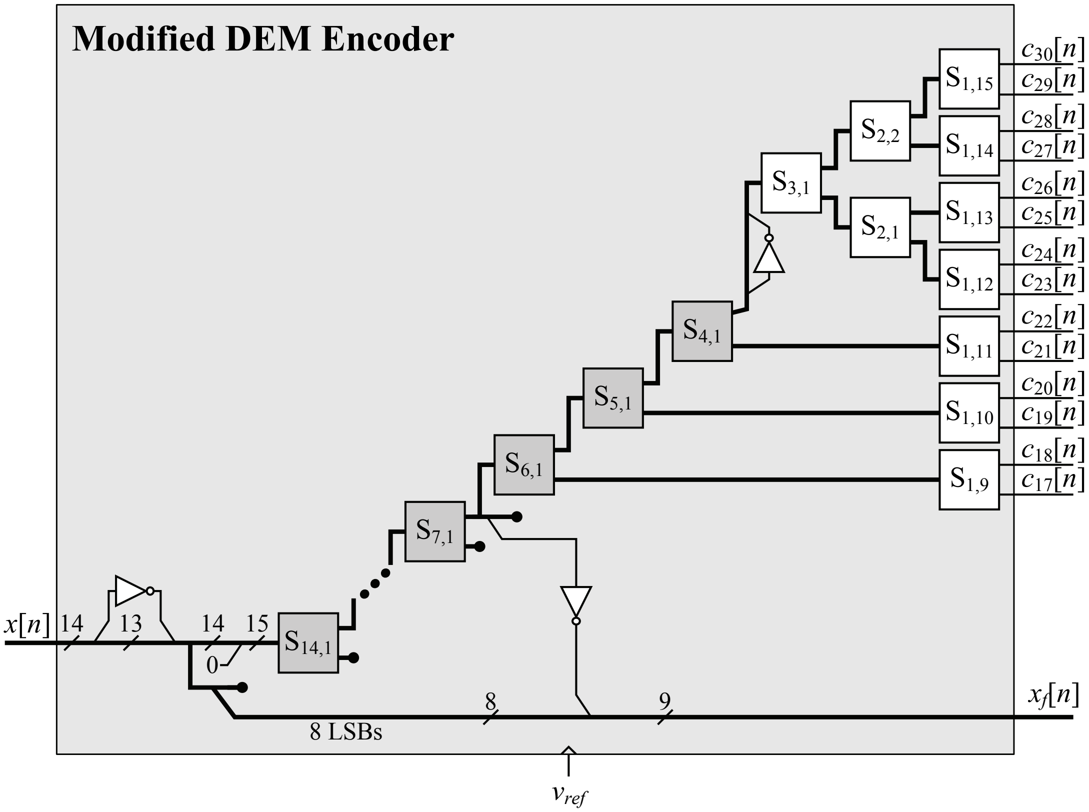


Figure 34: The adder-free modified DEM encoder for an adder-free MRQ-DEM encoder implementation

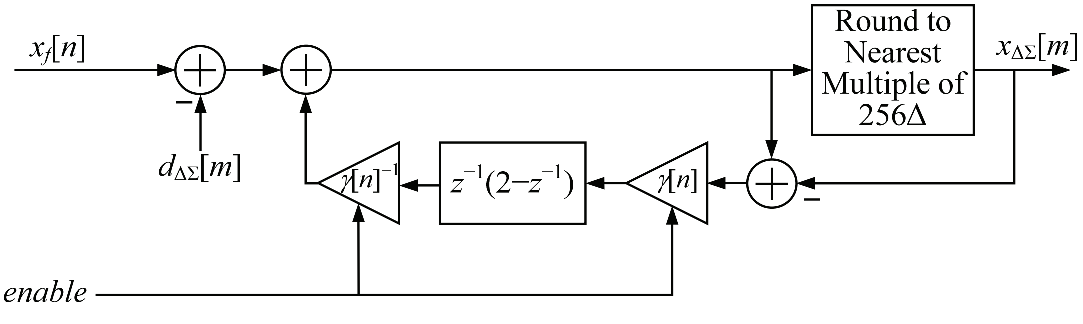


Figure 35: Functional diagram of the dithered second-order digital ΔΣ modulator with period distortion compensation. The *enable* signal is active once per divider period in correspondence to a stretched ΔΣ clock period. When active during the nth divider period, *enable* modifies the gain of the two elements it drives from a value of 1 to the value marked in their symbols

