UNIVERSITY OF CALIFORNIA, SAN DIEGO

High Speed, High Resolution Digital-to-Analog Converters

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

Doctor of Philosophy

in

Electrical Engineering (Electronic Circuits and Systems)

by

Kok Lim Chan

Committee in charge:

Professor Ian Galton, Chair
Professor Peter M. Asbeck
Professor Chung-Kuan Cheng
Professor Lawrence Larson
Professor Thomas T. Liu

2007

UMI Number: 3294746

UMI Microform 3294746

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

The dissertation of Kok Lim Chan is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2007

To my family

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

CHAPTER 2

CHAPTER 3

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my academic advisor, Professor Ian Galton, for his guidance and encouragement throughout this research. Not only has he been my best teacher, he has also been a great friend and a great mentor. It has been a great privilege to have worked with him, and he has been an excellent role model for me.

I would like to thank Professor Lawrence Larson of the Department of Electrical and Computer Engineering, Professor Peter Asbeck of the Department of Electrical and Computer Engineering, Professor Chung-Kuan Cheng of the Department of Computer Science and Engineering, and Professor Thomas Liu from the Department of Radiology for serving on my doctoral committee.

I would like to thank my lab-mates Sudhakar Pamarti, Ashok Swaminathan, Eric Siragusa, Erica Poole, Andrea Panigada, Kevin Wang, Jianyu Zhu, Neveda Rakuljic and Gerry Taylor for their friendship, advice, help and patience. I am also very grateful to Jianyu Zhu for his help with the my tape-out, and to Neveda Rakuljic for her help with my paper.

I would also like to thank A*Star, Singapore for sponsoring me throughout my PhD program.

The Three chapters that compose this dissertation are intended to be published as separate papers listed below. Kok Lim Chan was the principle researcher/author on these papers.

K. Chan, Jianyu 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*, submitted.

K. L. Chan, N. Rakuljic, and I. Galton, “On the Design of Dynamic Element Matching Encoders for High-Linearity Digital-to-Analog Conversion,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, in preparation.

K. Chan, I. Galton, “A 14b 100MS/s DAC with Fully Segmented Dynamic Element-Matching,” *IEEE International Solid-State Circuits Conference*, pp. 582-583, 675, February, 2006.

VITA

1998	B. Eng., Nayang Technological University, Singapore
2000	M. Eng., Nayang Technological University, Singapore
2000-2001	Marvell, Singapore
2001-2002	Institute of Communication Research, Singapore
2007	Ph. D., University of California, San Diego

PUBLICATIONS

K. Chan, Jianyu 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*, submitted.

K. L. Chan, N. Rakuljic, and I. Galton, "On the Design of Dynamic Element Matching Encoders for High-Linearity Digital-to-Analog Conversion," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, in preparation.

K. Chan, I. Galton, "A 14b 100MS/s DAC with Fully Segmented Dynamic Element-Matching," *IEEE International Solid-State Circuits Conference*, pp. 582-583, 675, February, 2006.

K. L. Chan, J. Zhu, I. Galton, "A 150MS/s 14-bit Segmented DEM DAC with Greater than 83dB of SFDR Across the Nyquist band," *Symposium on VLSI Circuits, Digest of Technical Papers*, June 14, 2007.

ABSTRACT OF THE DISSERTATION

High Speed, High Resolution Digital-to-Analog Converters

by

Kok Lim Chan

Doctor of Philosophy in Electrical Engineering

(Electronic Circuits and Systems)

University of California, San Diego, 2007

Professor Ian Galton, Chair

Dynamic element matching (DEM) is widely-used in multi-bit DACs to prevent mismatches among nominally identical components from introducing non-linear distortion. By scrambling the usage pattern of the components from sample to sample, DEM causes the error arising from mismatches to be white or spectrally shaped noise that is free of non-linear distortion. DEM has long been used as a performance-enabling technique in delta-sigma data converters which require low-resolution but high-linearity DACs. More recently, segmented DEM architectures with reduced complexity have been developed that have made high-resolution Nyquist-rate DEM DACs practical CMOS DACs play have been used in many applications.

Chapter 1 proves that properly-designed dynamic element matching (DEM) eliminates pulse shape, timing, and amplitude errors arising from component mismatches as sources of non-linear 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 previously published DEM en-

coders, 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 DACs. These analytical results are demonstrated experimentally with a 0.18 μm CMOS 14-bit DAC IC that has a sample-rate of 150 MHz and worst-case, single and two-tone spurious-free dynamic ranges of 83 dB and 84 dB, respectively, across the Nyquist band.

In Chapter 2 it is shown that that there is a fundamental input range restriction to a segmented DEM DAC, regardless of how the DEM encoder is implemented. A general method of designing DEM encoder for unity-weighted DACs and segmented DEM DACs is then presented. The DEM encoders designed are optimal in the sense that they have a range restriction no worst than that fundamental input range restriction due to segmentation. The methods are demonstrated via examples of a 13-level unity-weighted DEM DAC and a pair of 14-bit segmented DEM DACs. The power dissipation versus complexity tradeoff implied by segmentation is also studied through the 14-bit examples.

In chapter 3, a 14-b 100 Ms/s Nyquist-rate DAC using a segmented dynamic element matching technique involving all its DAC elements is demonstrated. The DAC is implemented in 0.18 μm CMOS process and worst-case SFDR across the Nyquist bands are 74.4 dB and 78.9 dB for sample-rates of 100 MS/s and 70 MS/s, respectively.does not grow exponentially with the number of bits of DAC resolution, so it is practical for high-resolution DACs.

Chapter 1

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

I. Introduction

In a typical high-resolution (≥ 12 bit) Nyquist-rate 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 non-linear 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 non-linear 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 mis-

matches. 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 non-linear distortion. Thus, DEM increases DAC linearity at the expense of decreasing its signal-to-noise ratio (SNR) somewhat. 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 psuedo-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 presents a highly-linear, 150 MS/s, 0.18 μm CMOS, 14-bit Nyquist-rate DAC IC enabled by a new DEM technique applied to all of the 1-bit DACs [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. In addition to demonstrating the technique experimentally, the paper provides an analytical proof that the technique eliminates pulse shape, timing, and amplitude errors arising from component mismatches as sources of non-linear distortion in high-resolution DACs. Although [13] demonstrated that a specific type of DEM can be used to eliminate mismatch-induced timing errors as sources of non-linear distortion, the general result that DEM can also be used to eliminate mismatch-induced pulse-shape errors as sources of non-linear distortion has not been published previously to the knowledge of the authors. More generally, the paper presents a set of conditions that are sufficient for an arbitrary DEM technique to satisfy to ensure this effect.

The remainder of the paper consists of three 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 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 in-

put 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 Figure 1.1. 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: -3.5Δ , -2.5Δ, -1.5Δ, -0.5Δ, 0.5Δ, 1.5Δ, 2.5Δ, and 3.5Δ. To satisfy (1), the encoder's output bits must satisfy

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

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 *i*th 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

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

where $e_{hi}(t)$ and $e_{li}(t)$ are *mismatch error pulses* caused by the component mismatches. It is assumed that the mismatch error pulses are non-zero only for the duration of the sample period. Otherwise, no assumptions are made about $e_{hi}(t)$ and $e_{li}(t)$.

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

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

which can be verified by substituting (6) into (5) to obtain (4). This result is illustrated graphically in Figure 1.2 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 $c_i[n]$.

As indicated in Figure 1.1, 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

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

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 non-linear, 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 non-linear 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 Figure 1.3a, 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 Figure 1.3b, an 8-level DAC can be implemented with the same 7 unity-weighted 1-bit DACs as in the DAC of Figure 1.3a, but with an encoder that individually controls their inputs under the constraint

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

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, $c_1[n]$, …, $c_7[n]$, 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 $c_1[n]$, …, $c_7[n]$ 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 $c_1[n]$, …, $c_7[n]$ 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.

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 non-linearly 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, $c_1[n]$, $c_2[n]$, …, $c_7[n]$, 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

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

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

$$\Pr\left\{\lambda_i[n], \lambda_j[m] \middle| x[n], x[m]\right\} = \Pr\left\{\lambda_i[n] \middle| x[n]\right\} \Pr\left\{\lambda_j[m] \middle| x[m]\right\}. \tag{12}$$

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

$$\Pr\left\{c_i[n] = 1, c_j[n] = 0\right\} = \Pr\left\{c_i[n] = 0, c_j[n] = 1\right\}, \tag{13}$$

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

$$\begin{aligned} \Pr\left\{c_i[n] = b\right\} &= \Pr\left\{c_i[n] = b, c_j[n] = 0\right\} + \Pr\left\{c_i[n] = b, c_j[n] = 1\right\}, \\ \Pr\left\{c_j[n] = b\right\} &= \Pr\left\{c_i[n] = 0, c_j[n] = b\right\} + \Pr\left\{c_i[n] = 1, c_j[n] = b\right\}, \end{aligned} \tag{14}$$

where b can be 0 or 1, so (13) implies that $c_i[n]$ and $c_j[n]$ have identical probability distri-

butions. 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 $\lambda_1[n]$, …, $\lambda_7[n]$ have identical probability distributions for each n. Therefore, taking the expectation of (11) further indicates that $\lambda_1[n]$, …, $\lambda_7[n]$ each have zero mean. This holds for each n regardless of $x[n]$, so it implies that $\lambda_1[n]$, …, $\lambda_7[n]$ 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

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

where

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

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 pulse sequence*, 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, $e_{DAC}(t)$, 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 non-linear distortion.

D. Sufficient Conditions for DEM to Prevent Non-Linearity 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 Figure 1.4 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.

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

Condition 1: The DEM encoder's output bits satisfy

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

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

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

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

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

where

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

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 a DAC noise pulse sequence, 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 $e_{DAC}(t)$, 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

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

during the nth sample period, $nT \le t < (n+1)T$. 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 Figure 1.4 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 20*th* 1-bit DAC which has a weight of 512. The 21*st* through 36*th* 1-bit DACs each have a weight of 1024.

The DEM encoder is shown in Figure 1.5. 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 Figure 1.5, 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 blocks. Each switching block has one input and two outputs. As indicated in Figure 1.5, each switching block calculates its two output sequences as a function of its input sequence and a so-called *switching sequence*. For the segmenting switching blocks the switching sequences are

$$s_{k,1}[n] = \begin{cases} 0, & \text{if } x_{k,1}[n] = \text{odd}, \\ 1, & \text{if } x_{k,1}[n] = \text{even}, d_k[n] = 1, \\ -1 & \text{if } x_{k,1}[n] = \text{even}, d_k[n] = 0, \end{cases} \tag{23}$$

and for the non-segmenting switching blocks the switching sequences are

$$s_{k,r}[n]=\begin{cases}0, & \text{if } x_{k,r}[n]=\text{even},\\ 1, & \text{if } x_{k,r}[n]=\text{odd},\ d_k[n]=1,\\ -1 & \text{if } x_{k,r}[n]=\text{odd},\ d_k[n]=0,\end{cases} \tag{24}$$

where $d_k[n]$, k = 1, 2, …, 14 are 14 pseudo-random 1-bit sequences that are designed to well-approximate white random processes that are independent from each other and $x[n]$, and take on values of 0 and 1 with equal probability.

For this DEM encoder, $x[n]$ can take on any value in the set $\{-8192\Delta, -8191\Delta, \ldots, 8192\Delta\}$. It can be verified by following the arithmetic operations shown in Figure 1.5 given (23) and (24) that for this set of input values, each output, $c_i[n]$, from the DEM encoder is restricted values of 0 and 1 as required.

Inspection of Figure 1.5 indicates that

$$\begin{aligned} c_{2j-1}[n]&=\frac{1}{2}\Big[1+s_{15-j,1}[n]-s_{1,j}[n]\Big], \quad \text{and}\\ c_{2j}[n]&=\frac{1}{2}\Big[1+s_{15-j,1}[n]+s_{1,j}[n]\Big], \end{aligned} \tag{25}$$

for j = 1, 2, …, 10. Following the arithmetic operations of the switching blocks in Figure 1.5, it can be verified that

$$\begin{aligned} c_{21+u}[n]=\frac{1}{2}\Bigg[& 2^{-13}\frac{x[n]}{\Delta}+1-\sum_{j=0}^{9}s_{14-j,1}[n]2^{j-13}+(2w-1)s_{4,1}[n]2^{-3}\\ & +(2v-1)s_{3,w+1}[n]2^{-2}+(2y-1)s_{2,2w+v+1}[n]2^{-1}\\ & +(2z-1)s_{1,11+4w+2v+y}[n]\Bigg], \end{aligned} \tag{26}$$

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

Therefore, (17) holds with

$$m_i=\begin{cases}0, & i=1,2,\ldots 20,\\ 2^{-14}, & i=21,\ldots 36,\end{cases} \tag{27}$$

and

$$\lambda_{2j-1}[n] = \frac{\Delta}{2}\Big[s_{15-j,1}[n] - s_{1,j}[n] \Big], \qquad \lambda_{2j}[n] = \frac{\Delta}{2}\Big[s_{15-j,1}[n] + s_{1,j}[n] \Big], \tag{28}$$

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

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

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 non-linear distortion in the overall DAC's output signal. It follows that from (22) and (27) that

$$\sum_{i=1}^{36} K_i m_i = \sum_{j=1}^{10} 2^{j-1}\left(m_{2j-1} + m_{2j} \right) + \sum_{j=11}^{18} 2^{10}\left(m_{2j-1} + m_{2j} \right) = 1, \tag{30}$$

and from (22), (28), and (29) that

$$\sum_{i=1}^{36} K_i \lambda_i[n] = \sum_{j=1}^{10} 2^{j-1}\left(\lambda_{2j-1}[n] + \lambda_{2j}[n] \right) + \sum_{j=11}^{18} 2^{10}\left(\lambda_{2j-1}[n] + \lambda_{2j}[n] \right) = 0, \tag{31}$$

so the DEM encoder satisfies Condition 1 as required. It follows from (23), (24) 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, (28) and (29) 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.

