UNIVERSITY OF CALIFORNIA SAN DIEGO

Inter-symbol Interference Mitigation in Dynamic Element Matching DACs

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

in

Electrical Engineering (Electronic Circuits and Systems)

by

Jason Remple

Committee in charge:

Professor Ian Galton, Chair
Professor Drew Hall
Professor William S. Hodgkiss
Professor Ryan Kastner
Professor Larry Milstein

2021

The dissertation of Jason Remple is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

DEDICATION

To my wife, who's patience and grace throughout my graduate studies surpassed my own.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Ian Galton. Before taking his circuits class, I had never considered pursuing a Ph.D. I admire his standards as they have helped to raise mine, and I am grateful for his patience through many, many, article revisions.

I would also like to thank Andrea Panigada, who's been my professional manager through much of my Ph.D studies. Without his support I would not have been able to build a DAC, and without his encouragement I would not have made it this far.

Chapter 1, in full, has been published in the IEEE Transactions on Circuits and Systems I: Regular Papers, volume 64, number 1, pages 14-23, January 2017, Remple, Jason; Galton, Ian. The chapter includes a correction to equation (12) and the sentence above it that are not in the published version. 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 Journal of Solid-State Circuits by Remple, Jason; Panigada, Andrea; Galton, Ian. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis of this paper.

VITA

2005	Bachelor of Science, University of California Los Angeles
2012	Master of Science, University of California San Diego
2021	Doctor of Philosophy, University of California San Diego

ABSTRACT OF THE DISSERTATION

Inter-symbol Interference Mitigation in Dynamic Element Matching DACs

by

Jason Remple

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

University of California San Diego, 2021

Professor Ian Galton, Chair

Unavoidable mismatches among the components in 1-bit DACs cause both static and dynamic errors in the 1-bit DAC output waveforms. When the 1-bit DACs are used to construct multi-bit continuous-time DACs the static and dynamic errors cause second- and higher-order distortion terms in the overall DAC output.

Chapter 1 shows that dynamic element matching (DEM) converts a portion of

the second-order distortion term, and all of the higher-order distortion terms into noise that limits the DAC's SNR. The remaining second-order distortion term is caused by a type of dynamic error called inter-symbol interference (ISI).

Chapter 2 proposes a technique called ISI scrambling which is a simple add-on to DEM that prevents the 1-bit DAC ISI errors from producing second-order distortion in the DAC output. Measurement results from a prototype 1-GS/s 14-bit DAC implemented in a 90 nm CMOS technology closely match the mathematical analysis of ISI scrambling that is presented in the chapter.

Chapter 3 proposes a technique called reduced noise DEM (RND) which achieves the same goals as DEM, and also reduces the noise in the DAC output caused by static 1-bit DAC errors.

Chapter 1

The Effects of Inter-Symbol Interference in Dynamic Element Matching DACs

Abstract— Dynamic element matching (DEM) is often applied to multi-bit DACs to avoid nonlinear distortion that would otherwise result from inevitable mismatches among nominally identical circuit elements. Unfortunately, for such a DEM DAC to fully achieve this objective its constituent 1-bit DACs must be free of inter-symbol interference (ISI), i.e., the error from each 1-bit DAC must not depend on prior samples of the DAC's input sequence. This paper provides the first quantitative general analysis of the effects of ISI on the continuous-time outputs of DEM DACs. The analysis provides some surprising insights such as the conclusion that for certain types of DEM the only nonlinear distortion caused by ISI is second-order distortion. The paper also presents a digital pre-distortion technique that cancels the second-order distortion in the DEM DAC's first Nyquist band if information about the 1-bit DAC mismatches is available.

I. Introduction

A multi-bit digital-to-analog converter (DAC) converts a digital input sequence, $x[n]$, to a continuous-time analog output waveform, $y(t)$. In most multi-bit DACs the digital input sequence is mapped to multiple 1-bit digital sequences that each drive a 1-bit DAC, and the outputs of the 1-bit DACs are summed to form $y(t)$. Ideally, each 1-

bit DAC generates a two-level analog output waveform that instantaneously switches between its two levels when its input bit value changes from 0 to 1 or 1 to 0. The mapping of $x[n]$ to the 1-bit DAC input sequences and the nominal output levels of the various 1-bit DACs are designed such that in the absence of non-ideal circuit behavior $y(t) = x[n]$ for all t within the nth clock period and each $n = 0, 1, 2, \ldots$.

In practice, non-ideal circuit behavior causes multi-bit DACs to deviate from this ideal behavior. Particularly significant types of non-ideal circuit behavior are component mismatches and nonlinear inter-symbol interference (ISI). Mismatches among nominally identical components that make up the 1-bit DACs inevitably occur during fabrication and cause signal-dependent error in the multi-bit DAC output. Additionally, practical 1-bit DACs do not transition instantaneously between their two levels, so they introduce transient errors. In many cases, a 1-bit DAC's transient error depends on one or more of its prior input bit values as well as its current input bit value. Such transient errors are said to contain ISI. Asymmetry between the rising and falling transient errors of the 1-bit DACs, which typically results from component mismatches, causes the ISI to contain nonlinear distortion [1-14].

Depending on the relative scaling of the 1-bit DACs in a multi-bit DAC, and depending on the value of $x[n]$, there can be multiple sets of 1-bit DAC input bit values that would yield the same value of $y(t)$ during the nth clock period in the absence of non-ideal circuit behavior. Dynamic element matching (DEM) is a digital technique that exploits such *redundancy* to manage the error caused by component mismatches [15-34]. In the absence of ISI, the only significant error caused by component

mismatches in such a DEM DAC is noise-like and uncorrelated with $x[n]$, whereas in a multi-bit DAC without DEM it would be nonlinear distortion. DEM can be implemented such that the noise-like error is spectrally shaped for use in oversampling DEM DACs or scrambled for use in Nyquist-rate DEM DACs.

Unfortunately, DEM does not completely eliminate nonlinear distortion from ISI [35]. Nevertheless, contrary to conventional wisdom, this paper shows that DEM does have a significant effect on the ISI error, and if the DEM satisfies certain properties the effect is beneficial.

The paper analyses the effects of ISI on the continuous-time outputs of DEM DACs. In contrast, previously published work analyzes the effects of ISI on DEM DACs by modelling the ISI as a discrete-time, input-referred error sequence [35, 36-37]. Such discrete-time models are useful for oversampling applications in which the DEM DAC output is integrated and sampled, e.g., as in continuous-time ΔΣ ADCs. The sampling operation aliases the ISI into a single Nyquist band, which makes it possible to accurately model the ISI as a discrete-time, input-referred sequence. However, in applications where the DAC output is not sampled such aliasing does not occur, so it is not possible to model the ISI accurately as a discrete-time sequence. Moreover, in most such applications only a single Nyquist band is of interest, and it is impossible to accurately analyze the effects of ISI on a single Nyquist band with a discrete-time, input-referred ISI model. In Nyquist-rate DEM DAC applications, such insight is particularly critical.

The results presented in this paper address this problem. The analysis is

general in that it applies to all types of DEM, and it is backward compatible in the sense that the results reduce to prior continuous-time DEM DAC results in the absence of ISI. It shows that the DEM DAC error resulting from ISI always consists of a linear term, a second-order distortion term, and a term that is noise-like and may or may not also contain nonlinear distortion depending on the type of DEM used. As the results are based on a continuous-time analysis they quantify the effects of ISI on all Nyquist bands of the DEM DAC and they are not subject to the aliasing problem described above that is inherent to prior discrete-time analyses.[1] Additionally, the continuous-time analysis enables some new observations about previously proposed techniques to mitigate ISI, and enables a new method of digitally cancelling nonlinear distortion from ISI in a mismatch-scrambling DEM DAC's first Nyquist band if information about the 1-bit DAC mismatches is known.

The remainder of the paper consists of five main sections. Section II briefly reviews the general form of a continuous-time multi-bit DAC, and the remaining sections present the new results of the paper. Section III defines the general form of the continuous-time error introduced by each of the 1-bit DACs. Section IV presents the ISI analysis with support from two appendices, and Section V applies the results to make observations about previously published ISI reduction techniques in DEM DACs. Section VI presents the new nonlinearity distortion cancellation technique using the results from Section V.

[1] The kth Nyquist band for $k = 1, 2, \ldots$, is defined as the set of frequencies that satisfy $(\pi(k-1))f_s < |\omega| < (\pi(k-1)+\pi)f_s$.

II. NYQUIST-RATE DAC OVERVIEW

The DAC input, $x[n]$, is an f_s-rate sequence of digital codewords, each of which is interpreted by design convention to have a numerical value in the range

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

where M is the number of DAC input levels minus one (so M is a positive integer), Δ is the DAC's minimum step-size, and f_s is the clock rate of the DAC. The continuous-time analog output of an ideal non-return-to-zero (NRZ) DAC is given by

$$y(t) = x\left[n_t\right] \tag{2}$$

where n_t is a continuous-time function of t given by

$$n_t = \lfloor f_s t \rfloor, \tag{3}$$

and $\lfloor f_s t \rfloor$ denotes the largest integer less than or equal to $f_s t$. Examples of the n_t and output waveforms from such a DAC are shown in Fig. 1.

The general form of a DAC that implements this behavior aside from error caused by non-ideal circuit behavior is shown in Fig. 2. It consists of an all-digital block called an encoder and N 1-bit DACs. The 1-bit DAC outputs, $y_i(t)$, are summed to form the overall DAC output, $y(t)$, i.e.,

$$y(t) = \sum_{i=1}^{N} y_i(t)\ . \tag{4}$$

In general, the output of the ith 1-bit DAC has the form

$$y_i(t) = x_i\left[n_t\right] K_i \Delta + e_i(t) \tag{5}$$

where

$$x_i[n_t] = \underbrace{c_i[n_t] - \frac{1}{2}}_{=\pm\frac{1}{2}}, \tag{6}$$

$c_i[n]$ is the 1-bit DAC input sequence (which is either 0 or 1 for each n), K_i is a constant called the 1-bit DAC's weight, and $e_i(t)$ represents any deviation from ideal NRZ 1-bit DAC operation, such as noise and distortion from non-ideal analog circuit behavior. By design, each K_i is an integer, $K_1 = 1$, and $K_i \geq K_{i-1}$ for $i = 2, 3, \ldots, N$. Examples of DACs with different choices of 1-bit DAC weights are shown in Fig. 3.

For each such DAC, the encoder sets its output bits, $c_i[n_t]$, such that

$$\sum_{i=1}^{N} x_i[n] K_i \Delta = x[n] \ . \tag{7}$$

Substituting (5) into (4) and (7) with (3) into the result shows that in the absence of deviations from ideal NRZ behavior (i.e., if the $e_i(t)$ terms in (5) were absent) the DAC would have the ideal behavior given by (2).

III. Mismatch, Transient Error, and ISI Model

Particularly significant types of non-ideal 1-bit DAC behaviors are mismatches from fabrication errors among nominally identical components that make up the 1-bit DACs, non-instantaneous rise and fall transitions, and inter-symbol interference (ISI).[2] In current-steering 1-bit DACs, these non-idealities can be realistically modelled as

[2] In this context, ISI occurs if $e_i(t)$ depends on not only the current value of $c_i[n]$ but also one or more past values of $c_i[n]$.

$$e_i(t) = \begin{cases} e_{11i}(t), & \text{if } c_i[n_t - 1] = 1,\ c_i[n_t] = 1, \\ e_{01i}(t), & \text{if } c_i[n_t - 1] = 0,\ c_i[n_t] = 1, \\ e_{00i}(t), & \text{if } c_i[n_t - 1] = 0,\ c_i[n_t] = 0, \\ e_{10i}(t), & \text{if } c_i[n_t - 1] = 1,\ c_i[n_t] = 0, \end{cases} \tag{8}$$

where $e_{11i}(t)$, $e_{01i}(t)$, $e_{00i}(t)$, and $e_{10i}(t)$, are periodic waveforms with period $T_s = 1/f_s$ that represent the error over each clock period corresponding to the four different possibilities of the current and previous 1-bit DAC input bit values. During any given T_s clock period, the 1-bit DAC error, $e_i(t)$, is equal to exactly one of the $e_{11i}(t)$, $e_{01i}(t)$, $e_{00i}(t)$, and $e_{10i}(t)$ waveforms.

Example 1-bit DAC waveforms are shown in Fig. 4. The arrows show which of the $e_{11i}(t)$, $e_{01i}(t)$, $e_{00i}(t)$, and $e_{10i}(t)$ periods are active, i.e., equal to $e_i(t)$. Each active period of the $e_{10i}(t)$ or $e_{01i}(t)$ waveform represents the transient error associated with an input bit transition from 1 to 0 or 0 to 1, respectively. Each active period of the $e_{00i}(t)$ and $e_{11i}(t)$ waveforms represents errors such as clock feedthrough that occur when the 1-bit DAC's current input bit is unchanged from the previous T_s clock period. The values of $e_{00i}(t)$ and $e_{10i}(t)$ at the end of each T_s clock period represent the final settling error of the 1-bit DAC when its input bit is 0, and those of $e_{11i}(t)$ and $e_{01i}(t)$ represent the final settling error of the 1-bit DAC when its input bit is 1.

Equation (8) models ISI because $e_i(t)$ depends on the 1-bit DAC's current and previous input bit values. In the special case where

$$e_{00i}(t) = e_{10i}(t) \quad \text{and} \quad e_{11i}(t) = e_{01i}(t) \tag{9}$$

it follows from (8) that $e_i(t)$ depends only on the 1-bit DAC's current input bit value, in which case ISI is avoided. This condition can be implemented in practice by resetting

the state of each 1-bit DAC at the end of every T_s clock period. For example, in a so-called *return-to-zero* (RZ) 1-bit DAC, the 1-bit DAC output is set to zero (or some other signal-independent level) for a portion of each T_s clock period. Example RZ 1-bit DAC waveforms are shown in Fig. 5. It can be verified visually from Fig. 5 that (9) holds in this case.

The model described above, and hence the results derived in the remainder of the paper, assume that each 1-bit DAC output only depends on the 1-bit DAC's input bit values during the current and immediately prior clock period. In all applications known to the authors, this is an accurate assumption. However, if necessary, the model can easily be extended. For example, if the 1-bit DAC output were to depend on the input bit values during the current and prior two clock periods, then $e_i(t)$ could be decomposed into 2^3=8 periodic waveforms, each with period T_s. The results presented in the remainder of the paper could be extended to such a model if necessary. Although this would complicate the equations, it would not require any new ideas or techniques.

IV. Effects on DEM DACs

Depending on the choice of 1-bit DAC weights, there can be multiple sets of encoder output bit values that satisfy (7) for a given DAC input value. Such DACs are said to have *redundancy*. For example, in the unity-weighted 9-level DAC shown in Fig. 3(b), $K_i = 1$ for $i = 1, 2, \ldots, 8$, so (7) is satisfied when $x[n] = -3\Delta$ provided exactly one of the encoder outputs is one and the rest are zero. Hence, in this case there are 8 different (redundant) sets of encoder output values that would yield the same DAC output in the absence of non-ideal circuit behavior. However, given that the 1-bit DAC

errors, $e_i(t)$, vary from one 1-bit DAC to another, each of these 8 choices has a different effect on the overall DAC error in general.

DEM DACs take advantage of redundancy to manipulate the usage pattern of the 1-bit DACs to impart desirable properties to the overall DAC error arising from the 1-bit DAC errors. A DEM DAC has the general form shown in Fig. 2 except the encoder is a *DEM encoder*. The DEM encoder selects its output bits according to a deterministic or pseudo-random algorithm that exploits the above-mentioned redundancy while still satisfying (7).

As explained in [22] the output bit sequences of a DEM encoder can be written as

$$c_i[n] = \frac{1}{\Delta}\left(m_i x[n] + \lambda_i[n]\right) + \frac{1}{2} \tag{10}$$

for $i = 1, 2, \ldots, N$, where each m_i is a constant, each $\lambda_i[n]$ is a sequence, and

$$\sum_{i=1}^{N} K_i m_i = 1 \quad \text{and} \quad \sum_{i=1}^{N} K_i \lambda_i[n] = 0 \ . \tag{11}$$

In Nyquist-rate DACs the DEM encoder usually is designed to ensure that the $\lambda_i[n]$ sequences well approximate zero-mean, white, random sequences that are uncorrelated with $x[n]$, i.e.,

$$\left.\begin{array}{l} \mathrm{E}\{\lambda_i[n]\} = 0 \\ \mathrm{E}\{\lambda_i[n]\lambda_i[m]\} = 0 \text{ for } m \neq n \end{array}\right\} \text{ regardless of } x[n] \tag{12}$$

where $\mathrm{E}\{u\}$ denotes the expected value of u.

As shown in Appendix A, the output of a DEM DAC can be written as

$$y(t) = \alpha(t) x\left[n_t\right] + \beta(t) + e_{DAC}(t) \tag{13}$$

where

$$\alpha(t) = 1 + \frac{1}{2\Delta}\sum_{i=1}^{N} m_i \left[e_{11i}(t) - e_{00i}(t) + e_{01i}(t) - e_{10i}(t) \right] \tag{14}$$

$$\beta(t) = \frac{1}{4}\sum_{i=1}^{N} \left[e_{11i}(t) + e_{00i}(t) + e_{01i}(t) + e_{10i}(t) \right] \tag{15}$$

and $e_{DAC}(t)$ is an error waveform that depends on the DAC input sequence, $x[n]$, the $e_i(t)$ waveforms, and the $\lambda_i[n]$ sequences. As $e_{11i}(t)$, $e_{01i}(t)$, $e_{00i}(t)$, and $e_{10i}(t)$ are T_s-periodic, so are $\alpha(t)$ and $\beta(t)$.

The desired signal component of the DAC output is the $\alpha(t)x[n_t]$ term in (13). It follows from the result shown in Appendix B that the continuous-time Fourier transform of the $\alpha(t)x[n_t]$ term in (13) is

$$\Im_{CT}\left\{ \alpha(t)x\left[n_t\right] \right\} = A_p\left(j\omega \right) X\left(e^{j\omega T_s} \right) \tag{16}$$

where $A_p(j\omega)$ is the continuous-time Fourier transform of $\alpha(t)$ set to zero outside the interval $0 \le t \le T_s$, i.e., the continuous-time Fourier transform of

$$\alpha_p(t) = \begin{cases} \alpha(t) & \text{if } 0 \le t \le T_s, \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

and $X(e^{j\omega T_S})$ is the discrete-time Fourier transform of $x[n]$. This can be interpreted as the result of applying a linear time-invariant (LTI) filter with frequency response $A_p(j\omega)$ to an ideal continuous-time version of $x[n]$. In the absence of non-ideal circuit behavior, $A_p(j\omega)$ is the frequency response of a zero-order hold operation, i.e., a sinc frequency response. Example waveforms for this case are shown in Fig. 6. Although non-ideal circuit behavior causes $A_p(j\omega)$ to deviate somewhat from the ideal sinc frequency response, the deviation is linear and time-invariant, and it also tends to be relatively small.

Hence, the deviation rarely results in a significant performance degradation.

The $\beta(t)$ term in (13) is T_s-periodic, so it consists only of tones at multiples of f_s. As these tones do not fall within any Nyquist band of the DAC output and do not depend on the DAC input, they do not cause significant problems in most DAC applications.

Unfortunately, the $e_{DAC}(t)$ term in (13) can be problematic in applications. As shown in Appendix A, it can be written as

$$e_{DAC}(t) = e_{MM}(t) + e_{ISI\text{-}linear}(t) + e_{ISI\text{-}nonlinear}(t) + e_{ISI\text{-}noise}(t) \tag{18}$$

where $e_{MM}(t)$ is error that arises from component mismatches and non-instantaneous rise and fall transitions but not ISI, and the three other terms are different types of error that arise from ISI. Therefore, these latter terms are all zero if (9) holds for all of the 1-bit DACs. Expressions for each of the terms in (18) are derived in Appendix A and explained below. The expressions show that $e_{ISI\text{-}linear}(t)$ is linearly related to a delayed version of the DAC input, and $e_{ISI\text{-}nonlinear}(t)$ is a second-order distorted version of the DAC input. They also show that if the DEM encoder is such that (12) holds, then $e_{MM}(t)$ and $e_{ISI\text{-}noise}(t)$ are noise-like terms that are uncorrelated with the DAC input.

The $e_{MM}(t)$ expression is

$$e_{MM}(t) = \sum_{i=1}^{N} \varepsilon_i(t)\lambda_i\left[n_t\right] \tag{19}$$

where each $\varepsilon_i(t)$ is a T_s-periodic waveform given by

$$\varepsilon_i(t) = \frac{e_{11i}(t) - e_{00i}(t) + e_{01i}(t) - e_{10i}(t)}{2\Delta} \quad . \tag{20}$$

Hence, if (12) is satisfied $e_{MM}(t)$ is a noise-like waveform that is zero-mean and

uncorrelated with the DAC input. Given that each $\lambda_i[n]$ sequence is white, the spectral shape of the noise is a weighted sum of the magnitude squared of the continuous-time Fourier transforms of the $\varepsilon_i(t)$ waveforms set to zero outside the time interval $0 \le t \le T_s$ (this follows from the result presented in Appendix B). If the 1-bit DACs do not introduce ISI, i.e., if each 1-bit DAC satisfies (9), then $e_{MM}(t)$ is equivalent to the $e_{DAC}(t)$ expression derived in [20].

The $e_{ISI\text{-}linear}(t)$ expression is

$$e_{ISI-linear}(t) = \underbrace{\left(\frac{1}{\Delta}\sum_{i=1}^{N} m_i \gamma_i(t)\right)}_{\triangleq \gamma(t)} x\left[n_t - 1\right] \tag{21}$$

where each $\gamma_i(t)$ is a T_s-periodic waveform given by

$$\gamma_i(t) = \frac{1}{2}\left[e_{11i}(t) - e_{00i}(t) - e_{01i}(t) + e_{10i}(t)\right]. \tag{22}$$

As $\gamma(t)$ is a linear combination of the $\gamma_i(t)$ waveforms, it too is T_s-periodic. It follows that $e_{ISI\text{-}linear}(t)$ has the same general form as the DEM DAC's desired signal except for a one-period delay and a factor of $\gamma(t)$ instead of $\alpha(t)$. Thus, similar to the DEM DAC's desired signal component, $e_{ISI\text{-}linear}(t)$ is equivalent to the result of applying an LTI filter to an ideal continuous-time version of $x[n-1]$. Although it is an error component, it represents linear error and also the magnitude of $\gamma(t)$ tends to be much smaller than $\alpha(t)$ in practice, so $e_{ISI\text{-}linear}(t)$ is rarely problematic in applications.

The $e_{ISI\text{-}nonlinear}(t)$ expression is

$$e_{ISI\text{-}nonlinear}(t) = \underbrace{\left(\frac{1}{\Delta^2} \sum_{i=1}^{N} m_i^{\,2} \eta_i(t) \right)}_{\triangleq \eta(t)} x[n_t]\, x[n_t - 1] \tag{23}$$

where each $\eta_i(t)$ is a T_s-periodic waveform given by

$$\eta_i(t) = e_{11i}(t) + e_{00i}(t) - e_{01i}(t) - e_{10i}(t)\,. \tag{24}$$

As $\eta(t)$ is a linear combination of the $\eta_i(t)$ waveforms, it too is T_s-periodic. It follows that $e_{ISI\text{-}nonlinear}(t)$ is equivalent to the result of applying an LTI filter to an ideal continuous-time version of $x[n]x[n-1]$. Unfortunately, this is a nonlinear function of the DAC input. As illustrated in Fig. 7, $x[n]x[n-1]$ is proportional to $(x[n] + x[n-1])^2 - (x[n] - x[n-1])^2$ so it can be viewed as the combination of two 2-tap FIR filtered versions of $x[n]$ each passed through a memoryless square law nonlinearity. Consequently, $e_{ISI\text{-}nonlinear}(t)$ is pure second-order distortion that is not addressed by DEM. It is present regardless of whether DEM is used.

The $e_{ISI\text{-}noise}(t)$ expression is

$$\begin{aligned} e_{ISI\text{-}noise}(t) = \sum_{i=1}^{N} \Bigg[\Big(& \lambda_i[n_t]\lambda_i[n_t - 1] + m_i x[n_t]\lambda_i[n_t - 1] + \\ & m_i x[n_t - 1]\lambda_i[n_t] \Big) \frac{\eta_i(t)}{\Delta^2} + \lambda_i[n_t - 1] \frac{\gamma_i(t)}{\Delta} \Bigg]. \end{aligned} \tag{25}$$

Each term in the expression is proportional to either $\lambda_i[n_t]$, $\lambda_i[n_t-1]$, or $\lambda_i[n_t]\lambda_i[n_t-1]$, so if (12) holds then $e_{ISI\text{-}noise}(t)$ is a noise-like waveform that is zero-mean and uncorrelated with $x[n]$ similar to $e_{MM}(t)$. Consequently, the $e_{ISI\text{-}noise}(t)$ term increases the noise power of the DEM DAC output relative to cases in which ISI is avoided, but it does not introduce harmonic distortion provided (12) is satisfied. However, DEM DACs designed to

spectrally shape the error arising from component mismatches do not satisfy (12). In such cases the term in $e_{ISI\text{-}noise}(t)$ proportional to $\lambda_i[n_t]\lambda_i[n_t-1]$ can sometimes introduce harmonic distortion via a mechanism similar to that explained in [38].

Computer simulation results that demonstrate the findings presented above for DACs with NRZ current-steering 1-bit DACs are shown in Figures 8 and 9. The figures show output power spectra of the 14-bit highly-segmented architecture presented in [20] with a –1 dBFS sinusoidal input signal of frequency $f_{in} = (0.091)f_s$. The simulated 1-bit DAC error waveforms, $e_{11i}(t)$, $e_{01i}(t)$, $e_{00i}(t)$, and $e_{10i}(t)$, model static current mismatches and first-order (exponential) current transients. The MSB-element current mismatches and transient time-constants were chosen randomly from Gaussian distributions with standard deviations of 0.3% and 5%, respectively. Figure 8 shows the DAC's output power spectrum with DEM disabled, i.e., with the encoder configured to be deterministic and memoryless and generate output bits that satisfy (7). Significant harmonic distortion from the 1-bit DAC errors in the form of spurious tones is visible in the power spectrum. Figure 9 shows the DAC's output power spectrum with DEM enabled such that (12) is satisfied. As predicted by the findings described above, only second-order harmonic distortion is visible in the power spectrum when DEM is enabled.

