Dynamic Element Matching to Prevent Nonlinear Distortion From Pulse-Shape Mismatches in High-Resolution DACs

Kok Lim Chan, Jianyu Zhu, and Ian Galton, *Member, IEEE*

***Abstract*—This paper shows analytically and experimentally that properly-designed dynamic element matching (DEM) eliminates pulse shape, timing, and amplitude errors arising from component mismatches as sources of nonlinear distortion in high-resolution DACs. A set of sufficient conditions on the DEM encoder that ensure this effect, and a specific segmented DEM encoder that satisfies the sufficient conditions are presented. Unlike most previously published DEM encoders, the new DEM encoder's complexity does not grow exponentially with the number of bits of DAC resolution, so it is practical for high-resolution Nyquist-rate DACs. These analytical results are demonstrated experimentally with a 0.18 μm CMOS 14-bit DAC IC that has a sample rate of 100 MHz and worst case, single and two-tone spurious-free dynamic ranges of 83 dB and 84 dB, respectively, across the Nyquist band.**

***Index Terms*—Dynamic element matching, digital-to-analog converters, CMOS current-steering DAC, high-speed high-resolution data converters.**

I. Introduction

In a typical high-resolution ($\geq$ 12 bit) Nyquist-rate digital-to-analog converter (DAC), the outputs of several parallel 1-bit DACs with various weights are summed to generate the overall output signal. Each 1-bit DAC consists of one or more *unit DAC elements* combined in parallel and controlled as a group; the number of unit DAC elements is the weight of the 1-bit DAC. Most commonly, each unit DAC element consists of a unit current steering cell and its associated switch driver circuitry.

The unit DAC elements inevitably are subject to random mismatches incurred during IC fabrication as well as possible systematic circuit and layout mismatches. Such mismatches cause amplitude, pulse shape, and timing errors in the 1-bit DAC output signals which usually introduce nonlinear distortion in the overall DAC output signal, often limiting the linearity of the overall DAC. Aside from careful layout strategies [1], [2], the primary published techniques to minimize the nonlinear distortion are calibration [3]–[6], trimming [7], and DEM [8]–[12].

The calibration and trimming techniques published to date work by reducing mismatch-induced amplitude errors in the 1-bit DACs, but do not address pulse shape and timing errors. Unfortunately, as a DAC's input signal frequency is increased, more DAC elements tend to be toggled from sample-to-sample, so the DAC's linearity becomes increasingly limited by pulse shape and timing mismatches rather than amplitude mismatches. Hence, DACs that incorporate these techniques tend to exhibit high linearity for low frequency input signals, but their linearity degrades as the input signal frequency is increased.

As described in detail in the next section, the idea behind DEM is to pseudo-randomly select one of several possible usage patterns of the 1-bit DACs each sample period such that the error arising from unit DAC element mismatches is scrambled from sample to sample. Such scrambling causes the error arising from DAC element mismatches to have a noise-like structure that, ideally, is free of nonlinear distortion. Thus, DEM increases DAC linearity at the expense of decreasing its signal-to-noise ratio (SNR). However, since DEM converts nonlinear distortion into noise, it tends to have little effect on the peak signal-to-noise-and-distortion ratio (SNDR) of the DAC. In many high-resolution DAC applications, maximizing linearity is more critical than maximizing SNR, so the tradeoff offered by DEM is worthwhile.

Unfortunately, previously published DEM techniques require circuitry whose complexity grows exponentially with the number of bits of DAC resolution. Therefore, it is not practical to apply these DEM techniques to all of the 1-bit DACs in a high-resolution Nyquist-rate DAC. Instead, a partial DEM approach is usually taken wherein the usage pattern of only the 1-bit DACs with large weights is chosen pseudo-randomly, the rational being that the 1-bit DACs with large weights contribute the bulk of the mismatch error. However, mismatch-induced errors from the other 1-bit DACs as a group are still significant, particularly pulse shape and timing errors, so the approach is only a partial solution. Moreover, it can actually degrade linearity for high frequency DAC signals as described in [9].

This paper derives a set of conditions that are sufficient for an arbitrary DEM technique to eliminate pulse shape, timing, and amplitude errors arising from component mismatches as sources of nonlinear distortion in high-resolution DACs. It is demonstrated in [13] that a specific type of DEM can be used to eliminate mismatch-induced timing errors as sources of nonlinear distortion, and proposed in [14] and [15] that mapping,

Manuscript received November 1, 2007; revised March 26, 2008. Current version published September 10, 2008. This work was supported by the National Science Foundation under Award 0515286, by the corporate members of the UCSD Center for Wireless Communications, and by the University of California Discovery Program.

K. L. Chan is with the Institute of Microelectronics, 117685 Singapore (e-mail: chankl@ime.a-star.edu.sg).

J. Zhu and I. Galton are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92092 USA (e-mail: galton@ucsd.edu).

Digital Object Identifier 10.1109/JSSC.2008.2001931

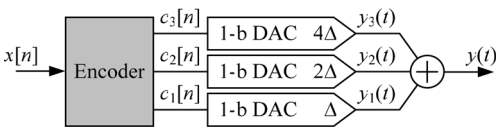

Fig. 1. High-level system diagram of a 3-bit power-of-two-weighted DAC.

a technique related to DEM, can be used to reduce the effects of mismatch-induced timing errors. However, the general result that DEM can be used to eliminate mismatch-induced pulse-shape errors as sources of nonlinear distortion has not been published previously to the knowledge of the authors. The paper also presents a highly linear, 100 MS/s, 0.18 μm CMOS, 14-bit Nyquist-rate DAC IC enabled by a new DEM technique that satisfies the sufficient conditions [12]. The complexity of the DEM technique does not grow exponentially with the number of bits of DAC resolution so it avoids the problems mentioned above.

The remainder of the paper consists of four sections. Section II defines the general architecture of a Nyquist-rate DAC, describes a model of mismatch-induced errors in the 1-bit DACs, explains how DEM works via a simple example, and derives the sufficient conditions mentioned above for the general architecture. Section III describes the specific DEM technique used in the IC prototype, and shows that it satisfies the sufficient conditions presented in the previous section. Section IV describes circuit details of the IC prototype. Section V describes measurement techniques and presents measured results.

II. Dynamic Element Matching in Continuous-Time DACs

A. Ideal Continuous-Time DAC Behavior

The input to a DAC is a sequence of digital codewords, each of which is interpreted by design convention to have a numerical value. In the following, the DAC's input sequence is *defined* to have $M+1$ possible values that range from $-M\Delta/2$ to $M\Delta/2$ in steps of Δ, where M is a positive integer and Δ is the minimum *step-size* of the input sequence. A typical circuit implementation of a DAC converts the sequence of input values into a sequence of continuous-time analog pulses. Ideally, the output of the DAC during the nth sample period, i.e., for $nT \le t < (n+1)T$, is

$$y(t) = a(t - nT)x[n] \tag{1}$$

where $x[n]$ is the value of the input sequence during the nth sample period, T is the duration of the sample period, and $a(t)$ is an analog pulse that is zero outside of $0 \le t < T$.

As a simple example, an 8-level (3-bit) *power-of-two-weighted DAC* architecture is shown in Fig. 1. It consists of an all-digital block called an *encoder* followed by three power-of-two-weighted 1-bit DACs, the outputs of which are summed to generate the overall DAC's output signal. Ideally, during the nth sample period, $nT \le t < (n+1)T$, the output of the ith 1-bit DAC is

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

where K_i is the *weight* of the 1-bit DAC. For this example, $K_i = 2^{i-1}, i = 1$, 2, and 3, and the input sequence can take on any of the values: $-3.5\Delta, -2.5\Delta, -1.5\Delta, -0.5\Delta$, 0.5 Δ, 1.5 Δ, 2.5 Δ, and 3.5 Δ. To satisfy (1), the encoder's output bits must satisfy

$$\Delta \sum_{i=1}^{3} 2^{i-1} \left(c_i[n] - \frac{1}{2} \right) = x[n]. \tag{3}$$

For each value of $x[n]$, there is exactly one set of encoder output bits $c_1[n]$, $c_2[n]$, and $c_3[n]$ that satisfy (3).

B. Mismatch Errors

As implied by (2), ideally only the sign of each pulse from the ith 1-bit DAC depends on its input bit, $c_i[n]$, and all of the 1-bit DACs have the same pulse shape, $a(t)$, scaled by $\Delta/2$ and their respective weights. Unfortunately, mismatches among nominally identical components used to implement the 1-bit DACs spoil this ideal performance in practice. The result is that the ideal behavior given by (2) is degraded to

$$y_i(t) = \begin{cases} K_i \dfrac{\Delta}{2} a(t - nT) + e_{\text{hi}}(t - nT), & \text{if } c_i[n] = 1 \\ -K_i \dfrac{\Delta}{2} a(t - nT) + e_{\text{li}}(t - nT), & \text{if } c_i[n] = 0 \end{cases} \tag{4}$$

where $e_{\text{hi}}(t)$ and $e_{\text{li}}(t)$ are *mismatch error pulses* caused by the component mismatches. It is assumed that $e_{\text{hi}}(t)$ and $e_{\text{li}}(t)$ are non-zero only for $0 \le t < T$. No other assumptions are made about $e_{\text{hi}}(t)$ and $e_{\text{li}}(t)$.