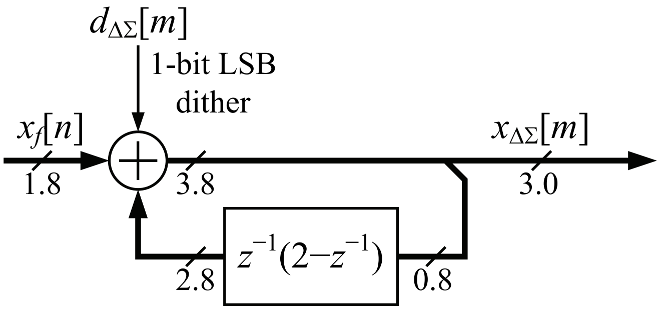


Figure 36: Signal processing diagram of the MRQ-DEM ΔΣ modulator as implemented in the synthesizer prototype

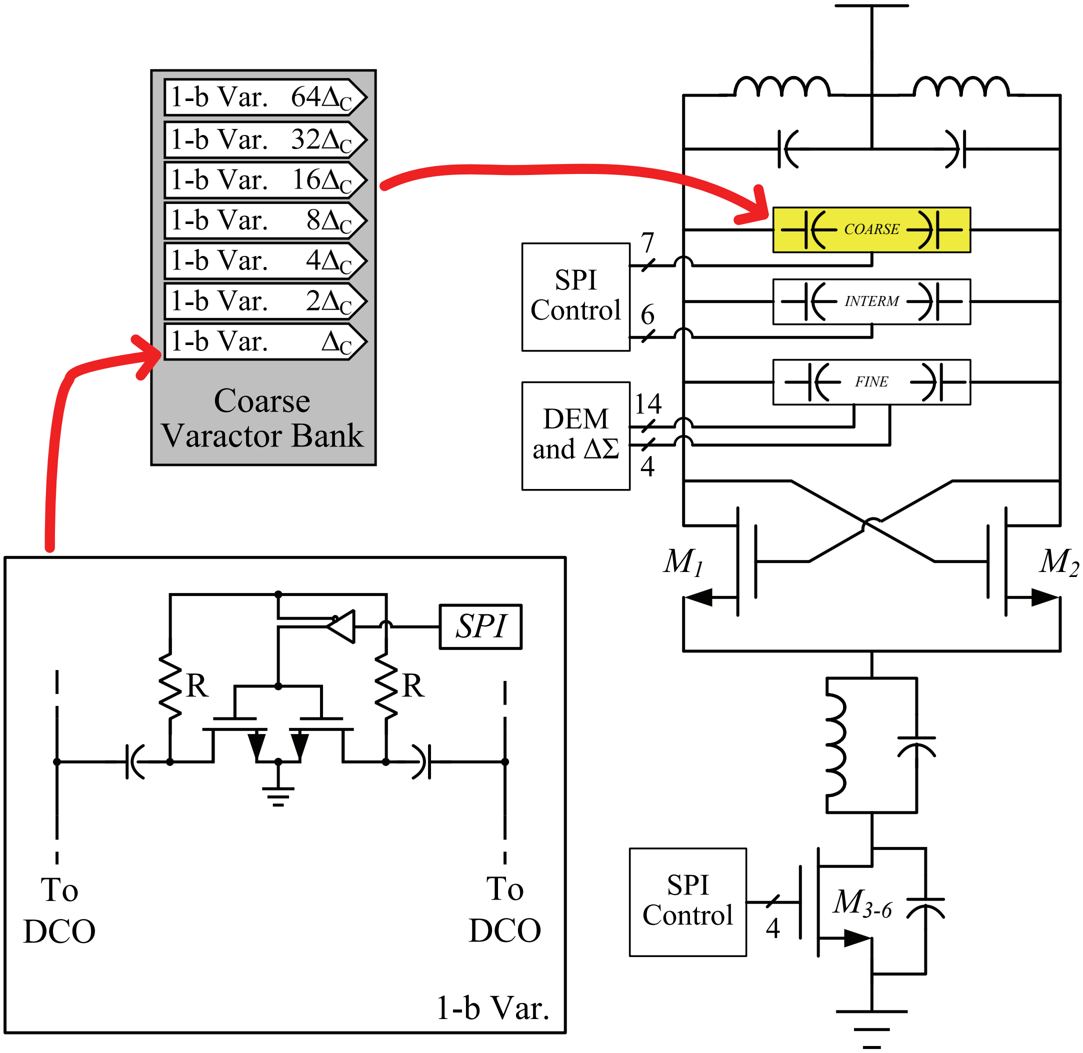


Figure 37: Digitally controlled oscillator – coarse tuning bank

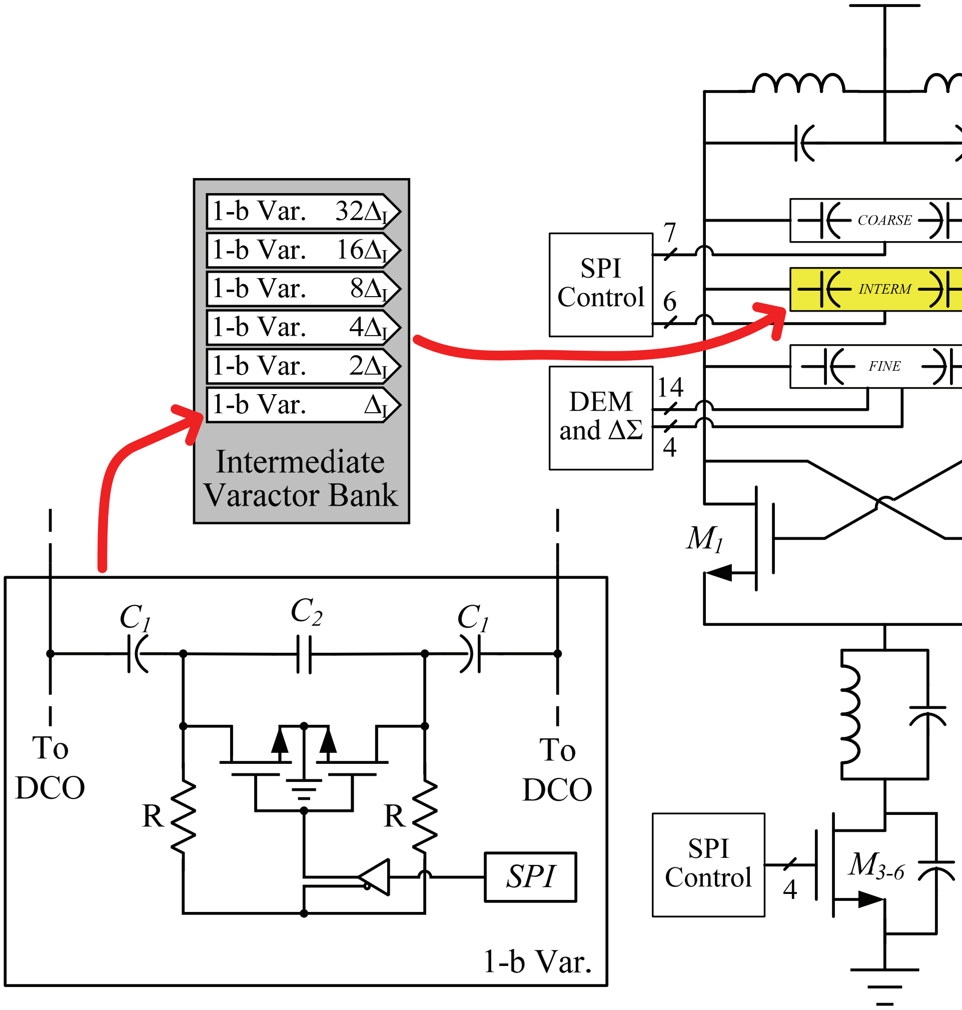