As mentioned above, $x[n]$ can take on any value in the set $\{-8192\Delta, -8191\Delta, \ldots,$

8192Δ}. A unity-weighted DEM DAC capable of handling such an input sequence would require 16,384 1-bit DACs each with a weight of 1. In contrast, the segmented DEM DAC described above requires only 36 1-bit DACs with weights, K_i, given by (22). In this sense, the segmentation implied by (22) provides a huge reduction in DEM encoder complexity compared to the unity-weighted DEM DAC.

However, a price is paid for this benefit. 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 18,430 unity-weighted 1-bit DACs, approximately 12% more than the number of 1-bit DACs required for the corresponding unity-weighted DEM DAC. Indeed, it can be verified by following the arithmetic operations shown in Figure 1.5 in conjunction with (23) and (24) that $c_{21}[n]$ to $c_{36}[n]$ have a non-zero probability of not being 0 or 1 whenever $|x[n]|$ exceeds 8192Δ. This could be prevented by further constraining the switching sequences beyond (23) and (24), but the result would be that some of the switching sequences would have non-zero means that are non-linear functions of $x[n]$. This would violate Condition 2. Moreover, simulations indicate that constraining the switching sequences in this fashion to allow a larger range of input values causes the mismatch error pulses from the 1-bit DACs to introduce harmonic distortion in the overall DAC's output signal.

IV. Circuit Details and Measurement Results

A block diagram of the DAC IC is shown in Figure 1.6. 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 [14], a 14-bit pseudo-random number generator of the type presented in [15], 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 full-scale 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.

A simplified circuit diagram of the first (lowest-weight) 1-bit DAC is shown in Figure 1.7 (a). It consists of a switch driver followed by a current-steering cell. The switch driver circuit 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 non-linear coupling 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 $\Delta = 2$ μA, and

$$a(t) = \begin{cases} 1, & 0 \le t < T/2, \\ 0, & \text{Otherwise.} \end{cases} \tag{32}$$

The unity weighted 1-bit DAC shown in Figure 1.7 (a) could have been used as a unit DAC element with which to construct the higher-weighted 1-bit DACs. For example, the 36*th* 1-bit DAC could have been constructed by combining 1024 of the 1-bit DACs of Figure 1.7 (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 Figure 1.7 (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 Figure 1.6, 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 $15 \leq i \leq 36$.

To simplify the circuit layout, the current-steering cells in the third through eighth 1-bit DACs are scaled up from that shown in Figure 1.7 (a) by increasing the transistor widths. The current-steering cells in the remaining 1-bit DACs consist of parallel combinations of copies of the ×8 current steering cell used in the seventh 1-bit DAC shown in Figure 1.7 (b). Copies of the 17*th* 1-bit DAC shown in Figure 1.7 (c) are used as unit elements in the 18*th* through 36*th* 1-bit DACs. For example, the 21*st* 1-bit DAC consists of 4 parallel copies of the 17*th* 1-bit DAC as shown in Figure 1.7 (d). The 17*th* 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 18*th* through 36*th* 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 error pulses from degrading the SFDR of the overall DAC, although a slight 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 non-linear distortion in the overall DAC's output signal, $y(t) = I_{out+}(t) - I_{out-}(t)$. 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 non-linear 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 is 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 para-

sitic 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 50Ω resistive differential load and coupled to a spectrum analyzer through a wideband transformer for testing as shown in Figure 1.8. The measured SFDR of the DAC versus input frequency is shown Figure 1.12, and representative PSD plots with and without DEM enabled are shown in Figure 1.9. With DEM enabled for a sample-rate of 150 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 56 dB, which is expected given the lack of attention paid to minimizing the mismatch error pulses. A full performance summary and die photograph are shown in Figures 1.10 and 1.11, respectively.

With DEM enabled, the DAC's linearity is limited by parasitic coupling between the digital circuitry and the clock buffer circuitry. Prior to fabrication, the DAC was simulated to predict its SFDR with transistor-level analog circuitry and a package model but with behavioral digital circuitry. The resulting SFDR prediction was 86dB. Unfortunately, the package model under-estimated the inductance of the exposed paddle ground connection. Subsequently, simulation with a revised package model was performed, and the predicted SFDR was reduced by 10 dB. Further investigation revealed that the digital circuitry generates signal dependent interference that couples into the clock buffer circuitry through the power supply lines. Because of this coupling problem, the DAC's SFDR was found to be fairly sensitive to the voltages of the power supplies feeding the

clock driver and the clock generator. To minimize the coupling, the clock supply voltage was set to 2 V, during measurement of the reported SFDR values. In contrast, the SFDR was almost constant over an analog power supply voltage range of 1.6 V to 1.9 V, so it was left at 1.8 V during measurement of the reported SFDR values.

In Figure 1.12 shows measured performance of recent state-of-the-art CMOS DACs [3, 5, 6, 16-18]. It can be seen that the SFDR of the DAC presented in this paper is relatively independent of the input signal frequency, as expected, whereas 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 mismatch tends to dominate. The DAC presented in this paper achieves higher linearity over the Nyquist Band for input signals above 45 MHz than all other CMOS DACs known to the authors.

V. Chapter Acknowledgements

The Text of this chapter is to be submitted for review and publication as a regular paper in the *IEEE Journal of Solid State Circuits*. The dissertation author was the primary researcher. Ian Galton supervised the research that forms the basis of the chapter.

K. Chan, Jianyu 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*, submitted.

VI. FIGURES

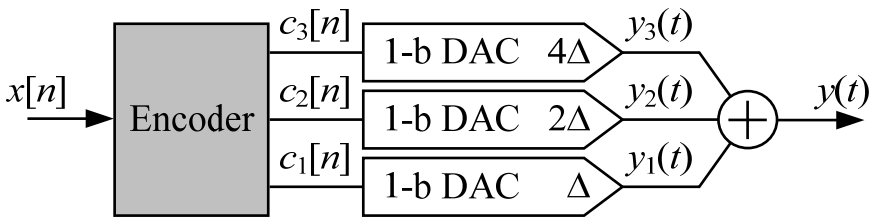


Figure 1.1: High-level system diagram of a 3-bit power-of-two-weighted DAC.

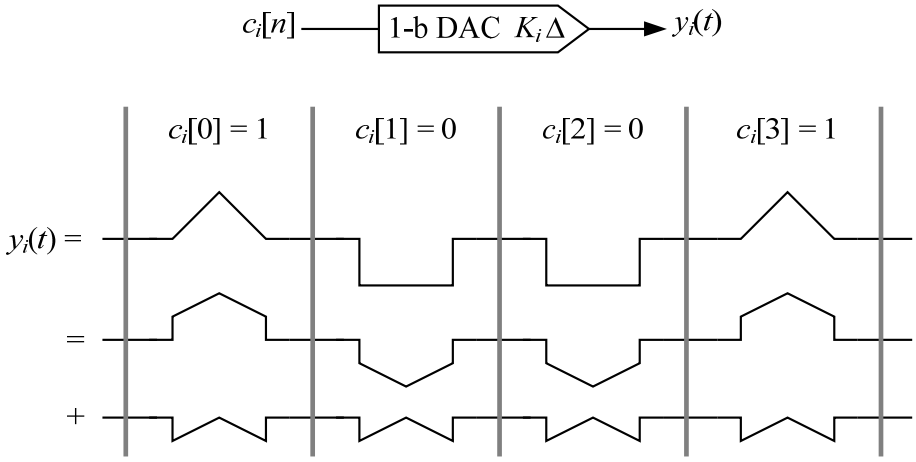


Figure 1.2: Decomposition of the output of a 1-bit DAC with mismatch error pulses into a linear pulse sequence and an offset pulse sequence.

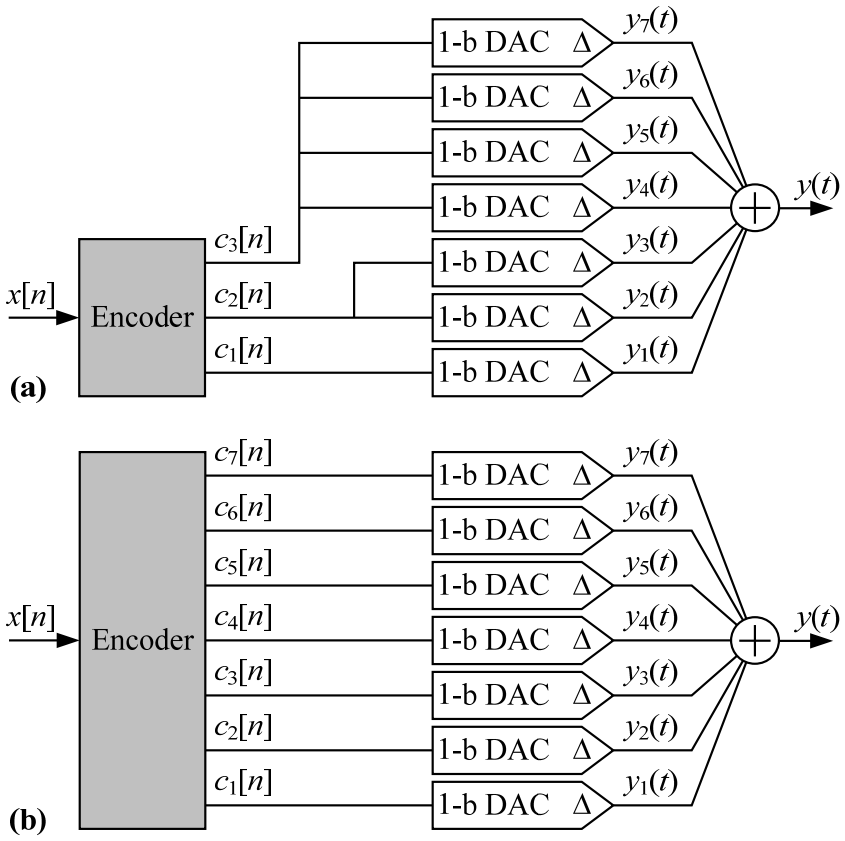


Figure 1.3: High-level system diagram of (a) a 3-bit power-of-two-weighted DAC and (b) a 3-bit unity-weighted DAC.

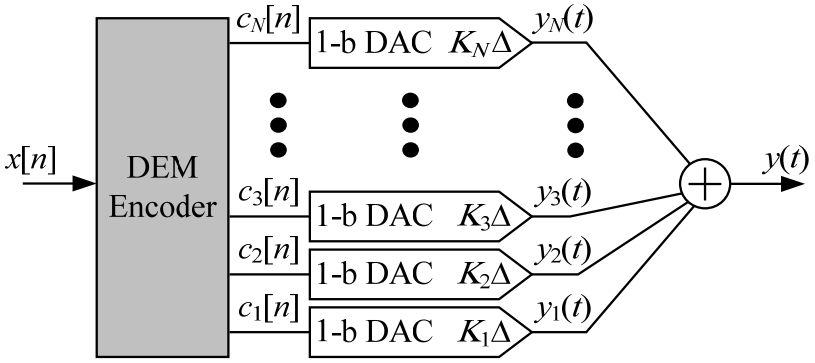
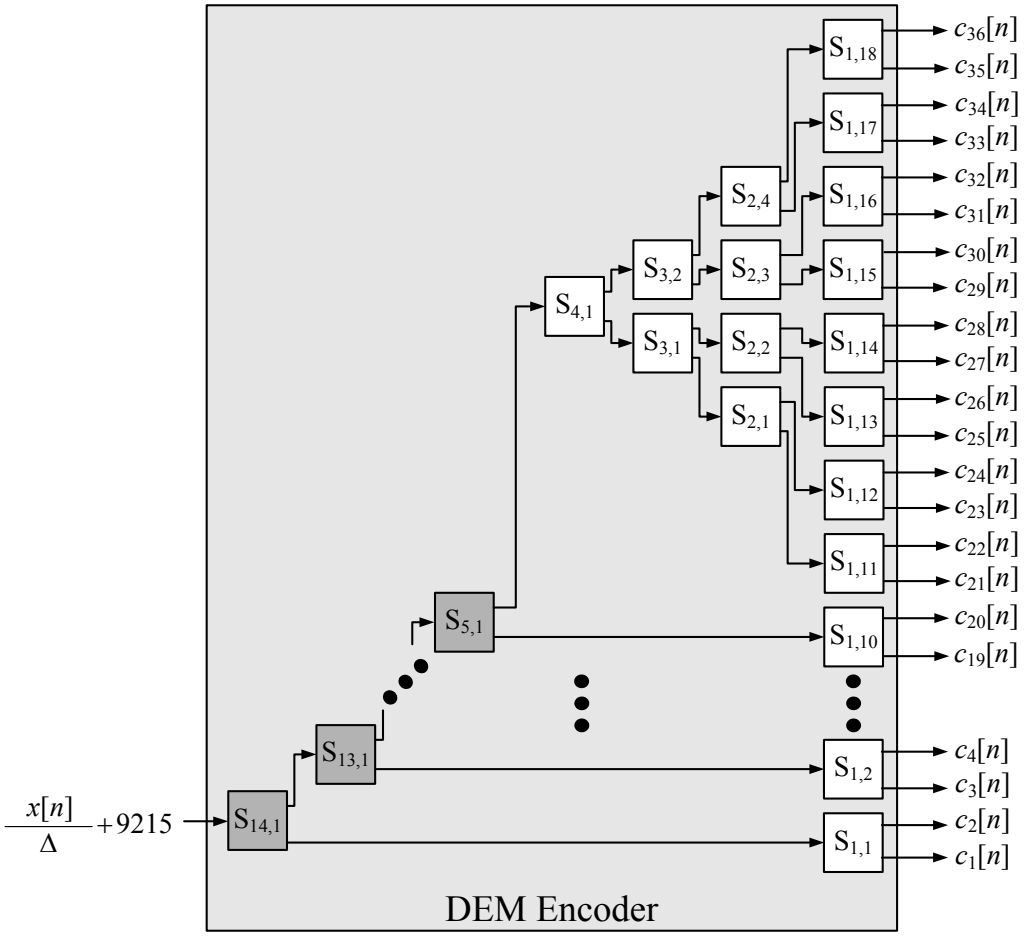
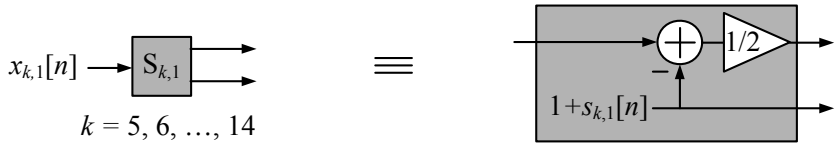
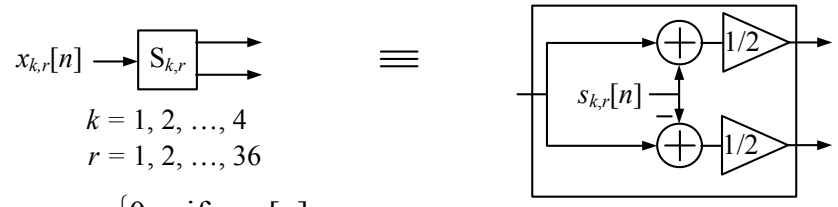


Figure 1.4: High-level system diagram of a general dynamic element matching DAC.





$$s_{k,1}[n]=\begin{cases}0, & \text{if } x_{k,1}[n]=\text{odd}\\ \pm 1, & \text{if } x_{k,1}[n]=\text{even (chosen randomly)}\end{cases}$$



$$s_{k,r}[n]=\begin{cases}0, & \text{if } x_{k,r}[n]=\text{even}\\ \pm 1, & \text{if } x_{k,r}[n]=\text{odd (chosen randomly)}\end{cases}$$

Figure 1.5: Details of the DEM Encoder in the 14-bit DAC IC.