Equivalently, (4) can be written as

$$y_i(t) = K_i \Delta \alpha_i (t - nT) \underbrace{\left(c_i[n] - \frac{1}{2} \right)}_{=\pm\frac{1}{2}} + \beta_i(t - nT) \tag{5}$$

where

$$\alpha_i(t) = a(t) + \frac{e_{\text{hi}}(t) - e_{\text{li}}(t)}{K_i \Delta} \quad \text{and} \quad \beta_i(t) = \frac{e_{\text{hi}}(t) + e_{\text{li}}(t)}{2} \tag{6}$$

which can be verified by substituting (6) into (5) to obtain (4). This result is illustrated graphically in Fig. 2 for a 1-bit DAC whose pulses corresponding to $c_i[n] = 1$ and $c_i[n] = 0$ have drastically different shapes (for illustration purposes) as a result of the mismatch error pulses. As shown in the figure, each pulse can be viewed as the sum of two pulses, one proportional to $\alpha_i(t)$ with a sign that depends on $c_i[n]$ and one equal to $\beta_i(t)$ that is independent of $c_i[n]$.

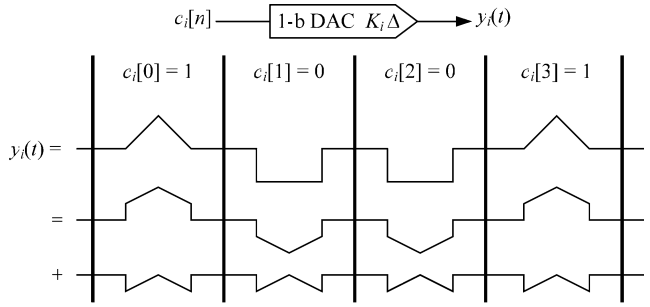

Fig. 2. Decomposition of the output of a 1-bit DAC with mismatch error pulses into a linear pulse sequence and an offset pulse sequence.

As indicated in Fig. 1, the overall DAC output is $y(t) = y_1(t) + y_2(t) + y_3(t)$, so it follows from (5) and (6) that during each sample interval, $nT \le t < (n+1)T$,

$$y(t) = a(t - nT)x[n] + \varepsilon(t) \tag{7}$$

where

$$\varepsilon(t) = \sum_{i=1}^{3} \left\{ \left(c_i[n] - \frac{1}{2} \right) [e_{\text{hi}}(t - nT) - e_{\text{li}}(t - nT)] + \beta_i(t) \right\} \tag{8}$$

is the error component in the output signal caused by the 1-bit DAC mismatch error pulses. Unfortunately, as implied by (8), $\varepsilon(t)$ is a nonlinear, deterministic function of the overall DAC's input sequence, $x[n]$, because it depends on $c_1[n]$, $c_2[n]$, and $c_3[n]$, which are deterministic nonlinear functions of $x[n]$.

C. A Dynamic Element Matching DAC Example

In the power-of-two-weighted DAC example described above, the 1-bit DACs with weights of 2 and 4 can be implemented as parallel combinations of 2 and 4 1-bit DACs with weights of 1, respectively. For example, $y_3(t)$ can be generated by adding the outputs of 4 such unity-weighted 1-bit DACs, each with the same input bit sequence, $c_3[n]$. Therefore, as depicted in Fig. 3(a), the power-of-two-weighted DAC example can (and, in practice, usually would) be implemented with 7 unity-weighted 1-bit DACs, one of which has an input of $c_1[n]$, two of which have an input of $c_2[n]$, and four of which have an input of $c_3[n]$.

Alternatively, as illustrated in Fig. 3(b), an 8-level DAC can be implemented with the same 7 unity-weighted 1-bit DACs as in the DAC of Fig. 3(a), but with an encoder that individually controls their inputs under the constraint

$$\Delta \sum_{i=1}^{7} \left(c_i[n] - \frac{1}{2} \right) = x[n]. \tag{9}$$

This type of DAC is called a *unity-weighted DAC* because it consists of unity-weighted 1-bit DACs whose inputs are controlled *individually* by the encoder.

It follows from (9) that whenever $x[n] = -3.5\Delta$ or $x[n] = 3.5\Delta$ the encoder's output bits, $c_1[n], \ldots, c_7[n]$, must all be zero or all be one, respectively. However, for all other input values there are more than one set of bit values for $c_1[n], \ldots, c_7[n]$ that satisfy (9). For example, to satisfy (9) when $x[n] = 2.5\Delta$ one of the $c_i[n]$ bits must be zero and the rest must be one, so in this case there are 7 possible sets of bit values for $c_1[n], \ldots, c_7[n]$ that satisfy (9). Therefore, in contrast to the power-of-two-weighted DAC, the encoder in the unity-weighted DAC has flexibility in choosing its output bit values for most input values.

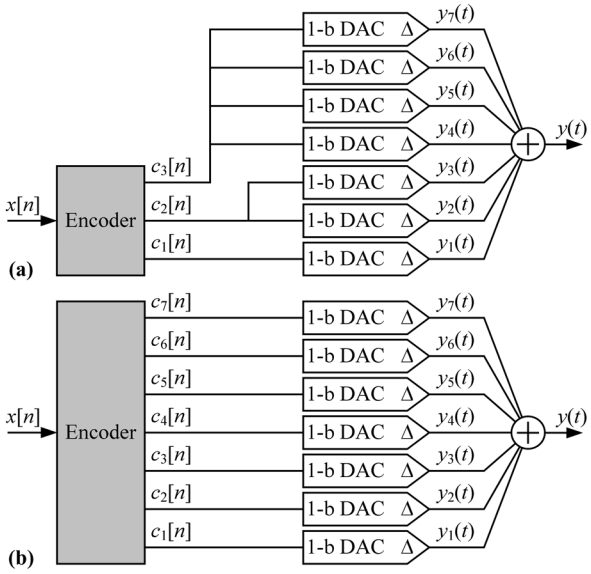

Fig. 3. High-level system diagram of (a) a 3-bit power-of-two-weighted DAC and (b) a 3-bit unity-weighted DAC.

The idea behind DEM is to exploit this flexibility to cause the error arising from mismatches to be white or spectrally shaped noise that is uncorrelated with the input signal instead of being nonlinearly related to the input signal. As an example, the 8-level unity-weighted DAC with an encoder designed to have the following properties is analyzed.

Property 1: Each sample period the encoder randomly chooses one of the possible sets of output bits, $c_1[n]$, $c_2[n], \ldots, c_7[n]$, that satisfy (9), such that its choice is statistically independent from the choices it makes in the other sample periods.

Property 2: For each input value, $x[n]$, the encoder makes its choice such that all of the possible sets of output bits that satisfy (9) have an equal probability of being chosen.

The analysis begins with the observation that there exists a set of sequences $\lambda_i[n]$, $i = 1, 2, \ldots, 7$, which are related to the encoder's input sequence and output bit sequences by

$$\Delta \left(c_i[n] - \frac{1}{2} \right) = \frac{1}{7} x[n] + \lambda_i[n] \tag{10}$$

and satisfy

$$\sum_{i=1}^{7} \lambda_i[n] = 0. \tag{11}$$

This can be verified by substituting (10) into (9) and applying (11). Properties 1 and 2 specify the statistics of the $\lambda_i[n]$ sequences through (10).

Property 1 implies that for any deterministic input sequence, $x[n]$, and each i, $\lambda_i[n]$, $n = 0, 1, \ldots$, is a sequence of random variables. As implied by (10), for each n the random variable $\lambda_i[n]$ depends on the value $x[n]$, so the statistical independence clause in Property 1 implies that the joint probability of $\lambda_i[n]$ and $\lambda_j[m]$ given $x[n]$ and $x[m]$ for any $m \neq n$ and any i and j is given by

$$\Pr\{\lambda_i[n], \lambda_j[m] | x[n], x[m]\} = \Pr\{\lambda_i[n] | x[n]\} \Pr\{\lambda_j[m] | x[m]\}. \quad (12)$$

Any two of the encoder's output bits can be interchanged without violating (9), so Property 2 implies that

$$\Pr\{c_i[n] = 1, c_j[n] = 0\} = \Pr\{c_i[n] = 0, c_j[n] = 1\} \quad (13)$$

for any $i \neq j$. The probability distributions of $c_i[n]$ and $c_j[n]$ can be written as

$$\begin{aligned} \Pr\{c_i[n] = b\} &= \Pr\{c_i[n] = b, c_j[n] = 0\} \\ &\quad + \Pr\{c_i[n] = b, c_j[n] = 1\}, \\ \Pr\{c_j[n] = b\} &= \Pr\{c_i[n] = 0, c_j[n] = b\} \\ &\quad + \Pr\{c_i[n] = 1, c_j[n] = b\}, \end{aligned} \quad (14)$$

where b can be 0 or 1, so (13) implies that $c_i[n]$ and $c_j[n]$ have identical probability distributions. Since this holds for every pair of the encoder's output bits, all of the encoder's output bits must have identical probability distributions for each n, so $\lambda_1[n], \ldots, \lambda_7[n]$ have identical probability distributions for each n. Therefore, taking the expectation of (11) further indicates that $\lambda_1[n], \ldots, \lambda_7[n]$ each have zero mean. This holds for each n regardless of $x[n]$, so it implies that $\lambda_1[n], \ldots, \lambda_7[n]$ are all uncorrelated with $x[n]$. Applying (12) further indicates that the expected value of $\lambda_i[n]\lambda_j[m]$ is zero regardless of the input sequence and regardless of i, j, n, and m, provided $m \neq n$.