Figure 38: Digitally controlled oscillator – intermediate tuning bank

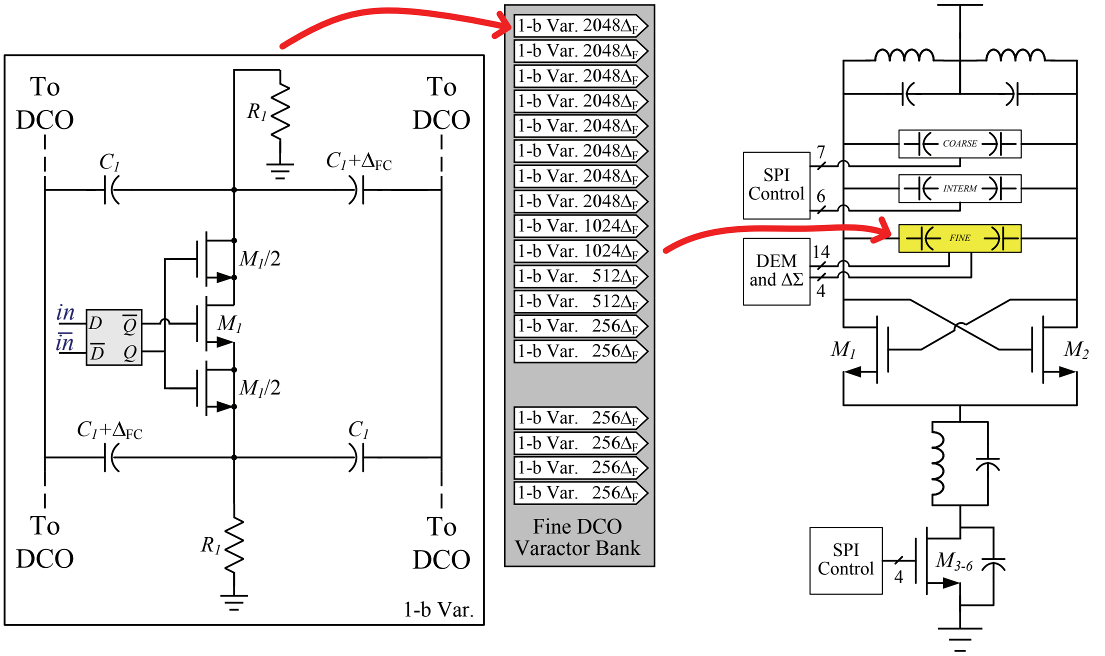