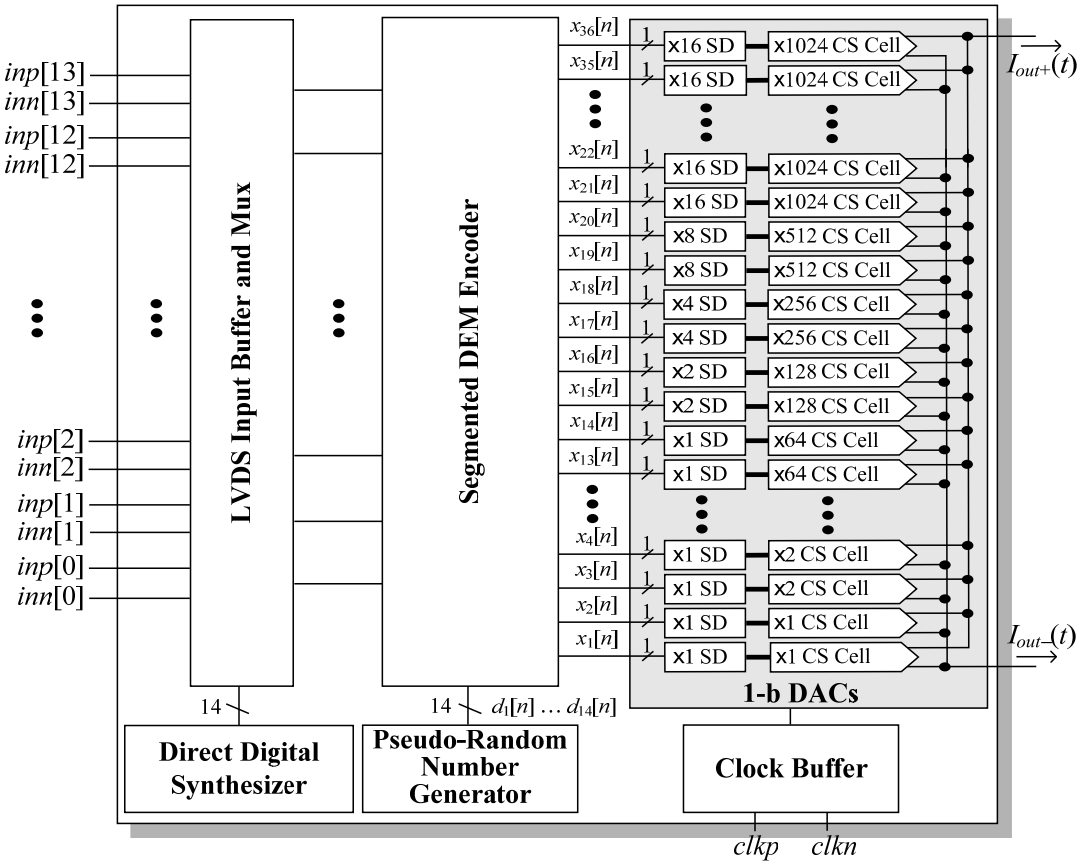


Figure 1.6: System diagram of the 14-bit DAC IC.

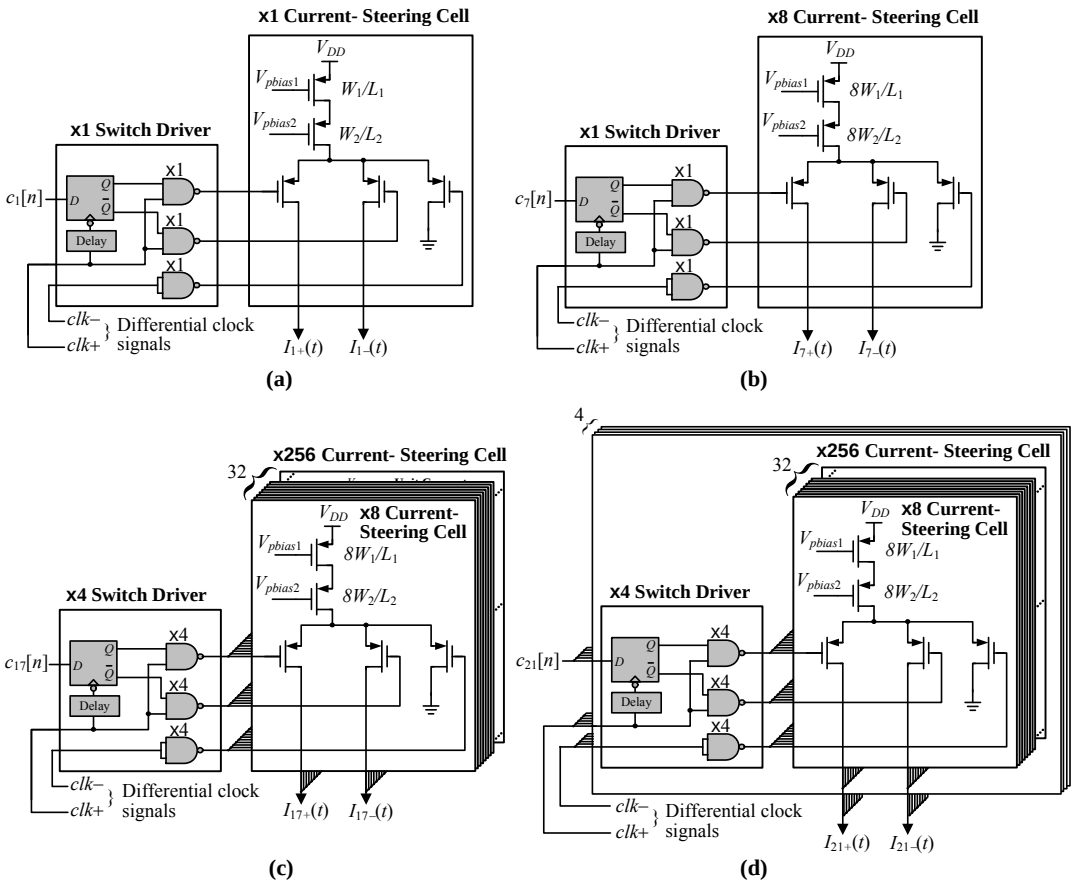


Figure 1.7: Simplified circuit diagram of the (a) first, (b) seventh, (c) seventeenth, and (d) twenty-first 1-bit DACs.

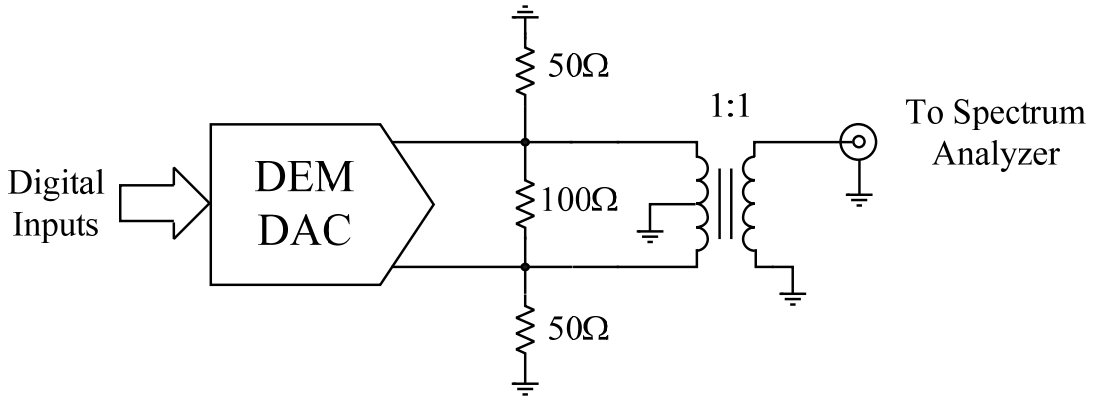


Figure 1.8: Differential-to-single-ended conversion using a wideband transformer.

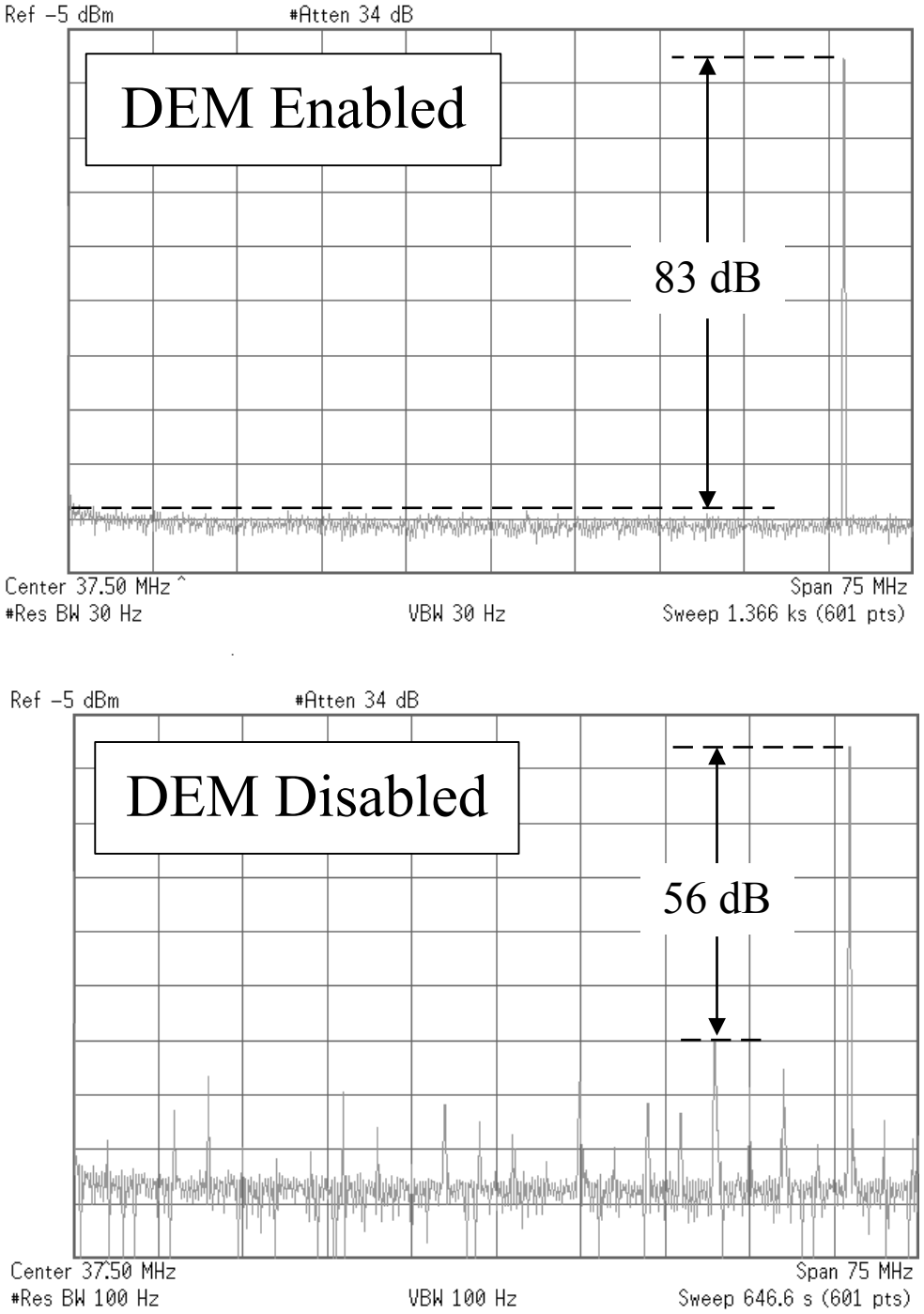


Figure 1.9: Representative PSD plots of the DAC output with DEM disabled and enabled.

Technology	TSMC 0.18 μm CMOS
Update Rate	150 MS/s
Package	QFN 64 with exposed paddle
Single-Tone SFDR @ 150 MS/s, 0 dBFS	>83 dB across the Nyquist Band >77 dB across 2nd Nyquist Band
Two-Tone SFDR @ 150 MS/s, −6 dBFS	>84 dB ($f_{out2} - f_{out1}$ = 1MHz)
DNL @ 150 MS/s	±1 LSB
INL @ 150 MS/s	+2.5/-3.5 LSB
SNR @ 150 MS/s	57 dB
Full-Scale Current	16 mA
Supply Voltages	Analog: 1.8 V, Digital: 1.8 V, Clock Generator: 2 V
Power Dissipation @ 150 MS/s	127 mW
Die Size (including bond pads)	4.8 mm x 2.4 mm
Active Area	3 mm^2

Figure 1.10: Performance table.