The output of the overall DAC is the sum of the outputs of its 7 1-bit DACs, so it follows from (5) and (10) that the DAC's output signal during the nth sample period can be written as

$$y(t) = \alpha(t - nT)x[n] + \beta(t - nT) + e_{\mathrm{DAC}}(t) \quad (15)$$

where

$$\begin{aligned} \alpha(t) &= \frac{1}{7}\sum_{i=1}^{7} \alpha_i(t), \\ \beta(t) &= \sum_{i=1}^{7} \beta_i(t), \\ e_{\mathrm{DAC}}(t) &= \sum_{i=1}^{7} \lambda_i[n]\alpha_i(t - nT), \end{aligned} \quad (16)$$

and $\alpha_i(t)$ and $\beta_i(t)$ are given by (6). Thus, the overall DAC's output signal has the three distinct components shown in (15). In the following, they are referred to as the *signal pulse sequence*, the *offset pulse sequence*, and the *DAC noise*, respectively.

The signal pulse sequence consists of the analog pulses, $\alpha(t - nT)$, scaled linearly by the input sequence. As indicated in (16), $\alpha(t)$, is the average of the $\alpha_i(t)$ pulses from the individual 1-bit DACs. Thus, $\alpha(t)$ deviates somewhat from the ideal pulse, $a(t)$, but in most applications this is not a serious problem because it has little effect on the SNR or spurious-free dynamic range (SFDR) of the overall DAC.

The offset pulse sequence consists of the analog pulses, $\beta(t - nT)$. As implied by (15) and (16) the offset pulses are identical from period to period, independent of $x[n]$. Consequently, they result only in spurious tones at multiples of the sample frequency, so they do not degrade the SNR or the in-band SFDR of the overall DAC.

The DAC noise pulse sequence, $e_{\mathrm{DAC}}(t)$, is so-named because it has a noise-like structure. As indicated by (16) it is the sum of the $\alpha_i(t)$ pulses from the individual 1-bit DACs modulated by the $\lambda_i[n]$ sequences. Since for each i, $\lambda_i[n]$, $n = 0, 1, \ldots$, is a sequence of zero-mean random variables that are uncorrelated with $x[n]$, it follows that the DAC noise pulse sequence has zero mean and is uncorrelated with $x[n]$. Moreover, since the expected value of $\lambda_i[n]\lambda_j[m]$ is zero for $m \neq n$, the DAC noise pulse sequence in each period is uncorrelated with itself in any other period. Thus, as desired, DEM in this example causes mismatch error pulses from the 1-bit DACs to introduce zero-mean random DAC noise that is uncorrelated with itself from period to period and uncorrelated with the input sequence instead of nonlinear distortion.

D. Sufficient Conditions for DEM to Prevent Nonlinearity from Mismatch Error Pulses

As described above, the encoder in the power-of-two-weighted DAC has no flexibility in choosing its output bits; for every input value only one set of output bits are valid. Therefore, DEM is not possible in a power-of-two-weighted DAC. Nevertheless, the architecture is very efficient in the sense that the encoder in a B -bit power-of-two-weighted DAC controls only B 1-bit DACs. In contrast, DEM is possible in a unity-weighted DAC, as shown above by example, but the architecture is not very efficient in that the encoder in a B -bit unity-weighted DAC controls $2^B - 1$ 1-bit DACs. For example, the encoder in a 14-bit unity-weighted DAC would have to control 16 383 1-bit DACs which would be impractical in many applications.

A compromise between the two extremes of the unity-weighted DAC and the power-of-two-weighted DAC is the so-called *segmented DAC*. Ideally, the weights of the 1-bit DACs in a segmented DAC are such that the encoder still has sufficient flexibility to implement DEM, but the number of 1-bit DACs it must control is not an exponential function of the number of DAC bits as with the unity-weighted DAC. Such a segmented DAC is presented in the next section.

In the mean time a general DEM DAC architecture that can represent any segmented or unity-weighted DEM DAC is shown in Fig. 4 and analyzed below. It consists of a *DEM encoder*, i.e., an encoder designed to implement DEM, followed by N 1-bit DACs that operate according to (4) with weights $K_i, i = 1, 2, \ldots, N$. The output of the overall DAC is the sum of the outputs of the 1-bit DACs.

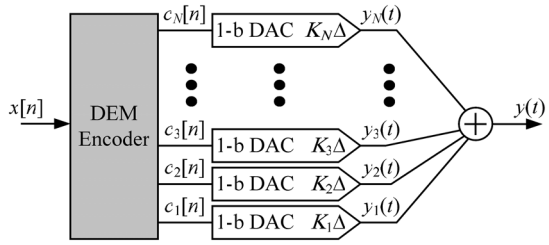

Fig. 4. High-level system diagram of a general dynamic element matching DAC.

As shown below, the following conditions on the DEM encoder are sufficient to prevent the mismatch error pulses from introducing nonlinear distortion in the overall DAC's output signal.

Condition 1: The DEM encoder's output bits satisfy

$$\Delta\left(c_i[n]-\frac{1}{2}\right)=m_i x[n]+\lambda_i[n] \tag{17}$$

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

$$\sum_{i=1}^{N}K_i m_i=1 \quad \text{and} \quad \sum_{i=1}^{N}K_i\lambda_i[n]=0. \tag{18}$$

Condition 2: The $\lambda_i[n]$ sequences for $i=1,2,\ldots,N$, are zero mean for each n regardless of $x[n]$, and the expected value of $\lambda_i[n]\lambda_j[m]$ is zero regardless of $x[n]$ and regardless of i,j,n, and m, provided $m\neq n$.

The DEM example described previously satisfies these conditions as a special case.

Since the output of the overall DAC is the sum of the outputs of its N 1-bit DACs, it follows from (5) and (17) that the DAC's output signal during the nth sample period can be written as

$$y(t)=\alpha(t-nT)x[n]+\beta(t-nT)+e_{\mathrm{DAC}}(t) \tag{19}$$

where

$$\begin{aligned}\alpha(t)&=\sum_{i=1}^{N}K_i m_i\alpha_i(t),\\ \beta(t)&=\sum_{i=1}^{N}\beta_i(t),\\ e_{\mathrm{DAC}}(t)&=\sum_{i=1}^{N}K_i\lambda_i[n]\alpha_i(t-nT),\end{aligned} \tag{20}$$

and $\alpha_i(t)$ and $\beta_i(t)$ are given by (6). Thus, as in the special case of the 8-level unity-weighted DEM DAC, the general DAC's output consists of a signal pulse sequence, an offset pulse sequence, and DAC noise, and the observations following (16) regarding the unity-weighted DEM DAC apply to the general case. The only differences are that N is not restricted to 7, the 1-bit DAC weights, K_i, appear in the expressions for $\alpha(t)$ and $e_{\mathrm{DAC}}(t)$, and the m_i constants appear in the expression for $\alpha(t)$.

Substituting the expression for $\alpha_i(t)$ from (6) into (20) and applying the rightmost equation in (18) gives

$$e_{\mathrm{DAC}}(t)=\frac{1}{\Delta}\sum_{i=1}^{N}\lambda_i[n]\left[e_{\mathrm{hi}}(t-nT)-e_{\mathrm{li}}(t-nT)\right] \tag{21}$$

during the nth sample period, $nT\leq t<(n+1)T$. Thus, the shape of the power spectrum of the DAC noise is determined by the power spectra of the mismatch error pulses from the 1-bit DACs. Nevertheless, Condition 2 ensures that if the DAC noise is sampled at a rate of $f_s=1/T$, the result is discrete-time white noise. Although it is beyond the scope of this work, Condition 2 can be relaxed to allow $\lambda_i[n]$ and $\lambda_i[m]$ to be correlated when $m\neq n$. Doing so would cause the $\lambda_i[n]$ sequences to be spectrally shaped, thereby affecting the spectral shape of the DAC noise pulse sequence.

III. Segmented Dynamic Element Matching

The specific DEM encoder used in the 14-bit DAC prototype IC is described in this section, and shown to satisfy the sufficient conditions presented above. It applies to the DEM DAC architecture shown in Fig. 4 with $N=36$ 1-bit DACs whose weights are

$$\begin{aligned}K_{2j-1}=K_{2j}=2^{j-1}, &\quad \text{for } j=1,2,\ldots,10, \text{ and}\\ K_j=1024, &\quad \text{for } j=21,\ldots,36.\end{aligned} \tag{22}$$

Thus, the first two 1-bit DACs each have a weight of unity, the next two each have a weight of 2, the next two each have a weight of 4, and so on, up to the 20th 1-bit DAC which has a weight of 512. The 21st through 36th 1-bit DACs each have a weight of 1024.