Figure 39: Digitally controlled oscillator – fine tuning bank

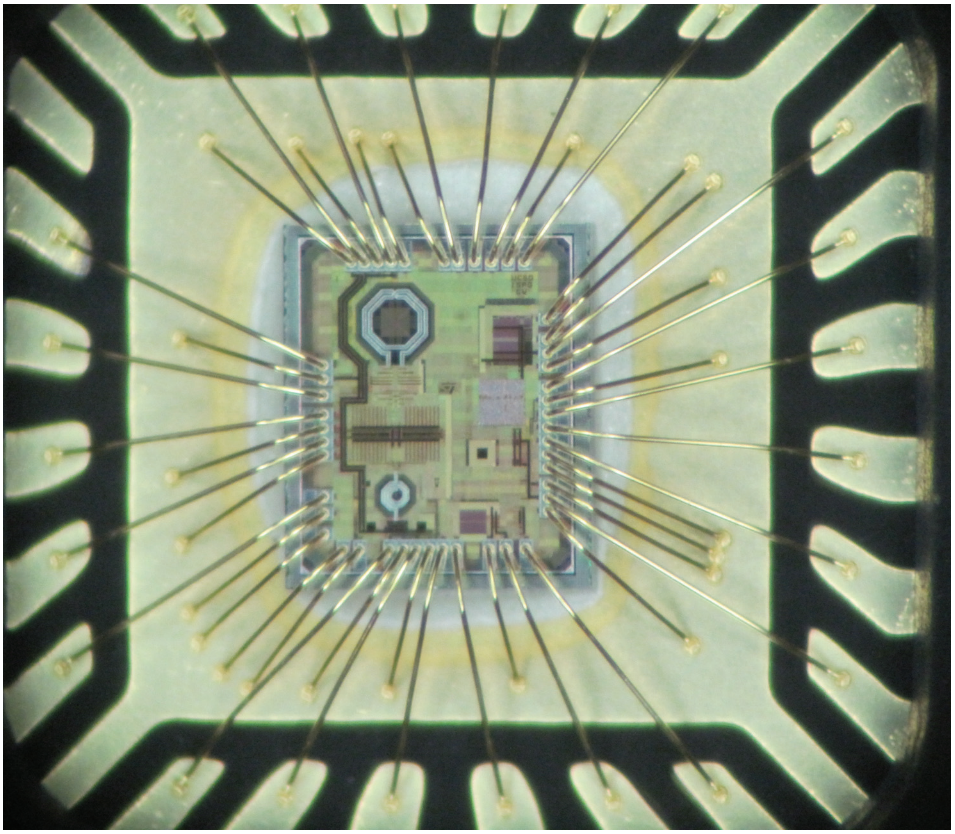
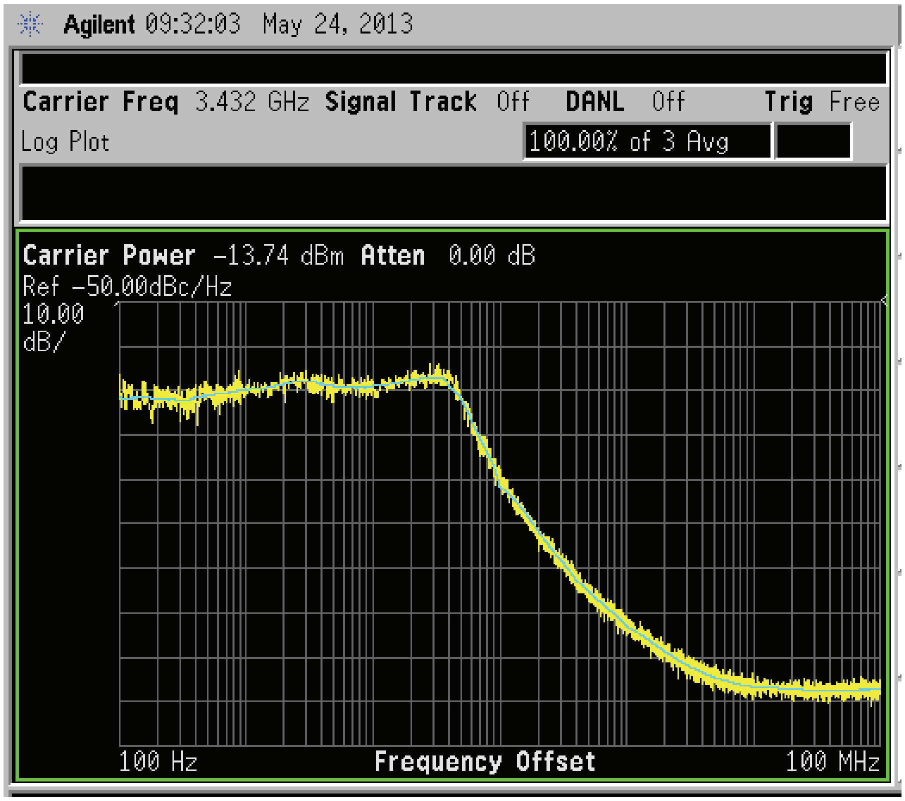