V. Prior DEM DAC ISI Error Mitigation Techniques

A. RZ 1-bit DACs

In principle, ISI error can be avoided altogether if a DEM DAC contains only

RZ 1-bit DACs, i.e., if (9) is satisfied by each 1-bit DAC. Unfortunately, such a DEM DAC typically has a lower desired signal power and a lower signal to noise and distortion ratio (SNDR) than a comparable DEM DAC based on NRZ 1-bit DACs. The lower signal power occurs because RZ 1-bit DACs cause the DEM DAC's desired signal component, $\alpha(t)x[n_t]$, to go to zero during the portion of each T_s clock period in which the RZ 1-bit DAC outputs go to zero. This happens because $\alpha(t)$ is a weighted sum of the $e_{11i}(t)$, $e_{01i}(t)$, $e_{00i}(t)$, and $e_{10i}(t)$ waveforms as indicated by (14). For example, with the RZ 1-bit DAC waveforms shown in Fig. 5, $\alpha(t)$ is nearly zero for almost half of every T_s clock period, so the power of the DEM DAC's desired signal component in any given Nyquist band is up to 6 dB lower than that of a comparable DEM DAC with NRZ 1-bit DACs.[3] The lower SNDR occurs because RZ 1-bit DACs and NRZ 1-bit DACs with comparable transient settling time constants have comparable transient errors. Hence, the difference in dB between the powers of the $e_{DAC}(t)$ waveforms for the two cases is smaller than the corresponding difference between the powers of the desired signal components.

These drawbacks of RZ 1-bit DACs typically are not offset by any reduction in power dissipation because RZ 1-bit DACs usually dissipate power even when the 1-bit DAC outputs go to zero. For example, current steering RZ 1-bit DACs typically implement the return-to-zero phase of each clock period by steering their output currents to dummy loads. Hence, their power consumption is not reduced during their

[3] The $A_p(j\omega)$ term in (16) for an RZ DAC that goes to zero for half of each clock period is a sinc function with half the amplitude of that of an NRZ DAC, although its roll-off is more gradual.

return-to-zero phase. Moreover, generating the extra clock phase required to implement the RZ behavior dissipates additional power.

For these reasons, NRZ 1-bit DACs are often favored in high-speed Nyquist-rate DACs. However, as quantified by the findings of this paper, this choice generally comes with ISI error.

B. Balanced Rise and Fall Transitions

It has been suggested in prior work that DAC error from ISI is avoided if each 1-bit DAC's rising transient error is precisely the opposite of its falling transient error, i.e., $e_{01i}(t) = -e_{10i}(t)$ in the notation of this paper [2]. The results presented above indicate that this is not completely true. To avoid error from ISI altogether, every $\eta_i(t)$ and $\gamma_i(t)$ waveform must be zero, and this happens if and only if (9) is satisfied by each 1-bit DAC, i.e., if and only if RZ 1-bit DACs are used. However, it follows from (23) and (25) that for $e_{ISI\text{-}nonlinear}(t)$ and most of the terms in $e_{ISI\text{-}noise}(t)$ to be zero, it is sufficient for just every $\eta_i(t)$ waveform to be zero. As can be seen from (24), this happens when

$$e_{11i}(t) - e_{01i}(t) = -\left[e_{00i}(t) - e_{10i}(t)\right] \tag{26}$$

for every 1-bit DAC. In practice, achieving (26) in every 1-bit DAC, while not eliminating error from ISI completely, eliminates its worst effects. Condition (26) reduces to the above-mentioned $e_{01i}(t) = -e_{10i}(t)$ condition for the special case where the only non-ideal 1-bit DAC behavior is from non-instantaneous output transitions.

C. ISI-Shaping

Another method of mitigating error from ISI is to combine an ISI-shaping

algorithm with DEM such that $e_{MM}(t) + e_{ISI\text{-}nonlinear}(t) + e_{ISI\text{-}noise}(t)$ is zero-mean with a power spectrum that is second-order highpass shaped [36]. In the context of the results presented above, the objective is to choose each $\lambda_i[n]$ such that the terms in $e_{ISI\text{-}noise}(t)$ which are proportional to the $\eta_i(t)$ waveforms cancel the corresponding terms in $e_{ISI\text{-}nonlinear}(t)$. Given that the $\eta_i(t)$ waveforms vary unpredictably with i, the above-mentioned cancellation must occur separately for each $\eta_i(t)$ term. It follows from (23) and (25) that for each $i = 1, 2, \ldots, N$, the factors of $\eta_i(t)$ in $e_{ISI\text{-}nonlinear}(t) + e_{ISI\text{-}noise}(t)$ can be grouped as

$$\big(m_i x[n] + \lambda_i[n]\big)\big(m_i x[n-1] + \lambda_i[n-1]\big) \tag{27}$$

Therefore, the ISI-shaping algorithm must ensure that (27) is a zero-mean sequence with a second-order highpass spectral shape for each $i = 1, 2, \ldots, N$.

Unfortunately, it follows from (27) that this can only happen if each $\lambda_i[n]$ is correlated with the DAC input. One consequence is that such ISI-shaping algorithms are not compatible with DEM encoders that implement (12), i.e., the type of DEM encoders most appropriate for high-speed Nyquist-rate DACs. Another consequence is that the power spectrum of $e_{DAC}(t)$ in DEM DACs with ISI-shaping algorithms inevitably contain harmonic distortion. This is most likely the cause of the spurious tones visible in the output spectra from the ISI-shaping DEM DAC presented in [36], although its overall performance is very impressive.

VI. ISI Nonlinear Distortion Cancellation

As shown in Section IV, ISI causes a second-order distortion term in the output

of a DEM DAC, i.e., $e_{ISI\text{-}nonlinear}(t) = \eta(t)x[n_t]x[n_t-1]$, where $\eta(t)$ is a T_s-periodic waveform that depends on the 1-bit DAC errors. In many applications, $e_{ISI\text{-}nonlinear}(t)$ is far more problematic than the DAC's other error terms. This section describes a digital technique that can suppress $e_{ISI\text{-}nonlinear}(t)$ within the DEM DAC's first Nyquist band provided $\alpha(t)$ and $\eta(t)$ are known a priori (e.g., through measurement as part of a calibration algorithm).

The idea is to pre-distort the DEM DAC input. This is done by setting the DEM DAC input sequence to be

$$x_{pd}[n] = x[n] + x_c[n] \tag{28}$$

where $x[n]$ is the original input sequence and $x_c[n]$ is a correction sequence that causes a term in the DEM DAC's output that cancels $e_{ISI\text{-}nonlinear}(t)$ over the first Nyquist band.

The $x_c[n]$ sequence is chosen such that

$$A_p(j\omega)X_c\left(e^{j\omega T_s}\right) + I_p(j\omega)X_{ISI-nonlinear}\left(e^{j\omega T_s}\right) \cong 0 \tag{29}$$

for all $|\omega| < \pi f_s$, where $X_c(e^{j\omega T_s})$ is the discrete-time Fourier transform of $x_c[n]$, $I_p(j\omega)$ is the continuous-time Fourier transform of $\eta(t)$ set to zero outside the interval $0 \le t \le T_s$, and $X_{ISI\text{-}nonlinear}(e^{j\omega T_s})$ is the discrete-time Fourier transform of $x_{pd}[n]x_{pd}[n-1]$. The first term on the left side of (29) is the part of the DEM DAC's desired signal component corresponding to $x_c[n]$, and the second term on the left side of (29) is the continuous-time Fourier transform of $e_{ISI\text{-}nonlinear}(t)$ with $x[n_t]$ replaced by $x_{pd}[n_t]$ in (23). Therefore when (29) is satisfied for all $|\omega| < \pi f_s$, $e_{ISI\text{-}nonlinear}(t)$ is approximately cancelled over the DEM DAC's first Nyquist band.

The discrete-time Fourier Transform of $x_c[n]$ is T_s-periodic whereas $A_p(j\omega)$ and $I_p(j\omega)$ are aperiodic functions that are not equal. Consequently, it is only possible to chose $x_c[n]$ such that (29) holds for one T_s-period. This is why $e_{ISI\text{-}nonlinear}(t)$ is only cancelled over the first Nyquist band. Alternatively, $x_c[n]$ could be chosen to cancel $e_{ISI\text{-}nonlinear}(t)$ over the kth Nyquist band for any particular $k = 2, 3, \ldots$, provided $A_p(j\omega)$ is non-zero over the kth Nyquist band.

For a DEM DAC with ideal 1-bit DACs, $\alpha(t) = 1$ and $\eta(t) = 0$, but 1-bit DAC errors cause $\alpha(t)$ and $\eta(t)$ to deviate from these ideals as described in Section IV. Still, in practical, well-designed DEM DACs the deviation tends to be small enough that the power of $e_{ISI\text{-}nonlinear}(t)$ is much lower than the power of the desired signal component over the first Nyquist band. For instance, the simulation results shown in Fig. 9 for the example DEM DAC indicate that the power of $e_{ISI\text{-}nonlinear}(t)$ in the first Nyquist band is more than 65 dB lower than that of the desired signal component. It follows that the mean squared value of the $x_c[n]$ sequence required to cancel $e_{ISI\text{-}nonlinear}(t)$ in the first Nyquist band is much smaller than the mean squared value of $x[n]$. This implies that

$$X_{ISI-nonlinear}\left(e^{j\omega T_s}\right) \cong \Im_{DT}\left\{x[n]x[n-1]\right\} \tag{30}$$

where $\Im_{DT}$ denotes the discrete-time Fourier transform.

Substituting (30) into (29) and solving for $X_c(e^{j\omega T_s})$ indicates that $x_c[n]$ could be synthesized by passing $x[n]x[n-1]$ through a digital filter with a frequency response that approximates $-I_p(j\omega)/A_p(j\omega)$ for $|\omega| < \pi f_s$. However, a practical problem arises in that a digital filter with this frequency response would be non-causal. This problem can be addressed by synthesizing a digital filter that approximates the desired frequency

response but with an integer-valued group delay, P, and either driving it with $x[n]x[n-1]$ advanced by P samples, or setting the DEM DAC input to be $x_{pd}[n] = x[n-P] + x_c[n]$.

A system-level diagram of an example implementation is shown in Fig. 10. The DEM DAC is identical to that described above. The digital filter is a 21-tap FIR filter with approximate frequency response

$$H_c\left(e^{j\omega T_s}\right) \cong -e^{-j\omega P T_s}\frac{I_p(j\omega)}{A_p(j\omega)} \quad \text{for} \quad |\omega| \le \pi f_s \tag{31}$$

where $P = 10$. The digital filter was designed in three steps. First, the Parks-McClellan algorithm was used to generate a zero-phase filter with optimum magnitude response [39][40]. Second, the zero-phase filter was delayed by P samples in order to be made causal. Finally, the causal filter was fractionally delayed [41] in order to better approximate the non-integer group delay implied by solving (31) for the randomly generated mismatches. The filter has fractional impulse response values, so its output has finer resolution than the 14-bit DEM DAC can accommodate directly. Therefore, dithered requantization was applied to quantize the filter's output to 6 bits without introducing harmonic distortion [42].

Figure 11 shows the simulated DEM DAC output spectrum for this system. The simulated DEM DAC and the corresponding $e_{11i}(t)$, $e_{01i}(t)$, $e_{00i}(t)$, and $e_{10i}(t)$ waveforms are identical to those used to generate the power spectrum shown in Fig. 9, but as can be seen from a comparison of Figures 9 and 11 the addition of $x_c[n]$ suppresses $e_{ISI\text{-}nonlinear}(t)$ in the first Nyquist band of the power spectrum of Fig. 11.

In practice $\alpha(t)$ and $\eta(t)$ are not known a priori so they would have to be measured as part of a calibration system. For example, this could be done with a foreground calibration algorithm that estimates $H_c(e^{j\omega Ts})$ at several uniformly spaced values of ω between 0 and π and then calculates the corresponding FIR filter coefficient values via an inverse fast Fourier transform of the vector of estimates. Each estimate would be obtained via an LMS algorithm [43]. For each value of ω, $x[n]$ in the system of Fig. 10 would be set to a full-scale sinusoid with a frequency of $\omega/2$, an ADC with appropriate filtering would measure the second harmonic of the sinusoid in the DAC's first Nyquist band, and the LMS algorithm would adjust the corresponding value of $H_c(e^{j\omega Ts})$ to zero the measured second harmonic.

VII. Conclusion

This paper quantifies the combined effects of 1-bit DAC mismatches, non-instantaneous rise and fall transitions, and ISI on the continuous-time outputs of DEM DACs. The results apply to most multi-bit DAC architectures and all types of DEM known to the authors, and they reduce to previously published continuous-time DEM DAC results in the absence of ISI. Previously published ISI analyses model ISI as an input-referred discrete-time sequence, so they do not quantify the effects of ISI within individual Nyquist bands. This is acceptable in applications such as ΔΣ ADCs wherein the DEM DAC outputs are sampled, but in other applications the signals of interest typically lie in individual Nyquist bands. The results of this paper address the needs of these latter applications in that they quantify the error within each Nyquist band. As demonstrated in the paper, they make it possible to devise algorithms that cancel error

in the Nyquist band of interest. The results also lead to several new insights such as the observation that for certain types of DEM the only nonlinear distortion caused by ISI is second-order distortion.

APPENDIX A

As indicated in (8), at any given time, t, the error of the ith 1-bit DAC, $e_i(t)$, is equal to one of $e_{11i}(t)$, $e_{01i}(t)$, $e_{00i}(t)$, or $e_{10i}(t)$, depending on the state of $c_i[n_t]$ and $c_i[n_t-1]$. Given that $c_i[n]$ is either 0 or 1 for each i and t, it follows that (8) can be rewritten as

$$\begin{aligned} e_i(t) = &\big(\quad c_i[n_t-1]\big)\big(\quad c_i[n_t]\big)e_{11i}(t) + \\ &\big(1-c_i[n_t-1]\big)\big(\quad c_i[n_t]\big)e_{01i}(t) + \\ &\big(1-c_i[n_t-1]\big)\big(1-c_i[n_t]\big)e_{00i}(t) + \\ &\big(\quad c_i[n_t-1]\big)\big(1-c_i[n_t]\big)e_{10i}(t)\ . \end{aligned} \tag{32}$$

Substituting (6) into (32) and the result into (5) gives an expression for the output of the ith 1-bit DAC. This expression can be arranged as

$$\begin{aligned} y_i(t) = x_i[n_t]\alpha_i(t)K_i\Delta + \beta_i(t) + x_i[n_t-1]\gamma_i(t) \\ + x_i[n_t]x_i[n_t-1]\eta_i(t) \end{aligned} \tag{33}$$

where

$$\alpha_i(t) = 1 + \frac{1}{2}\frac{e_{11i}(t) - e_{00i}(t) + e_{01i}(t) - e_{10i}(t)}{K_i\Delta} \tag{34}$$

$$\beta_i = \frac{1}{4}\left[e_{11i}(t) + e_{00i}(t) + e_{01i}(t) + e_{10i}(t)\right] \tag{35}$$

$\gamma_i(t)$ is given by (22), and $\eta_i(t)$ is given by (24). Fig. 12 shows example $\alpha_i(t)$, $\beta_i(t)$, $\gamma_i(t)$, and $\eta_i(t)$ waveforms corresponding to the 1-bit DAC example shown in Fig. 4.

Combining (6) and (10) gives

$$x_i[n_t]\Delta = m_i x[n_t] + \lambda_i[n_t] \ . \tag{36}$$

Substituting this into (33) and the result into (4) gives,

$$\begin{aligned} y(t) = \sum_{i=1}^{N} \big(m_i x[n_t] + \lambda_i[n_t]\big)\alpha_i(t)K_i + \beta_i(t) + \\ \big(m_i x[n_t - 1] + \lambda_i[n_t - 1]\big)\frac{\gamma_i(t)}{\Delta} + \\ \big(m_i x[n_t] + \lambda_i[n_t]\big)\big(m_i x[n_t - 1] + \lambda_i[n_t - 1]\big)\frac{\eta_i(t)}{\Delta^2} \ , \end{aligned} \tag{37}$$

collecting terms and applying (11) yields $y(t) = \alpha(t)x[n_t] + \beta(t) + e_{DAC}(t)$ where $\alpha(t)$ is given by (14), $\beta(t)$ is given by (15), and $e_{DAC}(t)$ is given by (18) through (25).

APPENDIX B

The result presented below in this sub-section is not new. However, the authors are not aware of a textbook or paper that presents it directly and it is necessary to explain the results of this paper, so it is derived below.

Let $a(t)$ be any T_s-periodic waveform, let $r[n]$ be any sequence, let

$$s(t) = a(t)r\big[\lfloor f_s t \rfloor\big] \tag{38}$$

and let

$$a_p(t) = \begin{cases} a(t) & \text{if } 0 \le t \le 1/f_s \ , \\ 0 & \text{otherwise.} \end{cases} \tag{39}$$

Then $s(t)$ can be written as

$$s(t) = \sum_{n=-\infty}^{\infty} a_p\left(t - \frac{n}{f_s}\right) r[n] \tag{40}$$

It follows that the continuous-time Fourier Transform of $s(t)$ is given by

$$S(j\omega) = A_p(j\omega)\sum_{n=-\infty}^{\infty} r[n]e^{-j\omega n/f_s} = A_p(j\omega)R\left(e^{j\omega/f_s}\right) \tag{41}$$

where $A_p(j\omega)$ is the continuous-time Fourier Transform of $a_p(t)$ and $R(e^{j\omega/fs})$ is the discrete-time Fourier transform of $r[n]$.

ACKNOWLEDGEMENTS

Chapter 1, in full, has been published in the IEEE Transactions on Circuits and Systems I: Regular Papers, volume 64, number 1, pages 14-23, Remple, Jason; Galton, Ian. The chapter includes a correction to equation (12) and the sentence above it that are not in the published version. 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

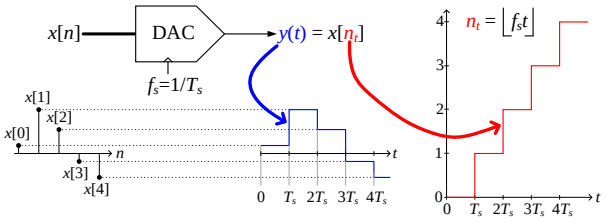

Figure 1 : Ideal NRZ DAC behavior.

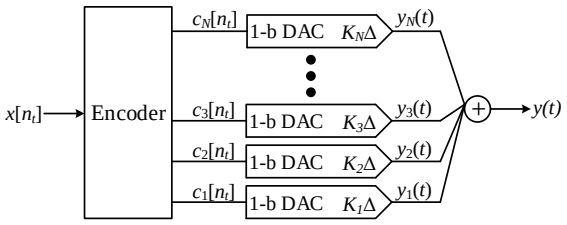

Figure 2 : General form of a multi-bit DAC.

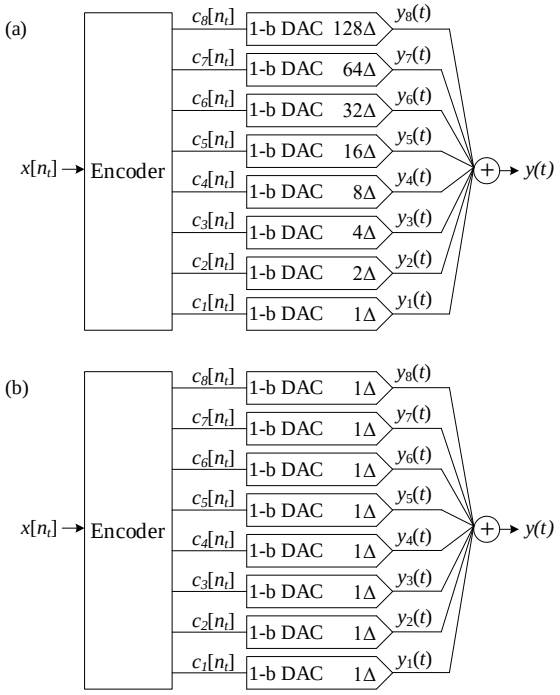

Figure 3 : Example DACs: (a) with power-of-two weighted 1-bit DACs, and (b) unity weighted 1-bit DACs.

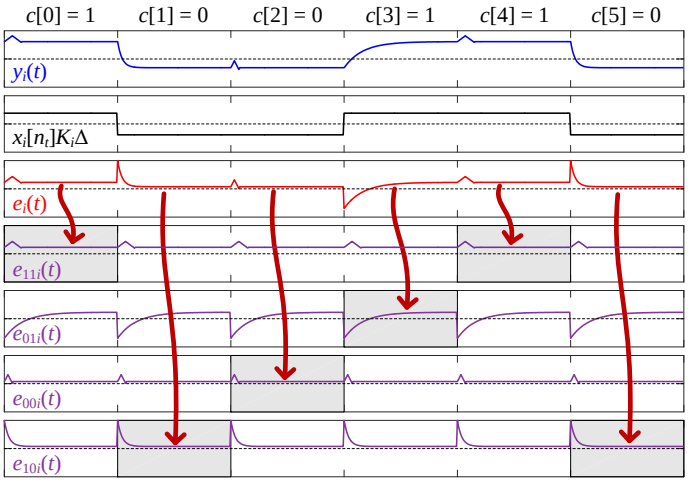

Figure 4 : Example NRZ 1-bit DAC waveforms.

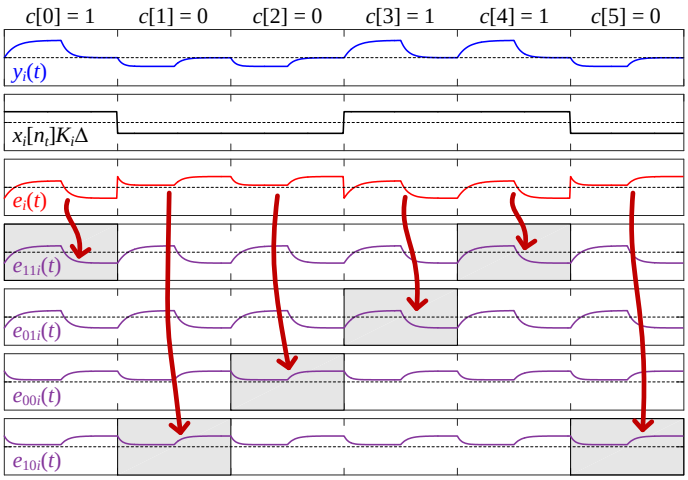

Figure 5 : Example RZ 1-bit DAC waveforms.

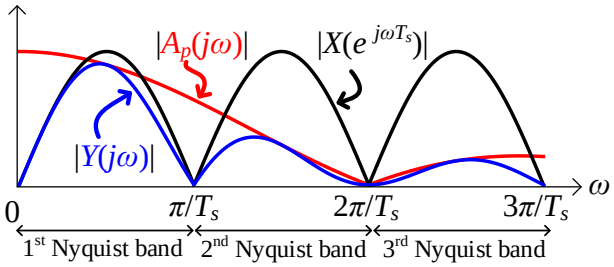

Figure 6 : Example of continuous-time Fourier transform of $\alpha(t)x[n_t]$.

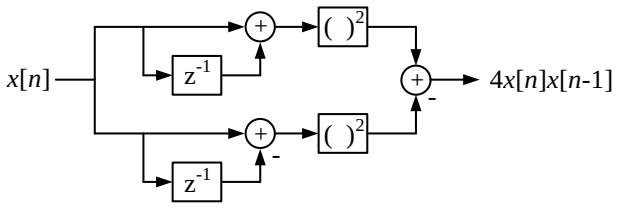

Figure 7 : Second-order nonlinearity caused by ISI.

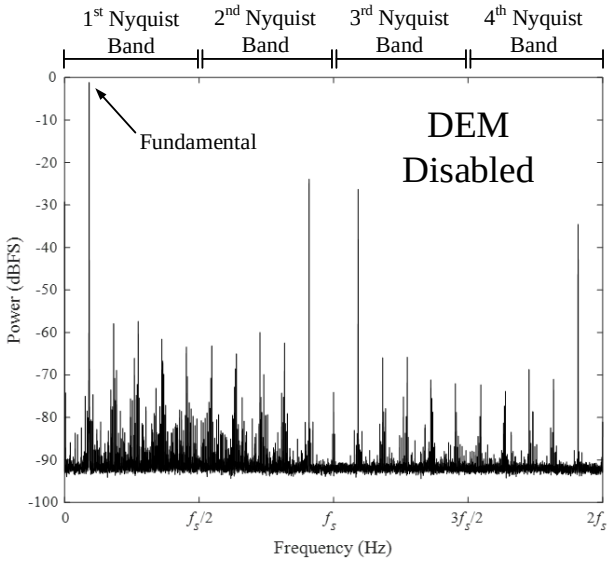

Figure 8 : Power spectrum of the simulated DAC output with DEM disabled.

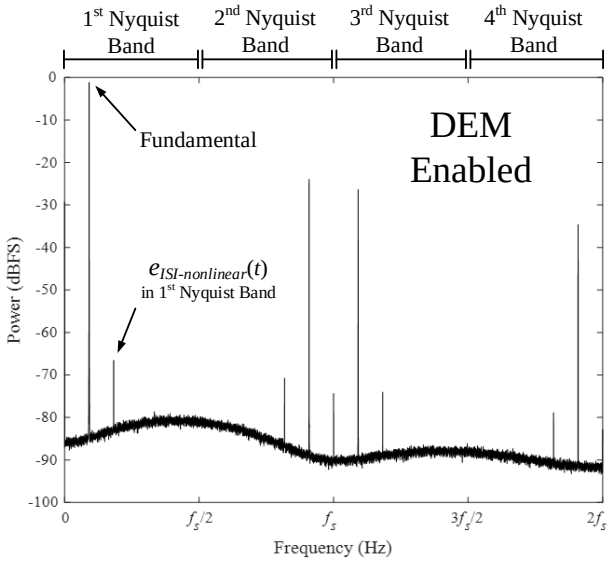

Figure 9 : Power spectrum of the simulated DAC output with DEM enabled.