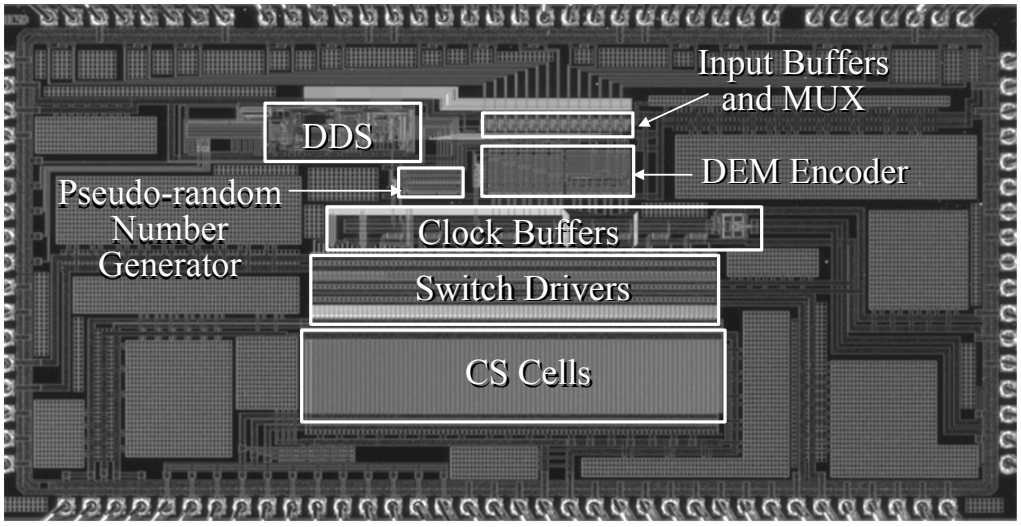


Figure 1.11: Die photograph.

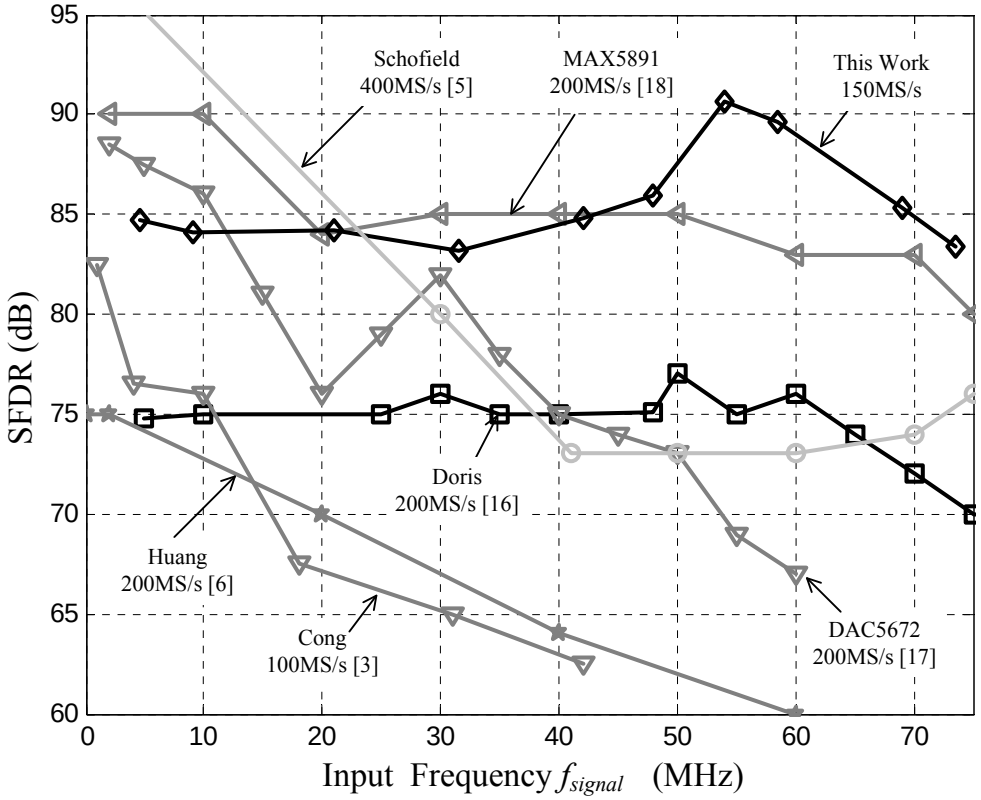


Figure 1.12: Comparison of recent state-of-the-art CMOS DACs.

VII. REFERENCES

1. J. Bastos, A. M. Marques, M. S. J. Steyaert, W. Sansen, "A 12-bit intrinsic accuracy high-speed CMOS DAC," *IEEE Journal of 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, G. G. E. Gielen, "A 14-bit intrinsic accuracy Q^2 random walk CMOS DAC," *IEEE Journal of Solid State Circuits,* vol. 34, no. 12, pp. 1708-1717, Dec, 1999.

3. Y. Cong, R. L. Geiger, "A 1.5-V 14-bit 100-MS/s self-calibrated DAC," *IEEE Journal of Solid State Circuits,* vol. 38, no. 12, pp. 2051-2060, December, 2003.

4. A. R. Bugeja, 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.

5. W. Schofield, D. Mercer, L. St. Onge, "A 16b 400MS/s DAC with <-80dBc IMD to 300MHz and <-160dBm/Hz noise power spectral density," *IEEE International Solid-State Circuits Conference*, pp. 126-127, February, 2003.

6. Q. Huang, P. A. Francese, C. Martelli, J. Nielsen, "A 200MS/s 14b 97mW DAC in 0.18µm CMOS," *IEEE International Solid-State Circuits Conference*, pp. 364-365, February, 2004.

7. J. Hyde, T. Humes, C. Diorio, M. Thomas, M. Figueroa, "A 300-MS/s 14-bit digital-to-analog converter in logic CMOS," *IEEE Journal of Solid State Circuits,* vol. 38, no. 5, pp. 734-740, May, 2003.

8. B. Jewett, J. Liu, K. Poulton, "A 1.2Gs/s 15b DAC for precision signal generation," *IEEE International Solid-State Circuits Conference*, pp. 110-111, February, 2005.

9. K. O. Sullivan, C. Gorman, M. Hennessy, 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.

10. Fujitsu MB86061 12-Bit 400MS/s Digital to Analog Converter.

11. K. L. Chan, I. Galton, "A 14b 100MS/s DAC with Fully Segmented Dynamic Ele-

ment Matching," *IEEE International Solid-State Circuits Conference*, pp. 582-583, 675, February, 2006.

12. K. L. Chan, J. Zhu, I. Galton, "A 150MS/s 14-bit Segmented DEM DAC with Greater than 83dB of SFDR Across the Nyquist band," *Symposium on VLSI Circuits*, June 14, 2007.

13. T. S. Kaplan, J. F. Jensen, C. H. Fields, M-C F. Chang, "A 2-GS/s 3-bit ΔΣ-modulated DAC with tunable bandpass mismatch shaping," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 3, pp. 603-610, March, 2005.

14. D. D. Caro and A. G. M. Strollo, "High-Performance Direct Digital Frequency Synthesizer in 0.25 μm CMOS Using Dual-Slope Approximation," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 11, pp. 2220-2227, November, 2005.

15. E. Fogleman, I. Galton, W. Huff, H. T. Jensen, "A 3.3V single-poly CMOS audio ADC delta-sigma modulator with 98dB peak SINAD and 105dB peak SFDR", *IEEE Journal of Solid State Circuits*, vol. 35, no. 3, pp. 297-307, March, 2000.

16. K. Doris, J. Briaire, D. Leenaerts, M. Vertregt, A. van Roermund, "A 12b 500MS/s DAC with >70dB SFDR up to 120MHz in 0.18μm CMOS," *ISSCC Dig. Tech. Papers,* pp. 116-117, 2005.

17. Texas Instruments: DAC5672: Dual 14-Bit 200 MSPS DAC.

18. Maxim MAX5891: 16-Bit, 600Msps High Dynamic Performance DAC with Differential LVDS Inputs.

Chapter 2

On the Design of Dynamic Element Matching Encoders for High-Linearity Digital-to-Analog Conversion

I. Introduction

In a dynamic element matching (DEM) DAC, a *DEM encoder* maps a digital input sequence to multiple 1-bit output sequences, each of which drives a 1-bit DAC. The outputs of the 1-bit DACs are summed to form the output of the DEM DAC. If the 1-bit DACs are nominally identical, then the structure is called a *unity-weighted* DEM DAC. In this case, for most input values there are several sets of DEM encoder output bit values that would result in the same DEM DAC output pulse in the absence of mismatches among the 1-bit DACs. The DEM encoder exploits this *redundancy* to scramble the usage pattern of the 1-bit DACs from sample-to-sample such that the error waveform resulting from mismatches among the 1-bit DACs has a noise-like structure that is free of non-linear distortion and spurious tones and has either a white or shaped power spectral density (PSD).

Such DEM DACs have long been used as enabling components in delta-sigma data converters which require low-resolution but high-linearity DACs. However, they are rarely used in applications that require input sequences with greater than about five bits of resolution, such as high-resolution Nyquist-rate DACs, because their circuit complexity grows exponentially with the number of bits required to represent their input sequence. Recently, *segmented* DEM DACs have been developed that overcome this prob-

lem.

Segmented DEM DACs have the same general structure as their unity-weighted counterparts except that the 1-bit DAC weights are not all equal and their DEM encoders function somewhat differently. By having groups of 1-bit DACs with equal weights, yet having the weights of the 1-bit DACs in each group be larger than those of the previous group, sufficient redundancy can be retained for DEM to be effective without having complexity that grows exponentially with the number of input bits. Given two DEM DACs capable of handling the same range of input values, the one with the smaller number of 1-bit DACs is said to have the *higher level of segmentation*. Typically, a higher level of segmentation implies lower DEM encoder complexity but a higher ratio of the largest to smallest 1-bit DAC weights.

Unfortunately, each segmented DEM DAC published to date has incurred a penalty in return for the complexity reduction achieved by segmentation: the error resulting from mismatches is free of non-linear distortion only if the range of values taken on by the input sequence is restricted relative to the set of all possible input values. Moreover, at least for the DEM DACs published to date, the higher the level of segmentation, the more the range must be restricted. For example, the DEM DAC with highest level of segmentation published to date can handle an input sequence which takes on any of 32,767 uniformly spaced values, but it is necessary to restrict the input sequence to the middle 16,385 values of this range to ensure that the error resulting from mismatches is free of non-linear distortion [1]. This corresponds to a 6 dB reduction in signal swing. In terms of the signal-to-noise ratio (SNR), the signal-swing reduction can be compensated by reducing the circuit noise from the 1-bit DACs by 6dB, but doing so usually dictates a

significant increase in power dissipation. Therefore, the published DEM DACs suggest that segmentation represents a tradeoff between circuit complexity and power dissipation.

Nevertheless, the previously published work on DEM DACs does not address whether the range restriction problem mentioned above is a fundamental limitation of segmentation or just a limitation of the proposed segmentation techniques. Moreover, while specific segmented DEM encoders have been described, general methods of synthesizing segmented DEM encoders and analyses of their tradeoffs have not been presented. This paper addresses these issues. It proves that the range restriction problem is an unavoidable side effect of segmentation, and quantifies the optimal range that can be achieved by a segmented DEM DAC for each possible set of 1-bit DAC weights. It also provides a general method of synthesizing segmented DEM DACs that achieve the optimal input range.

Although unity-weighted tree-structured DEM DACs have been shown to be general in the sense that they can be designed to mimic any possible DEM DAC including any segmented DEM DAC, no general technique for synthesizing segmented DEM DACs has been published previously [1-3,5]. In this paper, the tree-structured DEM DAC is generalized to form the basis of the synthesis technique. A side benefit of the approach is that it also provides a synthesis technique for unity-weighted DEM DACs with any number of 1-bit DACs; the original tree-structured DEM DAC architecture requires the number of 1-bit DACs to be a power of two.

II. A General Dynamic Element Matching DAC

A. Ideal Behavior

A general DEM DAC architecture is shown in Figure 2.1. It consists of an all-digital block called a *DEM encoder*, followed by N 1-bit DACs. Ideally, during the nth sample period, $nT \le t < (n+1)T$, the output of the ith 1-bit DAC is

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

where $c_i[n]$ is the input to the 1-bit DAC, K_i is the *weight* of the 1-bit DAC, and $a(t)$ is an analog pulse that is zero outside of $0 \le t < T$. By definition, $K_1 = 1$ and each K_i for $i = 2, 3, \ldots, N$ is a positive integer ordered such that $K_i \ge K_{i-1}$. The DEM encoder is designed to satisfy

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

where $x[n]$ is the digital input sequence to the DEM encoder. Therefore, the set of possible input values depends on the number of 1-bit DACs and their weights. With the ideal 1-bit DAC behavior given by (1), it follows from (2) and Figure 2.1 that the output of the DEM DAC during the nth sample period is

$$y(t) = a(t-nT)x[n]. \tag{3}$$

It can be shown from (2) that if

$$K_i \le \sum_{m=1}^{i-1} K_m + 1 \tag{4}$$

for $i = 2, 3, \ldots, N$, then $x[n]$ can take on all values in the set

$$\left\{ -\frac{M}{2}\Delta,\ -\left(\frac{M}{2}-1\right)\Delta,\ -\left(\frac{M}{2}-2\right)\Delta, \ldots,\ \frac{M}{2}\Delta \right\} \tag{5}$$

where

$$M = \sum_{i=1}^{N} K_i \ . \tag{6}$$

Alternatively, if (4) is not satisfied then the minimum and maximum possible values of $x[n]$ are still $-M\Delta/2$ and $M\Delta/2$, but some of the other values in (5) between these two extremes are not possible, so the range of possible input values is not *contiguous*. Given that having a contiguous range of input values is desirable in most applications, (4) is taken as a design requirement in this paper.

B. Behavior with Component Mismatches

In practice mismatches among nominally identical components used to implement the 1-bit DACs cause the ideal behavior given by (1) to degrade to

$$y_i(t) = \begin{cases} K_i \dfrac{\Delta}{2} a(t-nT) + e_{hi}(t-nT), & \text{if } c_i[n] = 1, \\ -K_i \dfrac{\Delta}{2} a(t-nT) + e_{li}(t-nT), & \text{if } c_i[n] = 0, \end{cases} \tag{7}$$

where $e_{hi}(t)$ and $e_{li}(t)$ are *mismatch error pulses* caused by the component mismatches. It is assumed in the remainder of the paper that the mismatch error pulses are non-zero only for the duration of the sample period. Otherwise, no assumptions are made about $e_{hi}(t)$ and $e_{li}(t)$.