Figure 40: Packaged die photograph of the synthesizer prototype



Agilent 09:32:03 May 24, 2013

Carrier Freq 3.432 GHz Signal Track Off DANL Off Trig Free

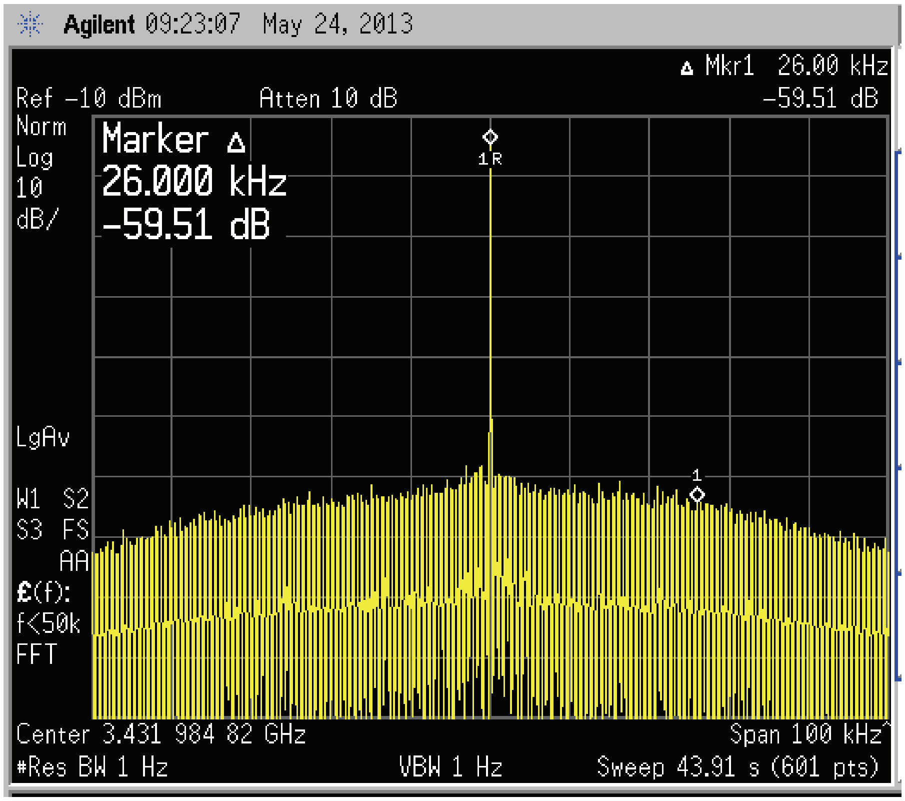
Log Plot 100.00% of 3 Avg

Carrier Power -13.74 dBm Atten 0.00 dB

Ref -50.00dBc/Hz

10.00 dB/

Figure 41: Representative phase noise plot of the synthesizer prototype for a 40 kHz bandwidth



Agilent 09:23:07 May 24, 2013

△ Mkr1 26.00 kHz
-59.51 dB

Ref -10 dBm Atten 10 dB

Norm
Log
10
dB/

Marker △
26.000 kHz
-59.51 dB

LgAv

W1 S2
S3 FS
AA
£(f):
f<50k
FFT

Center 3.431 984 82 GHz Span 100 kHz

#Res BW 1 Hz VBW 1 Hz Sweep 43.91 s (601 pts)

Figure 42: Best spurious tone performance achieved by the prototype. The fractional spur, expected to be at 26 kHz from the carrier, has a power that is below the spectrum analyzer noise floor. The reason for the spurious tone magnitude dependence on the fractional divisor modulus is currently being investigated