This combination of 1-bit DACs can represent 18431 levels. As described in Section II, the sequence of input codewords to a DEM DAC of the form shown in Fig. 4 with this set of 1-bit DACs can be interpreted as a sequence that takes on values in set $\{-9215\Delta,-9214\Delta,\ldots,9215\Delta\}$. This interpretation is useful when considering the behavior of the DEM DAC in the context of a larger signal processing system, such as a communication system, because it imparts a physical meaning to the input sequence in relation to the output waveform of the DAC as given by (19). However, the DEM encoder is implemented with digital logic, so it is convenient to base its signal processing operations on integer arithmetic. Therefore, the DEM encoder is designed to interpret the sequence of input codewords as a sequence of values given by

$$c[n]=\frac{x[n]}{\Delta}+9215 \tag{23}$$

which takes on values in the set $\{0,1,\ldots,18430\}$.

The DEM encoder is shown in Fig. 5. It consists of a tree of 35 blocks of digital logic called *switching blocks*, labeled $S_{k,r}$ for $k=1,\ldots,14$ and $r=1,\ldots,18$. As indicated in Fig. 5, there are two types of switching blocks. Switching blocks $S_{k,1}$ for $k=5,\ldots,14$ are called *segmenting* switching blocks, and the other switching blocks are called *non-segmenting* switching

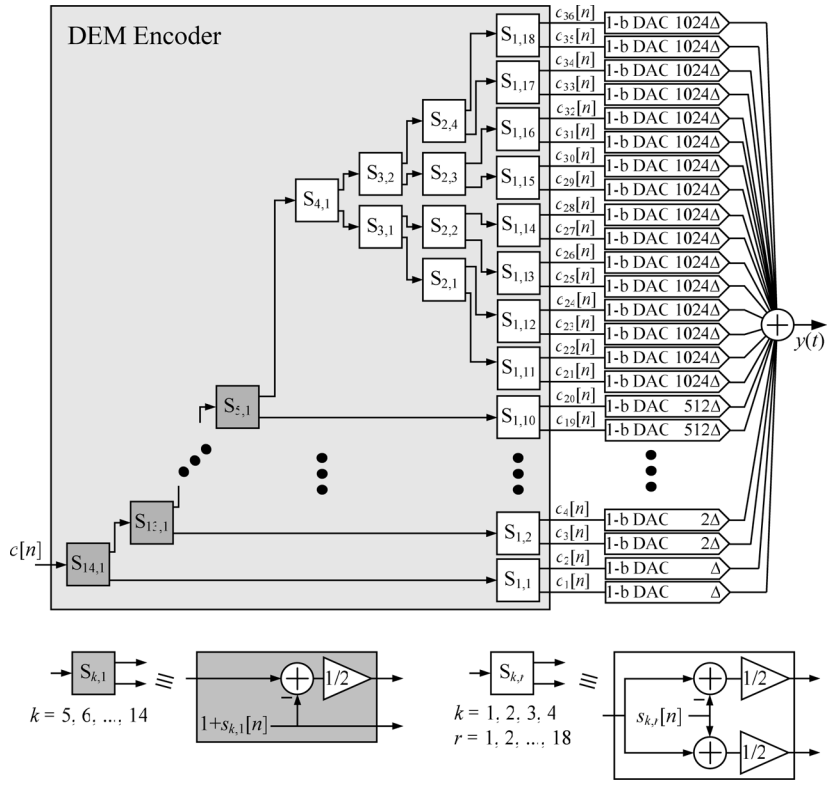

Fig. 5. Signal processing details of the DEM Encoder in the 14-bit DAC IC.

blocks. Each switching block calculates its two output sequences as a function of its input sequence and one of 14 pseudo-random 1-bit sequences, $d_k[n], k = 1, 2, \ldots, 14$, which are designed to well-approximate white random sequences that are independent from each other and $x[n]$, and each take on values of 0 and 1 with equal probability. The top and bottom outputs of the segmenting switching blocks $S_{k,1}$ are

$$\frac{1}{2}\left(c_{k,1}[n] - 1 - s_{k,1}[n]\right) \quad \text{and} \quad 1 + s_{k,1}[n] \tag{24}$$

respectively, where $c_{k,1}[n]$ is the switching block input sequence (with $c_{14,1}[n] = c[n]$), and

$$s_{k,1}[n] = \begin{cases} 0, \text{ if } c_{k,1}[n] = \text{odd} \\ 1, \text{ if } c_{k,1}[n] = \text{even},\ d_k[n] = 1 \\ -1 \text{ if } c_{k,1}[n] = \text{even},\ d_k[n] = 0. \end{cases} \tag{25}$$

Similarly, the top and bottom outputs of the non-segmenting switching blocks $S_{k,r}$ are

$$\frac{1}{2}\left(c_{k,r}[n] - s_{k,r}[n]\right) \quad \text{and} \quad \frac{1}{2}\left(c_{k,r}[n] + s_{k,r}[n]\right) \tag{26}$$

respectively, where

$$s_{k,r}[n] = \begin{cases} 0, \text{ if } c_{k,r}[n] = \text{even} \\ 1, \text{ if } c_{k,r}[n] = \text{odd}, d_k[n] = 1 \\ -1 \text{ if } c_{k,r}[n] = \text{odd}, d_k[n] = 0. \end{cases} \tag{27}$$

The $s_{k,r}[n]$ sequences are called *switching sequences*.

Inspection of Fig. 5 indicates that

$$c_{2j-1}[n] = \frac{1}{2}\left[1 + s_{15-j,1}[n] - s_{1,j}[n]\right]$$

and

$$c_{2j}[n] = \frac{1}{2}\left[1 + s_{15-j,1}[n] + s_{1,j}[n]\right] \tag{28}$$

for $j = 1, 2, \ldots, 10$. Following the arithmetic operations of the switching blocks in Fig. 5, it can be verified that

$$\begin{aligned} c_{21+u}[n] = \frac{1}{2}\Bigg[& 2^{-13}\frac{x[n]}{\Delta} + 1 - \sum_{j=0}^{9} s_{14-j,1}[n]2^{j-13} \\ & + (2w-1)s_{4,1}[n]2^{-3} \\ & + (2v-1)s_{3,w+1}[n]2^{-2} \\ & + (2y-1)s_{2,2w+v+1}[n]2^{-1} \\ & + (2z-1)s_{1,11+4w+2v+y}[n]\Bigg] \end{aligned} \tag{29}$$

where $u = 8w + 4v + 2y + z$, and $w, v, y, z \in \{0, 1\}$.

Therefore, (17) holds with

$$m_i = \begin{cases} 0, & i = 1, 2, \ldots 20 \\ 2^{-14}, & i = 21, \ldots 36 \end{cases} \tag{30}$$

and

$$\begin{aligned} \lambda_{2j-1}[n] &= \frac{\Delta}{2}\left[s_{15-j,1}[n] - s_{1,j}[n]\right] \\ \lambda_{2j}[n] &= \frac{\Delta}{2}\left[s_{15-j,1}[n] + s_{1,j}[n]\right] \end{aligned} \tag{31}$$

for $j = 1, 2, \ldots, 10$, and

$$\lambda_{21+u}[n] = -\frac{\Delta}{2}\left[\sum_{j=0}^{9} s_{14-j,1}[n]2^{j-13}\right.$$

$$+(2w-1)s_{4,1}[n]2^{-3} + (2v-1)s_{3,w+1}[n]2^{-2} + (2y-1)s_{2,2w+v+1}[n]2^{-1} + (2z-1)s_{1,11+4w+2v+y}[n]\Big] \quad (32)$$

for $u = 8w + 4v + 2y + z$ and $w, v, y, z \in \{0, 1\}$.

As described in the previous section, it is sufficient for the DEM encoder to satisfy Conditions 1 and 2 to ensure that mismatch error pulses from the 1-bit DACs do not introduce nonlinear distortion in the overall DAC's output signal. It follows from (22) and (30) that

$$\sum_{i=1}^{36} K_i m_i = \sum_{j=1}^{10} 2^{j-1}\left(m_{2j-1} + m_{2j}\right) + \sum_{j=11}^{18} 2^{10}\left(m_{2j-1} + m_{2j}\right) = 1 \quad (33)$$

and from (22), (31), and (32) that

$$\sum_{i=1}^{36} K_i \lambda_i[n] = \sum_{j=1}^{10} 2^{j-1}\left(\lambda_{2j-1}[n] + \lambda_{2j}[n]\right) + \sum_{j=11}^{18} 2^{10}\left(\lambda_{2j-1}[n] + \lambda_{2j}[n]\right) = 0 \quad (34)$$

so the DEM encoder satisfies Condition 1 as required. It follows from (25), (27), and the statistical properties of the $d_k[n]$ sequences that all the switching sequences are white and are zero mean for each n regardless of $x[n]$. Therefore, (31) and (32) indicate that the $\lambda_i[n]$ sequences must also be zero mean for each n regardless of $x[n]$ and the expected value of $\lambda_i[n]\lambda_j[m]$ must be zero regardless of $x[n]$ provided $m \neq n$, so the DEM encoder also satisfies Condition 2.