An equivalent form of (7) is

$$y_i(t) = K_i \Delta \alpha_i \left(t-nT\right) \underbrace{\left(c_i[n] - \frac{1}{2}\right)}_{=\pm\frac{1}{2}} + \beta_i \left(t-nT\right) \tag{8}$$

where

$$\alpha_i(t) = a(t) + \frac{e_{hi}(t) - e_{li}(t)}{K_i \Delta}, \qquad \text{and} \qquad \beta_i(t) = \frac{e_{hi}(t) + e_{li}(t)}{2}, \tag{9}$$

which can be verified by substituting (9) into (8) to obtain (7). It is straightforward to show that (2) holds if and only if there exist constants, m_i, and sequences, $\lambda_i[n]$, that satisfy

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

for $i = 1, 2, \ldots, N$, where

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

Therefore, the DEM DAC's output signal during the nth sample period can be written as

$$y(t) = \alpha(t - nT)x[n] + \beta(t - nT) + e_{DAC}(t) \tag{12}$$

where

$$\alpha(t) = \sum_{i=1}^{N} K_i m_i \alpha_i(t), \quad \beta(t) = \sum_{i=1}^{N} \beta_i(t), \quad e_{DAC}(t) = \sum_{i=1}^{N} K_i \lambda_i[n] \alpha_i(t - nT). \tag{13}$$

In the following, the three components of the DEM DAC's output signal in (12) are referred to as the *signal pulse sequence*, the *offset pulse sequence*, and the *DAC noise pulse sequence*, respectively.

The signal pulse sequence consists of the analog pulses, $\alpha(t - nT)$, scaled linearly by the input sequence. As indicated by the first equations in (11) and (20), $\alpha(t)$, is a weighted average of the $\alpha_i(t)$ pulses from the individual 1-bit DACs. Thus, the 1-bit DAC mismatch error pulses cause $\alpha(t)$ to deviate 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 (12) and (20) 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, $e_{DAC}(t)$, is so-named because the DEM encoder is designed to cause it to have a noise-like structure. As explained in the remainder of the paper, provided the number of 1-bit DACs and their weights are chosen with certain restrictions and then provided the input sequence stays within a certain range of values, the DEM encoder can be designed to ensure that $e_{DAC}(t)$ has zero mean, is uncorrelated with $x[n]$, and is free of spurious tones, i.e.,

$$\mathrm{E}\left\{e_{DAC}(t)\right\} = 0, \;\; \mathrm{E}\left\{e_{DAC}(t)\,x[n]\right\} = 0, \text{ and } \lim_{\tau\to\infty} \mathrm{E}\left\{e_{DAC}(t)\,e_{DAC}(t+\tau)\right\} = 0. \quad (14)$$

In this sense the DEM encoder can be designed to ensure that the DAC does not introduce non-linear distortion despite the mismatches among the 1-bit DACs.

C. Input Sequence Representation

In any practical DAC, each value of the input sequence is represented as a digital codeword. By design convention, each codeword is interpreted to represent one of the values in (5) as described above. 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 (12).

Nevertheless, (5) is a set of M+1 uniformly spaced numbers, so each of its values can be mapped to, and therefore represented at the digital circuit level as, an unsigned integer in the range 0 to M given by

$$c[n] = \frac{x[n]}{\Delta} + \frac{M}{2}. \quad (15)$$

Hence, while the sequence of input codewords can be interpreted as a sequence of values, $x[n]$, restricted to the set (5) as described previously, it can equivalently be interpreted as a sequence of values, $c[n]$, restricted to the set of integers ranging from 0 to M. It turns out that this latter interpretation is particularly convenient when designing the DEM encoder. With this interpretation, it follows from (12) and (15) that the DEM DAC's output signal during the nth sample period can be written in terms of $c[n]$ as

$$y(t) = \alpha(t - nT)\Delta c[n] + \beta'(t - nT) + e_{DAC}(t), \tag{16}$$

where

$$\beta'(t) = \beta(t) - \alpha(t)\frac{M}{2}\Delta. \tag{17}$$

A physical interpretation of $c[n]$ is as follows. In the absence of mismatches, the DEM DACs output pulse during the nth sample period is the same as would be produced by adding the output pulses during the nth sample period of M 1-bit DACs all of weight $K_1 = 1$, $c[n]$ of which have their input bits set to 1 and the rest of which have their input bits set to 0.

III. The Range Restriction Versus Segmentation Tradeoff

The following theorem quantifies a price that is paid for the reduction in circuit area offered by segmentation, regardless of how the DEM encoder is designed.

Theorem 1*:* A DEM DAC of the form shown in Figure 2.1 that satisfies (2) and (7) introduces DAC noise with the properties given by (14) only if $x[n]$ is always larger than the smallest $K_N - 2$ values of (5) and smaller than the largest $K_N - 2$ values of (5).

***Proof**:* Let p be the smallest integer for which $K_{p+1} = K_N$. Given that $K_i \geq K_{i-1}$, (2) implies that

$$x[n] = x_{2,1}[n] + x_{2,2}[n] \tag{18}$$

where

$$x_{2,1}[n] = \Delta \sum_{i=1}^{p} K_i \left(c_i[n] - \tfrac{1}{2} \right), \qquad \text{and} \qquad x_{2,2}[n] = K_N \Delta \sum_{i=p+1}^{N} \left(c_i[n] - \tfrac{1}{2} \right). \tag{19}$$

It is straightforward to show that (19) holds if and only if there exist constants, $m_{2,k,i}$, and sequences, $\lambda_{2,k,i}[n]$, for $k = 1$ and 2, and $i = a_k, a_k+1, \ldots, b_k$, that satisfy

$$\Delta \left(c_i[n] - \tfrac{1}{2} \right) = m_{2,k,i} x_{2,k}[n] + \lambda_{2,k,i}[n], \tag{20}$$

where

$$\sum_{i=a_k}^{b_k} K_i m_{2,k,i} = 1, \qquad \text{and} \qquad \sum_{i=a_k}^{b_k} K_i \lambda_{2,k,i}[n] = 0. \tag{21}$$

with

$$a_k = \begin{cases} 1, & \text{if } k = 1 \\ p+1, & \text{if } k = 2 \end{cases} \qquad \text{and} \qquad b_k = \begin{cases} p, & \text{if } k = 1 \\ N, & \text{if } k = 2 \end{cases} \tag{22}$$

Let

$$y_{2,k}(t) = \sum_{i=a_k}^{b_k} y_i(t) \,. \tag{23}$$

Substituting (8) and (20) into (23) implies that during the nth sample period

$$y_{2,k}(t) = \alpha_{2,k} \left(t - nT \right) x_{2,k}[n] + \beta_{2,k} \left(t - nT \right) + e_{DAC\,2,k} \left(t \right), \tag{24}$$

where

$$\begin{aligned} &\alpha_{2,k}(t) = \sum_{i=a_k}^{b_k} K_i m_{2,k,i} \alpha_i(t), \qquad \beta_{2,k}(t) = \sum_{i=a_k}^{b_k} \beta_i(t), \\ &e_{DAC\,2,k}(t) = \sum_{i=a_k}^{b_k} K_i \lambda_{2,k,i}[n] \alpha_i(t - nT). \end{aligned} \tag{25}$$

It follows from (22), (23), and Figure 2.1 that

$$y(t) = y_{2,1}(t) + y_{2,2}(t)\,, \tag{26}$$

from (20) and (25) that $\beta(t) = \beta_{1,1}(t) + \beta_{1,2}(t)$, and from (18) that for any G in the range $0 \le G \le 1$ there exists a sequence $s[n]$ for which

$$x_{2,1}[n] = (1-G)\,x[n] + \frac{s[n]}{2}\Delta, \qquad \text{and} \qquad x_{2,2}[n] = Gx[n] - \frac{s[n]}{2}\Delta\,. \tag{27}$$

Therefore, substituting (24) and (27) into (26) and collecting terms implies that (12) holds during the nth sample period with

$$\alpha(t) = (1-G)\,\alpha_{2,1}(t) + G\alpha_{2,2}(t)\,, \tag{28}$$

and

$$e_{DAC}(t) = \frac{\Delta}{2}\left[\alpha_{2,1}(t-nT) - \alpha_{2,2}(t-nT)\right]s[n] + e_{DAC\,2,1}(t) + e_{DAC\,2,2}(t)\,. \tag{29}$$

With (25) this can be written as

$$\begin{aligned} e_{DAC}(t) = &\sum_{i=1}^{p} K_i\alpha_i(t-nT)\left(\lambda_{2,1,i}[n] + \frac{\Delta}{2}s[n]m_{2,1,i}\right) \\ &+ \sum_{i=p+1}^{N} K_i\alpha_i(t-nT)\left(\lambda_{2,2,i}[n] - \frac{\Delta}{2}s[n]m_{2,2,i}\right) \end{aligned} \tag{30}$$

It follows that the expected value of $e_{DAC}(t)$ is zero for every set of mismatch error pulses only if the expected value of $s[n]$ is zero. For example, in the special case where the mismatch error pulses are zero for $i = 1, 2, \ldots, p$, it follows from (9) and (21) that (30) reduces to

$$e_{DAC}(t) = \frac{\Delta}{2}a(t-nT)s[n] + \sum_{i=p+1}^{N} K_i\alpha_i(t-nT)\left(\lambda_{2,2,i}[n] - \frac{\Delta}{2}s[n]m_{2,2,i}\right). \tag{31}$$

Since $\alpha_i(t)$ for $0 \le t < \mathrm{T}$ and each $i = p+1, p+2, \ldots, N$ is arbitrary because it depends on the mismatch error pulses, the expected value of $e_{DAC}(t)$ is zero only if the expected value of each term in the summation in (31) is zero. However, in this case (31) implies that the

expected value of $e_{DAC}(t)$ is zero only if the expected value of the first term in (31) and therefore of $s[n]$ is zero.

Equations (19) imply that $x_{2,1}[n]$ is restricted to the set

$$\left\{-\frac{M_1}{2}\Delta,\ -\left(\frac{M_1}{2}-1\right)\Delta,\ -\left(\frac{M_1}{2}-2\right)\Delta,\ldots,\ \frac{M_1}{2}\Delta\right\} \tag{32}$$

and $x_{2,2}[n]$ is restricted to the set

$$\left\{-\frac{M_2}{2}\Delta,\ -\left(\frac{M_2}{2}-K_N\right)\Delta,-\left(\frac{M_2}{2}-2K_N\right)\Delta,\ldots,\ \frac{M_2}{2}\Delta\right\} \tag{33}$$

where

$$M_1=\sum_{i=1}^{p}K_i \quad \text{and} \quad M_2=\left(N-p\right)K_N \tag{34}$$

Thus, for each value of $x[n]$ in (5) with $M = M_1 + M_2$, $s[n]$ must be chosen such that (27) yields values of $x_{2,1}[n]$ and $x_{2,2}[n]$ that are elements of (32) and (33), respectively.

For (14) to be satisfied, it is further necessary for the expected value of $s[n]$ to be zero regardless of $x[n]$. This can happen only if for each value of $x[n]$ either $s[n]$ is always zero, or $s[n]$ is a random variable that can take on non-zero positive and negative values. This is possible for a given value of $x[n]$ only if either $s[n]$ can be set to zero or there exist two non-zero values to which it can be set, one positive and one negative, such that (27) yields values of $x_{2,1}[n]$ and $x_{2,2}[n]$ that are elements of (32) and (33), respectively.

If $G|x[n]| \le M_2\Delta/2$, choosing

$$s[n]=\begin{cases}0, & \text{if } Gx[n]=q \\ \dfrac{2}{\Delta}u \text{ or } \dfrac{2}{\Delta}\left(u-K_N\Delta\right), & \text{if } Gx[n]=q+u, \text{ for } 0<u<K_N\Delta\end{cases} \tag{35}$$

where q is an element of (33), causes the right equation in (27) to yield values of $x_{2,2}[n]$ that are elements of (33). All other possible values of $s[n]$ have larger magnitudes than those specified by (35). With (35),

$$x_{2,2}[n]=\begin{cases}q & \text{if } s[n]=0 \text{ or } \dfrac{2}{\Delta}u \\ q+K_N\Delta, & \text{if } s[n]=\dfrac{2}{\Delta}\left(u-K_N\Delta\right)\end{cases} \tag{36}$$

If $G|x[n]| > M_2\Delta/2$, then $s[n]$ must be non-zero with a fixed sign for (27) to yield values of $x_{2,2}[n]$ in (33). Therefore, $G|x[n]| \le M_2\Delta/2$, is a necessary condition to satisfy (14).

Let

$$G=\frac{M_2}{M-2K_N+2}, \tag{37}$$

with which (27) becomes

$$x_{2,1}[n]=\frac{M_1-2K_N+2}{M-2K_N+2}x[n]+\frac{s[n]}{2}\Delta, \qquad \text{and} \qquad x_{2,2}[n]=\frac{M_2}{M-2K_N+2}x[n]-\frac{s[n]}{2}\Delta. \tag{38}$$

Therefore, when

$$x[n]=\pm\left(\frac{M}{2}-K_N+1\right)\Delta, \tag{39}$$

(35) implies that $s[n] = 0$, so

$$x_{2,1}[n]=\pm\left(\frac{M_1}{2}-K_N+1\right)\Delta, \qquad \text{and} \qquad x_{2,1}[n]=\pm\frac{M_2}{2}\Delta \tag{40}$$

which are elements of (32) and (33), respectively. Furthermore, whenever the value of $x[n]$ is an element of (5) between the two extremes of (39), it can be verified by substitution of (35) into (38) that every possible value of $s[n]$ specified by (35) gives values of

$x_{2,1}[n]$ and $x_{2,2}[n]$ that are elements of (32) and (33), respectively.