TABLES

Table 2: Parameters and evaluation settings of the prototype FDC-PLL

Design Parameters and Evaluation Settings	Value
f_{ref}	26 MHz
f_{PLL}	3.1 – 4.0 GHz
N	123 – 154
α	[-0.5, 0.5]
K_{DCO}	34 kHz
V_{DD}	1.0 V / 1.2 V
Δ	40 mV
ADC Input Range	0.5-0.7 V
C	1.25 pF
I_{CP}	180 μA
I_{OC}	$-I_{CP}$
T_{OC}	2 ns
T_{DZ}	1 ns
K_P	$768 \cdot 2^{-7}$
K_I	$768 \cdot 2^{-17}$
$\lambda_0, \lambda_1, \lambda_2, \lambda_3$	$768 \cdot (2^{-3}, 2^{-3}, 2^{-5}, 2^{-5})$
Loop-filter Word Width	32 bits
DCO Input Word Width	14 bits
DCO's ΔΣ Modulator Input Word Width	8 bits
DCO's ΔΣ Modulator Update Rate	$f_{PLL}/4, f_{PLL}/8, f_{PLL}/16$

Table 3: Measured phase noise performance of the synthesizer prototype

Board	PN @ 1 kHz	PN @ 10 kHz	PN @ 100 kHz	PN @ 1 MHz	PN @ 3 MHz
1	-75 dBc/Hz	-74 dBc/Hz	-86 dBc/Hz	-120 dBc/Hz	-125 dBc/Hz
2	-75 dBc/Hz	-72 dBc/Hz	-85 dBc/Hz	-120 dBc/Hz	-125 dBc/Hz
3	-75 dBc/Hz	-72 dBc/Hz	-86 dBc/Hz	-120 dBc/Hz	-125 dBc/Hz
4	-75 dBc/Hz	-74 dBc/Hz	-93 dBc/Hz	-122 dBc/Hz	-126 dBc/Hz
5	-74 dBc/Hz	-72 dBc/Hz	-86 dBc/Hz	-119 dBc/Hz	-125 dBc/Hz

Table 4: Measured spurious tone performance of the synthesizer prototype

Board	Spur Power α = 0.000125	Spur Power α = 0.00025	Spur Power α = 0.00050	Spur Power α = 0.00100	Spur Power α = 0.00150
1	-38 dBc	-38 dBc	-41 dBc	-41 dBc	-39 dBc
2	-37 dBc	-37 dBc	-38 dBc	-40 dBc	-36 dBc
3	-41 dBc	-41 dBc	-45 dBc	-47 dBc	-46 dBc
4	-46 dBc	-49 dBc	-43 dBc	-61 dBc	-60 dBc
5	-39 dBc	-39 dBc	-36 dBc	-41 dBc	-39 dBc

References

54. C.W. Mangelsdorf, “A 400-MHz input flash converter with error correction,” *IEEE Journal of Solid-State Circuits*, vol.25, no. 1, pp. 184-191, Feb 1990.

55. C.S. Vaucher, et al. “A family of low-power truly modular programmable dividers in standard 0.35-um CMOS technology,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 7, pp. 1039-1045, July 2000.

56. J. Welz, I. Galton, E. Fogleman, “Simplified logic for first-order and second-order mismatch-shaping digital-to-analog converters,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 11, pp. 1014-1028, November 2001.

57. A. Keady and C. Lyden, “Tree structure for mismatch noise-shaping multibit DAC,” *Electron. Lett.*, vol. 33, no. 17, pp. 1431–1432, Aug. 1997.

58. H. T. Jensen and I. Galton, “A reduced-complexity mismatch-shaping DAC for delta-sigma data converters,” *Proc. IEEE Int. Symp. Circuits and Systems*, May 31–June 3 1998, pp. 504–507.

59. Hegazi, et al., “A filtering technique to lower LC oscillator phase noise” *IEEE Journal of Solid-State Circuits*, vol.36, no.12, pp.1921,1930, Dec 2001.

60. G. Taylor, I. Galton, “A Reconfigurable Mostly-Digital Delta-Sigma ADC With a Worst-Case FOM of 160 dB” *IEEE Journal of Solid-State Circuits*, vol.48, no.4, pp.983,995, April 2013.