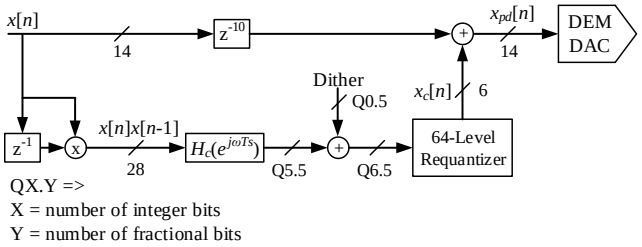

Figure 10 : ISI distortion cancellation technique.

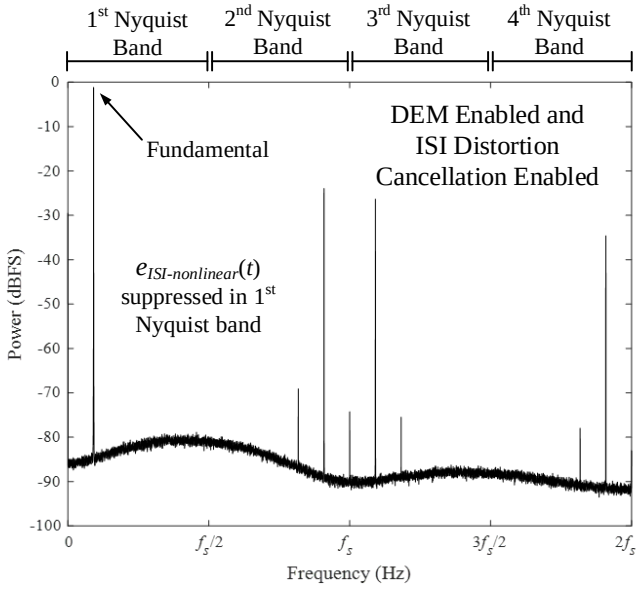

Figure 11 : DAC output with DEM enabled and ISI distortion cancellation enabled.

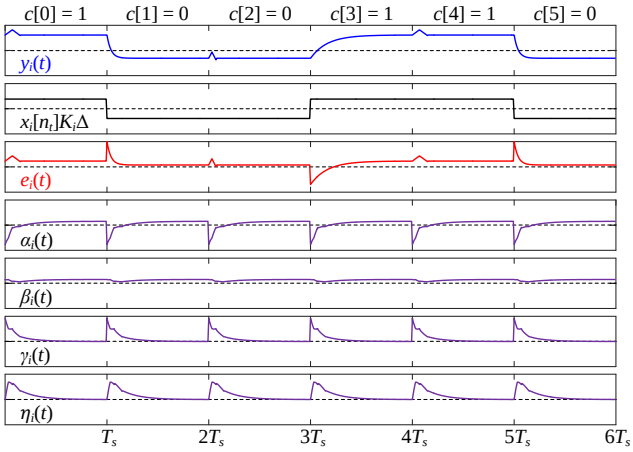

Figure 12 : $\alpha_i(t)$, $\beta_i(t)$, $\gamma_i(t)$, and $\eta_i(t)$ waveforms corresponding to the example $e_{11i}(t)$, $e_{01i}(t)$, $e_{00i}(t)$, and $e_{10i}(t)$, waveforms shown in Fig. 4.

REFERENCES

1. D. Mercer, "A 16-b D/A Converter with Increased Spurious Free Dynamic Range," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 10, pp. 1180-1185, October 1994.

2. R. Adams, K. Nguyen, and TewksBury, "Dual Return-To-Zero Pulse Encoding in a DAC Output Stage," U.S. Patent 6,061,010, May 2000.

3. S. Park, G. Kim, S.-C. Park, and W. Kim, "A Digital-to-Analog Converter Based on Differential-Quad Switching," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 10, pp. 1335-1338, October 2002.

4. K. Doris, J. Briaire, D. Leenaerts, M. Vertregt, and A. van Roermund, "A 12b 500 MS/s DAC with > 70 dB SFDR up to 120 MHz in 0.18 um CMOS", *IEEE ISSCC Digest of Technical Papers, 2005*, pp. 116-117.

5. B. Schafferer and R. Adams, "A 3V CMOS 400 mW 14b 1.4 GS/s DAC for Multi-Carrier Applications," *IEEE ISSCC Digest of Technical Papers, 2004*, pp. 360 – 362.

6. Y. Tang, J. Briaire, K. Doris, R. van Veldhoven, P. van Beek, H. Johannes, A. Hegt, and A. van Roermund, "A 14-bit 200 MS/s DAC with SFDR > 78 dBc, IM3 < -83 dBc and NSD < -163 dBm/Hz Across the Whole Nyquist Band Enabled by Dynamic-Mismatch Mapping", *IEEE Journal of Solid-State Circuits*, vol. 46, no. 6, pp. 1371-1381, June 2011.

7. A. Bugeja and B.-S. Song, "A Self-Trimming 14-b 100-MS/s CMOS DAC," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 12, pp. 1841-1852, December 2000.

8. A. Bugeja, B.-S. Song, P. Rakers, and S. Gillig, "A 14-b 100-MS/s CMOS DAC Designed for Spectral Performance," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 12, pp. 1719-1732, December 1999.

9. E. Olieman, A.-J. Annema, and B. Nauta, "An Interleaved Full Nyquist High-

Speed DAC Technique," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 3, pp. 704-713, March 2015.

10. D. Spicer and A. Rodger, "Antiglitch Digital-to-Analog Converter System," U.S. Patent 3,877,023, April 1975.

11. G. Engel, S. Kuo, and S. Rose, "A 14b 3/6GHz Current-Steering RF DAC in 0.18 μm CMOS with 66 dB ACLR at 2.9 GHZ," *IEEE ISSCC Digest of Technical Papers*, 2012, pp. 458-459.

12. W. Schofield, D. Mercer, and L. Onge, "A 16b 400MS/s DAC with <-80dBc IMD to 300MHz and <-160dBm/Hz Noise Power Spectral Density," *ISSCC Digest of Technical Papers*, 2003, pp. 126-482.

13. Q. Huang, P. Francese, C. Martelli, and J. Nielsen, "A 200MS/s 14b 97 mW DAC in 0.18 um CMOS," *IEEE ISSCC Digest of Technical Papers*, 2004, pp. 364-532.
14. W.-H. Tseng, C.-W. Fan, and J.-T. Wu, "A 12-Bit 1.25-GS/s DAC in 90 nm CMOS With > 70 dB SFDR up to 500 MHz," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 12, pp. 2845-2856, December 2011.

15. R. Adams, K. Nguyen, and K. Sweetland, "A 113 dB SNR Oversampling DAC with Segmented Noise-Shaped Scrambling," *IEEE ISSCC Digest of Technical Papers*, 1998, pp. 62-63.

16. R. van de Plassche, "Dynamic Element Matching for High-Accuracy Monolithic D/A Converters," *IEEE ISSCC Digest of Technical Papers,* 1976, pp. 148-149.

17. L. Carley, "A Noise-Shaping Coder Topology for 15+ Bit Converters," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 2, pp. 267-273, April 1989.

18. B. Jewett, J. Liu, and K. Poulton, "A 1.2 Gs/s 15b DAC for Precision Signal Generation," in *IEEE ISSCC Digest Technical Papers*, 2005, pp. 110-111.

19. K. O. Sullivan, C. Gorman, M. Hennessy, and V. Callaghan, "A 12-bit 320-MSample/s Current-Steering CMOS D/A Converter in 0.44 mm^2," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 7, pp. 1064-1072, July 2004.

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

21. I. Galton, "Spectral Shaping of Circuit Errors in Digital-to-Analog Converters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 10, pp. 808-817, October 1997.

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

23. W.-T. Lin, H.-Y. Huang, and T.-H. Kuo, "A 12-bit 40 nm DAC Achieving SFDR > 70 dB at 1.6 GS/s and IMD < -61dB at 2.8 GS/s With DEMDRZ Technique," *IEEE Journal of Solid-State Circuits*, Vol. 49, no. 3, pp. 708-717, March 2014.

24. K. Chan, N. Rakuljic, and I. Galton, "Segmented Dynamic Element Matching for High-Resolution Digital-to-Analog Conversion," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 11, pp. 3383-3392, December 2008.

25. F. Chen and B.H. Leung, "A High Resolution Multibit Sigma-Delta Modulator with Individual Level Averaging," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 4, pp. 453-460, April 1995.

26. I. Fujimori, A. Nogi, and 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.

27. R. Radke, A. Eshraghi, and T. Fiez, "A 15-Bit Current-Mode ∑Δ DAC Based Upon Rotated Data Weighted Averaging," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 8, pp. 1074-1084, August 2000.

28. R. T. Baird and T. S. Fiez, "Linearity Enhancement of Multibit ∑Δ A/D and D/A Converters Usign Data Weighted Averaging," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 12, pp. 752-

762, December 1995.

29. B. H. Leung and S. Sutarja, "Multibit ∑-Δ A/D Converter Incorporating A Novel Class of Dynamic Element Matching Techniques," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 39, no.1, pp. 35-51, January 1992.

30. M. 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, April 11, 1992.

31. W. Redman-White and D. J. L. Bourner, "Improved Dynamic Linearity in Multi-Level ∑-Δ Converters by Spectral Dispersion of D/A Distortion Products," *European Conference on Circuit Theory and Design*, September 1989, pp. 205-208.

32. R. W. Adams and T. W. Kwan, "Data-Directed Scrambler for Multi-Bit Noise-Shaping D/A Converters," U.S. Patent 5,404,142, April 4 1995.

33. E. Tuijl, J. Homberg, D. Reefman, C. Bastiaansen, and L. Dussen, "A 128 fs, multi-bit ∑Δ CMOS audio DAC with real-time DEM and 115 dB SFDR," *IEEE ISSCC Digest of Technical Papers*, 2004, pp. 368-369.

34. T. S. Kaplan, J. F. Jensen, C. H. Fields, and M. F. Chang, "A 2-GS/s 3-bit Δ∑-Modulated DAC With Tunable Bandpass Mismatch Shaping," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 3, pp. 6-3-610, March 2005.

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

37. A. Sanyal and N. Sun, "Dynamic Element Matching Techniques for Static and

Dynamic Errors in Continuous-Time Multi-Bit $\Delta\sum$ modulators," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 5, no. 4, pp. 598-611, December 2015.

38. E. Familier, I. Galton, "A Fundamental Limitation of DC-Free Quantization Noise With Respect to Nonlinearity-Induced Spurious Tones," *IEEE Transactions on Signal Processing*, vol. 61, no. 16, pp. 4172 - 4180, August 2013.

39. T. Parks and J. McClellan, "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase," *IEEE Transactions on Circuit Theory*, vol. 19, no. 2, pp. 189-194, March 1972.

40. J. McClellan and T. Parks, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters," *IEEE Transactions on Audio and Electroacoustics*, vol. 21, no. 6, pp. 506-526, December 1973.

41. T. Laakso, V. Valimaki, M. Karjalainen, and U. Laine, "Spliting the Unit Delay," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30-60, January 1996.

42. R. A. Wannamaker, S. P. Lipshitz, J. Vanderkooy, and J. N. Wright, "A Theory of Nonsubtractive Dither," *IEEE Transactions on Signal Processing*, vol. 48, no. 2, pp. 449-516, February 2000.

43. B. Widrow, J. Glover, J. McCool, J. Kaunitz, C. Williams, R. Hearn, J. Zeidler, E. Dong, R. Goodlin, "Adaptive Noise Canceling: Principles and Applications," Proceedings of the IEEE, vol. 64, no. 12, pp. 1692-1716, December 1975.

Chapter 2

An ISI Scrambling Technique for Dynamic Element Matching Current-Steering DACs

Abstract— The linearity of high-resolution current-steering digital-to-analog converters (DACs) is often limited by inter-symbol interference (ISI). While dynamic element matching (DEM) can be applied to convert a portion of the ISI to uncorrelated noise instead of nonlinear distortion, DEM alone fails to prevent ISI from at least introducing strong second-order nonlinear distortion. This paper addresses this problem by proposing, analyzing, and experimentally demonstrating a low-cost add-on technique, called ISI scrambling, that, in conjunction with DEM, causes a DAC's ISI to be free of nonlinear distortion. The ISI scrambling technique is demonstrated in a 1-GS/s, 14-bit DEM DAC implemented in 90 nm CMOS technology. The DAC's measured linearity is in line with the state of the art and its measured output power spectra closely match those predicted by the paper's theoretical results.

I. Introduction

Inter-symbol interference (ISI) often limits the linearity of high-resolution current-steering digital-to-analog converters (DACs). It is caused by parasitic memory effects within the DAC's constituent 1-bit DACs, which cause each 1-bit DAC output waveform to depend not only on the 1-bit DAC's current input bit value but also on one or more of its prior input bit values.

Dynamic element matching (DEM) is often applied to multi-bit current-steering DACs to cause error from clock skew and component mismatches to be noise-like waveforms instead of nonlinear distortion [44-52]. It also causes some of the ISI to be a pseudo-random waveform, but even with DEM the ISI contains at least a strong second-order distortion component [53].

The most effective previously-published means of mitigating DAC ISI is to implement the constituent 1-bit DACs as return-to-zero (RZ) 1-bit DACs. RZ 1-bit DACs are reset to a signal-independent state at the end of each clock period. This mitigates ISI because it reduces the dependence of the 1-bit DACs on past input bit values. Unfortunately, the technique's efficacy degrades with clock frequency because of the reduced time available to discharge signal-dependent 1-bit DAC circuit nodes during the reset phase. RZ 1-bit DACs are also significantly more sensitive to clock jitter than their non-return-to-zero (NRZ) counterparts, particularly at high clock frequencies, and they typically consume more than twice the power of comparable NRZ 1-bit DACs.

Other previously published ISI-mitigation techniques measure and then suppress ISI by trimming the delays in the 1-bit DAC switch drivers [54], canceling part of the ISI in either the digital or analog domain [53, 55-57], or dynamically reordering the 1-bit DACs to minimize the ISI [58-59]. The downsides of these techniques are that they require analog-to-digital converter (ADC) based ISI measurement circuitry, their accuracy is limited by that of the measurement circuitry, and they are foreground calibration techniques so they do not track changes in voltage or temperature.

All-digital ISI-mitigation techniques have also been published in which a

modified DEM algorithm spectrally shapes the ISI [60-63]. These techniques are useful in oversampling continuous-time delta-sigma data converters. However, as proven in [53], it is not possible for any such technique to suppress nonlinear distortion across the full Nyquist band, so they are not well-suited to wideband applications that utilize the full Nyquist bandwidth of the DAC.

This paper presents a simple, low-cost, add-on technique called ISI scrambling that works in conjunction with DEM to address these problems. By pseudo-randomly scrambling each 1-bit DAC's transient error, the technique converts the ISI error component that would otherwise have been nonlinear distortion into a noise-like waveform that is free of nonlinear distortion. The paper presents a rigorous mathematical analysis of the technique, presents a 90 nm CMOS, 1-GS/s, 14-bit DEM DAC enabled by the technique that achieves linearity in line with the present state of the art, and shows that the measured results closely match the performance predicted by the mathematical analysis. Furthermore, the paper presents the first published experimental demonstration of a key theoretical result presented in [53] in that when the technique is disabled, the ISI nonlinearity manifests primarily as second-order distortion.

II. ISI Scrambling Technique

A conventional DEM DAC consists of a DEM encoder that drives multiple 1-bit DACs, the outputs of which are summed to form the DEM DAC's output [64]. The output of the i^{th} conventional 1-bit DAC is

$$y'_i(t) = \left(c'_i[n_t] - \tfrac{1}{2}\right) K_i \Delta + e'_i(t), \tag{42}$$

where $c'_i[n]$, which takes on values of 0 and 1, is the 1-bit DAC's binary input sequence, K_i is the 1-bit DAC's weight, Δ is the overall DEM DAC's minimum step-size, $e'_i(t)$ represents the 1-bit DAC's error waveform, $n_t = \lfloor f_s t \rfloor$ is the largest integer less than or equal to $f_s t$, and f_s is the 1-bit DAC's sample rate [53].†

The ISI scrambling technique is a supplement to DEM that converts what would otherwise have been nonlinear ISI error into pseudo-random noise. It incorporates 1-bit DACs that are modified to periodically swap the transient errors that cause nonlinear distortion in conventional 1-bit DACs. Each modified 1-bit DAC, called an ISI scrambling 1-bit DAC in the remainder of the paper, can be configured in real-time to be taken offline and to swap its transient errors.

The idea is to control each of the ISI-scrambling 1-bit DACs such that it spends an equal amount of time with its transient errors swapped and not swapped on average, which causes the average rise and fall transient errors to be symmetric. This is illustrated in Fig. 13. Fig. 13(a) shows an example 1-bit DAC output waveform, Fig. 13(b) shows the corresponding output waveform with its transient errors swapped, and Fig. 13(c) shows the average of the two cases.

Each ISI scrambling 1-bit DAC is taken offline when it transitions from swapping to not swapping its transients, or vice versa. This reduces the disturbance to the overall DAC output caused by inverting the swap state. When an ISI scrambling 1-bit DAC is offline, an auxiliary 1-bit DAC is used temporarily in its place.

A block diagram of the implemented ISI scrambling DEM DAC is shown in

† The prime character is used in this paper to denote variables that pertain to conventional 1-bit DACs.

Fig. 14. It consists of the digital DEM encoder presented in [47], a digital ISI scrambling controller, 20 conventional 1-bit DACs, and 17 ISI scrambling 1-bit DACs. The ISI scrambling 1-bit DACs are shaded in Fig. 14. The conventional 1-bit DACs have weights 1, 1, 2, 2, 4, 4, … , 512, 512, and the ISI scrambling 1-bit DACs all have a weight of 1024. As proven in Section IV, only the 1024-weight 1-bit DACs contribute nonlinear ISI error to the overall DEM DAC output, so the other 1-bit DACs need not be ISI scrambling 1-bit DACs. The top 1024-weight ISI scrambling 1-bit DAC shown in Fig. 14 is the auxiliary 1-bit DAC. The remaining 1024-weight 1-bit DACs are called the *primary* ISI scrambling 1-bit DACs.

The ISI scrambling controller coordinates taking the primary ISI scrambling 1-bit DACs offline and inverting their swap states. It generates the $o_i[n_t]$ and $w_i[n_t]$ binary sequences that control the online state and the swap state, respectively, of each ISI scrambling 1-bit DAC.[‡] The ISI scrambling controller waits a random number of clock cycles between taking randomly-selected individual primary ISI scrambling 1-bit DACs offline. The number of wait cycles is chosen from 1 to N_{Delay} with equal probability where N_{Delay} is a register setting that ranges from 16 to 65536. The purposes of the random wait time are to prevent the ISI errors from being correlated with the input and to avoid introducing periodic disturbances to the overall DEM DAC output that could result in spurs. When selected by the ISI scrambling controller, each primary ISI scrambling 1-bit DAC is taken offline for 8 clock cycles and its swap state is inverted

[‡] A sequence indexed with n_t is technically a continuous-time function, but is referred to as a sequence because it remains constant over each sample period.

after being offline for 4 clock cycles. The auxiliary 1-bit DAC is offline when it is not taking the place of one of the primary ISI scrambling 1-bit DACs. When the auxiliary 1-bit DAC is offline, the ISI scrambling controller randomly inverts or does not invert its swap state with the objective of ensuring that its rise and fall transient errors are symmetric on average.

A behavioral block diagram of each ISI scrambling 1-bit DAC is shown in Fig. 15. It contains a conventional 1-bit DAC with additional components that implement the ISI-scrambling features. The components within the shaded box together behave as a 3-level DAC with ideal outputs –512Δ, 0, and 512Δ, where the output level of 0 corresponds to the offline state ($o_i[n_t] = 0$). The error, $e_{3,i}(t)$, represents the error of the 3-level DAC. It includes the error of the conventional 1-bit DAC as well as that of the $o_i[n_t]$ multiplier.

The ISI scrambling controller swaps the ISI errors of the conventional 1-bit DAC by setting $w_i[n_t] = 1$, which causes two inversions in the signal path of $c_i[n_t]$. The first inversion is caused by the XOR gate and the second inversion is caused by the $p_i[n_t]$ multiplier. In the ideal case where $e_{3,i}(t) = 0$ and the multipliers are error-free, $y_i(t)$ is an exact analog representation of $c_i[n_t]$ when the 1-bit DAC is online ($o_i[n_t] = 1$), because the input signal passes through the XOR gate and the multiplier unchanged if swapping is disabled ($w_i[n_t] = 0$), and the two inversions in the signal path cancel each other out if swapping is enabled ($w_i[n_t] = 1$). The error, $e_{3,i}(t)$, only passes through a single inversion when the transients are swapped which allows its polarity in $y_i(t)$ to be controlled by the ISI scrambling controller.

III. Circuit Details

In conventional DEM DACs with NRZ 1-bit DACs, the nonlinear portion of the ISI error primarily consists of second-order and third-order distortion [53, 65]. However, to the extent that the output of each 1-bit DAC is negligibly affected by the states of the other 1-bit DACs, the ISI error consists of just second-order distortion [53]. The ISI scrambling technique mitigates second-order distortion, so the circuit architecture was chosen to separately mitigate third-order distortion by minimizing the extent to which the 1-bit DACs influence each other as described below.

Figure 16 shows the current-steering cell and switch driver that comprise each 1-bit DAC. Transistor M_1 sets the signal-bearing portion of the 1-bit DAC current, so its dimensions are large to facilitate good matching [66]. The dimensions of M_2 are comparatively small to reduce the current source's parasitic output capacitance and thereby reduce ISI [67].

Transistors $M_{k\,a}$ and $M_{k\,b}$ for k = 3, 4, and 5 steer the 1-bit DAC's signal-bearing current to one or the other of its two output terminals or divert it away from both output terminals depending on the states of $c_i[n_t]$ and $o_i[n_t]$ according to the timing diagram shown in Fig. 17. A quad switching technique is used wherein the transistors $M_{k\,a}$ and $M_{k\,b}$ for k = 3, 4, or 5 each conduct the 1-bit DAC's signal-bearing current for half of each clock cycle [68-71]. This improves DAC linearity because, as implied by Fig. 17, exactly one switch turns on and exactly one switch turns off at each clock edge which causes the disturbance to the sources of $M_{k\,a}$ and $M_{k\,b}$ to be largely independent of the 1-bit DAC input [69]. The gate voltages of $M_{k\,a}$ and $M_{k\,b}$ are such that each transistor

is in saturation when it conducts current, which increases the 1-bit DAC's output impedance.

Thick oxide transistors M_6 through M_9 implement the 1-bit DAC's transient swapping feature. When the 1-bit DAC is online, i.e., when $o_i[n_t] = 1$, they swap or do not swap the connections between their sources and the two 1-bit DAC outputs depending on $w_i[n_t]$. Toggling $w_i[n_t]$ while the 1-bit DAC is online would result in different transient errors than toggling $c_i[n_t]$ and contribute significant distortion to the DAC output. Hence, $w_i[n_t]$ is only toggled when the 1-bit DAC is offline. Toggling $w_i[n_t]$ when the 1-bit DAC is offline still causes unwanted charge to be injected into the DAC's output terminals, but much of this charge is cancelled because rising and falling transitions of the gate voltages of M_6 and M_9 coincide with falling and rising transitions of the gate voltages of M_7 and M_8, respectively. Furthermore, d and $\overline{d}$ change state at random times so any error from charge injection is independent of the DAC's input code and does not contribute harmonic distortion.

The level-shifter in Fig. 16(b) that drives the gates of M_6 through M_9 is powered by an on-chip LDO similar to that presented in [72], except that an external bypass capacitor is used instead of an internal Miller capacitor. The LDO output voltage is such that when d is high and $\overline{d}$ is low, M_6 and M_9 are in saturation and M_7 and M_8 are off, and when d is low and $\overline{d}$ is high, M_6 and M_9 are off and M_7 and M_8 are in saturation. Keeping M_6 through M_9 in saturation when conducting the 1-bit DAC's signal-bearing current increases the 1-bit DAC's output impedance, thereby reducing the dependence of its transient error on the overall DEM DAC's output [65, 73]. Hence, in addition to

implementing the transient swapping feature, M_6 through M_9 perform the function of the cascode stages commonly used in conventional current-steering 1-bit DACs [50, 55-56, 58-59, 65, 70, 74]. As such, they improve overall DEM DAC linearity at the expense of a small reduction in headroom. The $I_{trickle}$ current sources shown in Fig. 16(a) prevent M_6 through M_{10} from entering triode when not conducting the 1-bit DAC's signal-bearing current, thereby reducing nonlinear distortion that would otherwise be caused by code-dependent output capacitance variations [74].

Two versions of the IC were fabricated that differ only in that each 1-bit DAC in the second version includes an extra cascode stage. The second version was fabricated to evaluate the effect of further increasing the 1-bit DAC output impedances. The measured performance of the two versions was found to be indistinguishable, which supports the assumption made in the analysis in Section IV that the error caused by finite 1-bit DAC output impedance is negligible, at least relative to the linearity achieved by the IC as reported in Section V. Hence, this paper presents the details of only the first version of the chip.

The main sources of transient errors that cause nonlinear ISI are the skews between the rising and falling edges of the gate voltages of M_{3a}, M_{3b}, M_{4a}, and M_{4b}, and mismatches among M_{3a}, M_{3b}, M_{4a}, and M_{4b}. The transient swapping feature implemented by M_6 through M_9 swaps these transient errors, but does not swap any additional transient error introduced by mismatches among M_6 through M_9. However, the impedance looking into the sources of M_6 through M_9 is relatively low, so mismatches among M_6 through M_9 do not cause significant transient errors. Monte Carlo simulations

support this assertion. With mismatches applied only to M_6 through M_9, they predict a second harmonic of less than −97 dBc for a full-scale sinusoidal input sequence. In contrast, with mismatches applied to all transistors except M_6 through M_9, ISI scrambling disabled, and the same full-scale input sequence, they predict a second harmonic of −72 dBc.