It follows from (18), (32), and (33) that whenever the value of $x[n]$ is any of the K_N smallest elements of (5), the value of $x_{2,2}[n]$ must be the smallest element of (33). Otherwise, the value of $x_{2,1}[n]$ would be smaller than the smallest element of (32) to satisfy (18). It follows from (36) that $s[n]$ is forced to be either zero or a non-zero positive value when $x[n]$ is any of the K_N smallest elements of (5). Moreover, since $0 \le G \le 1$, (5) and (35) imply that $s[n]$ is zero for at most one value of $x[n]$ out of the K_N smallest elements of (5). By similar reasoning, $s[n]$ is forced to be either zero or a non-zero negative value when $x[n]$ is any of the K_N largest elements of (5), and $s[n]$ is zero for at most one of these values of $x[n]$. For the choice of G given by (37), it is shown above that $s[n] = 0$ when $x[n]$ has the values given by (39). However, these values are the largest of the K_N smallest elements of (5) the smallest of the K_N largest elements of (5). This implies that whenever the value of $x[n]$ is any of the $K_N - 1$ smallest elements of (5) or any of the any of the $K_N - 1$ largest elements of (5), $s[n]$ is non-zero with a fixed sign.

IV. Synthesis of Optimal-Range DEM Encoders

It will be shown that a DEM encoder can be built from digital blocks called *switching blocks*. In Figure 2.2, $S_{k,r}$ is a switching block which details will be presented later, and DAC$_j$ is a DAC of with N_j 1-bit DACs of weight K_j where $j \in \{i, i+1\}$. K_{i+1} and K_i are chosen such that $K_i' \equiv (K_{i+1} / K_i) \in \mathbb{Z}^+$. Each DAC$_j$ consists of a DEM encoder which ensures that DAC$_j$'s output has the form of (24), and its DAC noise has the properties of (14). The inputs to DAC$_i$ and DAC$_{i+1}$ can be interpreted as $x_{k\text{-}1,2r}[n]$ and $x_{k\text{-}1,2r\text{-}1}[n]$,

respectively, restricted to the set

$$\left\{-\frac{R_j K_j}{2}\Delta,\ -\left(\frac{R_j K_j}{2}-K_j\right)\Delta,\ -\left(\frac{R_j K_j}{2}-2K_j\right)\Delta,\ldots,\ \frac{R_j K_j}{2}\Delta\right\}, \tag{41}$$

where $j \in \{i, i+1\}$, $R_j \in \mathbb{N}$ and $R_j \le N_j$. Equivalently the inputs to DAC$_i$ and DAC$_{i+1}$ can be interpreted as $c_{k\text{-}1,2r}[n]$ and $c_{k\text{-}1,2r\text{-}1}[n]$, respectively, restricted to the set

$$\{A_j,\ A_j+1,\ldots,\ B_j-1, B_j\},\ j \in \{i, i+1\}, \tag{42}$$

where $A_j, B_j \in \mathbb{N}$ and $A_j, B_j \le N_j$. R_j, A_j and B_j account for any range restriction that might be imposed by the DEM encoder in DAC$_j$. As mentioned previously, it is more convenient to work with unsigned integers when implementing the DEM encoder. $c_{k\text{-}1,2r}[n]$ and $c_{k\text{-}1,2r\text{-}1}[n]$ also have the physical meaning of the number of 1-bit DACs of weight K_i and K_{i+1}, respectively, that have their input bits set to 1. The two interpretations are related by

$$\begin{aligned} x_{k-1,2r-1}[n] &= \left(c_{k-1,2r-1}[n]-\frac{N_i}{2}\right)K_i\Delta, \\ x_{k-1,2r}[n] &= \left(c_{k-1,2r}[n]-\frac{N_{i+1}}{2}\right)K_{i+1}\Delta, \end{aligned} \tag{43}$$

and R_j, A_j and B_j are related by

$$A_j = \frac{N_j - R_j}{2} \quad \text{and} \quad B_j = \frac{N_j + R_j}{2}. \tag{44}$$

The input/output relationships of the switching block $S_{k,r}$ are given by

$$\begin{aligned} c_{k-1,2r-1}[n] &= \left(1-G_{k,r}\right)c_{k,r}[n] + \frac{s_{k,r}[n]}{2} + b_{k,r}, \\ c_{k-1,2r}[n] &= \frac{1}{K_i'}\left(G_{k,r}c_{k,r}[n] - \frac{s_{k,r}[n]}{2} - b_{k,r}\right), \end{aligned} \tag{45}$$

where

$$G_{k,r} = \frac{K_i^{'}\left(B_{i+1} - A_{i+1}\right)}{K_i^{'}\left(B_{i+1} - A_{i+1}\right) + \left(B_i - A_i\right) - 2K_i^{'} + 2}, \tag{46}$$

$$b_{k,r} = \frac{G_{k,r}N_i - (1 - G_{k,r})N_{i+1}K_i^{'}}{2}. \tag{47}$$

For a given range restriction in DAC$_i$, N_i is chosen such that

$$\left(B_i - A_i\right) \geq 2K_i^{'} - 2, \text{ or equivalently, } 0 \leq G_{k,r} \leq 1. \tag{48}$$

$s_{k,r}[n]$ is called the *switching sequence*, and is given by

$$s_{k,r}[n] = \begin{cases} 0, & \text{if } G_{k,r}c_{k,r}[n] - b_{k,r} = K_i^{'}v \\ 2w \text{ or } 2\left(w - K_i^{'}\right), & \text{if } G_{k,r}c_{k,r}[n] - b_{k,r} = K_i^{'}v + w, \text{ for } 0 < w < K_i^{'} \end{cases} \tag{49}$$

where v is a non-negative integer. The flexibility in (49) is exploited to ensure that

$$\mathrm{E}\left\{s_{k,r}[n]\right\} = 0, \ \mathrm{E}\left\{s_{k,r}[n]c_{k,r}\left[n\right]\right\} = 0, \text{ and } \lim_{\tau \to \infty} \mathrm{E}\left\{s_{k,r}[n]s_{k,r}[n+\tau]\right\} = 0. \tag{50}$$

Corollary 1: The DEM DAC in Figure 2.2 introduces DAC noise with the properties given by (14), if and only if its input $c_{k,r}[n]$ is restricted to the set of

$$c_{k,r}[n] \in \left\{A_{k,r}, A_{k,r} + 1, ..., B_{k,r} - 1, B_{k,r}\right\} \tag{51}$$

where

$$A_{k,r} = K_i^{'}A_{i+1} + A_i + K_i^{'} - 1 \ \text{ and } \ B_{k,r} = K_i^{'}B_{i+1} + B_i - K_i^{'} + 1_i. \tag{52}$$

Proof: From (43), $x_{k\text{-}1,2r}[n]$ and $x_{k\text{-}1,2r\text{-}1}[n]$ can be expressed as

$$x_{k-1,2r-1}[n] = \Delta^{'}\sum_{l=1}^{N_i}\left(c_l[n] - \tfrac{1}{2}\right), \ \text{ and } \ x_{k-1,2r}[n] = K^{'}\Delta^{'}\sum_{i=N_i+1}^{N_i+N_{i+1}}\left(c_l[n] - \tfrac{1}{2}\right). \tag{53}$$

where $\Delta^{'} = K_i\Delta$, $K_i^{'} = \left(K_{i+1} / K_i\right)$ and $c_l[n]$ is the input to the 1-bit DAC in Figure 2.2. It can be seen that (53) is just (19) with a change of variables and

with $K_i = 1,\ i \in \{1,2,..,p\}$, in (19). Let *D* be a digital logic block implementing (27) such that

$$x_{k-1,2r-1}[n] = (1-G)\,x_{k,r}[n] + \frac{s[n]}{2}\Delta' \ \text{ and } \ x_{k-1,2r}[n] = G x_{k,r}[n] - \frac{s[n]}{2}\Delta', \quad (54)$$

where $x_{k,r}$ is the input to the DAC. From (41),

$$x_{k-1,2r-1} \in \left\{-\frac{R_i}{2}\Delta', -\left(\frac{R_i}{2}-1\right)\Delta', -\left(\frac{R_i}{2}-2\right)\Delta', ..., \frac{R_i}{2}\Delta'\right\},$$
$$x_{k-1,2r} \in \left\{-\frac{R_{i+1}K'}{2}\Delta', -\left(\frac{R_{i+1}K'}{2}-K'\right)\Delta', -\left(\frac{R_{i+1}K'}{2}-2K'\right)\Delta', ..., \frac{R_{i+1}K'}{2}\Delta'\right\}. \quad (55)$$

Thus following similar reasoning in the proof in *Theorem 1*, if *G* is chosen to be

$$G = \frac{K'R_{i+1}}{K'R_{i+1} + R_i - 2K' - 2}, \quad (56)$$

and $s[n]$ is chosen using (35) while satisfying

$$\mathrm{E}\{s[n]\} = 0,\ \ \mathrm{E}\{s[n]x[n]\} = 0, \text{ and } \lim_{\tau\to\infty} \mathrm{E}\{s[n]s[n+\tau]\} = 0, \quad (57)$$

then the DAC noise of the DAC will have properties given by (14) if and only if $x_{k,r}[n]$ is restricted to be, $x_{k,r}[n] \in$

$$\left\{-\left(\frac{K'R_{i+1}+R_i}{2} - K' + 1\right)\Delta', -\left(\frac{K'R_{i+1}+R_i}{2} - K'\right)\Delta', ..., \left(\frac{K'R_{i+1}+R_i}{2} - K' + 1\right)\Delta'\right\}. \quad (58)$$

From (54),

$$x_{k,r}[n] = x_{k-1,2r-1}[n] + x_{k-1,2r}[n]. \quad (59)$$

Using (43), (59),

$$x_{k,r}[n] = \left(c_{k,r}[n] - \frac{N_i + N_{i+1}K'}{2}\right)\Delta'. \quad (60)$$

Substituting (43) and (60) into (54), it can be shown that (45) and (47) are equivalent to (54). It can also be seen that (46) and (56) are identical. Therefore, the switching block

$S_{k,r}$ is equivalent to the logic block D. Using (44) and (60), it can also be shown that the restriction in $x_{k,r}[n]$ in (58) is equivalent to the restriction in $c_{k,r}[n]$ in (51) and (52). With (47) and (60), it can be shown that (49) is equivalent to (35). Using (60), it can also be seen that the conditions on $s[n]$ in (57) are equivalent to the conditions on $s_{k,r}[n]$ in (50). Thus, with G chosen to be (46) and $s_{k,r}[n]$ chosen with (49) and (50), the DAC noise of the DAC has properties given by (14) if and only if $c_{k,r}[n]$ is within the range specified by (51) and (52).

A. Synthesis of DEM Encoders for Unity-Weighted DACs

If the DEM DAC in Figure 2.2 has $K_i = K_{i+1} = 1,\ A_i = A_{i+1} = 0$, $B_i = N_i,$ and $B_{i+1} = N_{i+1}$, it follows from *Corollary 1* that the permitted range of $c_{k,r}[n]$ is given by

$$c_{k,r}[n] \in \{0,1,2,\ldots,N_i + N_{i+1}\}. \tag{61}$$

Thus, a unity-weighted DEM DAC of the form shown in Figure 2.1 with $K_i = 1,\ i \in \{1,2,..,N\}$, can be partitioned into 2 smaller DACs of N_i+1 and N_{i+1}+1 levels with a switching block $S_{k,r}$ without any range restriction on $c_{k,r}[n]$. The smaller DACs can in turn be further partitioned into yet smaller DACs. The process can be repeated until each 1-bit DAC in Figure 2.1 is terminated to a switching block. Since $K_i = K_{i+1} = 1,\ A_i = A_{i+1} = 0,\ B_i = N_i,$ and $B_{i+1} = N_{i+1}$ at each step of the partitioning, *Corollary 1* ensures that there is no range restriction at each step. Thus, DEM encoders synthesized with switching blocks for unity-weighted DACs are optimal in the sense that they do not impose any restrictions on $c_{k,r}[n]$. An example for a 13-level DEM encoder is

shown in Figure 2.3.

B. Example: A 13-level DEM DAC

Figure 2.3 shows a DEM encoder consisting of 11 switching blocks for a 13-level DEM DAC. Each switching block $S_{k,r}$ implements equations (45) - (47), and its switching sequence $s_{k,r}[n]$ is given by (49). Since $K_i = K_{i+1} = 1,\ A_i = A_{i+1} = 0,$ $B_i = N_i,$ and $B_{i+1} = N_{i+1}$ for each $S_{k,r}$, (45) - (47) reduce to

$$\begin{aligned} c_{k-1,2r-1}[n] &= \left(1 - G_{k,r}\right) c_{k,r}[n] + \frac{s_{k,r}[n]}{2}, \\ c_{k-1,2r}[n] &= G_{k,r} c_{k,r}[n] - \frac{s_{k,r}[n]}{2}, \end{aligned} \tag{62}$$

where

$$G_{k,r} = \frac{N_{i+1}}{N_{i+1} + N_i}. \tag{63}$$

For switching block $S_{k,r}$, where $(k,r) \in \{(4,1),(3,1),(3,2),(1,1),(1,2),(1,3),(1,4)\}$, $G_{k,r}$ is 1/2, and $s_{k,r}[n]$ can be chosen to be

$$s_{k,r}[n] = \begin{cases} 0 & \text{when } c_{k,r}[n] \text{ is even,} \\ \pm 1 & \text{when } c_{k,r}[n] \text{ is odd.} \end{cases} \tag{64}$$

The flexibility in $s_{k,r}[n]$ when $c_{k,r}[n]$ is odd is used to ensure that (50) is met. It can be observed that the switching blocks in this case reduce to that the tree-structured switching blocks in [4]. For $S_{k,r}$, $k = 2,\ r \in \{1,2,3,4\}$, $G_{k,r}$ = 2/3, and for each input, Figure 2.4 shows the corresponding $s_{k,r}[n]$. Similarly, (50) is satisfied through the flexibility in the choices of $s_{k,r}[n]$.