Each DEM encoder output sequence drives a 1-bit DAC, so it must be restricted to values of 0 and 1. It can be verified from (25), (27), (28), (29), and Fig. 5 that this always occurs provided $1023 \leq c[n] \leq 17407$. However, it can also be verified that if $c[n]$ is ever less than 1023 or greater than 17407, some of the DEM encoder outputs have a non-zero probability of taking on values that differ from 0 and 1. Therefore, it is necessary to restrict $c[n]$ to the set of values $\{1023, 1024, \ldots, 17407\}$, which means that $x[n]$ must be restricted to the set of values $\{-8192\Delta, -8191\Delta, \ldots, 8192\Delta\}$. This implies that the *usable input range* of the DEM DAC is 16385 levels.

This range restriction represents a fundamental tradeoff associated with segmentation. Recall that a 1-bit DAC with an integer weight of K_i is equivalent to the parallel combination of K_i unity-weighted 1-bit DACs. Thus, (22) implies that the segmented DEM DAC requires the equivalent of 18430 unity-weighted 1-bit DACs, 12% more than the 16384 1-bit DACs required by a unity-weighted DEM DAC with the same usable input range. This causes the segmented DEM DAC to have higher analog current consumption than the unity-weighted DEM DAC. Furthermore, the extra parallel unity-weighted 1-bit DACs introduce extra circuit noise, so it is often necessary to further increase the current consumption of each 1-bit DAC to compensate for this increase in noise. On the other hand, the segmented DEM DAC controls only 36 1-bit DACs whereas the unity-weighted DEM DAC controls 16384 1-bit DACs. In this sense, the segmented DEM DAC benefits from huge complexity reductions in the DEM encoder, 1-bit DAC control circuitry, and circuit layout.

The range restriction described above is an unavoidable consequence of segmentation. As shown in [16], DEM can prevent mismatches from causing nonlinear distortion only if $K_N - 1 \leq c[n] \leq M - K_N + 1$, where K_N denotes the largest 1-bit DAC weight and M is the equivalent number of unity-weighted 1-bit DACs, i.e.,

$$M = \sum_{i=1}^{N} K_i. \quad (35)$$

Thus, the greater the value of K_N, the more the usable input range of the DAC is reduced relative to its maximum possible range of $M + 1$ levels.

The segmented DEM DAC described above represents a balance between the conflicting goals of having a high level of segmentation to reduce complexity, which implies a large K_N, and having a small range restriction, which implies a small K_N. In general, the larger the number of 1-bit DACs with the maximum weight of K_N, the larger the number of 1-bit DACs overall, but the smaller the value of K_N. The segmented DEM DAC described above has 16 1-bit DACs with the maximum weight of K_N, thereby keeping the range restriction relatively small while only requiring 36 1-bit DACs overall.

IV. Circuit Details

A block diagram of the DAC IC is shown in Fig. 6. In addition to the DEM encoder and the bank of 36 1-bit DACs described above, it includes a low-voltage differential signaling (LVDS) receiver, a direct digital synthesizer (DDS) of the type presented in [17], a 14-bit pseudo-random number generator of the type presented in [18], and a clock buffer. Optionally, the digital input signal can be provided from off chip via the LVDS receiver, or by the DDS which generates a high spectral-purity sinusoid. The purpose of the pseudo-random number generator is to provide the 14 pseudo-random bits required by the DEM encoder as described in the previous section.

The usable input range of the DAC is 16385 levels as described above. However, a 14-bit number can only represent 16384 levels, so the input range of the DAC is further restricted by 1 level to accommodate a 14-bit input signal. This choice has a negligible affect on the SNDR of the DAC, but it saves two pins into the LVDS receiver.

A simplified circuit diagram of the first (lowest-weight) 1-bit DAC is shown in Fig. 7(a). It consists of a switch driver followed by a current-steering cell. The switch driver consists of a flip-flop to retime the $c_1[n]$ bit from the DEM encoder and NAND gates to generate the current-steering cell switch signals. The current-steering cell consists of a pMOS cascode current source with current steering switches. The differential clock signals and NAND gate circuitry are designed to achieve a low current-steering crossover point to minimize nonlinear coupling

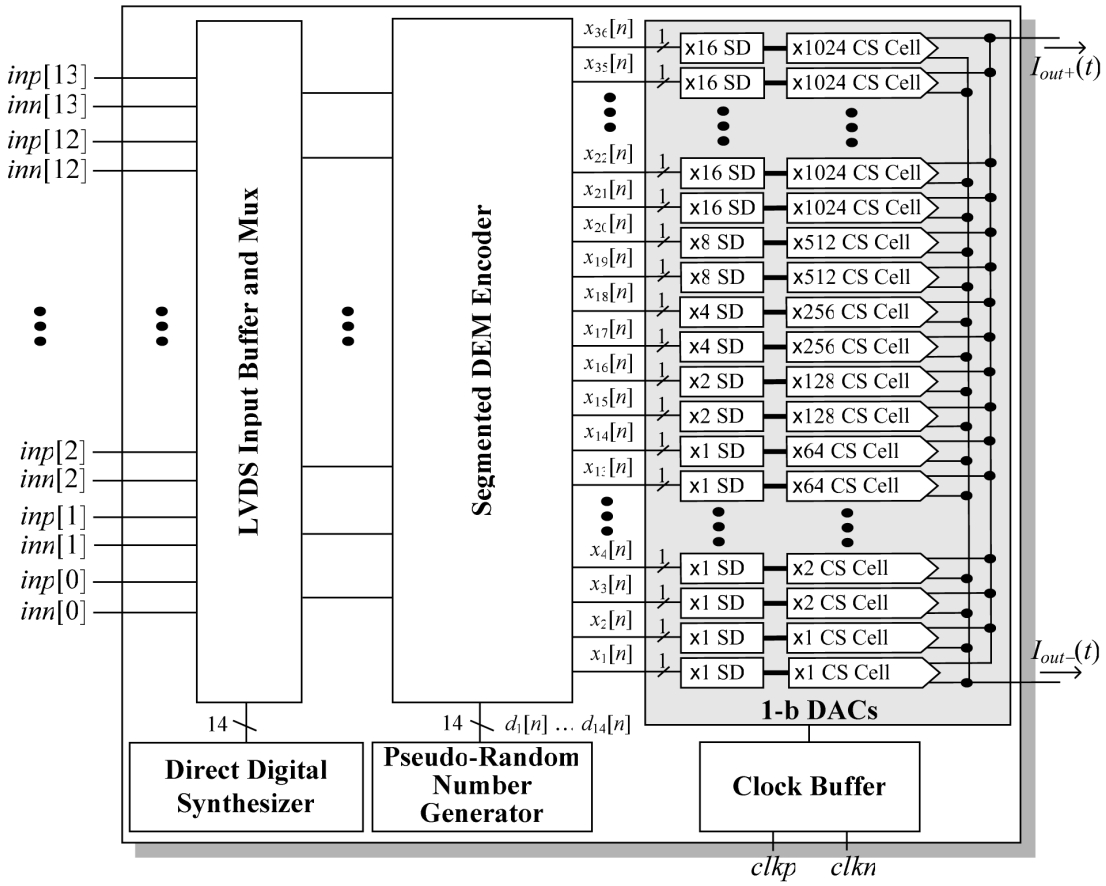

Fig. 6. System diagram of the 14-bit DAC IC.

of the differential outputs. The 1-bit DAC generates differential return-to-zero output current pulses, thereby minimizing the dependence of each pulse on previous values of the input sequence. During the first half of each sample period, the current-steering cell nominally steers 1 μA of current to the positive output or the negative output depending upon the 1-bit DAC's input bit value, and during the second half of the sample period it steers the current to a dummy load so that no current flows from either output.

The weights of the remaining 1-bit DACs are scaled relative to the first 1-bit DAC according to (22). Thus, with the differential output of the ith DAC written as $y_i(t) = I_{i+}(t) - I_{i-}(t)$, the ideal output of the 1-bit DAC is given by (2) with $\Delta = 2\mu$A, and

$$a(t) = \begin{cases} 1, & 0 \le t < T/2 \\ 0, & \text{otherwise.} \end{cases} \tag{36}$$

The unity weighted 1-bit DAC shown in Fig. 7(a) could have been used as a unit DAC element with which to construct the higher-weighted 1-bit DACs. For example, the 36th 1-bit DAC could have been constructed by combining 1024 of the 1-bit DACs of Fig. 7(a) in parallel. However, doing so would have wasted circuit area because the switch drivers would have been much larger than necessary. The unit switch driver in Fig. 7(a) is larger than needed to control the current-steering cell, but it cannot be made smaller because of technology limitations. Therefore, area can be saved by not scaling the switch drivers up in lock step with the 1-bit DAC weights. As indicated in Fig. 6, the switch drivers for the 14 lowest-weight 1-bit DACs have weights of unity, and those for the other 1-bit DACs have weights of $2^{-6}K_i$ where K_i is the weight of the ith 1-bit DAC for $15 \le i \le 36$.