As explained in Section II each 1-bit DAC has a weight K_i = 1, 2, 4, 8, ..., or 1024. The overall DEM DAC's minimum-step size, Δ, is 2.4 μA, so the nominal signal-bearing current sourced by M_1 and M_2 in the K_i = 1 1-bit DAC is 1.2 μA. The current-steering cells in the K_i = 2, 4, and 8 1-bit DACs were implemented by increasing the widths of M_1 and M_2 by factors of 2, 4, and 8, respectively, relative to those in the K_i = 1 1-bit DAC while keeping the dimensions of the other transistors unchanged. The current-steering cells of weights K_i = 16, 32, ..., and 256, were implemented by replicating each transistor in the K_i = 8 current-steering cell 2, 4, 8, ..., and 32 times, respectively, with the replicated transistors connected in parallel. The 1-bit DACs of weight 512 and 1024 were implemented as parallel copies of the K_i = 256 1-bit DAC.

In the current-steering cells of weight K_i = 1, 2, 4, and 8 in which M_6 through M_{10} are each implemented with unit-weight transistors, the nominal value of $I_{trickle}$ is 280 nA. In the current-steering cells of weights K_i = 16, 32, ..., and 256, in which M_6 through M_{10} are each implemented by connecting $2^{-3}K_i$ unit-weight transistors in parallel, the nominal value of $I_{trickle}$ is 3% of the signal-bearing current.

Better matching could have been achieved by implementing each K_i-weight current-steering cell for $K_i \geq 2$ by simply connecting K_i unit-weight current-steering

cells in parallel. However, doing so would have significantly increased the required current-steering cell drive strength for $K_i \geq 2$, which would have correspondingly increased the area and current consumption of the switch drivers.

In the absence of other considerations, the best switch driver scaling strategy to match the 1-bit DAC transients is to have the weight of each switch driver be proportional to the weight of the current-steering cell it drives. The switch driver latches are as described in [74], and in TSMC 90 nm technology a minimum-size latch has more drive strength than is required to drive a $K_i = 1$ current-steering cell. Consequently, if the latches were scaled in proportion to the current-steering cells they would consume far more area and power than necessary. Instead, all current-steering cells are driven by copies of a switch driver that is optimized to drive a $K_i = 256$ current-steering cell. Dummy transistors are used to load the switch drivers in each 1-bit DAC with a weight of $K_i = 128$ and lower, so the loads driven by all the switch drivers are approximately equal to the load of a $K_i = 256$ current-steering cell.

As proven in Section IV, ISI scrambling is not necessary in 1-bit DACs 1-20. Hence, the ISI scrambling feature is disabled in these 1-bit DACs by setting $o_i[n_t]$ and $w_i[n_t]$ to 1 and 0, respectively, for $i = 1, 2, \ldots, 20$ and all t. Alternatively, the ISI-scrambling circuitry could have been omitted from 1-bit DACs 1-20, but this would have degraded matching.

The clock input buffer is a two-stage differential to single-ended amplifier. The first stage is a differential pair with diode-connected load transistors, and the second stage is a differential pair with a current mirror load. The clock input buffer drives a

clock tree that distributes the clock to the 1-bit DACs. The targeted total jitter of the clock input buffer and clock tree is 100 fs RMS, and was verified via simulation. The jitter target ensures that the jitter does not limit the noise performance of the DAC when DEM is enabled.

The placed and routed (P/R) digital block, clock input buffer, bias circuitry, and set of switch drivers are each powered by their own power domain to reduce coupling through the supplies. The power and ground are distributed via wide traces to reduce supply impedance, and extensive on-chip decoupling fills most of the unused area in the chip.

IV. Analysis

This section along with the appendices presents a mathematical derivation that quantifies the behavior of the ISI scrambling technique. It also develops theoretical DAC output PSD expressions that are compared against and closely match the corresponding measured power spectra in Section V. The section and appendices may be skipped without loss of continuity by those who are not interested in the mathematical details.

A. ISI Scrambling 1-Bit DAC Output Model

In conventional current-steering DACs, particularly significant types of errors include mismatches among the 1-bit DACs, transient errors, ISI, and clock feedthrough. For most current-steering DACs, these errors are dominant and $e'_i(t)$ in (42) is well-

modeled as

$$e'_i(t) = \begin{cases} e_{11i}(t), & \text{if } c'_i[n_t - 1] = 1,\ c'_i[n_t] = 1, \\ e_{01i}(t), & \text{if } c'_i[n_t - 1] = 0,\ c'_i[n_t] = 1, \\ e_{00i}(t), & \text{if } c'_i[n_t - 1] = 0,\ c'_i[n_t] = 0, \\ e_{10i}(t), & \text{if } c'_i[n_t - 1] = 1,\ c'_i[n_t] = 0, \end{cases} \tag{43}$$

where $e_{11i}(t)$, $e_{01i}(t)$, $e_{00i}(t)$, and $e_{10i}(t)$, are T_s-periodic ($T_s = 1/f_s$) waveforms that correspond to the errors made by the 1-bit DAC for the four possible combinations of the current and previous 1-bit DAC input bit values [53]. During each clock cycle, $e'_i(t)$ is equal to one of the four T_s-periodic error waveforms. Formulating $e'_i(t)$ in terms of these waveforms simplifies the subsequent analysis. It does not impose any restrictions on the 1-bit DAC input sequence, nor does it cause $e'_i(t)$ to be periodic.

As shown in [53], an equivalent form of (42) is

$$\begin{aligned} y'_i(t) = {} & x'_i[n_t]\alpha_i(t)K_i\Delta + \beta_i(t) + x'_i[n_t - 1]\gamma_i(t) \\ & + x'_i[n_t]x'_i[n_t - 1]\eta_i(t), \end{aligned} \tag{44}$$

where

$$x'_i[n_t] = c'_i[n_t] - \tfrac{1}{2}, \tag{45}$$

and $\alpha_i(t)$, $\beta_i(t)$, $\gamma_i(t)$, and $\eta_i(t)$ are linear combinations of $e_{11i}(t)$, $e_{01i}(t)$, $e_{00i}(t)$, and $e_{10i}(t)$ so they are T_s-periodic waveforms. The $x'_i[n_t]x'_i[n_t - 1]\eta_i(t)$ term in (44) is a second-order nonlinearly distorted version of the 1-bit DAC's input sequence and is caused by ISI. When the rise and fall transients of the 1-bit DAC output waveform are asymmetric, such as illustrated in Fig. 13(a), $\eta_i(t)$ is nonzero which causes the 1-bit DAC to introduce second-order nonlinear ISI error. Without DEM, this would cause the overall

DAC to introduce second-order and several higher-order nonlinear distortion terms, but, as shown in [53], DEM prevents all but second-order nonlinear distortion.

It follows from Fig. 15 that $p_i^2[n_t] = 1$,

$$p_i[n_t] = 1 - 2w_i[n_t], \quad \text{and } c'_i[n_t] = \left(c_i[n_t] - \tfrac{1}{2}\right) p_i[n_t] + \tfrac{1}{2}, \tag{46}$$

so (42) and Fig. 15 imply that

$$y_i(t) = x_i[n_t]o_i[n_t]K_i\Delta + e_i(t), \tag{47}$$

where

$$x_i[n_t] = c_i[n_t] - \tfrac{1}{2}, \tag{48}$$

and $e_i(t)$ represents all error from non-ideal behavior. To the extent that the $p_i[n_t]$ multiplier is ideal, Fig. 15 implies that

$$e_i(t) = p_i[n_t]e_{3,i}(t). \tag{49}$$

Given that $e_{3,i}(t)$ during the nth clock period depends on whether the ISI scrambling 1-bit DAC is offline or online during the nth and $(n-1)$th clock periods, it is convenient for the following analysis to define sequences that are 1 or 0 depending on the four combinations of offline and online statuses during the two clock periods. Specifically, these sequences are defined as

$$o_{++,i}[n_t] = o_i[n_t - 1]o_i[n_t], \tag{50}$$

$$o_{+\times,i}[n_t] = o_i[n_t - 1]\left(1 - o_i[n_t]\right), \tag{51}$$

$$o_{\times+,i}[n_t] = \left(1 - o_i[n_t - 1]\right)o_i[n_t], \tag{52}$$

and

$$o_{\times\times,i}[n_t] = \left(1 - o_i[n_t - 1]\right)\left(1 - o_i[n_t]\right), \tag{53}$$

which have the property that one of $o_{++,i}[n_t]$, $o_{+\times,i}[n_t]$, $o_{\times+,i}[n_t]$, and $o_{\times\times,i}[n_t]$ is 1 and the rest are 0 during each clock cycle.[4]

In the error formulation of the conventional 1-bit DAC, a specific T_s-periodic error waveform is defined for each combination of the 1-bit DAC's current and previous input bit values. This results in the $2^2 = 4$ T_s-periodic error waveforms used in (43) to formulate $e'_i(t)$. Applying similar reasoning to the 3-level DAC in the shaded box in Fig. 15 results in $3^2 = 9$ T_s-periodic error waveforms used to formulate

$$\begin{aligned} e_{3,i}(t) = {} & o_{++,i}[n_t]e'_i(t) \\ & + o_{+\times,i}[n_t]\left[c'_i[n_t-1]e_{1\times i}(t) + \left(1-c'_i[n_t-1]\right)e_{0\times i}(t)\right] \\ & + o_{\times+,i}[n_t]\left[c'_i[n_t]e_{\times 1i}(t) + \left(1-c'_i[n_t]\right)e_{\times 0i}(t)\right] \\ & + o_{\times\times,i}[n_t]e_{\times\times i}(t), \end{aligned} \tag{54}$$

where $e_{1\times i}(t)$, $e_{0\times i}(t)$, $e_{\times 1i}(t)$, $e_{\times 0i}(t)$, and $e_{\times\times i}(t)$ are 5 of the 9 T_s-periodic error waveforms and the remaining 4 T_s-periodic error waveforms are contained in the definition of $e'_i(t)$, which is given by (43).

Thus, $e_i(t)$ is given by (49) with $e_{3,i}(t)$ given by (54) provided the error introduced by the $p_i[n_t]$ multiplier in Fig. 15 is negligible. Otherwise, it is necessary to define two error waveforms, $e_{3,i}^{+}(t)$ and $e_{3,i}^{-}(t)$, that have forms similar to the right side of (54) except that their constituent T_s-periodic error waveforms correspond to the cases of $p_i[n_t] = 1$ and $p_i[n_t] = -1$ respectively. Then, $e_i(t) = e_{3,i}^{+}(t)$ when $p_i[n_t] = 1$ and $e_i(t) = -e_{3,i}^{-}(t)$ when $p_i[n_t] = -1$. In this case ISI scrambling does not completely eliminate ISI-induced second-order nonlinear distortion because $e_{3,i}^{+}(t) \neq e_{3,i}^{-}(t)$. Yet as supported by

[4] The subscript characters + and × denote "online" and "offline" respectively.

the experimental results presented in Section V, the error introduced by the $p_i[n_t]$ multiplier in Fig. 15 is indeed negligible. The reason is that the $p_i[n_t]$ multiplier is implemented as a swapper cell as explained in Section III so it only introduces error when $p_i[n_t]$ changes state and this only happens when the 1-bit DAC is offline.

B. ISI Scrambling DEM DAC Output Model

Equations (47)-(54) were formulated to model ISI scrambling 1-bit DACs, but with $o_i[n_t] = 1$ and $w_i[n_t] = 0$ for all t they can also be used to model conventional 1-bit DACs. This is because a conventional 1-bit DAC is equivalent to an ISI scrambling 1-bit DAC that never goes offline and never has its transient errors swapped. Consequently, in the following analysis all the 1-bit DACs in Fig. 14 are modelled via (47)-(54) but with $o_i[n_t] = 1$ and $p_i[n_t] = 1$ for $i = 1, 2, \ldots, 20$ and for all t.

As proven in [47], the DEM encoder's outputs, $c_i[n_t]$, cause the $x_i[n_t]$ sequences, which are given by (48) and take on values that are restricted to ½ and −½, to satisfy

$$x_i[n_t] = \frac{m_i x[n_t] + \lambda_i[n_t]}{\Delta}, \tag{55}$$

where

$$\sum_{i=1}^{36} K_i m_i = 1, \text{ and } \sum_{i=1}^{36} K_i \lambda_i[n_t] = 0. \tag{56}$$

Given that the ISI scrambling technique causes the auxiliary 1-bit DAC to sometimes take the place of one of the primary ISI scrambling 1-bit DACs, (56) implies

$$\sum_{i=1}^{37} K_i m_i o_i[n_t] = 1 \text{ and } \sum_{i=1}^{37} K_i \lambda_i[n_t] o_i[n_t] = 0. \tag{57}$$

At the circuit level, the summation operation in Fig. 14 is implemented by connecting the outputs of the current-steering 1-bit DACs. Under the assumption that the impedance of each 1-bit DAC is high enough that its output is negligibly affected by the states of the other 1-bit DACs, an assumption which is supported by the measured results presented in Section V, it follows that the output of the overall DAC is given by

$$y(t) = \sum_{i=1}^{37} y_i(t) \,. \tag{58}$$

Substituting (47) with $e_i(t) = 0$ and (55) into (58), and simplifying the result using (57) results in

$$y(t) = x[n_t] \,, \tag{59}$$

which represents the overall DEM DAC output in the absence of non-ideal behavior. In contrast, as shown in Appendix A, the 1-bit DAC errors cause the overall DAC output to degrade to

$$y(t) = \alpha(t)x[n_t] + \beta(t) + e_{DAC}(t) \,, \tag{60}$$

where $\alpha(t)$ and $\beta(t)$ are T_s-periodic waveforms and $e_{DAC}(t)$ represents the overall DAC's remaining non-ideal performance.

The first term on the right side of (60) is the desired signal component of the DAC output. As shown in [53], its continuous-time Fourier transform (CTFT) is

$$\Im_{CT}\left\{\alpha(t)x[n_t]\right\} = A_p\left(j\omega\right) X\left(e^{j\omega T_s}\right), \tag{61}$$

where $A_p(j\omega)$ is the CTFT of

$$\alpha_p(t) = \begin{cases} \alpha(t) & \text{if } 0 \le t < T_s, \\ 0 & \text{otherwise,} \end{cases} \tag{62}$$

and $X(e^{j\omega T_S})$ is the discrete-time Fourier transform (DTFT) of $x[n]$. Ideally, $\alpha(t) = 1$ in which case the $\alpha(t)x[n_t]$ term in (60) reduces to the right side of (59), and (16)-(17) imply that $|A_p(j\omega)| = \sin(\pi T_s f)/\pi f$. This corresponds to the classic zero-order-hold frequency response roll-off of an ideal DAC. In practice, 1-bit DAC errors cause $\alpha(t)$ to deviate from unity, but, as implied by (16) and (17), this just affects the frequency response roll-off without introducing nonlinear distortion. Moreover, the effect on the frequency roll-off typically is not significant in current steering DACs [53].

The $\beta(t)$ term in (60) introduces fixed tones in the DAC output at integer multiples of f_s. Such tones occur in all types of DACs, e.g., as a result of clock feedthrough. They do not fall within any Nyquist band of the DAC output and do not depend on the DAC input, so they do not cause problems in typical DAC applications [53].

In contrast, the $e_{DAC}(t)$ term in (60) limits performance in typical applications. As proven in Appendix A it can be written as

$$e_{DAC}(t) = e_{MM}(t) + e_{ISI-linear}(t) + e_{ISI-noise}(t), \tag{63}$$

where $e_{MM}(t)$ is caused by mismatches among the 1-bit DACs and the remaining two terms are caused by ISI.

The expression for $e_{MM}(t)$ is

$$e_{MM}(t) = \frac{1}{\Delta}\sum_{i=1}^{37} o_{++,i}[n_t]\lambda_i[n_t]\varepsilon_i(t), \tag{64}$$

where each $\varepsilon_i(t)$ is a T_s-periodic waveform defined in Appendix A and DEM causes the $\lambda_i[n_t]$ sequences to be zero-mean, pseudo-random sequences that are uncorrelated with the DAC's input sequence, $x[n_t]$, i.e.

$$\left.\begin{array}{l} \mathrm{E}\{\lambda_i[n_t]\} = 0 \\ \mathrm{E}\{\lambda_i[n_t]\lambda_i[m_t]\} = 0 \text{ for } m_t \neq n_t \end{array}\right\} \text{ regardless of } x[n_t], \tag{65}$$

where E{u} denotes the expected value of u [47]. Without DEM the $\lambda_i[n_t]$ sequences would be nonlinearly related to the DAC input, so $e_{MM}(t)$ would contain nonlinear distortion. With DEM $e_{MM}(t)$ is a zero-mean pseudo-random noise waveform because of the behavior of the $\lambda_i[n_t]$ sequences [53].

The expression for $e_{ISI\text{-}linear}(t)$ is

$$e_{ISI-linear}(t) = x[n_t - 1]\gamma(t), \tag{66}$$

where $\gamma(t)$ is a T_s-periodic waveform defined in Appendix A. The 1-bit DAC errors cause $\gamma(t)$, and hence $e_{ISI\text{-}linear}(t)$, to deviate from 0, but the argument applied above to show that the $\alpha(t)x[n_t]$ term in (60) does not introduce nonlinear distortion also applies to $e_{ISI\text{-}linear}(t)$. While the 1-bit DAC errors cause $\alpha(t)$ to deviate slightly from its ideal value of $\alpha(t) = 1$, they cause $\gamma(t)$ to deviate slightly from its ideal value of $\gamma(t) = 0$, so in addition to not introducing nonlinear distortion, $e_{ISI\text{-}linear}(t)$ typically has much lower power than $\alpha(t)x[n_t]$.

The expression for $e_{ISI\text{-}noise}(t)$ is

$$e_{ISI-noise}(t) = e_{DEM}(t) + e_{IS}(t), \tag{67}$$

where $e_{DEM}(t)$ and $e_{IS}(t)$ are error waveforms resulting from ISI that comprise noise instead of nonlinear distortion because of DEM and ISI scrambling, respectively. The expression for $e_{DEM}(t)$ is

$$
\begin{aligned}
e_{DEM}(t) = \frac{1}{\Delta}\sum_{i=1}^{37} \lambda_i[n_t - 1]\Big[o_{++,i}[n_t]\gamma_i(t) + z_i[n_t]\lambda_i[n_t]\Big] \\
+\frac{1}{\Delta}\sum_{i=21}^{37}\Big(o_{+\times,i}[n_t]\gamma_{+\times,i}(t)\lambda_i[n_t - 1] \\
+\Delta \cdot o_{\times+,i}[n_t]\varepsilon_{\times+,i}(t)\lambda_i[n_t] \\
+2^{-14}\cdot z_i[n_t]\big[x[n_t - 1]\lambda_i[n_t] + x[n_t]\lambda_i[n_t - 1]\big]\Big),
\end{aligned}
\tag{68}
$$

where $\varepsilon_{\times+,i}(t)$ and $\gamma_{+\times,i}(t)$ are defined in Appendix A, and $z_i[n_t] = o_{++,i}[n_t]p_i[n_t]\eta_i(t)/\Delta$. Every term on the right side of (68) contains one or both of $\lambda_i[n_t]$ and $\lambda_i[n_t - 1]$, so, as in the case of $e_{MM}(t)$, DEM causes $e_{DEM}(t)$ to comprise zero-mean pseudo-random noise instead of nonlinear distortion.

The expression for $e_{IS}(t)$ is

$$
\begin{aligned}
e_{IS}(t) = \sum_{i=21}^{37}\Big(e_i(t) + p_i[n_t]\Big[o_{++,i}[n_t]\beta_i(t) + o_{+\times,i}[n_t]\beta_{+\times,i}(t) \\
+ o_{\times+,i}[n_t]\beta_{\times+,i}(t) + o_{\times\times,j}[n_t]e_{\times\times i}(t)\Big] \\
+2^{-14}x[n_t]\Big[\tilde{o}_{++,i}[n_t]\varepsilon_i(t) + \tilde{o}_{\times+,i}[n_t]\varepsilon_{\times+,i}(t)\Big] \\
+2^{-14}\Delta^{-1}x[n_t - 1]\Big[\tilde{o}_{++,i}[n_t]\gamma_i(t) + \tilde{o}_{+\times,i}[n_t]\gamma_{+\times,i}(t)\Big]\Big),
\end{aligned}
\tag{69}
$$

where

$$
e_i(t) = \frac{1}{2^{28}\Delta^2} q_i[n_t]x[n_t]x[n_t - 1]\eta_i(t), \tag{70}
$$

$$
q_i[n_t] = o_{++,i}[n_t]p_i[n_t], \tag{71}
$$

and the factors in (69) that are not defined above are defined in Appendix A. To show that $e_{IS}(t)$ does not introduce nonlinear distortion, it is sufficient to show that

$$
\mathrm{E}\big\{ e_{IS}(t)\, f(x[n_t], x[n_t - 1], x[n_t - 2], ...)\big\} = 0 \tag{72}
$$

holds for every deterministic nonlinear function of the DEM DAC's input sequence,

$f(x[n_t], x[n_t - 1], x[n_t - 2], \ldots)$. Given that $x[n_t]$ is deterministic, it follows from (72) that it is sufficient to show

$$\mathrm{E}\{e_{IS}(t)\} = 0 \text{ regardless of } x[n_t] \text{ for all } t. \tag{73}$$

Every term on the right side of (69) contains either $p_i[n_t]$, $\tilde{o}_{++,i}[n_t]$, $\tilde{o}_{+\times,i}[n_t]$, or $\tilde{o}_{\times+,i}[n_t]$, all of which, by definition, are zero-mean random sequences that are independent of $x[n_t]$ for all t. In addition, $p_i[n_t]$ is independent of the other stochastic sequences in (69), and so it follows from (69) that (73) holds.

If ISI scrambling were not enabled, $p_i[n_t] = 1$ and $o_{++,i}[n_t] = 1$ for all t, which would cause the $e_i(t)$ terms in (69) to contribute second-order nonlinear distortion. In contrast, the other terms in (69) as well as those in (64) and (68) either have zero means regardless of $x[n_t]$ or have no dependence on $x[n_t]$, so they do not contribute nonlinear distortion in either the absence or presence of ISI scrambling.

As shown in [47], $m_i = 0$ for $i = 1, 2, \ldots, 20$, so (55) and (48) imply that the $c_i[n_t]$ outputs of the DEM encoder for these values of i do not contain components proportional to $x[n_t]$. This is why the summation in (69) starts from $i = 21$, and why the bottom 20 1-bit DACs in Fig. 14 need not be ISI scrambling 1-bit DACs.

To show that $e_{IS}(t)$ does not introduce spurious tones, it is sufficient to show that

$$\lim_{\tau \to \infty} \mathrm{E}\{e_{IS}(t) e_{IS}(t+\tau)\} = 0 \qquad \text{regardless of } x[n_t] \text{ for all } t. \tag{74}$$

By definition, $x[n_t]$ is deterministic, and, for sufficiently large τ and any i and j, $o_i[n_t]$, $o_j[n_{t+\tau}]$, $p_i[n_t]$, and $p_j[n_{t+\tau}]$ are independent, and $p_i[n_t]$ and $p_j[n_{t+\tau}]$ are zero mean. As explained in Appendix A, $\tilde{o}_{**,i}[n]$ is the zero-mean portion of $o_{**,i}[n]$ where $**$ is a

placeholder for $_{++}$, $_{+\times}$, $_{\times+}$, or $_{\times\times}$. Therefore, (50)-(53) imply that $\delta_{ab,i}[n_t]$, $\delta_{cd,j}[n_{t+\tau}]$, $p_i[n_t]$, and $p_j[n_{t+\tau}]$ are independent for sufficiently large τ and any i and j, where a, b, c, and d are any combination of + and ×. Expanding $\mathrm{E}\{e_{IS}(t)e_{IS}(t+\tau)\}$ using (69) and applying the above observations, verifies that (74) holds.

C. Effect of ISI Scrambling for Sinusoidal Inputs

As explained above, the $e_i(t)$ terms in (69) would contribute nonlinear distortion if it were not for ISI scrambling, so the properties of these terms are of particular interest. The power spectral density (PSD) of $e_i(t)$ is derived in Appendices B and C for a full-scale input signal, $x[n_t] = 8192\Delta\sin(2\pi n_t f_0/f_s)$, where f_0/f_s satisfies

$$0 < f_0/f_s < \tfrac{1}{2} \quad \text{and} \quad f_0/f_s \neq \tfrac{1}{4}, \tag{75}$$

to avoid the degenerate cases of either the fundamental or the second harmonic of the input signal aliasing to zero frequency in $x[n_t]$.

The derivation involves two key components. Appendix B proves that the $q_i[n_t]$ factor of $e_i(t)$ with n_t replaced by n is a wide-sense stationary (WSS) discrete-time random process, and derives an expression for its autocorrelation, $R_{q,i}[k]$. Appendix C applies this result to show that $e_i(t)$ is a cyclo-stationary continuous-time random process, and derives expressions for its time-average autocorrelation, $\bar{R}_{e,i}(\tau)$, and PSD, $S_{e,i}(j\omega)$.

Given that each $e_i(t)$ term in (69) is proportional to $p_j[n_t]$ if and only if $j = i$, and, by definition, $p_i[n]$ is independent of $p_j[n]$ when $i \neq j$, it follows that $e_i(t)$ and $e_j(t)$ are independent when $i \neq j$. Consequently, the PSD of the portion of $e_{DAC}(t)$ that would

be nonlinear distortion if ISI scrambling were not applied is

$$S_{e_{IS}}(j\omega) = \sum_{i=21}^{37} S_{e,i}(j\omega). \tag{76}$$

Theoretical curves calculated from (76) and its supporting equations in Appendices B and C are shown in Section V to closely match measurement results. The theoretical results prove and the measured results demonstrate that the ISI scrambling technique converts what would otherwise be ISI-induced second-order harmonic distortion spurs to noise "bumps" centered at the spur frequencies, e.g., at frequencies of 0 and $2f_0$ Hz in the DAC's first Nyquist band. The peak amplitudes of the noise bump PSDs decrease as the average ISI scrambling 1-bit DAC transient swapping rate is increased. Hence, they decrease as N_{Delay} is decreased.