C. Synthesis of DEM Encoders for Segmented DACs

In this section, the synthesis of a segmented DEM encoder using switching blocks is presented. It is also be shown that the encoder is optimal in the sense that it has a range restriction no worst than that stated in *Theorem 1*. Figure 2.5 shows a segmented DEM DAC comprising of DAC$_i$ of weight K_i, $i \in \{1,2,...,J\}$. K_i is chosen such that $K_i^{'} \equiv (K_{i+1} / K_i) \in \mathbb{Z}^+$ and $K_1 = 1$. Each DAC$_i$ consists of a DEM encoder and N_i 1-bit DACs of weight K_i. The DEM encoder in DAC$_i$ is implemented with switching blocks as described in Section A such that each DAC$_i$'s output has the form of (24), and its DAC noise has the properties of (14). It has been shown in Section A that a unity-weighted DEM encoder implemented with switching blocks does not restrict the input range of the DAC, hence the input to DAC$_i$ is given by

$$c_i[n] \in \{0,1,...,N_j\}, \quad i \in \{1,2,...,J\}. \tag{65}$$

Each switching block $S_{k,1}$ $k \in \{1,2,...,J-1\}$ in Figure 2.5 implements

$$\begin{aligned} c_{J-k}[n] &= (1-G_{k,1})c_{k,1}[n] + \frac{s_{k,1}[n]}{2} + b_{k,1}, \\ c_{k-1,1}[n] &= \frac{1}{K_{J-K}^{'}}\left(G_{k,1}c_{k,1}[n] - \frac{s_{k,1}[n]}{2} - b_{k,1}\right) \end{aligned} \tag{66}$$

where $c_{0,1}[n] \equiv c_J[n]$,

$$G_{k,1} = \frac{K_{J-K}^{'}(B_{k-1,1} - A_{k-1,1})}{K_{J-K}^{'}(B_{K-1,1} - A_{k-1,1}) + N_{J-K} - 2K_{J-K}^{'} + 2}, \tag{67}$$

$$b_{k,1} = \frac{G_{k,1}N_{J-K} - (1-G_{k,1})N_{J-K+1}K_{J-K}^{'}}{2}. \tag{68}$$

$B_{k-1,1}$ and $A_{k-1,1}$ represent the end points of the set which the input to the switching block

$S_{k,1}$ is restricted where $k \in \{2,3,...,J\}$, and $B_{0,1}$ and $A_{0,1}$ represent end points of the permitted range of DAC$_J$. Therefore

$$c_{k-1,1}[n] \in \{A_{k-1,1}, A_{k-1,1}+1, ..., B_{k-1,1}+1, B_{k-1,1}\}. \tag{69}$$

N_i is chosen such that

$$N_i \geq 2K_i^{'} - 2, \text{ or equivalently, } 0 \leq G_{k,1} \leq 1. \tag{70}$$

The switching sequence to $S_{k,1}$ is given by

$$s_{k,1}[n] = \begin{cases} 0, & \text{if } G_{k,1}c_{k,1}[n] - b_{k,1} = K_{J-K}^{'}v \\ 2w \text{ or } 2\left(w - K_{J-K}^{'}\right), & \text{if } G_{k,1}c_{k,1}[n] - b_{k,1} = K_{J-K}^{'}v + w, \text{ for } 0 < w < K_{J-K}^{'} \end{cases} \tag{71}$$

where v is a non-negative integer. Similarly, the flexibility in (71) is exploited to ensure that (50) is met.

The design procedure begins with $S_{1,1}$. $G_{1,1}$ and $b_{1,1}$ can be found from (67) and (68). Since (66) - (71) are just (45) - (49) with different indices, the range restriction on $c_{1,1}[n]$ is given by *Corollary 1*. Therefore

$$c_{1,1}[n] \in \{A_{1,1}, A_{1,1}+1, ..., B_{1,1}-1, B_{1,1}\} \tag{72}$$

where

$$A_{1,1} = K_{J-1}^{'} - 1, \text{ and } B_{1,1} = K_{J-1}^{'}N_J + N_{J-1} - K_{J-1}^{'} + 1. \tag{73}$$

Continuing with $S_{2,1}$, $G_{2,1}$ and $b_{2,1}$ can be found with (67), (68) and (73). *Corollary 1* can then be used to find the range restriction on $c_{2,1}[n]$. Knowing the range restriction on $c_{2,1}[n]$, $G_{3,1}$ and $b_{3,1}$ can be found. In this way, all the switching blocks can be designed.

Theorem 2: The segmented DEM DAC in Figure 2.5 will introduce DAC noise with the properties given by (14) if and only if the input is given by

$$c[n] \equiv c_{J-1,1} \in \left\{K_J - 1, K_J, K_J + 1, ..., \sum_{l=1}^{J} N_l K_l - K_J + 1\right\} \tag{74}$$

***Proof*:** From *corollary 1*, a DEM DAC will introduce DAC noise with the properties given by (14) if and only if the input is given by

$$c[n] \equiv c_{J-1,1}[n] \in \left\{ A_{J-1,1}, A_{J-1,1}+1, ..., B_{J-1,1}-1, B_{J-1,1} \right\} \tag{75}$$

where

$$A_{J-1,1} = \frac{K_2}{K_1} A_{J-2,1} + A_1 + \frac{K_2}{K_1} - 1, \text{ and } B_{J-1,1} = \frac{K_2}{K_1} B_{J-2,1} + B_1 - \frac{K_2}{K_1} + 1. \tag{76}$$

However, since $K_1 = 1$, $A_1 = 0$ and $B_1 = N_1$, (76) becomes

$$A_{J-1,1} = K_2 A_{J-2,1} + K_2 - 1, \text{ and } B_{J-1,1} = K_2 B_{J-2,1} + N_1 - K_2 + 1. \tag{77}$$

It can be seen from (75) and (77) that if $S_{J\text{-}1,1}$ has been connected to a DAC_2 directly, then $A_{J-2,1} = 0$ and $B_{J-2,1} = N_2$, and (76) becomes

$$A_{J-1,1} = K_2 - 1, \text{ and } B_{J-1,1} = K_2 N_2 + N_1 - K_2 + 1. \tag{78}$$

However, the top output of $S_{J\text{-}1,1,}$ $c_{J\text{-}2,1}[n]$ is connected to another switching block $S_{J\text{-}2,1,}$ and this imposes restrictions on $A_{J-2,1}$ and $B_{J-2,1}$ which in turn impose restrictions on $A_{J-1,1}$ and $B_{J-1,1}$ as indicated in (77). For $S_{J\text{-}2,1}$,

$$c_{J-2,1}[n] \in \left\{ A_{J-2,1}, A_{J-2,1}+1, ..., B_{J-2,1}-1, B_{J-2,1} \right\} \tag{79}$$

where

$$A_{J-2,1} = \frac{K_3}{K_2} A_{J-3,1} + A_2 + \frac{K_3}{K_2} - 1, \text{ and } B_{J-2,1} = \frac{K_3}{K_2} B_{J-3,1} + B_2 - \frac{K_3}{K_2} + 1. \tag{80}$$

Substituting (80) with $A_2 = 0$ and $B_2 = N_2$ in (77), (77) becomes

$$\begin{aligned} A_{J-1,1} &= K_2 \left(\frac{K_3}{K_2} A_{J-3,1} + \frac{K_3}{K_2} - 1 \right) + K_2 - 1 = K_3 A_{J-3,1} + K_3 - 1, \\ B_{J-1,1} &= K_2 \left(\frac{K_3}{K_2} B_{J-3,1} + N_2 - \frac{K_3}{K_2} + 1 \right) + N_1 - K_2 + 1 \\ &= K_3 B_{J-3,1} + N_2 K_2 + N_1 - K_3 + 1. \end{aligned} \tag{81}$$

By recursively representing $A_{J-i,1}$ and $B_{J-i,1}$ in terms of $A_{J-i-1,1}$ and $B_{J-i-1,1}$, $i \in \{3,4,...,J-1\}$, respectively,

$$A_{J-1,1} = K_J - 1, \text{ and } B_{J-1,1} = \sum_{l=1}^{J} N_l K_l - K_J + 1. \quad (82)$$

Thus, if and only if the input to the DAC is restricted to the set given by (75) and (82), it follows from *Corollary 1* that the DEM DAC introduces DAC noise with the properties given by (14).

D. Example: A Pair of 14-bit DEM DACs

In this section, the implementation of a fully-segmented 14-bit DEM DAC [1], and a highly-segmented 14-bit DEM DAC [5] is presented. Figure 2.6 and Figure 2.7 show the segmented DEM encoder for [1] and [5], respectively. In implementing the switching blocks, $S_{k,1}$, $k \in \{2,3,...,14\}$ in Figure 2.6, and $k \in \{5,6,...,14\}$ in Figure 2.7, the complexity of the DEM encoder can be significantly reduced with the proper choice of N_{J-i} and K'_{J-i} in Figure 2.5. Choosing

$$N_{J-i} = 2K'_{J-i} - 2, \quad i \in \{1,2,...,J-1\}, \quad (83)$$

from (67), $G_{i,1}$ =1, and (66) becomes

$$c_{i-1,1}[n] = \frac{1}{K'_{J-i}}\left\{c_{i,1}[n] - \frac{s_{i,1}[n]}{2} - b_{i,1}\right\} \text{ and } c_{J-i}[n] = \frac{s_{i,1}[n]}{2} + b_{i,1} \quad (84)$$

where $b_{i,1} = N_{J-i}/2$ from (68). Furthermore choosing $K'_{J-i} = N_{J-i} = 2$, (84) reduces to

$$c_{i-1,1}[n] = \frac{1}{2}\left\{c_{i,1}[n] - \frac{s_{i,1}[n]}{2} - 1\right\} \text{ and } c_{N-i}[n] = \frac{s_{i,1}[n]}{2} + 1 \quad (85)$$

as shown in Figure 2.6 and Figure 2.7. Also, now (71) simplifies to

$$s_{k,1}[n] = \begin{cases} 0 & \text{when } c_{k,r}[n] \text{ is odd,} \\ \pm 2 & \text{when } c_{k,r}[n] \text{ is even ,} \end{cases} \tag{86}$$

Following from the previous argument, the flexibility in $s_{k,r}[n]$ when $c_{k,r}[n]$ is even is used to ensure that (50) is satisfied. The switching blocks $S_{k,r,}$, $k = 1$, $r \in \{1, 2, ..., 28\}$ in Figure 2.6, and $k \in \{1, 2, ..., 4\}$, $r \in \{1, 2, ..., 36\}$ in Figure 2.7, can be synthesized following the procedure in Section A and B.

In implementing a DEM DAC with a given input range, having more switching blocks with $K_i^{'} > 1$ in (45) results in a smaller number of 1-bit DACs. Hence, switching blocks with $K_i^{'} > 1$ can be called *Segmenting Switching Block* as the number of segmenting block determines the level of segmentation. However, from *Theorem 2*, $2K_J - 2$ of input range is lost where K_J is the weight of the largest 1-bit DAC. In a typical application of current-steering DAC as in [1] and [5], each 1-bit DAC of weight K_i is implemented with K_i unity-weighted 1-bit DACs controlled as a group. Thus, more unity-weighted 1-bit DACs would be required in a DAC with a higher level of segmentation although the number of 1-bit DACs that need to be controlled would be reduced. Therefore there exists a trade-off between the complexity and the power dissipation of the DEM DAC. Figure 2.8 shows the trade-off for a 14-bit DEM DAC. For a DAC of 14-bit, a minimum of 16384 unity-weighted 1-bit DACs would be required, and it has been shown in Section A that if the 14-bit DAC is implemented with unity-weighted 1-bit DACs, there is no range restriction. However, the DEM encoder controlling the unity-weighted DACs would require 16383 switching blocks, which is clearly impractical. On

the other hand, a DEM DAC with the highest level of segmentation is shown in Figure 2.6 [1]. It contains a total of 27 switching blocks including 13 segmenting blocks and 28 1-bit DACs which are implemented with 32766 unity-weighted DACs. From *Theorem 2*, since the largest 1-bit DAC has a weight of 8192, there is a loss of 16362. This translates to having a almost 100% increase in the power dissipation as compared to the minimum case. From Figure 2.8, it can be seen that having 10 segmenting switching blocks as shown in Figure 2.7 [5] provides a good compromise with 35 switching blocks and 18430 unity-weighted 1-bit DACs. This corresponds to just a 12.5% increase in power consumption compared to the minimum case.

V. CONCLUSION

This paper shows that that there is a fundamental input range restriction to a segmented DEM DAC, regardless of how the DEM encoder is implemented. A general method of designing DEM encoder for unity-weighted DACs and segmented DEM DACs is then presented. The DEM encoders designed are optimal in the sense that they have a range restriction no worst than that fundamental input range restriction due to segmentation. The methods are demonstrated via examples of a 13-level unity-weighted DEM DAC and a pair of 14-bit segmented DEM DACs. The power dissipation versus complexity tradeoff implied by segmentation is also studied through the 14-bit examples.

VI. Chapter Acknowledgements

The Text of this chapter is in preparation to be submitted for review and publication as a regular paper in the *Transactions on Circuits and Systems II: Analog and Digital Signal Processing*. The dissertation author was the primary researcher. Ian Galton supervised the research that forms the basis of the chapter.

K. L. Chan, N. Rackuljic, and I. Galton, "On the Design of Dynamic Element Matching Encoders for High-Linearity Digital-to-Analog Conversion," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, in preparation.

VII. FIGURES

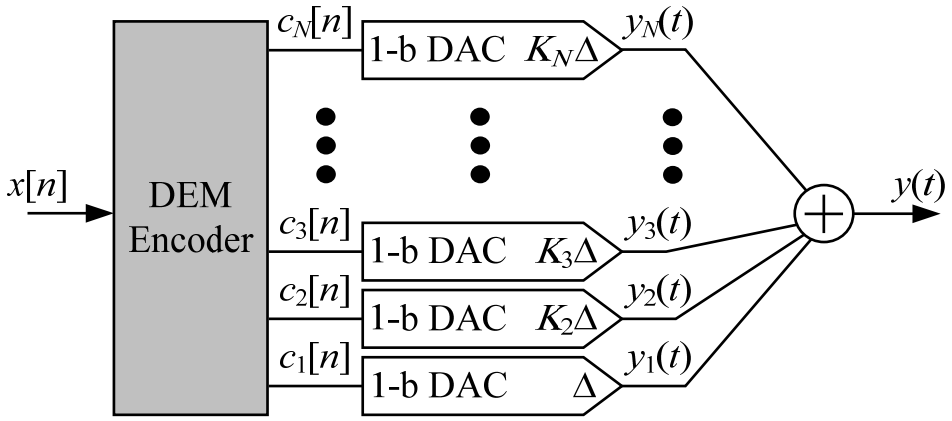


Figure 2.1: High-level system diagram of a general dynamic element matching DAC.