To simplify the circuit layout, the current-steering cells in the third through eighth 1-bit DACs are scaled up from that shown in Fig. 7(a) by increasing the transistor widths. The current-steering cells in the remaining 1-bit DACs consist of parallel combinations of copies of the x8 current steering cell used in the seventh 1-bit DAC shown in Fig. 7(b). Copies of the 17th 1-bit DAC shown in Fig. 7(c) are used as unit elements in the 18th through 36th 1-bit DACs. For example, the 21st 1-bit DAC consists of four parallel copies of the 17th 1-bit DAC as shown in Fig. 7(d). The 17th 1-bit DAC is laid out in an 840 μm by 21 μm column, and the columns are replicated and connected in parallel as necessary for the 18th through 36th 1-bit DACs.

Had the 1-bit DACs been constructed as parallel copies of the first 1-bit DAC, not only would the circuit area have increased as described above, but the switch driver current consumption and ground bounce would have increased. This would have increased coupling of data-dependent signals into the clock and bias circuitry and likely would have degraded the SFDR of the overall DAC. However, by not constructing the 1-bit DACs as parallel copies of the first 1-bit DAC, systematic mismatch components inevitably have been added to the mismatch error pulses. DEM is relied upon to prevent the increased mismatch

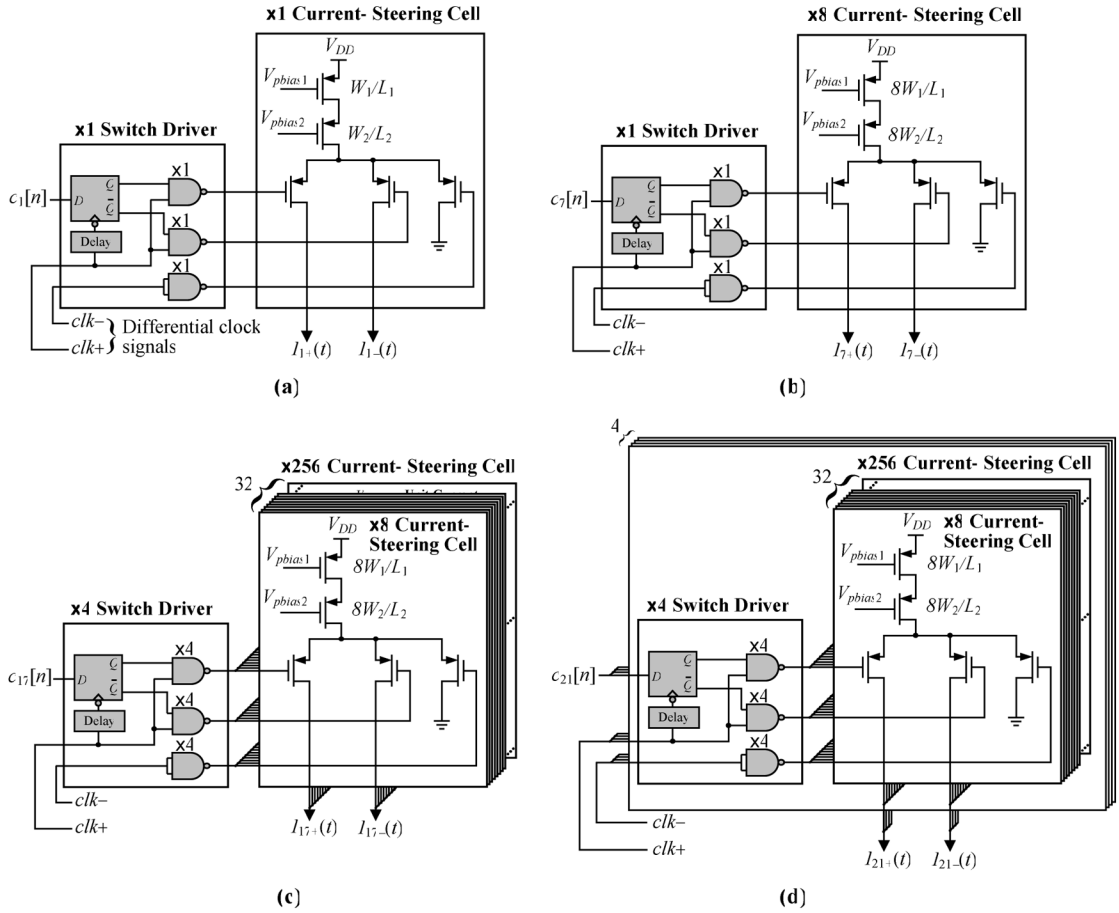

Fig. 7. Simplified circuit diagram of the (a) 1st, (b) 7th, (c) 17th, and (d) 21st 1-bit DACs.

error pulses from degrading the SFDR of the overall DAC, although some degradation of the SNR is inevitable.

As derived in the previous section, the DEM encoder prevents the mismatch error pulses introduced by the 1-bit DACs from causing nonlinear distortion in the overall DAC's output signal, $y(t) = I_{\text{out}+}(t) - I_{\text{out}-}(t)$. The definition of the mismatch error pulses in (4) includes the effects of all systematic and random mismatches among the unit switch driver and unit current-steering cells. Thus, DEM prevents pulse-shape, timing, and amplitude mismatches and the glitches they cause from introducing nonlinear distortion in the DAC's output signal.

This eases several circuit design issues because linearity no longer depends on good component matching and low glitch power; mismatches become a secondary concern so the 1-bit DACs can be optimized for low parasitic capacitance and high output impedance to improve linearity at high frequencies [5]. It also obviates the need for glitch reduction techniques such as having a single high-linearity return-to-zero switch circuit following the summed current-steering cell outputs [6].

The DAC was fabricated in a 0.18 μm CMOS process and is packaged in a QFN 64 package with ground down-bonding. The emphasis of the floor plan is to ensure that coupling from digital to analog circuits is minimized or data-independent. Wide supply and ground lines, and multiple supply pins with double bonding are used to minimize parasitic resistance and inductance in the supply lines. All ground lines are down-bonded to the exposed paddle of the QFN package to reduce parasitic inductance. ESD protection is implemented on all pads of the IC. The DAC's output current is converted to a voltage through an off-chip 50 Ω resistive differential load and coupled to a spectrum analyzer through a wideband transformer for testing as shown in Fig. 8.

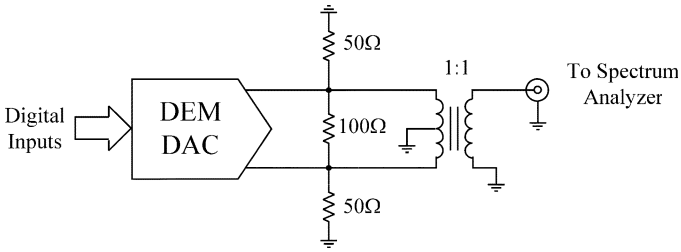

Fig. 8. Differential-to-single-ended conversion using a wideband transformer.

V. Measurement Results

The circuit design of the DAC is identical to that presented in [12], but the version presented in this paper has a modified layout which reduces some systematic mismatches within the prior layout to improve the SNDR. Although the new version of the IC exhibits similar linearity to the earlier version for a sample rate of 150 MHz, the measured results reported below are for a sample rate of 100 MHz. A clock coupling problem in

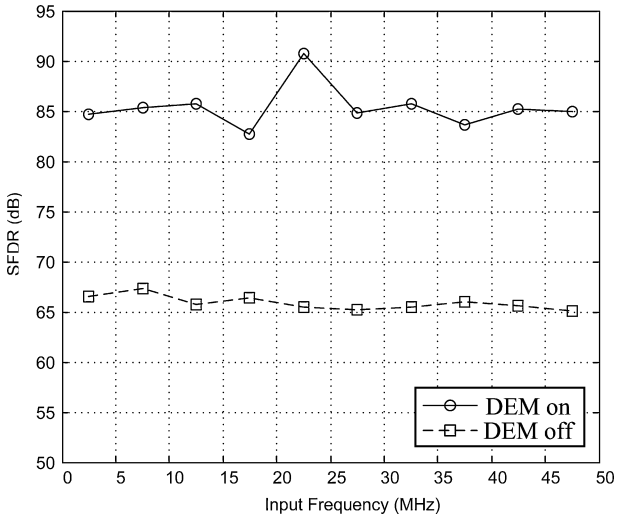

Fig. 9. Measured SFDR versus frequency with DEM enabled and disabled.