V. Measurement Results

The IC was implemented in a TSMC 90 nm process. A die photograph of it is shown in Fig. 18. The die measures 2.3 mm × 2.45 mm and its active area is 1.48 mm^2. The incremental circuit area required to implement ISI scrambling is 0.08 mm^2 of which 0.03 mm^2 corresponds to digital logic. The IC was tested in a QFN64 package and all grounds were down-bonded to the package's ground paddle. In addition to the DAC core, the IC contains LVDS interface circuitry interspersed throughout the pad ring and a direct digital synthesizer (DDS) integrated in the P/R digital block. The DDS was used for all measurements presented in this section.

The packaged IC was mounted to a test circuit board with an Ironwood

elastomer socket. The clock signal applied to the test circuit board was generated by passing the single-ended output of a low-jitter laboratory signal generator through a passive bandpass filter to suppress noise and spurious tones. A balun and associated passive matching circuitry on the test circuit board converts the clock signal to differential form prior to the IC. A transformer and associated passive matching circuitry on the test circuit board converts the differential DAC output to a single-ended signal that was measured via a laboratory signal analyzer to obtain the results presented in this section.

Fig. 19 shows representative DAC output power spectra for a 481.4 MHz single-tone input sequence. Compared to the case with both DEM and ISI scrambling disabled, the results indicate that enabling just DEM improves the spurious-free dynamic range (SFDR) by only 4 dB whereas enabling both DEM and ISI scrambling improves the SFDR by 14 dB. With DEM enabled and ISI scrambling disabled, the measured SFDR is limited by an aliased second harmonic caused by ISI as predicted by the corresponding theoretical result presented in [53]. As predicted by the theoretical results presented in this paper, the measured second harmonic is highly attenuated when both DEM and ISI scrambling are enabled.

Similar results also hold for other input signals. Fig. 20 shows measured SFDR values versus signal frequency for single-tone and two-tone DAC input sequences. The data demonstrate that ISI scrambling significantly increases the measured SFDR values relative to the cases where ISI scrambling is disabled.

Fig. 21 shows several measured and theoretically calculated power spectra

corresponding to a 481.4 MHz single-tone DAC input sequence with DEM and ISI scrambling enabled. To demonstrate the correspondence between theory and measurement, the power spectra are shown over a zoomed-in frequency band centered on the frequency at which the limiting aliased second harmonic occurs in the absence of ISI scrambling. A pair of power spectra, one measured and one calculated, are superimposed for each of N_{Delay} = 16, 256, 4096, and 65536. The calculated power spectra were obtained via the equations derived in Section IV and the appendices with a noise floor added to match that of the measured power spectra, $\alpha(t)$ approximated as unity (its ideal value), and each $\eta_i(t)$ approximated as a constant taken to be that which yielded the best overall match between the calculated and measured power spectra.

As predicted by the analysis in Section IV and demonstrated in Fig. 21, ISI scrambling converts what would otherwise be second-order nonlinear distortion into a spectral noise "bump" and the height of the bump decreases with N_{Delay}. Except for a −82 dBc residual second-order spur in the measured results, the measured and corresponding calculated power spectra shown in Fig. 21, are in very close agreement.

The authors believe that the residual second-order spur is the result of differential path mismatches from transistors M_6 through M_9 in Fig. 16(a) through the test circuit board's output network up to the transformer. Evidence in support of this belief is that the residual spur was not predicted by simulations in the absence of such mismatches, and it was found to vary somewhat across different copies of the IC and test circuit board.

With DEM and ISI scrambling enabled, the DAC's SFDR for input frequencies

above 300 MHz is limited by higher third-order distortion than was predicted by simulations prior to IC fabrication. The authors subsequently realized they had made a mistake in the simulation setup which caused the impedance of the ground bond wires to be underestimated. The measured third-order distortion was reproduced in simulation when the mistake was corrected. Nearly identical third order distortion was observed in simulation when ISI scrambling was disabled and M_7 and M_8 in Fig. 16(a) were removed, which shows that ISI scrambling is not the cause of the third-order distortion. If the problem caused by the DAC ground impedance were fixed, simulations suggest that the third-order distortion would be limited by the 1-bit DAC output impedance [65, 73-75].

Fig. 22 shows the effect of ISI scrambling on the second-harmonic for full-scale single tone input sequences. With DEM enabled and ISI scrambling disabled, the ratio of the power of the desired signal component to the power of the second-harmonic decreases with input frequency. In contrast, with both DEM and ISI scrambling enabled, the ratio of the two components does not increase with frequency.

These observations are predicted by the analysis presented in Section IV. The CTFT of the DAC's desired signal component given by (16) is proportional to $A_p(j\omega)$, which is the CTFT of the T_s-duration pulse, $\alpha_p(t)$, equal to one period of $\alpha(t)$. The second-order distortion caused by ISI is represented by the $e_i(t)$ terms in (69). By the same reasoning that led to (16), the CTFT of $e_i(t)$ is proportional to $H_{p,i}(j\omega)$, which is the CTFT of a T_s-duration pulse, $\eta_{p,i}(t)$, equal to one period of $\eta_i(t)$. While $\alpha_p(t)$ is approximately constant for most of $0 \le t < T_s$, $\eta_{p,i}(t)$ is the result of 1-bit DAC transient errors

so it is non-zero mainly over a short interval close to $t = 0$. Consequently, $|H_{p,i}(j\omega)|$ decreases less over the first Nyquist band than $|A_p(j\omega)|$ such that the second-order distortion exhibits less frequency roll-off than the desired signal component. In contrast, the residual second-order distortion term that remains when ISI-scrambling is enabled is not caused by ISI, so it does not appear in $e_i(t)$ and is not subject to this effect.

Five randomly-selected copies of the IC were tested. For full-scale single-tone input sequences with frequencies spanning the Nyquist band, the worst measured second harmonic of the five copies is –64 dBc with DEM enabled and ISI scrambling disabled and –78 dBc with both DEM and ISI scrambling enabled. The power spectra and SFDR measurements presented in this section are from the IC copy with the worst second-order distortion when ISI scrambling is disabled.

The full-scale single-tone noise performance of the IC is shown in Table I. Enabling DEM reduces the SNDR by about 3 dB, which is expected; DEM increases the number of 1-bit DAC transitions and causes the 1-bit DAC rise and fall transition mismatches to increase the overall DEM DAC's output noise. The same degradation in SNDR was observed in simulation when DEM was enabled. In contrast, the RZ 1-bit DACs used in [47, 51] have the same number of output transitions regardless of whether DEM is enabled, so enabling DEM does not degrade the SNDR for full-scale input signals in those cases. Enabling ISI scrambling in addition to DEM reduces the SNDR by up to 0.5 dB, which shows that the noise caused by DEM dominates the noise floor. As expected, the noise performance of the IC is worse than those of DACs that incorporate 1-bit DAC mismatch error calibration [50-51, 56, 58-59, 77]. However, the

ISI scrambling technique's low circuit area and power consumption makes it inexpensive to combine with previously published techniques that calibrate 1-bit DAC mismatch errors.

Table II and Fig. 23 present key specifications of the IC along with those of several other published state-of-the-art DACs. The P/R digital block includes the DEM encoder, ISI scrambling controller and DDS, and consumes 105 mW from a 1.2 V supply. The analog circuitry consumes the remaining 133 mW from 1.2 V and 3.3 V supplies (the DAC output current is sourced from the 3.3 V supply). The data show that the IC exhibits better SFDR performance than all but the DACs presented in [50], [51], and [56]. The DAC presented in [56] achieves high linearity, but it requires manual laboratory measurements to iteratively configure its foreground calibration technique which limits its applicability. The DAC presented in [51] uses RZ 1-bit DACs to avoid being limited by nonlinear ISI (measurements show that its SFDR drops by 10 dB when its 1-bit DACs are operated in NRZ mode). Had the DAC presented in [51] been augmented with ISI scrambling, it is likely that it could have used NRZ 1-bit DACs and thereby avoided the downsides of RZ 1-bit DACs without sacrificing linearity. The DAC presented in [50] uses NRZ 1-bit DACs and no ISI-mitigation techniques are mentioned in [50]. This suggests that excellent design and layout practices as well as the advanced 16 nm CMOS IC technology in which it is implemented are likely responsible for keeping its nonlinear ISI in check. Yet the analysis in Section IV shows that its NRZ 1-bit DACs must be introducing significant nonlinear ISI, so it is reasonable to expect that it too could have benefited from ISI scrambling.

VI. Appendix A

Substituting (48) into the right-most equation in (46) and substituting the result into (54) yields

$$\begin{aligned} e_{3,i}(t) = {} & o_{++,i}[n_t]e'_i(t) \\ & + o_{+\times,i}[n_t]\left[\beta_{+\times,i}(t) + x_i[n_t-1]p_i[n_t-1]\gamma_{+\times,i}(t)\right] \\ & + o_{\times+,i}[n_t]\left[\beta_{\times+,i}(t) + x_i[n_t]p_i[n_t]\varepsilon_{\times+,i}(t)\Delta\right] \\ & + o_{\times\times,i}[n_t]e_{\times\times i}(t), \end{aligned} \tag{77}$$

where

$$\beta_{+\times,i}(t) = \frac{1}{2}\left[e_{1\times i}(t) + e_{0\times i}(t)\right], \quad \beta_{\times+,i}(t) = \frac{1}{2}\left[e_{\times 1i}(t) + e_{\times 0i}(t)\right], \tag{78}$$

$$\gamma_{+\times,i}(t) = e_{1\times i}(t) - e_{0\times i}(t), \text{ and } \varepsilon_{\times+,i}(t) = \frac{1}{\Delta}\left[e_{\times 1i}(t) - e_{\times 0i}(t)\right]. \tag{79}$$

As explained in Section II, $p_i[n_t]$ only changes when the ISI scrambling 1-bit DAC has been offline for multiple clock cycles. This implies that $p_i[n_t] = p_i[n_t-1]$ whenever $o_{\times\times,i}[n] = 0$. In such cases, both $p_i[n_t]p_i[n_t-1] = 1$ and $p_i^2[n_t] = 1$, because $p_i[n_t]$ takes on values of only 1 and −1. Consequently, (77) implies

$$\begin{aligned} p_i[n_t]e_{3,i}(t) = {} & o_{++,i}[n_t]p_i[n_t]e'_i(t) \\ & + o_{+\times,i}[n_t]\left[p_i[n_t]\beta_{+\times,i}(t) + x_i[n_t-1]\gamma_{+\times,i}(t)\right] \\ & + o_{\times+,i}[n_t]\left[p_i[n_t]\beta_{\times+,i}(t) + x_i[n_t]\varepsilon_{\times+,i}(t)\Delta\right] \\ & + o_{\times\times,i}[n_t]p_i[n_t]e_{\times\times i}(t). \end{aligned} \tag{80}$$

As explained in Section IV-A, $o_{+\times,i}[n_t] = o_{\times+,i}[n_t] = o_{\times\times,i}[n_t] = 0$ when $o_{++,i}[n_t] = 1$. Thus, (47), (49), and (80) imply that the output of the ith ISI scrambling 1-bit DAC

is

$$y_i(t) = x_i[n_t]K_i\Delta + p_i[n_t]e'_i(t), \tag{81}$$

when $o_{++,i}[n_t] = 1$. Performing an analysis nearly identical to that applied in [53] to derive (44) from (42) and (43), and applying $p_i[n_t]p_i[n_t-1] = 1$ and $p_i^2[n_t] = 1$ results in

$$y_i(t) = x_i[n_t]\alpha_i(t)K_i\Delta + e_{++,i}(t), \tag{82}$$

when $o_{++,i}[n_t] = 1$, where

$$\begin{aligned} e_{++,i}(t) = {} & p_i[n_t]\beta_i(t) + x_i[n_t-1]\gamma_i(t) \\ & + p_i[n_t]x_i[n_t]x_i[n_t-1]\eta_i(t). \end{aligned} \tag{83}$$

Given that one of $o_{++,i}[n_t]$, $o_{+\times,i}[n_t]$, $o_{\times+,i}[n_t]$, and $o_{\times\times,i}[n_t]$ is 1 and the rest are 0 during each clock cycle and $o_i[n_t] = 1$ when $o_{++,i}[n_t] = 1$, it follows from (47), (49), (80), and (82) that

$$\begin{aligned} y_i(t) = {} & x_i[n_t]o_i[n_t]K_i\Delta \\ & + o_{++,i}[n_t]\left[x_i[n_t]\varepsilon_i(t)\Delta + e_{++,i}(t)\right] \\ & + o_{+\times,i}[n_t]\left[p_i[n_t]\beta_{+\times,i}(t) + x_i[n_t-1]\gamma_{+\times,i}(t)\right] \\ & + o_{\times+,i}[n_t]\left[p_i[n_t]\beta_{\times+,i}(t) + x_i[n_t]\varepsilon_{\times+,i}(t)\Delta\right] \\ & + o_{\times\times,i}[n_t]p_i[n_t]e_{\times\times i}(t), \end{aligned} \tag{84}$$

where

$$\varepsilon_i(t) = K_i(\alpha_i(t) - 1). \tag{85}$$

The random sequences $o_{++,i}[n_t]$, $o_{+\times,i}[n_t]$, $o_{\times+,i}[n_t]$, and $o_{\times\times,i}[n_t]$ can each be written as

$$o_{**,i}[n] = \tilde{o}_{**,i}[n] + \bar{o}_{**,i}, \tag{86}$$

where $**$ is a placeholder for $++$, $+\times$, $\times+$, or $\times\times$, $\tilde{o}_{**,i}[n]$ is the zero-mean portion of $o_{**,i}[n]$,

and $\bar{o}_{**,i}$ is the mean of $o_{**,i}[n]$.

The DEM encoder causes $m_i = 0$ for $1 \le i \le 20$ and $m_i = 2^{-14}$ for $21 \le i \le 36$ [47]. Therefore, substituting (84) with (86) and (55) into (58), simplifying the result with (57), and applying $o_i[n_t] = 1$ for all $i = 1, 2, \ldots, 20$ results in (60), (63), (64), and (66)-(69), where

$$\alpha(t) = 1 + 2^{-14} \sum_{i=21}^{37} \left(\bar{o}_{++,i} \varepsilon_i(t) + \bar{o}_{\times+,i} \varepsilon_{\times+,i}(t) \right), \tag{87}$$

$$\beta(t) = \sum_{i=1}^{20} \beta_i(t), \tag{88}$$

and

$$\gamma(t) = \frac{2^{-14}}{\Delta} \sum_{i=21}^{37} \left(\bar{o}_{++,i} \gamma_i(t) + \bar{o}_{+\times,i} \gamma_{+\times,i}(t) \right). \tag{89}$$

VII. Appendix B

The autocorrelation of $q_i[n]$ for $i = 21, 22, \ldots, 37$ is derived in this appendix using a Markov chain to model the states of each ISI-scrambling 1-bit DAC. An analysis is first presented that applies to the primary 1-bit DACs, i.e., to $q_i[n]$ for $i = 21, 22, \ldots, 36$. Then an analysis is presented that applies to the auxiliary 1-bit DAC, i.e., to $q_i[n]$ for $i = 37$.

Fig. 24 shows a Markov chain state diagram that applies to the ith ISI-scrambling 1-bit DAC for the example case of $N_{Delay} = 4$, where $i = 21, 22, \ldots, 36$. The example uses $N_{Delay} = 4$ to simplify the figure and its explanation, but the results derived below apply to the general case.

Each of the states in Fig. 24 with a single output state transition probability of 1 implements a one clock delay wait period. For example, States 10-17 implement the 8 clock delays during which the *i*th 1-bit DAC is offline and its transients are changed from not swapped to swapped.

The hashmarks in Fig. 24 denote states in which the *i*th 1-bit DAC's transients are swapped. For example, the 1-bit DAC's transients are swapped in the second half of States 10-17 as indicated by the hashmarks on the symbols for States 14-17.

As explained in Section II, whenever a primary ISI-scrambling 1-bit DAC goes from its offline to online state, all of the primary ISI-scrambling 1-bit DACs remain online for X clock periods where X is a random variable that takes on integer values from 1 to N_{Delay}, each with probability $q = 1/N_{Delay}$. Hence, if the *i*th 1-bit DAC is in State 1 at time index n, then the probability that one of the primary ISI-scrambling 1-bit DACs will be taken offline at time index n+1 is q. As there are 16 such 1-bit DACs, the probability that the *i*th 1-bit DAC will be taken offline at time index n+1 is q/16, and the probability that one of the other 1-bit DACs will be taken offline at time index n+1 is 15q/16. These two cases correspond States 10 and 2, respectively, in Fig. 24. If none of the primary 1-bit DACs are taken offline for another 1, 2, or 3 clock cycles, then the *i*th 1-bit DAC enters State 38, 37, or 36, respectively. Similar reasoning can be applied to each of the subsequent states to verify that the Markov chain correctly models the states of each primary ISI-scrambling 1-bit DAC per the explanation in Section II.

The number of Markov chain states, N_{States}, depends on N_{Delay}, which can be set

via the SPI and can be as large as 65536. The example described above uses $N_{Delay} = 4$ which resulted in $N_{States} = 42$. In general, $N_{States} = 34+2N_{Delay}$ because the consecutive clock periods during which none of the primary 1-bit DACs are offline are represented by $2N_{Delay}+2$ possible states. Specifically, these states are States 1, 18, *C*-*D*, and *E*-*F*, where

$$\begin{aligned} C &= 35, \qquad & D &= 35 + N_{Delay} - 1, \\ E &= 35 + N_{Delay}, \qquad & F &= 35 + 2N_{Delay} - 1. \end{aligned} \tag{90}$$

It follows from (50), (71), and the Markov chain's definition that

$$q_i[n] = \begin{cases} 1, & \text{if } s_i[n] \in S_1, \\ -1, & \text{if } s_i[n] \in S_{-1}, \\ 0, & \text{otherwise,} \end{cases} \tag{91}$$

where $s_i[n]$ is the state of the Markov chain during the nth clock cycle, $S_1 = \{2, 3, \ldots, 9, C, C+1, \ldots, D\}$, and $S_{-1} = \{19, 20, \ldots, 26, E, E+1, \ldots, F\}$. Therefore,

$$\begin{aligned} E\{q_i[n]q_i[n+k]\} = &\sum_{u \in S_1} \sum_{v \in S_1} \Pr\left(s_i[n] = u,\, s_i[n+k] = v\right) \\ &+ \sum_{u \in S_{-1}} \sum_{v \in S_{-1}} \Pr\left(s_i[n] = u,\, s_i[n+k] = v\right) \\ &- \sum_{u \in S_1} \sum_{v \in S_{-1}} \Pr\left(s_i[n] = u,\, s_i[n+k] = v\right) \\ &- \sum_{u \in S_{-1}} \sum_{v \in S_1} \Pr\left(s_i[n] = u,\, s_i[n+k] = v\right), \end{aligned} \tag{92}$$

where $\Pr(s_i[n] = u, s_i[n+k] = v)$ is the joint probability that the Markov chain states at time indices n and $n+k$ are u and v, respectively.

The values of $\Pr(s_i[n] = u, s_i[n+k] = v)$ can be derived via the Markov chain's $N_{States} \times N_{States}$ state transition matrix, $\mathbf{P} = [p_{u,v}]$, where $p_{u,v}$ is the element on the uth row and vth column of $\mathbf{P}$ and is the probability that the Markov chain will be in State v

at any time index m given that it was in State u at time index $m-1$. Although **P** has a relatively large dimension, most of its elements are zero. Specifically, the Markov chain's definition implies that the only non-zero elements of **P** are:

$$\begin{gathered} p_{34,1} = p_{m,\,m+1} = 1 \text{ for all } m \notin \{1,9,18,26,34,D,E\}, \\ p_{1,\,2} = p_{18,\,19} = 15q/16,\ p_{1,10} = p_{18,27} = q/16, \\ p_{D,\,2} = p_{F,\,19} = 15/16,\ p_{D,10} = p_{F,27} = 1/16, \\ p_{9,\,C+m} = p_{26,\,E+m} = q, \quad \text{for } 0 \le m < N_{Delay}, \\ p_{1,\,C+m} = p_{18,\,E+m} = q, \quad \text{for } 1 \le m < N_{Delay}. \end{gathered} \tag{93}$$

The Markov chain's definition further implies that there is a non-zero probability of the system successively entering States 1-9, *C*-*D*, 10-26, *E*-*F*, 27-34, 1, which is a loop that includes all N_{States} states. This implies that every state can be reached from every other state, so the Markov chain satisfies the definition of being *irreducible*.

By the same reasoning, regardless of the system's state at any given time index n, there is a non-zero probability that it will return to the same state at time index $n+m$, where $m = N_{States}$. The Markov chain's definition also implies that there is a non-zero probability of the system successively entering the states in the order listed above but skipping either State *C* or State *E*. This implies that regardless of the system's state at any given time index n, there is a non-zero probability that it will return to the same state at time index $n+m$, where $m = N_{States}-1$. Given that the greatest common divisor of N_{States} and $N_{States}-1$ is 1 regardless of the value to which N_{States} is set, this implies that the Markov chain satisfies the definition of being *aperiodic*.

As the Markov chain is irreducible, aperiodic, and has a finite number of states, it approaches a steady state in that the sequence of its states' probability

distributions as a function of time index n converges to a unique steady-state probability distribution as $n \to \infty$ [76]. This implies that

$$\begin{bmatrix} \pi_1 & \pi_2 & \cdots & \pi_{N_{States}} \end{bmatrix} = \begin{bmatrix} \pi_1 & \pi_2 & \cdots & \pi_{N_{States}} \end{bmatrix} \mathbf{P} \tag{94}$$

where π_u is the steady-state probability of the system being in State u. This matrix equation along with the probability distribution property that $\pi_1 + \pi_2 + \cdots + \pi_{N_{States}} = 1$ can be solved to find the values of $\pi_1, \pi_2, \cdots \pi_{N_{States}}$.

As the objective of this appendix is to derive the steady-state autocorrelation of $q_i[n]$, the Markov chain is taken to have converged to its steady state in the following analysis, i.e.,

$$\Pr\left(s_i[n] = u\right) = \pi_u \quad \text{for all } n. \tag{95}$$

Therefore,

$$\Pr\left\{s_i[n] = u,\ s_i[n+k] = v\right\} = \pi_u p_{u,v}(k) \tag{96}$$

where $p_{u,v}(k)$ is the probability that the Markov chain's state at time index $n+k$ is v given that it was in State u at time index n. Substituting this into (92) leads to an expression for $\mathrm{E}\{q_i[n]q_i[n+k]\}$ that is independent of n. This implies that $\mathrm{E}\{q_i[n]q_i[n+k]\} = \mathrm{E}\{q_i[n]q_i[n-k]\}$. Furthermore, the definition of $q_i[n]$ implies that $\mathrm{E}\{q_i[n]\} = 0$, so the mean of $q_i[n]$ is also independent of n. It follows that $q_i[n]$ is WSS. Substituting (96) into (92) and applying these observations implies that the autocorrelation of $q_i[n]$ can be written as

$$R_{q,i}[k] = \sum_{u \in S_1} \pi_u \left[\sum_{v \in S_1} p_{u,v}\left(|k|\right) - \sum_{w \in S_{-1}} p_{u,w}\left(|k|\right) \right] + \sum_{r \in S_{-1}} \pi_r \left[\sum_{s \in S_{-1}} p_{r,s}\left(|k|\right) - \sum_{t \in S_1} p_{r,t}\left(|k|\right) \right]. \tag{97}$$

The properties of Markov chains imply that

$$\left[p_{u,1}(k) \quad p_{u,2}(k) \quad \ldots \quad p_{u,N_{States}}(k) \right] = \mathbf{b}_u \mathbf{P}^k, \tag{98}$$

where $\mathbf{b}_u$ is a row vector of length N_{States} with a 1 in column u and zeros in all other columns [76]. Matrix equation (98) with the definition of $\mathbf{P}$ via (93) is used to calculate the values of $p_{u,v}(k)$ in (97).

The above analysis applies only to the primary ISI-scrambling 1-bit DACs, but it can be modified to apply to the auxiliary ISI-scrambling 1-bit DAC as follows. The properties of the $o_{37}[n]$ and $p_{37}[n]$ stochastic sequences described in Section II imply that the auxiliary ISI-scrambling 1-bit DAC's states can be modeled via a Markov chain with an $N_{States} \times N_{States}$ state transition matrix, $\mathbf{P} = [p_{u,v}]$, that is similar to that described above for the primary ISI-scrambling 1-bit DACs. The only non-zero elements of the state transition matrix are still given by (93) except with $p_{1,2} = p_{18,19} = p_{1,10} = p_{18,27} = q/2$ and $p_{D,2} = p_{F,19} = p_{D,10} = p_{F,27} = 1/2$ because of the 50% chance that the swap state of the auxiliary ISI-scrambling 1-bit DAC gets inverted each time it goes offline.

By the same reasoning applied previously, $\mathbf{P}$ is irreducible and aperiodic, and $q_{37}[n]$ is given by (91) with $i = 37$, $S_1 = \{3, 4, \ldots, 9, 28, 29, \ldots, 34\}$, and $S_{-1} = \{11, 12, \ldots, 17, 20, 21, \ldots, 26\}$. Consequently, $R_{q,37}[k]$ is given by (97) with probability

distributions calculated via (94) and (98) using the version of **P** corresponding to the auxiliary ISI-scrambling 1-bit DAC.

VIII. APPENDIX C

Suppose $x[n_t] = 8192\Delta\sin(\omega_0 n_t)$, where $\omega_0 = 2\pi f_0/f_s$, and f_0/f_s satisfies (75). As $e_i(t)$ has zero mean and $x[n_t]$ is deterministic, the autocorrelation of $e_i(t)$ can be written as

$$R_{e,i}(t,\tau) = g_i(t)g_i(t+\tau)\,\mathrm{E}\{q_i[n_t]q_i[n_{t+\tau}]\}, \tag{99}$$

with

$$g_i(t) = 2^{-2}\eta_i(t)\sin(\omega_0 n_t)\sin(\omega_0(n_t - 1)). \tag{100}$$

The DAC's input sequence is generated digitally, so f_0/f_s is a rational number. This implies that $g_i(t)$ is periodic with a period, T_g, where T_g is an integer multiple of T_s. As $\eta_i(t)$ is periodic with a period of T_s by definition, this implies that $g_i(t)g_i(t+\tau)$ is periodic in t with a period, T_g.

As proven in Appendix B, the discrete-time stochastic sequence, $q_i[n]$, is WSS, so the expectation term in (99) can be written as

$$\mathrm{E}\{q_i[n_t]q_i[n_{t+\tau}]\} = R_{q,i}[n_{t+\tau} - n_t], \tag{101}$$

the right side of which is given by (97) with k replaced by $n_{t+\tau} - n_t$. The quantity $n_{t+\tau}$ can be written as $n_{t+\tau} = \lfloor n_t + \langle f_s t\rangle + n_\tau + \langle f_s\tau\rangle\rfloor$ where, for any real number x, $\lfloor x\rfloor$ denotes the largest integer less than or equal to x, and $\langle x\rangle = x - \lfloor x\rfloor$ is the fractional part of x. For any real numbers x and y, $\lfloor x\rfloor - \lfloor y\rfloor$ can be written as $\lfloor x - \lfloor y\rfloor\rfloor$, so

$$n_{t+\tau} - n_t = \left\lfloor \langle f_s t \rangle + n_\tau + \langle f_s \tau \rangle \right\rfloor . \tag{102}$$

The only term that contains t on right side of (102) is $\langle f_s t \rangle$, which is T_s-periodic, so (101) must be T_s-periodic in t. Given $g_i(t)g_i(t+\tau)$ is periodic in t with a period, T_g, and T_g is an integer multiple of T_s, it follows from (99) and (101) that $R_{e,i}(t, \tau)$ is periodic in t with a period of T_g. Consequently, $e_i(t)$ is a cyclostationary random process and its average PSD is the Fourier transform of the average of $R_{e,i}(t,\tau)$ over one T_g period of t, i.e., the Fourier transform of

$$\bar{R}_{e,i}(\tau) = \frac{1}{T_g} \int_0^{T_g} g_i(t) g_i(t+\tau) R_{q,i}[n_{t+\tau} - n_t] dt . \tag{103}$$

Substituting (100) into (103) and applying sinusoid product-to-sum identities shows that the integrand of (103) can be written as

$$\begin{aligned} f_\tau(t) \Big[& 2\cos^2(\omega_0) \\ & + \cos\left(2\omega_0 \left(n_{t+\tau} - n_t\right)\right) + \cos\left(2\omega_0 \left(n_{t+\tau} + n_t - 1\right)\right) \\ & - 2\cos(\omega_0)\cos\left(2\omega_0 n_t - \omega_0\right) \\ & - 2\cos(\omega_0)\cos\left(2\omega_0 n_{t+\tau} - \omega_0\right) \Big], \end{aligned} \tag{104}$$

where

$$f_\tau(t) = 2^{-7} R_{q,i}\left[n_{t+\tau} - n_t\right] \eta_i(t) \eta_i(t+\tau) . \tag{105}$$

Both $\cos(2\omega_0 n_t - \omega_0)$ and $\cos(2\omega_0 n_{t+\tau} + \omega_0)$ are constant with respect to t over successive intervals of T_s. Furthermore, they average to zero over intervals of T_g in t because the restrictions on ω_0 imposed by (75) prevent them from aliasing to zero frequency. By the same reasoning, $\cos(2\omega_0(n_{t+\tau} + n_t - 1))$ averages to zero over intervals of T_g in t, and it is constant with respect to t over two fixed sub-intervals of every successive T_s time

interval. Given that $f_\tau(t)$ is T_s-periodic in t, the above observations imply that the last three terms on the right side of (104) average to zero in (103), so they do not contribute to $\bar{R}_{e,i}(\tau)$.

As explained above, (102) and, hence, $\cos(2\omega_0(n_{t+\tau}-n_t))$ are T_s-periodic in t. Given that $f_\tau(t)$ is T_s-periodic in t and $\cos^2(\omega_0)$ is constant, it follows that all the terms in (104) which contribute to $\bar{R}_{e,i}(\tau)$ are T_s-periodic in t, so (103) can be written as

$$\bar{R}_{e,i}(\tau) = \frac{1}{T_s}\int_0^{T_s} f_\tau(t)\left[2\cos^2(\omega_0) + \cos\left(2\omega_0\left(n_{t+\tau} - n_t\right)\right)\right]dt\,. \tag{106}$$

Equation (102) implies that

$$n_{t+\tau} - n_t = \begin{cases} n_\tau, & \text{if } \langle f_s t\rangle < 1 - \langle f_s \tau\rangle, \\ n_\tau + 1, & \text{otherwise,} \end{cases} \tag{107}$$

so (106) with (105) can be rewritten as

$$\begin{aligned} \bar{R}_{e,i}(\tau) = w_i[n_\tau]&\int_0^{T_s(1-\langle f_s\tau\rangle)} \eta_i(t)\eta_i(t+\tau)dt \\ &+ w_i[n_\tau+1]\int_{T_s(1-\langle f_s\tau\rangle)}^{T_s} \eta_i(t)\eta_i(t+\tau)dt, \end{aligned} \tag{108}$$

where

$$w_i[n_\tau] = \frac{2^{-7}R_{q,i}[n_\tau]}{T_s}\left[2\cos^2(\omega_0) + \cos\left(2\omega_0 n_\tau\right)\right]. \tag{109}$$

As $\eta_i(t)$ is T_s-periodic, the $\eta_i(t+\tau)$ integrand factors in the first and second integrals on the right side of (108) can be replaced by $\eta_i(t+\tau-n_\tau T_s)$ and $\eta_i(t+\tau-(n_\tau+1)T_s)$, respectively, without changing $\bar{R}_{e,i}(\tau)$. By definition, $\langle f_s\tau\rangle = f_s\tau - n_\tau$, so these replacements can be written as $\eta_i(t+T_s\langle f_s\tau\rangle)$ and $\eta_i(t-T_s(1-\langle f_s\tau\rangle))$, respectively. Hence, with these replacements, both integrals have integrands of the form $\eta_i(t)\eta_i(u)$ and the limits of integration

are such that t and u are limited to and, together, span the range [0, T_s]. Therefore, (108) can be rewritten as

$$\begin{aligned}\bar{R}_{e,i}(\tau) &= w_i[n_\tau]\eta_{p^*,i}\left(\tau - n_\tau T_s\right) \\ &\quad + w_i[n_\tau + 1]\eta_{p^*,i}\left(\tau - \left(n_\tau + 1\right)T_s\right),\end{aligned} \tag{110}$$

where

$$\eta_{p^*,i}(u) = \int_{-\infty}^{\infty} \eta_{p,i}(t)\eta_{p,i}\left(t + u\right)dt \tag{111}$$

and

$$\eta_{p,i}(t) = \begin{cases} \eta_i(t) & \text{if } 0 \le t < T_s, \\ 0 & \text{otherwise.} \end{cases} \tag{112}$$

By definition, $\eta_{p^*,i}(\tau)$ is nonzero only when $-T_s < \tau < T_s$, so (110) can be re-written as

$$\bar{R}_{e,i}(\tau) = \sum_{n=-\infty}^{\infty} w_i[n]\eta_{p^*,i}\left(\tau - nT_s\right). \tag{113}$$

Taking the CTFT of (113) yields

$$\begin{aligned}S_{e,i}\left(j\omega\right) &= H_{p^*,i}\left(j\omega\right)\sum_{n=-\infty}^{\infty} w_i[n]e^{-j\omega nT_s} \\ &= H_{p^*,i}\left(j\omega\right)W_i\left(e^{j\omega T_s}\right),\end{aligned} \tag{114}$$

where $H_{p^*,i}(j\omega)$ is the CTFT of $n_{p^*,i}(t)$ and $W_i(e^{j\omega Ts})$ is the DTFT of $w_i[n]$. Taking the DTFT of (109) with n_t replaced by n yields

$$\begin{aligned}W_i\left(e^{j\omega T_s}\right) = \frac{2^{-8}}{T_s}\Big[&4\cos^2\left(\omega_0\right)S_{q,i}\left(e^{j\omega T_s}\right) \\ &+ S_{q,i}\left(e^{j(\omega+2\omega_0)T_s}\right) + S_{q,i}\left(e^{j(\omega-2\omega_0)T_s}\right)\Big]\end{aligned} \tag{115}$$

where $S_{q,i}(e^{j\omega Ts})$ is the DTFT of $R_{q,i}[k]$ which is given by (97).

ACKNOWLEDGEMENTS

Chapter 2, in full, has been submitted for review to the IEEE Journal of Solid-State Circuits by Remple, Jason; Panigada, Andrea; Galton, Ian. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis of this paper.

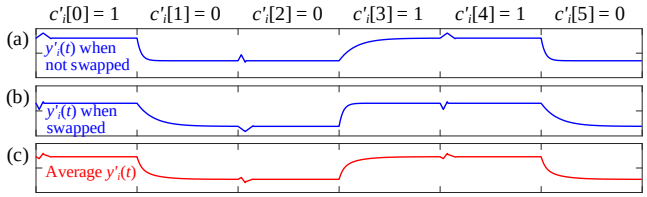

Figure 13 : Example 1-bit DAC outputs with transient errors: (a) not swapped, (b) swapped, and (c) averaged.

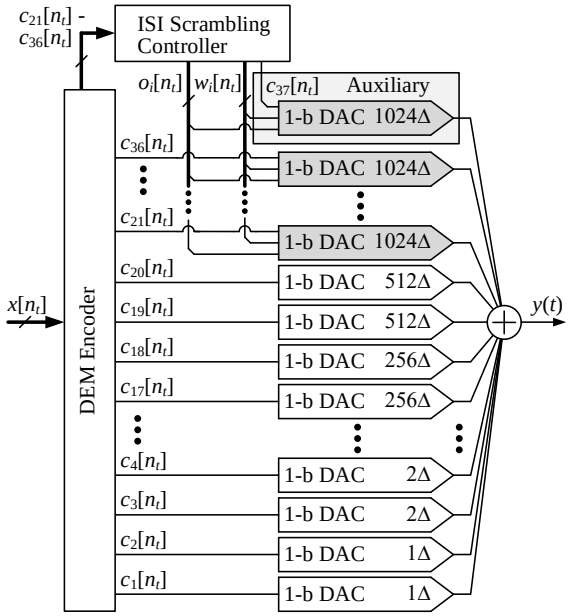

Figure 14 : ISI scrambling DEM DAC block diagram.

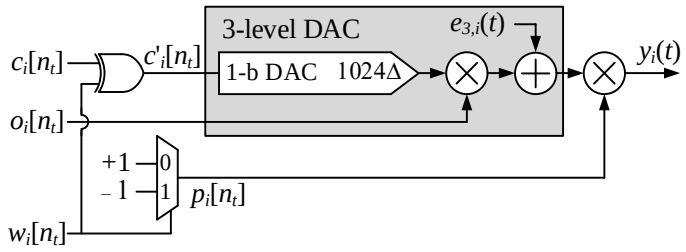

Figure 15 : ISI scrambling 1-bit DAC block diagram.

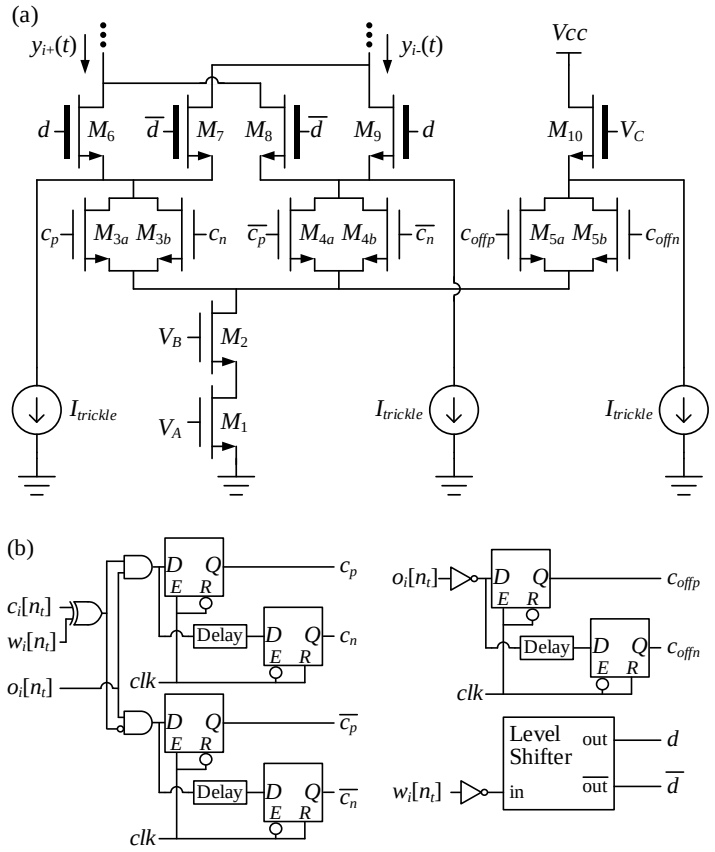

Figure 16 : ISI scrambling 1-bit DAC implementation, (a) current-steering cell, (b) switch-driver.

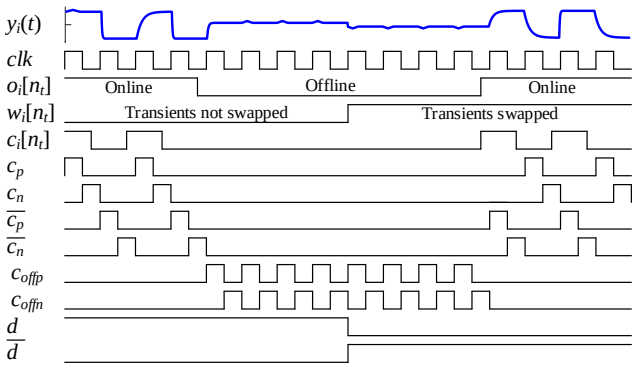

Figure 17 : Switch driver timing diagram.

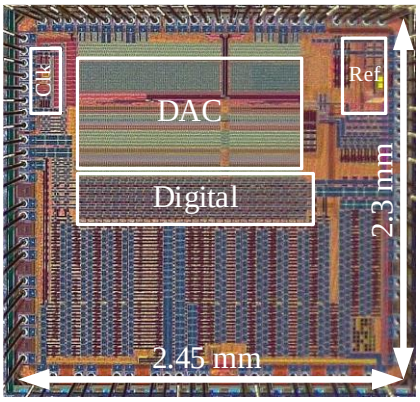

Figure 18 : Die photograph.

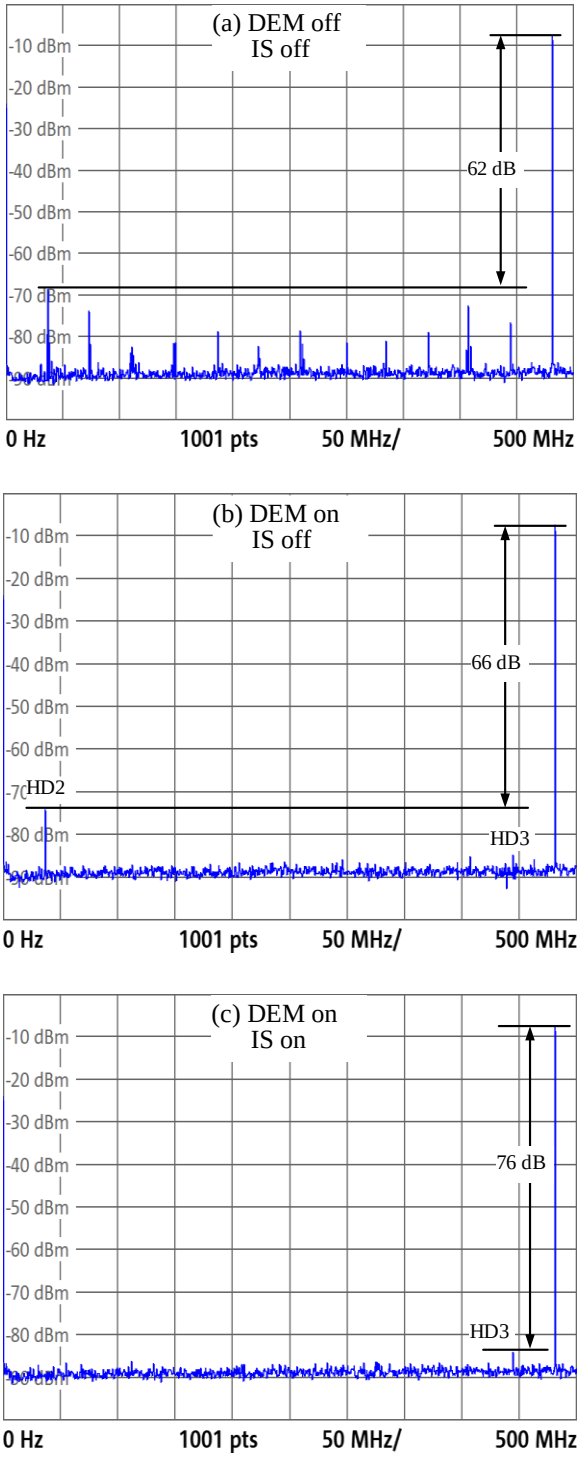

Figure 19 : Measured output power spectra for a full-scale 481.4 MHz input signal.

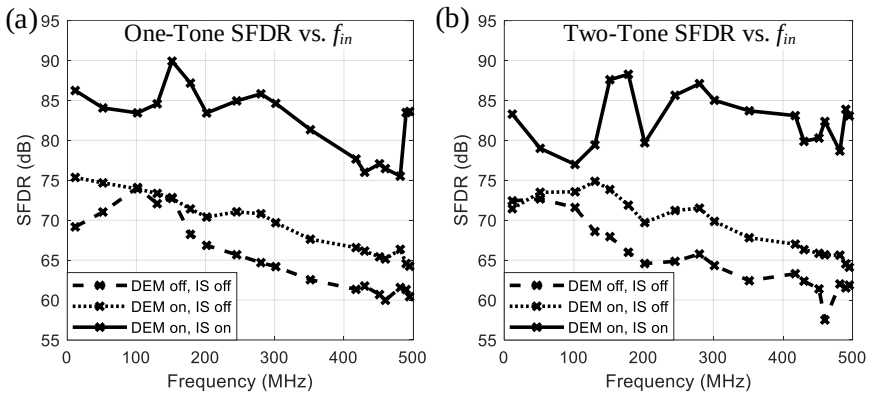

Figure 20 : Measured SFDR vs. frequency (a) one-tone input signals, and (b) two-tones input signals separated by 3.52 MHz.

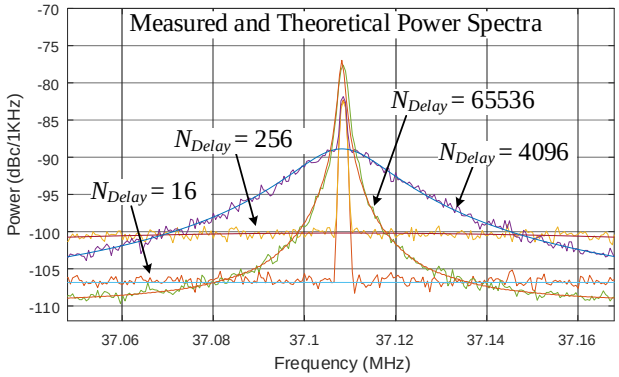

Figure 21 : Measured and theoretical power spectra around aliased second harmonic for full-scale 481.4 MHz single-tone input signal.

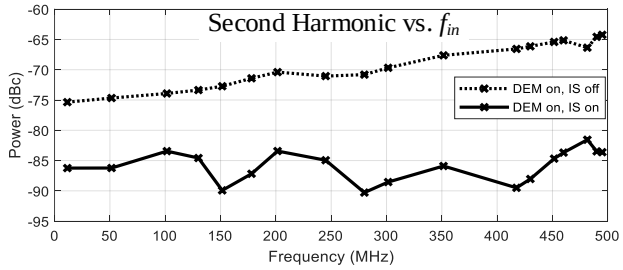

Figure 22 : Measured second harmonic versus frequency for a full-scale single tone input signal.

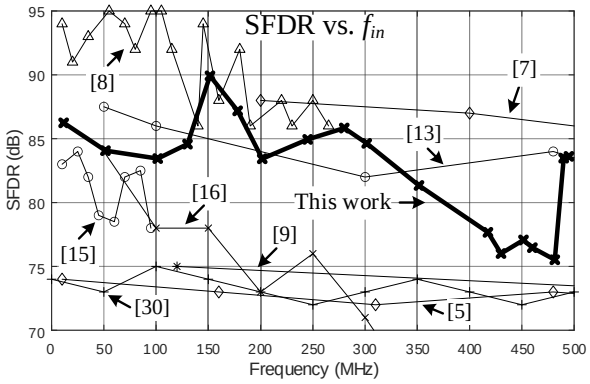

Figure 23 : SFDR comparison of recent state-of-the-art DACs.

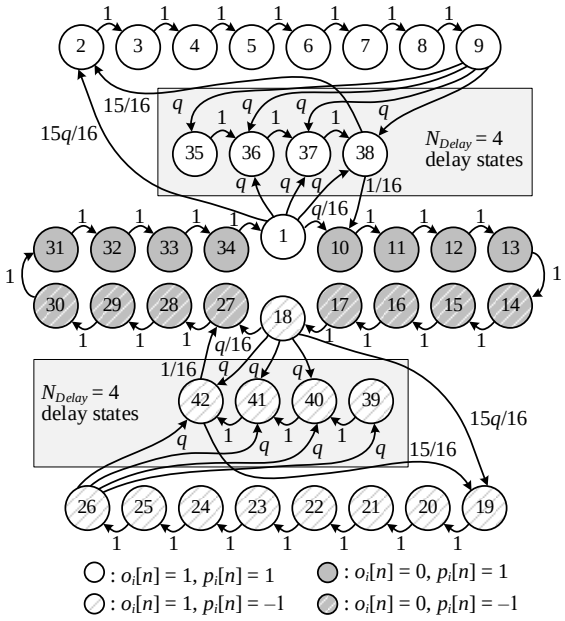

Figure 24 : Markov state transition diagram corresponding to the primary ISI scrambling 1-bit DACs.

TABLES

Table 1 NSD/SNDR measured over first Nyquist band.

f_{in} (MHz)	NSD (dBc/Hz)			SNDR (dB)		
	DEM off IS off	DEM on IS off	DEM on IS on	DEM off IS off	DEM on IS off	DEM on IS on
51	–163.4	–147.1	–146.3	63.6	59.9	59.4
180	–158.1	–145.8	–145.2	62.4	58.6	58.3
481	–150.8	–144.8	–144.8	60.3	57.8	57.8

Table 2 Key specifications of recent state-of-the-art DACs.

	Process (nm)	Resolution (bits)	Sample Rate (GHz)	Full Scale (mA)	Power (mW)	Technique
This Work	90	14	1000	20	238*	DEM / IS
[5]	40	16	1600	16/20	40	DEMDRZ
[7]	16	16	6000	40	350	Static Cal
[8]	22	14	600	16	202	MNC
[9]	28	14	10000	16	162	OIC
[13]	65	16	9000	16	1080	DPD
[15]	140	14	200	20	270	DMM
[16]	65	16	3200	20	240	3DSC
[30]	90	12	1250	16	128	DRRZ

REFERENCES

44. R. van de Plassche, "Dynamic element matching for high-accuracy monolithic D/A converters," *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 1976, pp. 148-149.

45. K. O. Sullivan, C. Gorman, M. Hennessy and V. Callaghan, "A 12-bit 320-MSample/s current-steering CMOS D/A converter in 0.44 mm^2," *IEEE J. of Solid-State Circuits*, vol. 39, no. 7, pp. 1064-1072, July 2004.

46. B. Jewett, J. Liu and K. Poulton, "A 1.2 Gs/s 15b DAC for precision signal generation," *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2005, pp. 110-111.

47. K. L. Chan, J. Zhu and I. Galton, "Dynamic element matching to prevent nonlinear distortion from pulse-shape mismatches in high-resolution DACs," *IEEE J. of Solid-State Circuits*, vol. 43, no. 9, pp. 2067-2078, September 2008.

48. W.-T. Lin, H.-Y. Huang and T.-H. Kuo, "A 12-bit 40 nm DAC achieving SFDR > 70 dB at 1.6 GS/s and IMD < -61dB at 2.8 GS/s with DEMDRZ technique," *IEEE J. of Solid-State Circuits*, Vol. 49, no. 3, pp. 708-717, March 2014.

49. W. Mao, Y. Li, C. Heng and Y. Lian, "High dynamic performance current-steering DAC design with nested-segment structure," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 26, no. 5, pp. 995-999, May 2018.

50. C.-H. Lin, K. L. J. Wong, T.-Y. Kim, G. R. Xie, D. Major, G. Unruh, S. R. Dommaraju, H. Eberhart and A. Venes, "A 16b 6 GS/s Nyquist DAC with IMD < -90dBc up to 1.9 GHz in 16 nm CMOS," *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2018, pp. 360 – 362.

51. D. Kong, K. Rivas-Rivera and I. Galton, "A 600 MS/s DAC with over 87- dB SFDR and 77-dB peak SNDR enabled by adaptive cancellation of static and dynamic mismatch error," *IEEE J. of Solid-State Circuits*, vol. 54, no. 8, pp. 2219-2229, August 2019.

52. H.-Y. Huang and T.-H. Kuo, "A 0.07-mm^2 162-mW DAC achieving >65 dBC

SFDR and < –70 dBc IM3 at 10 GS/s with output impedance compensation and concentric parallelogram routing," *IEEE J. of Solid-State Circuits*, vol. 55, no. 9, pp. 2478-2488, September 2020.

53. J. Remple and I. Galton, "The effects of inter-symbol interference in dynamic element matching DACs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 1, pp. 14-23, Jan. 2017.

54. T. He, M. Ashburn, S. Ho, Y. Zhang and G. Temes, "A 50MHZ-BW continuous-time ΔΣ ADC with dynamic error correction achieving 79.8dB SNDR and 95.2dB SFDR," *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2018, pp. 230-232.

55. S. Su and M. S.-W. Chen, "A 12-bit 2 GS/s dual-rate hybrid DAC with pulse-error pre-distortion and in-band noise cancellation achieving > 74 dBc SFDR and <–80 dBc IM3 up to 1 GHz in 65 nm CMOS," *IEEE J. of Solid-State Circuits*, vol. 51, no. 12, pp. 2963-2978, December, 2016.