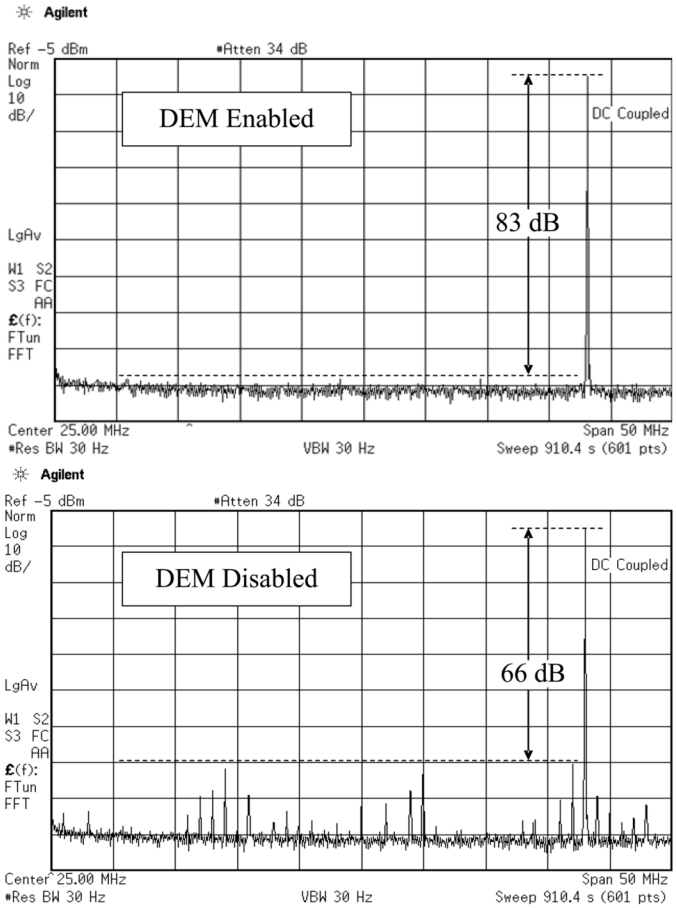

Fig. 10. Representative measured PSD plots of the DAC output with DEM enabled and disabled.

the original version of the IC causes the measured SFDR to be sensitive to the voltages of the power supply feeding the clock buffer. In [12] the coupling was minimized by setting the clock buffer's power supply to 2 V during measurement of the reported SFDR values. The same problem occurs with the new version of the IC at a sample rate of 150 MHz. However, at a sample rate of 100 MHz, the sensitivity is significantly reduced and excellent results are achieved with the power supplies set to their nominal 1.8 V design value. Despite the reduced clock frequency, the measured results represent state-of-the-art performance, and support the thesis of the paper, so giving up some speed for not having to adjust the power supply voltages is viewed by the authors as a worthwhile conservative tradeoff.

Measured SFDR values of the DAC versus input frequency with and without DEM enabled are shown in Fig. 9, and representative PSD plots are shown in Fig. 10. With DEM enabled for a sample rate of 100 MHz, the worst case SFDR value across the Nyquist band is 83 dB. As expected, the measured SFDR values show little dependence on signal frequency. With DEM disabled, the SFDR drops to less than 66 dB, which is expected given the lack of attention paid to minimizing the mismatch error pulses.

Measured peak SNDR values of the DAC versus input frequency with and without DEM enabled for full-scale and half-scale input sequences are shown in Fig. 11. The data suggests that DEM has relatively little effect (i.e., < 2 dB) on the SNDR for sinusoidal input signals with at least half scale amplitudes. This is an expected result because DEM simply converts nonlinear distortion into noise. For small-amplitude input signals, the results are more variable. They depend strongly on the DC offset of the input sequence because in both cases the DC offset affects which of the 1-bit DACs change state most frequently from sample to sample.

When DEM is enabled, the measured peak SNDR is identical to the measured peak SNR because DEM converts nonlinear distortion to noise as described above. Unfortunately, high-quality present-day spectrum analyzers have insufficient dynamic range to simultaneously measure the signal power and noise power of the DAC for a full-scale sinusoidal input signal, so measurement of the peak SNR must be performed in two steps. The signal power is measured with the input attenuation level of the spectrum analyzer increased to the point that overloading of the spectrum analyzer is avoided. In this case, the signal power can be measured, but the spectrum analyzer's noise obscures the DAC noise. For example, the noise floors shown in Fig. 10 are those of the spectrum analyzer, not the DAC. The power of the DAC noise can be measured by removing the sinusoidal input signal with a notch filter prior to the spectrum analyzer and increasing the gain of the spectrum analyzer's preamplifier (with the attenuation disabled) to the point that the DAC noise dominates the noise from the spectrum analyzer.

Although a similar procedure was used to obtain the SNR reported in [12], it must be noted that a mistake was made in [12]. In performing the noise power measurement, a low-pass filter was used to remove the sinusoidal signal component of the DAC prior to the spectrum analyzer. The noise removed by the low-pass filter was calculated on the assumption that its PSD is flat. After publication of [12], the authors obtained specialized notch filters with which to remove signal components at various frequencies and repeated the measurements. It was found that for sinusoidal input sequences the portion of the DAC noise previously removed by the low-pass filter has a PSD that is not flat but increases with frequency. In some cases this effect degrades the SNR to as low as 48 dB.

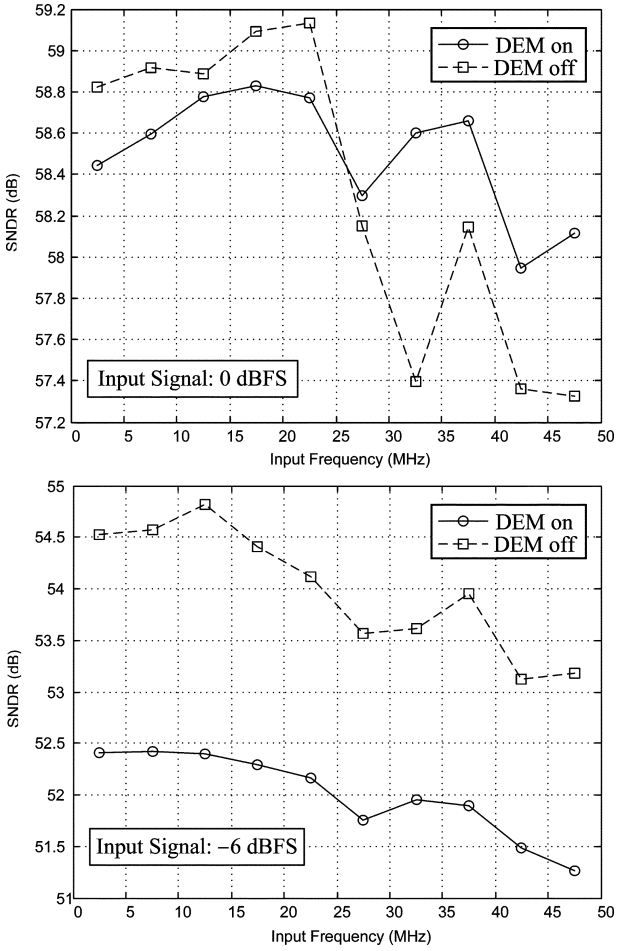

Fig. 11. Measured SNDR versus frequency with DEM enabled and disabled.

TABLE I
PERFORMANCE SUMMARY

Technology	TSMC 0.18 μm CMOS
Update Rate	100 MS/s
Package	QFN 64 with exposed paddle
Single-Tone SFDR @ 100 MS/s, 0 dBFS	≥ 83 dB across the Nyquist Band > 80 dB across 2nd Nyquist Band
Two-Tone SFDR @ 100 MS/s, −6 dBFS	> 84 dB ($f_{out2} - f_{out1}$ = 1MHz)
SNR @ 100 MS/s, 0 dBFS	≥ 58 dB across the Nyquist Band
Full-Scale Current	16 mA
Supply Voltages	Analog: 1.8 V, Digital: 1.8 V Clock Generator: 1.8 V
Power Dissipation @ 100 MS/s	125 mW (excluding the LVDS buffers)
Die Size (including bond pads)	4.8 mm × 2.4 mm
Active Area	3 mm^2

The noise measurements used to obtain the data shown in Fig. 11 were made without low-pass filters, so the measurement problem mentioned above has been avoided. The data indicate that layout improvements in the version of the IC reported in this paper result in much improved SNR values. This is largely because the measured DAC noise PSD from the new version of the IC does not increase significantly with frequency for sinusoidal input signals.

Table I summarizes the physical details of the IC and its measured performance. A die photograph is shown in Fig. 12. Fig. 13 shows measured performance of recent state-of-the-art CMOS DACs [3], [5], [6], [19]–[21]. It can be seen that while

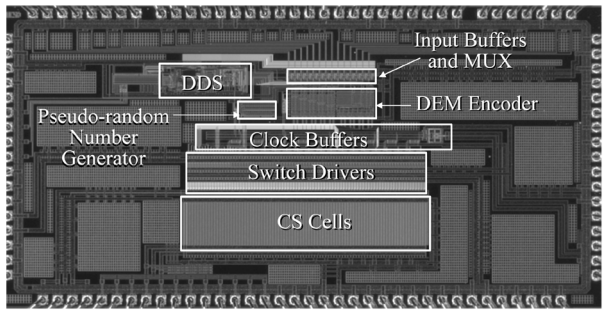

Fig. 12. Die photograph.