56. S. Su and M. S.-W. Chen, "A 16-bit 12-GS/s single-/dual-rate DAC with a successive bandpass delta-sigma modulator achieving <−67-dBc IM3 within DC to 6-GHz tunable passbands," *IEEE J. of Solid-State Circuits*, vol. 53, no. 12, pp. 3517-3527, Dec. 2018.

57. D. Basak, S. Kalani, P. Kinget and K. Pun, "An on-chip static and dynamic DAC error correction technique for high speed multibit delta-sigma modulators," *IEEE Int. Symposium on Circuits and Systems (ISCAS)*, Florence, 2018, pp. 1-5.

58. Y. Tang, J. Briaire, K. Doris, R. van Veldhoven, P. C. W. van Beek, H. J. A. Hegt and A. H. M. van Roermund, "A 14-bit 200 MS/s DAC with SFDR > 78 dBc, IM3 < –83 dBc and NSD < –163 dBm/Hz across the whole nyquist band enabled by dynamic-mismatch mapping", *IEEE J. of Solid-State Circuits*, vol. 46, no. 6, pp. 1371-1381, June 2011.

59. H. Van de Vel, J. Briaire, C. Bastiaansen, P. van Beek, G. Geelen, H. Gunnink, Y. Jin, M. Kaba, K. Luo, E. Paulus, B. Pham, W. Relyveld and P. Zijlstra, "A 240mW 16b 3.2GS/s DAC in 65nm CMOS with <–80dBc IM3 up to 600 MHz," *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2014, pp. 206-

208.

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

61. L. Risbo, R. Hezar, B. Kelleci, H. Kiper and M. Fares, “Digital approaches to ISI-mitigation in high-resolution oversampled multi-level D/A converters,” *IEEE J. of Solid-State Circuits*, vol. 46, no. 12, pp. 2892-2903, December 2011.

62. A. Sanyal, P. Wang and N. Sun, "A thermometer-like mismatch shaping technique with minimum element transition activity for multibit ΔΣ DACs," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 7, pp. 461-465, July 2014.

63. A. Sanyal, L. Chen and N. Sun, “Dynamic element matching with signal-independent element transition rates for multibit ΔΣ modulators,” *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 62, no. 5, pp. 1325-1334, May 2015.

64. I. Galton, “Why Dynamic-Element-Matching DACs Work,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 2, pp. 69–74, Feb. 2010.

65. L. Lai, X. Li, Y. Fu, Y. Liu and H. Yang, "Demystifying and Mitigating Code-Dependent Switching Distortions in Current-Steering DACs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 68-81, January 2019.

66. A. Van den Bosch, M. A. F. Borremans, M. S. J. Steyaert and W. Sansen, “A 10-bit 1-GSample/s Nyquist current-steering CMOS D/A converter,” *IEEE J. of Solid-State Circuits,* vol. 36, no. 3, pp. 315-324, March 2001.

67. W. Schofield, D. Mercer and L. Onge, “A 16b 400MS/s DAC with <-80dBc IMD to 300MHz and <-160dBm/Hz noise power spectral density,” *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2003, pp. 126-482.

68. S. Park, G. Kim, S.-C. Park and W. Kim, “A digital-to-analog converter based on differential-quad switching,” *IEEE J. of Solid-State Circuits,* vol. 37, no. 10, pp. 1335-1338, October 2002.

69. G. Engel, S. Kuo and S. Rose, “A 14b 3/6GHz current-steering RF DAC in 0.18 μm CMOS with 66 dB ACLR at 2.9 GHZ,” *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2012, pp. 458-460.

70. G. Engel, M. Clara, H. Zhu and P. Wilkins, "A 16-bit 10Gsps current steering RF DAC in 65nm CMOS achieving 65dBc ACLR multi-carrier performance at 4.5GHz Fout," *Symposium on VLSI Circuits*, Kyoto, 2015, pp. C166-C167.

71. E. Olieman, A. Annema and B. Nauta, "An interleaved full Nyquist high-speed DAC technique," *IEEE J. of Solid-State Circuits*, vol. 50, no. 3, pp. 704-713, March 2015.

72. C. J. Leo, M. K. Raja and J. Minkyu, "An ultra low-power capacitor-less LDO with high PSR," *IEEE MTT-S International Microwave Workshop Series on RF and Wireless Technologies for Biomedical and Healthcare Applications*, Singapore, 2013, pp. 1-3.

73. T. Chen and G. Gielen, "The Analysis and Improvement of a Current-Steering DAC's Dynamic SFDR—II: The Output-Dependent Delay Differences," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 2, pp. 268-279, February 2007.

74. C.-H. Lin, F. M. L. van der Goes, J. R. Westra, J. Mulder, Y. Lin, E. Arslan, E. Ayranci, X. Liu and K. Bult, "A 12 bit 2.9 GS/s DAC with IM3 $<$ –60 dBc beyond 1 GHz in 65 nm CMOS," *IEEE J. of Solid-State Circuits*, vol. 44, no. 12, pp. 3285-3293, December 2009.

75. S. Luschas and H. -. Lee, "Output impedance requirements for DACs," *IEEE Int. Symposium on Circuits and Systems (ISCAS)*, Bangkok, Thailand, 2003, pp. 861-864.

76. A. Leon-Garcia, “Markov Chains,” in *Probability and Random Processes for*

Electrical Engineering, 2nd ed. New York, NY, USA: Addison Wesley Longman, 1994, pp. 459-498.

77. W.-H. Tseng, C.-W. Fan and J.-T. Wu, "A 12-bit 1.25-GS/s DAC in 90 nm CMOS with > 70 dB SFDR up to 500 MHz," *IEEE J. of Solid-State Circuits*, vol. 46, no. 12, pp. 2845-2856, December 2011.

Chapter 3

Reduced Noise Dynamic Element Matching

Abstract— Dynamic element matching (DEM) is often applied to DACs to convert what would otherwise be nonlinear distortion caused by 1-bit DAC mismatch errors into noise. The resulting noise is often more tolerable than nonlinear distortion, but is still undesirable and typically limits the noise performance of the DAC. This paper proposes a new technique called reduced noise DEM (RND) which reduces the noise in the DAC output caused by 1-bit DAC mismatch errors while still ensuring the DAC output is free of nonlinear distortion. RND is a foreground calibration technique that incorporates an ADC to measure the 1-bit DAC mismatch errors. An algorithm uses the measurement results to optimize the 1-bit DAC input sequences in order to reduce the noise in the DAC output. An analysis is presented that proves the RND technique prevents nonlinear distortion and spurious tones in the DAC output. Matlab simulation results of a 14-bit DAC incorporating RND show a 7 dB improvement in SNR relative to a 14-bit segmented DEM DAC.

I. Introduction

High-resolution multi-bit DACs typically comprise a digital encoder and multiple 1-bit DACs, the outputs of which are summed to form the overall DAC output. The encoder drives the inputs of the 1-bit DACs such that in the absence of errors, the DAC output waveform is an exact analog representation of the DAC input sequence.

The 1-bit DACs are typically implemented as one or more parallel copies of unit DAC elements controlled by the same encoder output signal. The number of unit DAC elements connected in parallel is the weight of the 1-bit DAC.

Depending on the weights of the 1-bit DACs and the DAC input value, there can be more than one vector of 1-bit DAC input values that would yield the desired DAC output if the 1-bit DACs were error-free. For these DAC input values the encoder is free to choose any of the valid vectors of 1-bit DAC input values. In practice, each valid vector results in a different error in the DAC output because the 1-bit DACs incur random errors during fabrication, and the error produced by each 1-bit DAC depends on its input value. The encoder has some flexibility in choosing the vector of 1-bit DAC input values it generates, and therefore it has some control over how the 1-bit DAC errors impact the DAC performance.

Thermometer encoders are often used to drive 1-bit DACs that have the same weight. When the input to the thermometer encoder is such that the inputs to k 1-bit DACs must be 1, a thermometer encoder sets the inputs of the first k 1-bit DACs to 1 and sets the inputs of the remaining 1-bit DACs to 0. Consequently, every 1-bit DAC produces the same error for a given DAC input value, and the resulting error in the DAC output is nonlinear distortion. In many applications the DAC's linearity requirement is the most difficult requirement to meet, thus motivating the design of alternative encoders in order to mitigate the effect of the 1-bit DAC errors.

Mismatch-mapping (MM) is a technique that aims to improve the DAC linearity by optimizing the vectors of 1-bit DAC input values generated by the encoder [78-

80]. MM DACs incorporate an MM encoder, an MM algorithm, and an ADC. The ADC measures the 1-bit DAC errors and provides the measurement results to the MM algorithm. The MM algorithm optimizes the vectors of 1-bit DAC input values produced by the MM encoder. The optimized vectors of 1-bit DAC input values minimize the net error in the DAC output by utilizing 1-bit DACs with positive errors in conjunction with 1-bit DACs with negative errors. As a result, a portion of the 1-bit DAC errors are cancelled when the 1-bit DAC outputs are summed to form the overall DAC output. Like thermometer encoders, MM encoders generate the same vector of 1-bit DAC input values for a given DAC input value. Thus, the error in the DAC output that is not cancelled is still correlated with the DAC input sequence, resulting in nonlinear distortion that still limits the DAC linearity.

Dynamic element matching (DEM) encoders differ from thermometer and MM encoders in that they do not always generate the same vector of 1-bit DAC input values for every DAC input value [81-89]. For each DAC input value, if there is more than one valid vector of 1-bit DAC input values, typical DEM encoders used in Nyquist-rate DACs will pseudo-randomly choose one of the valid vectors. As a result, DEM is able to prevent the 1-bit DAC errors from producing nonlinear distortion or spurious tones in the overall DAC output. However, the 1-bit DAC errors instead manifest as noise in the DAC output, and tend to limit the noise performance of DEM DACs. In many applications noise is more tolerable than nonlinear distortion but it is still undesirable.

This paper presents a new technique called reduced noise DEM (RND) that

combines the key ideas of MM and DEM. A RND DAC incorporates a RND encoder, a RND algorithm, and an ADC. Just like a DEM encoder, a RND encoder pseudo-randomly selects the vector of 1-bit DAC input values it generates when there is more than one valid vector. Similarly to MM, an ADC measures the 1-bit DAC errors and an algorithm uses the measurement results to optimize the vectors of 1-bit DAC input values generated by the encoder. As a result, the output of a RND DAC is free of non-linear distortion and spurious tones caused by 1-bit DAC errors just like the output of a DEM DAC, but the noise in the output of a RND DAC is lower than if the 1-bit DACs were driven by a DEM encoder.

Section II provides a brief overview of thermometer and DEM encoders in the context of a 17-level DAC. Section III presents the RND technique in the context of a 17-level DAC, and shows how it can be applied to a 14-bit hybrid DAC incorporating DEM. A mathematical analysis shows the output of the 14-bit hybrid DAC is free of nonlinear distortion. Section IV presents simulation results.

II. Thermometer and DEM Encoder Overview

The DAC output is a continuous-time waveform, $v(t)$, which is updated at the DAC sample rate, f_s. In discrete-time switched-capacitor circuits, the DAC output is sampled once per sample period, $T_s = 1/f_s$. For the purposes of such discrete-time systems, the DAC output can be interpreted as the sequence, $y[n] = v(nT_s)$, where each sample takes on an analog value. Similarly, in low-speed continuous-time DACs where the DAC output is not sampled but is settled for a large portion of each sample period,

the DAC output can be reasonably approximated as the sequence, $y[n]$. To simplify the analysis, this paper assumes the DAC output can be interpreted, or at least well-approximated, as the analog-valued sequence, $y[n]$.

The input to a DAC is a sequence of digital codewords, $x[n]$, updated at the same rate as the DAC output, f_s. A DAC contains N 1-bit DACs, the outputs of which are summed to form the DAC output. Figure 25 shows an example of a 17-level DAC containing $N = 16$ unit-weight 1-bit DACs. For the DAC in Fig. 25, $x[n]$, takes on values from the set

$$\left\{-\frac{N}{2}\Delta,\left(-\frac{N}{2}+1\right)\Delta,\left(-\frac{N}{2}+2\right)\Delta,...,\frac{N}{2}\Delta\right\}, \tag{116}$$

where Δ is the DAC's minimum step-size. In each clock cycle, the encoder sets its binary output sequences, $c_i[n]$, such that in the absence of errors,

$$y[n] = x[n]. \tag{117}$$

The vector of 1-bit DAC input values produced by the encoder during clock cycle n consists of the values of $c_i[n]$ for $1 \le i \le N$.

Figure 26 shows an example of the DAC output values produced when the encoder in the 17-level DAC of Fig. 25 is implemented as a thermometer encoder and the output of each 1-bit DAC contains one of two randomly selected errors, depending on its input value. The line connecting the DAC output values when $x[n] = \pm 8\Delta$ is described by $y = \alpha x + \beta$ where α is the slope and β is the offset. Figure 26 implies that the DAC output sequence can be expressed as

$$y[n] = \alpha x[n] + \beta + e_{Therm}[n] \tag{118}$$

where $e_{Therm}[n]$ represents the deviation of the DAC output value from the line, $y = \alpha x + \beta$, for the given DAC input value. The DAC output in (118) deviates from the ideal given in (117) because the 1-bit DAC errors cause α, β, and $e_{Therm}[n]$ to deviate from their ideal values of 1, 0, and 0, respectively. Equation (118) describes a DAC with a gain of α, an offset of β, and an error sequence, $e_{Therm}[n]$, that is a deterministic nonlinear function of the DAC input sequence. In many applications, small deviations in the gain and offset from the ideal values don't cause problems [90]. However, $e_{Therm}[n]$ is problematic because it represents code-dependent error that results in nonlinear distortion in the DAC output.

For all but the minimum and maximum input values to the 17-level DAC shown in Fig. 25, there is more than one valid vector of 1-bit DAC input values that would yield the ideal DAC output value if the 1-bit DACs were error-free. For example, when the input to a single 1-bit DAC must be 1 and the input to the fifteen other 1-bit DACs must be 0, the encoder is free to choose which of the sixteen valid vectors of 1-bit DAC input values to produce. Each valid vector of 1-bit DAC input values yields a different DAC output value because the error produced by each 1-bit DAC depends on the value of its input. Figure 27 shows all of the possible DAC output values of a 17-level DAC with the same 1-bit DAC errors shown in Fig. 26.

DEM encoders typically do not use all of the valid vectors of 1-bit DAC input values as this is not required to achieve the goal of preventing nonlinear distortion and spurious tones caused by 1-bit DAC errors [84]. Figure 28 shows an example of the DAC output values produced when the encoder in Fig. 25 is implemented with the

DEM technique described in [84] and the 1-bit DAC errors are the same as shown in Fig. 26 and Fig. 27. By definition, the DAC output values corresponding to the full-scale input values are on the line, $y = \alpha x + \beta$. For all other input values, the DEM encoder pseudo-randomly selects one valid vector of 1-bit DAC input values from a subset of the valid vectors so that the average of the DAC output values produced is on the line, $y = \alpha x + \beta$. As a result, the output of a DEM DAC is given by (118) where $e_{Therm}[n]$ is replaced by $e_{DEM}[n]$ and $e_{DEM}[n]$ is a zero-mean, noise-like stochastic sequence that is uncorrelated with the DAC input sequence [84]. Thus, by producing a subset of the valid vectors of 1-bit DAC input values, a DEM encoder prevents nonlinear distortion and spurious tones caused by 1-bit DAC errors.

III. Reduced Noise DEM

The goals of DEM can be achieved using different subsets of the valid vectors of 1-bit DAC input values, and some subsets result in less error in the DAC output than others. The objective of the RND technique is to constrain the RND encoder to produce a subset of the valid vectors of 1-bit DAC input values that achieve the goals of DEM, and that minimize the error in the DAC output.

Figure 29 shows an example of the output values a 17-level RND DAC produces given the same 1-bit DAC errors shown in Fig. 26 – Fig. 28. For all but the full-scale DAC input values, the RND encoder pseudo-randomly selects valid vectors of 1-bit DAC input values such that the average of the DAC output values produced is on the line, $y = \alpha x + \beta$. Thus, the output of a RND DAC is given by (118) where $e_{Therm}[n]$

is replaced by $e_{RND}[n]$ and $e_{RND}[n]$ is a zero-mean noise-like stochastic sequence that is uncorrelated with the DAC input sequence. For all but the four most extreme DAC input values, $x[n] = \pm 8\Delta$ and $x[n] = \pm 7\Delta$, the mean squared deviation of the RND DAC output values from the line, $y = \alpha x + \beta$, is lower than the mean squared deviation of the DEM DAC output values from the same line. For the four most extreme DAC input values, the analysis in the following sub-sections shows that the RND encoder cannot optimize the vectors of 1-bit DAC input values and must use the same vectors as the DEM encoder in order to prevent nonlinear distortion. However, typical DAC input sequences are not composed exclusively of these four most extreme DAC input values. Thus for typical DAC input sequences, $E\{e_{RND}[n]^2\} < E\{e_{DEM}[n]^2\}$ and the noise in the output of a RND DAC is lower than the noise in the output of a DEM DAC.

A. 14-bit Segmented Hybrid DAC

High-resolution, multi-bit DACs typically contain several large, equal-weight 1-bit DACs and several smaller, non-uniformly-weighted 1-bit DACs. For example, Fig. 30 shows a block diagram of the 14-bit segmented DEM DAC presented in [84] which contains sixteen 1024-weight 1-bit DACs and twenty smaller-weight 1-bit DACs. A larger-weight 1-bit DAC contributes more error to the DAC output than a smaller-weight 1-bit DAC [91], thus the sixteen 1024-weight 1-bit DACs contribute significantly more error to the DAC output than the remaining 1-bit DACs. Therefore, a large benefit is achieved by applying RND to the top sixteen 1-bit DACs in Fig. 30. Figure 31 shows a block diagram of the proposed 14-bit hybrid DAC which

incorporates a RND encoder and DEM switching blocks.

The outputs of the top sixteen 1-bit DACs in the hybrid DAC can be directed to an ADC to be measured. The ADC measurement results are provided to the RND algorithm which calculates and stores optimized vectors of 1-bit DAC input values in a memory contained in the RND encoder. For each DAC input value, the RND encoder pseudo-randomly selects one of the valid vectors of 1-bit DAC input values from memory and routes the bits of the vector to the inputs of the 1024-weight 1-bit DACs.

B. Requirements to Prevent Nonlinear Distortion

The hybrid encoder in Fig. 31 generates its output bit sequences pseudo-randomly, so the $c_i[n]$ sequences are stochastic. As shown in [84], the $c_i[n]$ sequences must have certain statistical properties in order to prevent nonlinear distortion. The switching blocks that drive the bottom twenty 1-bit DACs in Fig. 31 cause $c_i[n]$ for $1 \le i \le 20$ to have the required properties [84]. The statistical properties that are stated in [84] are restated in this section, and extended to derive the expected value of $c_i[n]$ for $21 \le i \le 36$ that prevents nonlinear distortion caused by errors in the top sixteen 1-bit DACs. The next two sub-sections present the RND encoder and algorithm that achieve the desired expected value, thus preventing nonlinear distortion from all 1-bit DACs in Fig. 31.

The output of the ith 1-bit DAC is

$$y_i[n] = (c_i[n] - \tfrac{1}{2})K_i\Delta + e_i[n], \tag{119}$$

where K_i is the weight of the 1-bit DAC and $e_i[n]$ represents the error made by the 1-

bit DAC during the *n*th clock cycle. In the applications described in Section II in which the DAC output can be modeled as a sequence, the most significant types of 1-bit DAC errors are mismatches among the 1-bit DACs. Such mismatches are well-modeled by

$$e_i[n] = \begin{cases} e_{1i}, & \text{if } c_i[n] = 1, \\ e_{0i}, & \text{if } c_i[n] = 0, \end{cases} \tag{120}$$

where e_{1i} and e_{0i} are constants that represent the error made by the 1-bit DAC when its input bit value is 1 and 0, respectively.

An equivalent form of the 1-bit DAC output given in (119) is

$$y_i[n] = K_i \Delta \alpha_i (c_i[n] - \tfrac{1}{2}) + \beta_i , \tag{121}$$

where

$$\alpha_i = 1 + \frac{e_{1i} - e_{0i}}{K_i \Delta}, \text{ and } \quad \beta_i = \frac{1}{2}\left[e_{1i} + e_{0i}\right]. \tag{122}$$

As shown in Fig. 31, the outputs of the 1-bit DACs are summed to form the overall DAC output, i.e.,

$$y[n] = \sum_{i=1}^{N} y_i[n] . \tag{123}$$

The output sequences of the encoder must satisfy

$$x[n] = \Delta \sum_{i=1}^{36} K_i \left(c_i[n] - \tfrac{1}{2}\right), \tag{124}$$

which constrains the encoder to producing only the valid vectors of 1-bit DAC input values that yield the desired DAC output value in the absence of 1-bit DAC errors. To verify that (124) restricts the encoder to producing only valid vectors of 1-bit DAC values, the equation for the ideal 1-bit DAC output, i.e., (119) with $e_i[n] = 0$, is

substituted into (123), then (124) is substituted into the result, which gives the ideal DAC output behavior in (117).

As shown in [84], the output sequences of the segmented DEM encoder in Fig. 30, and thus also the output sequences of the hybrid encoder in Fig. 31, satisfy

$$c_i[n] = \frac{1}{\Delta}\left(m_i x[n] + \lambda_i[n]\right) + \frac{1}{2}, \tag{125}$$

where

$$m_i = \begin{cases} 0, & i = 1, 2, ..., 20, \\ 2^{-14}, & i = 21, 22, ..., 36, \end{cases} \tag{126}$$

each $\lambda_i[n]$ is a stochastic sequence, and

$$\sum_{i=1}^{36} K_i m_i = 1, \text{ and } \quad \sum_{i=1}^{36} K_i \lambda_i[n] = 0. \tag{127}$$

The DAC output when the 1-bit DACs are not error-free is obtained by substituting the left equation of (122) into (121), the result into (123), then substituting (125) into the result and simplifying the outcome using (127) to give

$$y[n] = x[n]\alpha + \beta + e_{DAC}[n] \ , \tag{128}$$

where

$$\alpha = 1 + \frac{1}{\Delta}\sum_{i=1}^{36} m_i \left(e_{1i} - e_{0i}\right) \ , \tag{129}$$

$$\beta = \sum_{i=1}^{36} \beta_i \ , \tag{130}$$

and

$$e_{DAC}[n] = \frac{1}{\Delta}\sum_{i=1}^{36} \lambda_i[n]\left(e_{1i} - e_{0i}\right). \tag{131}$$

The 1-bit DAC errors cause α and β to deviate from their ideal values of 1 and 0, respectively, but as discussed in the previous section, errors in α and β do not cause problems in most DAC applications.

The hybrid encoder prevents nonlinear distortion and spurious tones if each $\lambda_i[n]$ sequence is a zero-mean, noise-like sequence that is uncorrelated with the DAC input sequence, i.e.,

$$\left.\begin{aligned} &E\{\lambda_i[n]\} = 0 \\ &E\{\lambda_i[n]\lambda_i[m]\} = 0 \quad \text{for } m \neq n \end{aligned}\right\} \text{regardless of } x[n]. \tag{132}$$

If (132) holds, then because each term in the summation of (131) is multiplied by $\lambda_i[n]$, $e_{DAC}[n]$ represents zero-mean noise that is uncorrelated with the DAC input sequence instead of nonlinear distortion.

The first expected value in (132) holds if the expected value of $\lambda_i[n]$ is zero given every possible input value to the RND encoder. The RND encoder is implemented in digital logic that operates on unsigned integers so it is convenient to define its input sequence as

$$c_{RND}[n] = \sum_{k=21}^{36} c_k[n], \tag{133}$$

where $c_{RND}[n]$ is the number of 1024-weight 1-bit DACs that have their inputs set to 1 in each clock cycle. As there are sixteen 1024-weight 1-bit DACs shown in Fig. 31, it follows that $c_{RND}[n]$ takes on values from the set

$$\Psi = \{0,1,2,...,16\}. \tag{134}$$

Therefore, the first expected value in (132) holds if

$$E\{\lambda_i[n] \mid c_{RND}[n] = Q\} = 0 \qquad \text{for } Q \in \Psi . \tag{135}$$

The conditional expectation in (135) is derived in terms of the conditional expectation of $c_i[n]$, which is used to constrain the design of the RND encoder. Equation (125) is solved for $\lambda_i[n]$, and the summation in (124) is equivalently stated as the sum of two summations with limits that together span the limits of the original summation, and substituted into the result to give,

$$\begin{aligned} \lambda_i[n] = &\left(c_i[n] - \frac{1}{2} \right) \Delta \\ &- m_i \left(\Delta \sum_{j=1}^{20} K_j \left(c_j[n] - \tfrac{1}{2} \right) + \Delta \sum_{k=21}^{36} K_k \left(c_k[n] - \tfrac{1}{2} \right) \right). \end{aligned} \tag{136}$$

Substituting (133) into (136) given $c_{RND}[n] = Q$, and taking the expected value of the result gives,

$$\begin{aligned} E\left\{ \lambda_i[n] \mid c_{RND}[n] = Q \right\} = &E\left\{ c_i[n] \mid c_{RND}[n] = Q \right\} \Delta - \frac{1}{2} \Delta \\ &- m_i \Delta E \left\{ \left(\sum_{i=j}^{20} K_j \left(c_j[n] - \tfrac{1}{2} \right) \right) \mid c_{RND}[n] = Q \right\} \\ &- m_i \cdot 1024 \Delta \cdot (Q - 8). \end{aligned} \tag{137}$$

The second conditional expectation on the right of (137) is simplified using a result of [84], which shows that the switching blocks in Fig. 31 ensure $E\{\lambda_j[n]\} = 0$ for $1 \le j \le 20$. Substituting (125) with (126) into the second conditional expectation on the right of (137), and applying the result from [84] shows that

$$E\left\{ \left(\sum_{i=j}^{20} K_j \left(c_j[n] - \tfrac{1}{2} \right) \right) \mid c_{RND}[n] = Q \right\} = 0 . \tag{138}$$

Substituting (138) along with (126) for $21 \le i \le 36$ into (137), then substituting the

result into (135) and solving for the conditional expected value of $c_i[n]$ gives,

$$E\left\{c_i[n]=1 \mid c_{RND}[n]=Q\right\}=\frac{Q}{16} \quad \text{for } 21 \le i \le 36 \text{ and } Q \in \Psi. \tag{139}$$

Therefore, the $\lambda_i[n]$ sequences are zero-mean and the first expected value in (132) holds provided the RND encoder ensures (139) holds. In addition, if the RND encoder ensures (139) holds regardless of $\lambda_i[m]$ for $n \neq m$, then $\lambda_i[n]$ is zero-mean regardless of $\lambda_i[m]$ and the second expected value in (132) also holds.