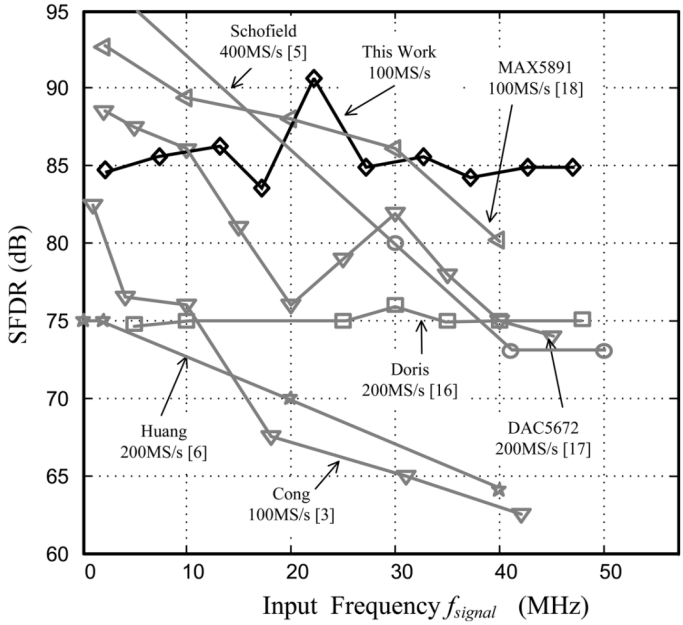

Fig. 13. Comparison of recent state-of-the-art CMOS DACs.

the SFDR of the DAC presented in this paper is relatively independent of the input signal frequency, as expected, most other DACs have SFDRs that degrade with signal frequency. For low input signal frequencies, the DACs that use calibration [3], [5], [6] tend to exhibit high SFDRs because the effect of the switching transient is reduced whereas the effect of static mismatches tends to dominate. The DAC presented in this paper achieves higher linearity over the Nyquist Band for input signals above 32 MHz than other CMOS DACs known to the authors.

ACKNOWLEDGMENT

The authors would like to thank B. Jewett at Agilent for his advice regarding analog circuit and layout design of the IC, A. Barlow and B. Matthews at Analog Devices for their assistance with the IC fabrication process, D. Kimball at UCSD for his advice regarding test equipment, and C. Vu at UCSD for populating test boards.

REFERENCES

[1] J. Bastos, A. M. Marques, M. S. J. Steyaert, and W. Sansen, "A 12-bit intrinsic accuracy high-speed CMOS DAC," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 1959–1969, Dec. 1998.

[2] G. A. M. Van Der Plas, J. Vandenbussche, W. Sansen, M. S. J. Steyaert, and G. G. E. Gielen, "A 14-bit intrinsic accuracy Q^2 random walk CMOS DAC," *IEEE J. Solid-State Circuits*, vol. 34, no. 12, pp. 1708–1717, Dec. 1999.
[3] Y. Cong and R. L. Geiger, "A 1.5-V 14-bit 100-MS/s self-calibrated DAC," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2051–2060, Dec. 2003.
[4] A. R. Bugeja and B. S. Song, "A self-trimming 14-b 100-MS/s CMOS DAC," *IEEE J. Solid-State Circuits*, vol. 35, no. 12, pp. 1841–1852, Dec. 2000.
[5] W. Schofield, D. Mercer, and L. St. Onge, "A 16 b 400 MS/s DAC with < -80 dBc IMD to 300 MHz and < -160 dBm/Hz noise power spectral density," in *IEEE ISSCC Dig. Tech. Papers*, 2003, pp. 126–127.
[6] Q. Huang, P. A. Francese, C. Martelli, and J. Nielsen, "A 200 MS/s 14b 97 mW DAC in 0.18 μm CMOS," in *IEEE ISSCC Dig. Tech. Papers*, 2004, pp. 364–365.
[7] J. Hyde, T. Humes, C. Diorio, M. Thomas, and M. Figueroa, "A 300-MS/s 14-bit digital-to-analog converter in logic CMOS," *IEEE J. Solid-State Circuits*, vol. 38, no. 5, pp. 734–740, May 2003.
[8] B. Jewett, J. Liu, and K. Poulton, "A 1.2 Gs/s 15b DAC for precision signal generation," in *IEEE ISSCC Dig. Tech. Papers*, 2005, pp. 110–111.
[9] 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. Solid-State Circuits*, vol. 39, no. 7, pp. 1064–1072, Jul. 2004.
[10] MB86061 12-bit 400 MS/s digital to analog converter. Fujitsu.
[11] K. L. Chan and I. Galton, "A 14b 100 MS/s DAC with fully segmented dynamic element matching," in *IEEE ISSCC Dig. Tech. Papers*, 2006, pp. 582–583, 675.
[12] K. L. Chan, J. Zhu, and I. Galton, "A 150 MS/s 14-bit segmented DEM DAC with greater than 83 dB of SFDR across the Nyquist band," in *Symp. VLSI Circuits*, Jun, 14, 2007.
[13] T. S. Kaplan, J. F. Jensen, C. H. Fields, and M.-C. F. Chang, "A 2-GS/s 3-bit $\Delta\Sigma$ -modulated DAC with tunable bandpass mismatch shaping," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 603–610, Mar. 2005.
[14] K. Doris, C. Lin, D. Leenaerts, and A. van Roermund, "D/A conversion: amplitude and time error mapping optimization," in *8th IEEE Int. Conf. Electronics, Circuits and Systems*, Sep. 2001, vol. 2, pp. 863–866.
[15] Y. Tang, H. Hegt, A. van Roermund, K. Doris, and J. Briaire, "Statistical analysis of mapping technique for timing error correction in current-steering DACs," in *IEEE Int. Symp. Circuits and Systems (ISCAS'07)*, May 2007, pp. 1225–1228.
[16] K. L. Chan, N. Rakuljic, and I. Galton, "Segmented dynamic element matching for high-resolution digital-to-analog conversion," *IEEE Trans. Circuits Syst. I: Reg. Papers*, submitted, under second review.
[17] D. D. Caro and A. G. M. Strollo, "High-performance direct digital frequency synthesizer in 0.25 μm CMOS using dual-slope approximation," *IEEE J. Solid-State Circuits*, vol. 40, no. 11, pp. 2220–2227, Nov. 2005.
[18] E. Fogleman, I. Galton, W. Huff, and H. T. Jensen, "A 3.3 V single-poly CMOS audio ADC delta-sigma modulator with 98 dB peak SINAD and 105 dB peak SFDR," *IEEE J. Solid-State Circuits*, vol. 35, no. 3, pp. 297–307, Mar. 2000.
[19] 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 μm CMOS," in *IEEE ISSCC Dig. Tech. Papers*, 2005, pp. 116–117.
[20] DAC5672: Dual 14-Bit 200 MSPS DAC. Texas Instruments, Inc.
[21] MAX5891: 16-Bit, 600 Msps high dynamic performance DAC with differential LVDS inputs. Maxim.

Kok Lim Chan received the B.Eng. and M.Eng. degrees in electrical engineering from Nayang Technological University, Singapore, in 1998 and 2000, respectively, and the Ph.D. degree in electrical engineering from University of California at San Diego in 2008.

Currently, he is a Senior Research Engineer at the Institute of Microelectronics, Singapore. His research interests include mixed-signal IC design and high-speed high-resolution data converters.

Jianyu Zhu received the B.S. degree from Tsinghua University, Beijing, China, in 2000 and the M.S. degree from the University of Arizona, Tucson, in 2002, both in electrical engineering.

He worked as a hardware engineer at HP's All-in-One inkjet printer division from 2002 to 2005. He is currently working on his Ph.D. degree at the University of California at San Diego. His research interests include high-speed and high-resolution data converters.

Ian Galton (M'92) received the Sc.B. degree from Brown University, Providence, RI, in 1984, and the M.S. and Ph.D. degrees from the California Institute of Technology, Pasadena, in 1989 and 1992, respectively, all in electrical engineering.

Since 1996 he has been a Professor of electrical engineering at the University of California at San Diego where he teaches and conducts research in the field of mixed-signal integrated circuits and systems for communications. Prior to 1996, he was with University of California Irvine, and prior to 1989 he was with Acuson and Mead Data Central. His research involves the invention, analysis, and integrated circuit implementation of critical communication system blocks such as data converters, frequency synthesizers, and clock recovery systems. In addition to his academic research, he regularly consults at several semiconductor companies and teaches industry-oriented short courses on the design of mixed-signal integrated circuits.

Dr. Galton has served on a corporate Board of Directors, on several corporate Technical Advisory Boards, as the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING, as a member of the IEEE Solid-State Circuits Society Administrative Committee, as a member of the IEEE Circuits and Systems Society Board of Governors, as a member of the IEEE International Solid-State Circuits Conference Technical Program Committee, and as a member of the IEEE Solid-State Circuits Society Distinguished Lecturer Program.