C. RND Encoder

The RND encoder and algorithm are designed to ensure that (139) holds. The vectors of 1024-weight 1-bit DAC input values produced by the RND encoder are stored in a two-dimensional array of 16-bit memory words, $\mathbf{M} = [m_{u,v}]$, where $m_{u,v}$ is the vth memory word in the uth row of the array. Each row stores the possible vectors of 1024-weight 1-bit DAC input values produced by the encoder for a particular value of $c_{RND}[n]$. Thus, because $c_{RND}[n]$ can take on one of the 17 values in {0, 1, ..., 16}, there are 17 rows in the memory array. For reasons explained shortly, each row does not contain the same number of memory words. In each clock cycle, the RND encoder selects one memory word from the array and routes the ith bit of the selected memory word to the input of the ith 1024-weight 1-bit DAC. The $c_{RND}[n]$ sequence provides the row index of the memory word to select. A random sequence, $d_m[n]$, is used to select one memory word from the selected row such that all memory words on the row have an equal probability of being selected. Any two distinct samples, $d_m[n]$ and $d_m[m]$ for $n \neq m$, are independent which ensures (139) holds regardless of $\lambda_i[m]$ in order to prevent

spurious tones.

The number of memory words in each row, and the contents of the memory words required to prevent nonlinear distortion, follow from the analysis in the previous sub-section. Equation (139) implies that in every 16 clock cycles in which $c_{RND}[n] = Q$, on average each 1024-weight 1-bit DAC input sequence must be 1 for Q clock cycles and 0 for the remaining clock cycles. The desired conditional expected value of the 1024-weight 1-bit DAC input sequence can be achieved by allocating 16 memory words on the row selected when $c_{RND}[n] = Q$, and setting each bit, i, in Q of these memory words. However, when the greatest common factor of Q and 16, $GCF_{Q,16}$, is greater than one, the conditional expected value in (139) can also be achieved by allocating $16/GCF_{Q,16}$ memory words on the row selected when $c_{RND}[n] = Q$ and setting each bit i in $Q/GCF_{Q,16}$ of these memory words, i.e.,

$$Q / GCF_{Q,16} = \sum_{v=1}^{16/GCF_{Q,16}} m_{Q+1,v}{}^{(i)} \quad \text{for } 0 \le Q \le 16 \text{ and } 1 \le i \le 16, \tag{140}$$

where $m_{Q+1,v}{}^{(i)}$ denotes the ith bit of the vth memory word in the row selected when $c_{RND}[n] = Q$. Thus, a significant amount of memory is saved by allocating only the minimum required amount of memory, i.e., $16/GCF_{Q,16}$ memory words per row, $Q + 1$.

For the hybrid encoder to satisfy (124) and produce only the valid vectors of 1-bit DAC input values that yield the ideal DAC output in the absence of errors, each memory word on row $Q + 1$ must have Q bits set, i.e.,

$$Q = \sum_{i=1}^{16} m_{Q+1,v}{}^{(i)} \qquad \text{for } 0 \le Q \le 16 \text{ and } 1 \le v \le 16 / GCF_{Q,16}. \tag{141}$$

The RND encoder logic retrieves memory words from the memory array and routes the bits of the memory word to the 1024-weight 1-bit DACs inputs according to

$$c_i[n] = m_{c_{RND}[n]+1, d_m[n]+1}^{(i-20)}, \tag{142}$$

where

$$d_m[n] = d[n] \bmod \frac{16}{GCF_{c[n],16}}, \tag{143}$$

and $d[n]$ is a sequence of random integers uniformly distributed between 0 and 15, and "mod" denotes the modulo operator. Not every row contains the same number of memory words, so the modulo operation in (143) restricts ($d_m[n]$ + 1) in (142) to selecting one of the memory words on the selected row.

D. RND Algorithm

The RND algorithm has an initialization phase that programs the memory array so that it satisfies (140) and (141) in order to prevent nonlinear distortion in the DAC output, and an optimization phase that makes incremental changes to the memory array to reduce the variance of the error of the 1024-weight 1-bit DACs while still ensuring (140) and (141) hold.

The initialization phase of the RND algorithm resets each memory word in the array to zero, then programs the memory one row at a time. For each vth memory word on row Q + 1, the algorithm sets Q bits sequentially starting at bit ($Q(v-1)$ mod 16) + 1. If the operation of sequentially setting bits attempts to set the 17*th* bit, i.e., a nonexistent bit of the memory word, the operation wraps around and continues to set bits sequentially starting at the first bit until Q bits are set in the memory word. The modulo

arithmetic used to calculate the starting bit for sequentially setting bits ensures (140) holds. By setting Q bits in each memory word on row $Q + 1$ the algorithm ensures (141) holds.

The optimization phase of the RND algorithm optimizes each row, $Q + 1$, of the memory array, except for the first two and last two rows, as explained shortly. To optimize a row, a bit value of 1 in a memory word is swapped with a bit value of 1 in a different memory word on the same row, i.e., $m_{Q+1,v}^{(i)}$ and $m_{Q+1,w}^{(j)}$ are cleared and $m_{Q+1,v}^{(j)}$ and $m_{Q+1,w}^{(i)}$ are set. The indices of the bits and memory words, i, j, v, and w, are randomly selected such that $i \neq j$ and $v \neq w$, and each set or clear operation causes a change to the memory word, i.e., bit values of 0 and 1 are never cleared or set, respectively. After performing the swapping operation, the algorithm calculates the variance of the error of the 1024-weight 1-bit DACs. If the variance increased the swap is undone. The process of swapping bits and evaluating the variance of the error of the 1024-weight 1-bit DACs is repeated N_{swaps} times for each row, where N_{swaps} is a parameter of the RND algorithm.

The error of the 1024-weight 1-bit DACs is equal to $e_{DAC}[n]$ if the bottom twenty 1-bit DACs in Fig. 31 were assumed not to contribute to the DAC output, i.e., e_{1i} and e_{0i} are zero for $1 \leq i \leq 20$, and the first summation on the right of (136) is zero. Then the error of the 1024-weight 1-bit DACs is given by the summation in (131) except with the lower limit set to 21 instead of 1, i.e.,

$$e_{1024\Delta}[n] = \sum_{i=21}^{36} \tilde{\lambda}_i[n]\left(e_{1i} - e_{0i}\right), \tag{144}$$

and $\tilde{\lambda}_i[n]$ is given by substituting (133) and (126) into (136) and setting the first summation on the right of (136) to zero to give,

$$\tilde{\lambda}_i[n] = c_i[n]\Delta - \frac{c_{RND}[n]\Delta}{16}. \tag{145}$$

The error of the 1024-weight 1-bit DACs given by (144) with (145) depends on $c_i[n]$, which is specified by (142) and depends on $c_{RND}[n]$ and $d_m[n]$. Thus substituting (142) into (145), the result into (144), and setting $c_{RND}[n] = Q$ and $d_m[n] = R$, gives the conditional error of the 1024-weight 1-bit DACs,

$$e_{1024\Delta|c_{RND}[n]=Q,d_m[n]=R}[n] = \sum_{i=21}^{36}\left(m_{Q+1,R+1}^{(i-20)}\Delta - \frac{Q\Delta}{16}\right)\left(e_{1i} - e_{0i}\right). \tag{146}$$

The $d_m[n]$ sequence selects each memory word on the row with an equal probability, thus the variance of the error of the 1024-weight 1-bit DACs for a given value of $c_{RND}[n] = Q$ is the mean squared of the error produced by the 1024-weight 1-bit DACs for each memory word on the row,

$$E\left\{e_{1024\Delta}[n]^2 \mid c_{RND}[n] = Q\right\} = \frac{GCF_{Q,16}}{16}\sum_{v=0}^{16/GCF_{Q,16}-1}\left(e_{1024\Delta|c_{RND}[n]=Q,d_m[n]=v}[n]\right)^2 \tag{147}$$

The optimization phase of the algorithm optimizes one row of the memory array, $Q + 1$, at a time, which contains the vectors of 1024-weight 1-bit DAC input values produced by the RND encoder when $c_{RND}[n] = Q$. Thus, to minimize the variance of the 1024-weight 1-bit DAC error when $c_{RND}[n] = Q$, the optimization phase of the algorithm evaluates (147) in each iteration after swapping bits. The RND algorithm requires the e_{1i} and e_{0i} errors of the 1024-weight 1-bit DACs to be measured with the

ADC in order to calculate the conditional variance of the error of the 1024-weight 1-bit DAC specified in (147).

The optimization phase of the RND algorithm optimizes all but the first two and last two rows of the memory array. The first row contains a single memory word containing the vector of 1024-weight 1-bit DAC input values produced when $c_{RND}[n] = 0$. The swapping operation requires a minimum of two memory words in the row, so this row is skipped by the optimization phase of the algorithm. The second row contains 16 memory words, but this row contains the vectors of 1024-weight 1-bit DAC input values produced when $c_{RND}[n] = 1$, so only a single bit is set in each memory word. As a result, swapping bits between memory words only changes the order of the memory words in the row and does not alter the variance of the DAC error. Similarly, the RND algorithm cannot optimize the last two rows of the memory array. As a result of these limitations, the RND encoder cannot reduce the error in the DAC output relative to a DEM encoder for the two largest and two smallest input values to the RND encoder.

IV. Simulation Results

Figure 32 shows the simulated output power spectra of the segmented DEM DAC shown in Fig. 30 and the hybrid DAC shown in Fig. 31 for a full-scale, single-tone input signal. The DACs were simulated in Matlab using the equations presented in Section III-A. The errors of the 1-bit DACs were chosen from Gaussian distributions in order to model the 58 dB SNR of the segmented DEM DAC presented in [84] if the only 1-bit DAC errors were due to the current sources in the 1-bit DACs. Consequently,

the standard deviation used to select the unit-weight 1-bit DAC errors was 0.12Δ, and as shown in [91], the standard deviation was increased by a factor of $\sqrt{2}$ for each doubling of the weight of the 1-bit DACs. In the high-speed DAC presented in [84], each 1-bit DAC has several significant sources of dynamic error, such as the switch drivers and clock skew, which were not modeled in the Matlab simulations presented in this section. However, as the sample rate of a DAC is decreased, the dynamic errors become less significant relative to the error of the current sources in the 1-bit DACs. The ADC in Fig. 31 was modeled as a 6-bit ideal quantizer with a dynamic range of 27Δ, centered at 1024Δ.

The power spectra of the hybrid DAC and segmented DEM DAC outputs shown in Fig. 32 are free of spurious tones as expected. However, the hybrid DAC shows a 7 dB improvement in SNR relative to the segmented DEM DAC because the vectors of 1024-weight 1-bit DAC input values produced by the RND encoder have been optimized by the RND algorithm.

Figure 33 shows the simulated SNR of the hybrid DAC vs. the N_{swaps} parameter of the RND algorithm, which determines the numbers of swaps evaluated on each row of the memory array. The SNR improves as N_{swaps} is increased until the maximum benefit of RND is achieved, which occurs at N_{swaps} = 1000 for the particular randomly selected mismatches.

The improvement in SNR of a hybrid DAC relative to the SNR of a segmented DEM DAC increases as the number of 1-bit DACs driven by the RND encoder (and corresponding switching blocks in the segmented DEM DAC) is increased. The 17-

level RND encoder shown in Fig. 31 can be replaced with a RND encoder that drives a different number of 1024-weight 1-bit DACs to yield a hybrid DAC with a resolution other than 14-bits. Figure 34 shows the simulated SNR of a hybrid DAC architecture versus the number of 1024-weight 1-bit DACs driven by the RND encoder, compared to the simulated SNR of a segmented DEM DAC architecture with the same 1-bit DACs. RND keeps the noise in the DAC output relatively insensitive to the number of 1-bit DACs driven by the RND encoder. As a result, the simulated SNR of a hybrid DAC increases by nearly 6 dB each time the number of 1024-weight 1-bit DACs driven by the RND encoder is doubled. Doubling the number of 1-bit DACs driven by the RND encoder increases the amplitude of the desired signal component by a factor of two, resulting in a 6 dB increase in signal power. In contrast, the noise in the output of a segmented DEM DAC increases with the number of 1024-weight 1-bit DACs, thus the simulated SNR of the segmented DEM DAC increases by approximately 3 dB each time the number of 1024-weight 1-bit DACs is doubled.

The memory required by the RND encoder increases nearly exponentially with the number of 1024-weight 1-bit DACs it drives, as shown in Table 3. Thus, there is a tradeoff between the performance improvement of an RND encoder that drives a greater number of 1-bit DACs, and the memory required to implement the RND encoder.

V. Conclusion

RND is a foreground calibration technique that can reduce the net effect of 1-

bit DAC errors in the DAC output, while still ensuring the DAC output is free of non-linear distortion and spurious tones. When applied to a high-resolution hybrid DAC, RND significantly improves the SNR relative to a segmented DEM DAC when dynamic errors are insignificant

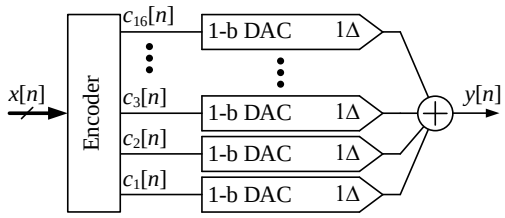

Figure 25 : Example 17-level DAC.

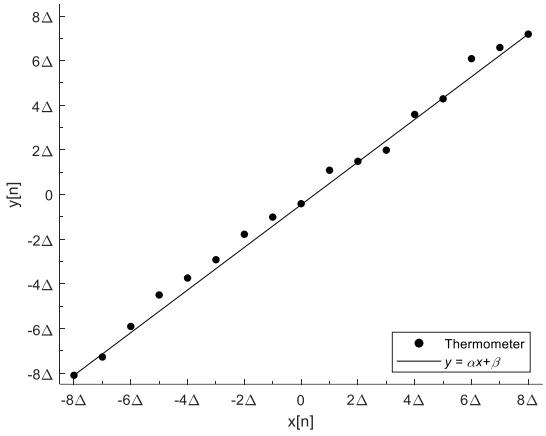

Figure 26 : Example output values of 17-level thermometer DAC.

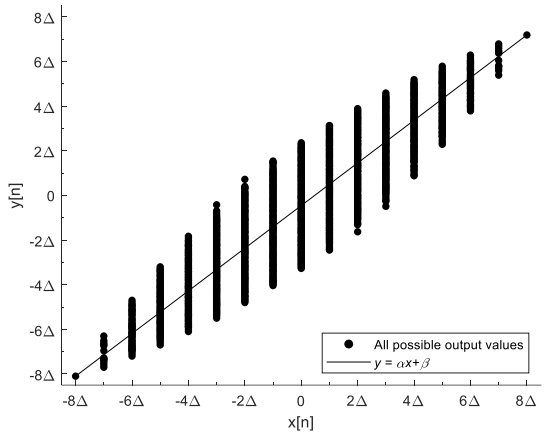

Figure 27 : Example of all possible output values of 17-level DAC.

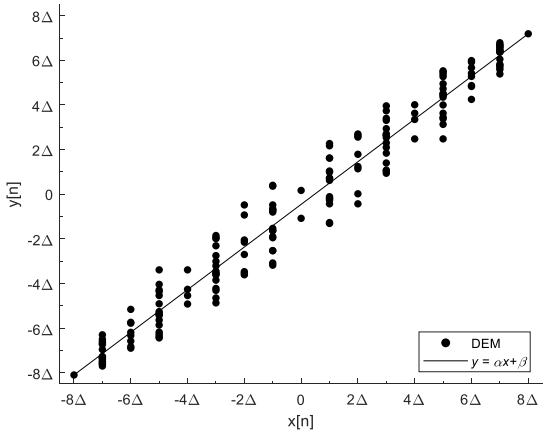

Figure 28 : Example output values of 17-level DEM DAC.

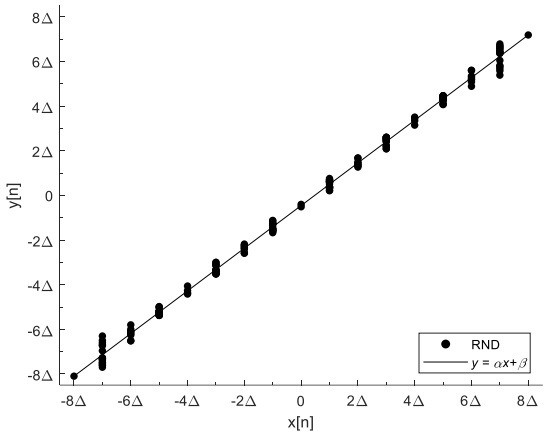

Figure 29 : Example output values of 17-level RND DAC.

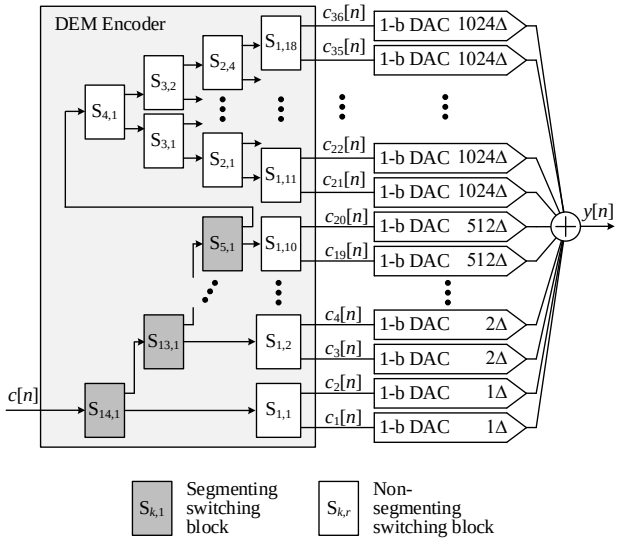

Figure 30 : Block diagram of segmented DEM DAC from [84].

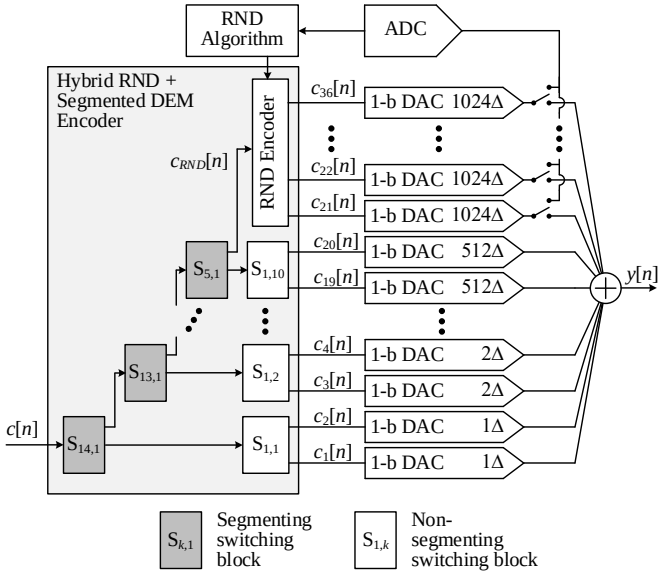

Figure 31 : Block diagram of hybrid DAC.

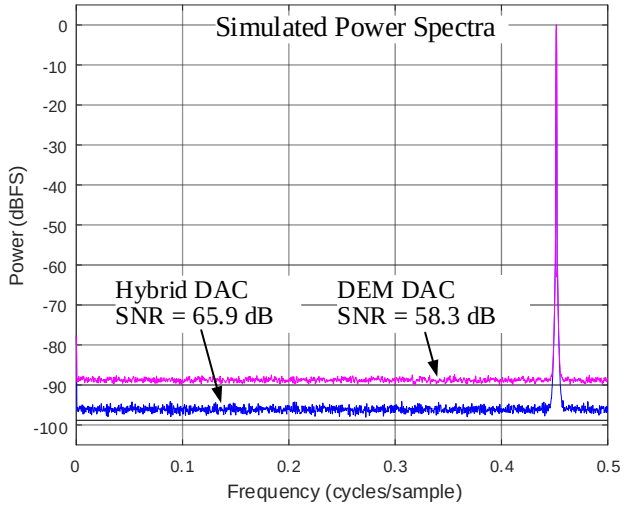

Figure 32 : Simulated power spectra of hybrid DAC and segmented DEM DAC.

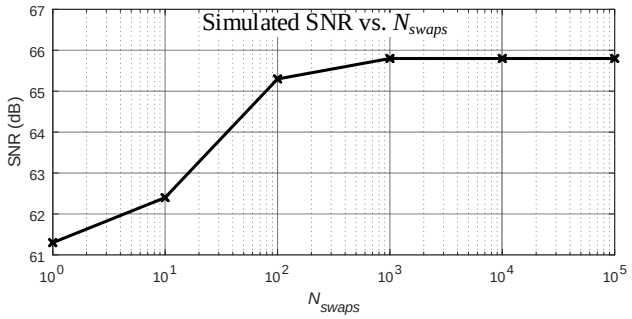

Figure 33 : Simulated SNR of hybrid DAC vs. N_{swaps}.

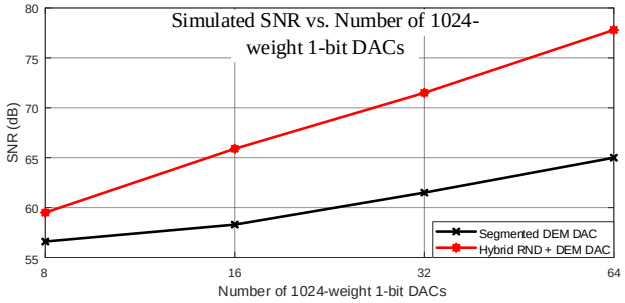

Figure 34 : Simulated power spectra of hybrid DAC and segmented DAC.

TABLES

Table 3 RND Memory Requirements.

# of 1-bit DACs driven by RND encoder	Required Memory (bytes)
8	44
16	344
32	2736
64	21856

REFERENCE

78. T. Chen, G. Gielen, "A 14-bit 200 MHz current-steering DAC with switching-sequence post-adjustment calibration," *IEEE J. Solid-State Circuits*, vol. 42, no. 11, pp. 2386-2394, Nov. 2007.

79. Y. Tang, J. Briare, K. Doris, R. van Veldhoven, P. van Beek, H. Hegt, A. Roermund, "A 14 bit 200 MS/s DAC with SFDR > 78 dBc, IM3 < -83 dBc and NSD < -163 dBm/Hz across the whole Nyquist band enabled by dynamic mismatch mapping," *IEEE J. Solid-State Circuits*, vol. 46, no.6, pp. 1371-1381, June 2011.

80. H. Van de Vel, J. Briaire, C. Bastiaansen, P. van Beek, G. Geelen, H. Gunnink, Y. Jin, M. Kaba, K. Luo, E. Paulus, B. Pham, W. Relyveld and P. Zijlstra, "A 240mW 16b 3.2GS/s DAC in 65nm CMOS with <–80dBc IM3 up to 600 MHz," *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2014, pp. 206-208.

81. R. van de Plassche, "Dynamic element matching for high-accuracy monolithic D/A converters," *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 1976, pp. 148-149.

82. K. O. Sullivan, C. Gorman, M. Hennessy and V. Callaghan, "A 12-bit 320-MSample/s current-steering CMOS D/A converter in 0.44 mm^2," *IEEE J. of Solid-State Circuits*, vol. 39, no. 7, pp. 1064-1072, July 2004.

83. B. Jewett, J. Liu and K. Poulton, "A 1.2 Gs/s 15b DAC for precision signal generation," *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2005, pp. 110-111.

84. K. L. Chan, J. Zhu and I. Galton, "Dynamic element matching to prevent nonlinear distortion from pulse-shape mismatches in high-resolution DACs," *IEEE J. of Solid-State Circuits*, vol. 43, no. 9, pp. 2067-2078, September 2008.

85. W.-T. Lin, H.-Y. Huang and T.-H. Kuo, "A 12-bit 40 nm DAC achieving SFDR > 70 dB at 1.6 GS/s and IMD < -61dB at 2.8 GS/s with DEMDRZ technique," *IEEE J. of Solid-State Circuits*, Vol. 49, no. 3, pp. 708-717, March 2014.

86. W. Mao, Y. Li, C. Heng and Y. Lian, "High dynamic performance current-steering DAC design with nested-segment structure," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 26, no. 5, pp. 995-999, May 2018.

87. C.-H. Lin, K. L. J. Wong, T.-Y. Kim, G. R. Xie, D. Major, G. Unruh, S. R. Dommaraju, H. Eberhart and A. Venes, "A 16b 6 GS/s Nyquist DAC with IMD < -90dBc up to 1.9 GHz in 16 nm CMOS," *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2018, pp. 360 – 362.

88. D. Kong, K. Rivas-Rivera and I. Galton, "A 600 MS/s DAC with over 87- dB SFDR and 77-dB peak SNDR enabled by adaptive cancellation of static and dynamic mismatch error," *IEEE J. of Solid-State Circuits*, vol. 54, no. 8, pp. 2219-2229, August 2019.

89. H.-Y. Huang and T.-H. Kuo, "A 0.07-mm^2 162-mW DAC achieving >65 dBC SFDR and < –70 dBc IM3 at 10 GS/s with output impedance compensation and concentric parallelogram routing," *IEEE J. of Solid-State Circuits*, vol. 55, no. 9, pp. 2478-2488, September 2020.

90. J. Remple and I. Galton, "The effects of inter-symbol interference in dynamic element matching DACs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 1, pp. 14-23, January 2017.

91. A. van den Bosch, M. A. F. Borremans, M. S. J. Steyaert and W. Sansen, "A 10-bit 1-GSample/s Nyquist current-steering CMOS D/A converter," *IEEE J. of Solid-State Circuits*, vol. 36, no. 3, pp. 315-324, March 2001.