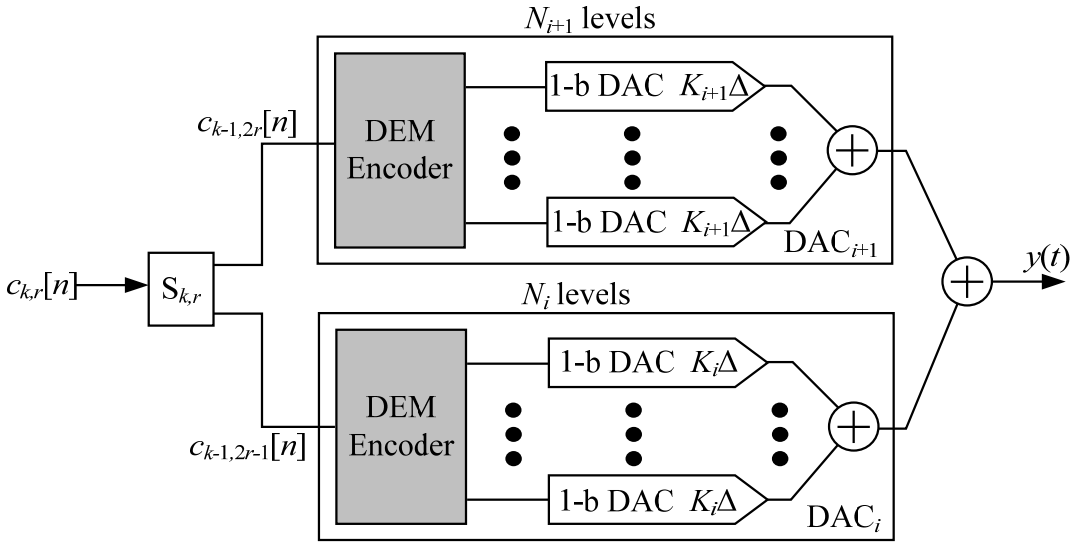


Figure 2.2: DEM DAC with a switching block.

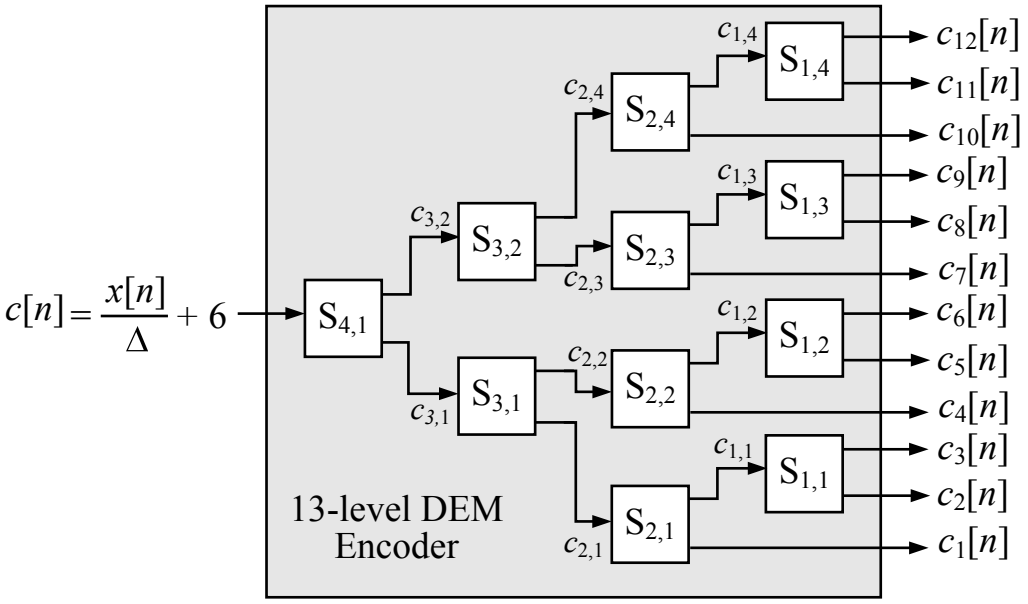


Figure 2.3: A 13-level DEM encoder.

$S_{2,i}$ switching block, $G_{2,i}$=2/3, $c_{1,i}[n]$=2/3$c_{2,i}[n]$-$s_{2,i}[n]$/2, and $c_{1+3(i-1)}[n]$=1/3$c_{2,i}[n]$+$s_{2,i}[n]$/2, $i \in \{1,2,3,4\}$			
$c_{2,i}[n]$	$c_{1,i}[n]$	$c_{1+3(i-1)}[n]$	$s_{2,i}[n]/2$
0	0	0	0
1	1	0	-1/3 where P($s_{2,i}[n]/2$ = -1/3) =2/3
	0	1	2/3 where P($s_{2,i}[n]/2$ = 2/3) =1/3
2	2	0	-2/3 where P($s_{2,i}[n]/2$ = -2/3) =1/3
	1	1	1/3 where P($s_{2,i}[n]/2$ = 1/3) =2/3
3	2	1	0

Figure 2.4: Details of switching blocks $S_{2,i}$.

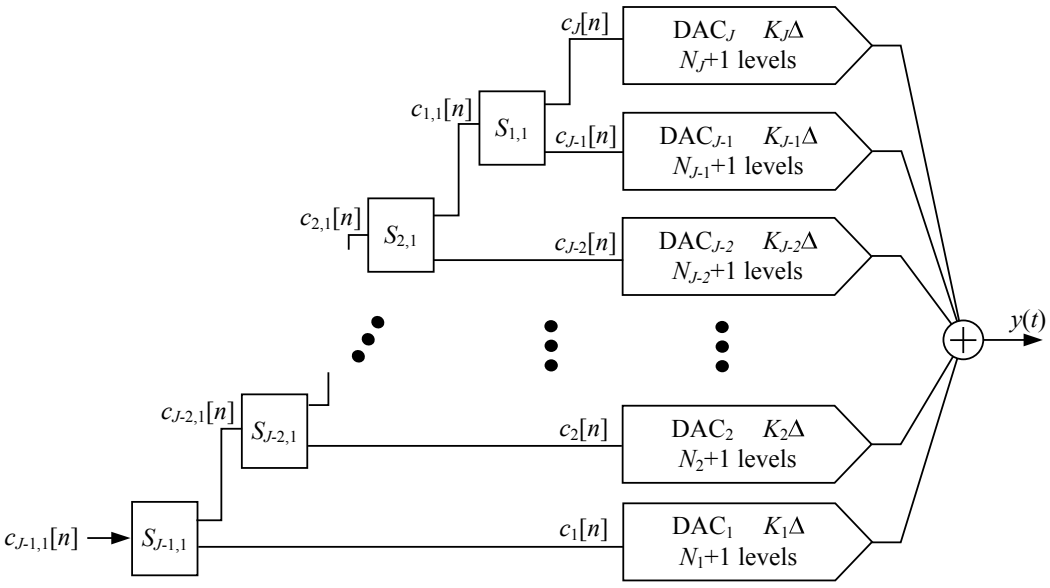


Figure 2.5: A Segmented DEM DAC.

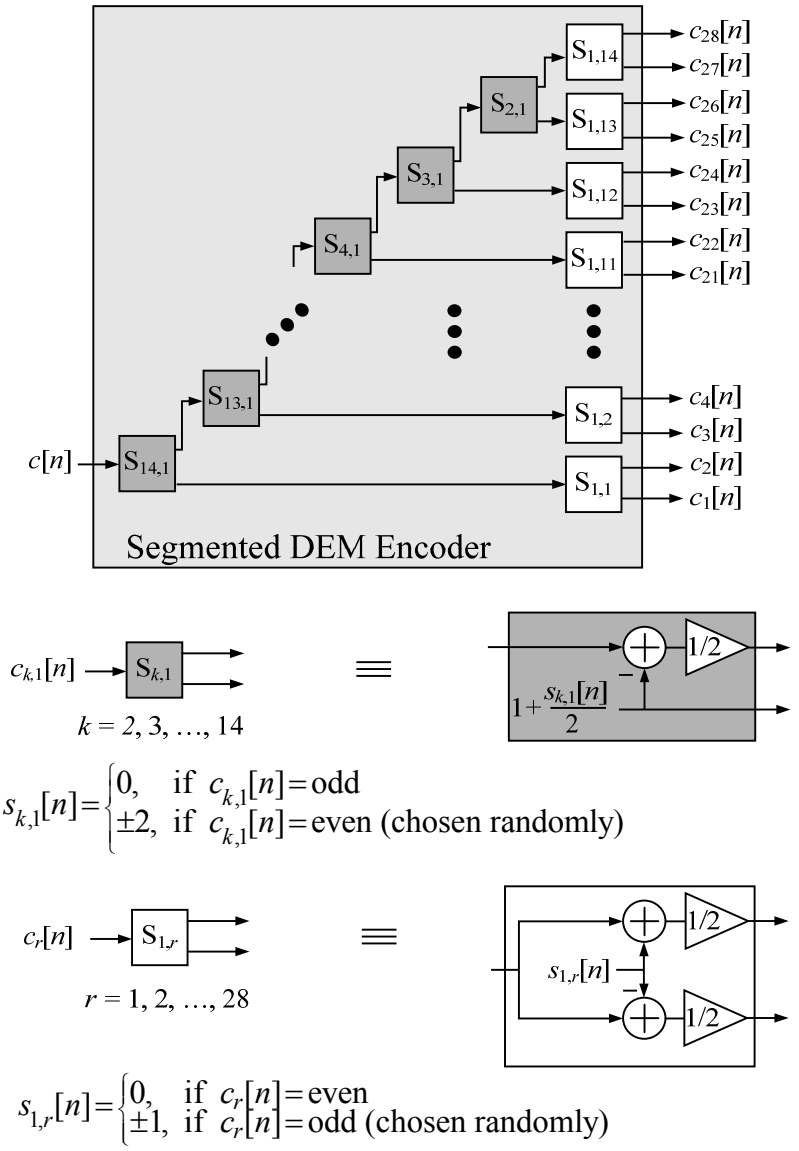
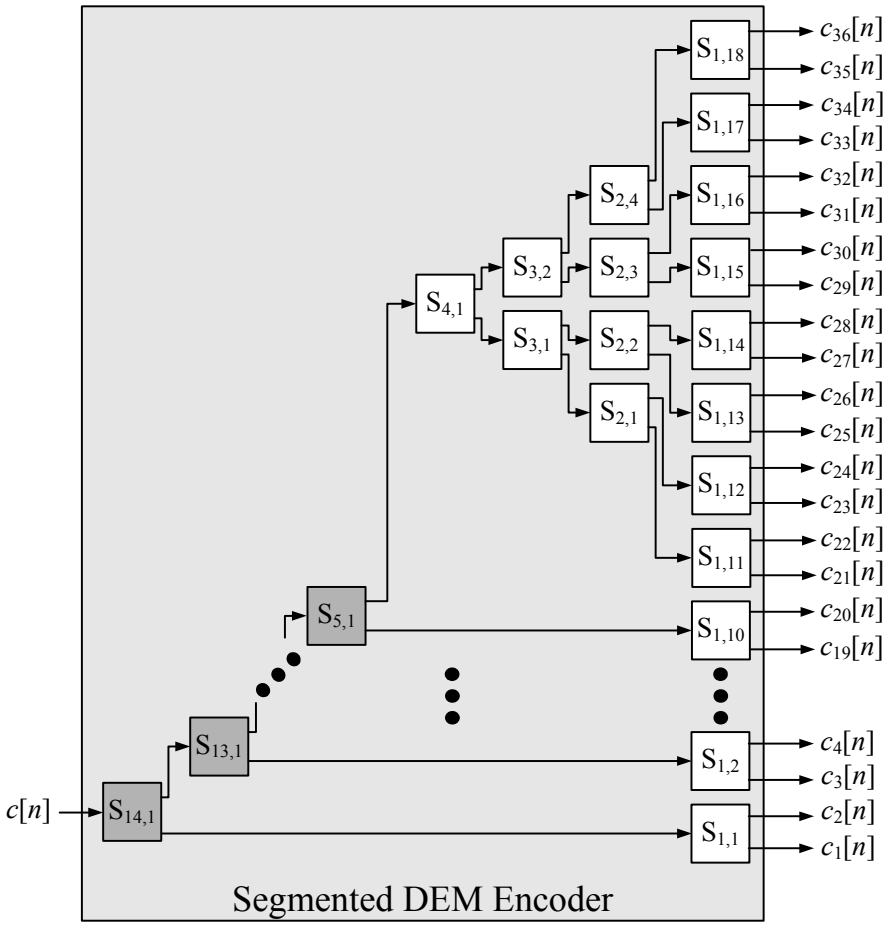
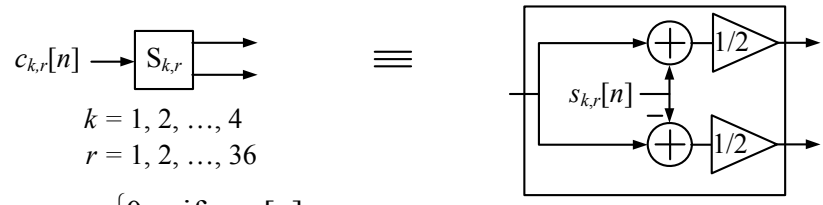


Figure 2.6: A Fully-segmented DEM DAC.



$$s_{k,1}[n]=\begin{cases}0, & \text{if } c_{k,1}[n]=\text{odd}\\ \pm 2, & \text{if } c_{k,1}[n]=\text{even (chosen randomly)}\end{cases}$$



$$s_{k,r}[n]=\begin{cases}0, & \text{if } c_{k,r}[n]=\text{even}\\ \pm 1, & \text{if } c_{k,r}[n]=\text{odd (chosen randomly)}\end{cases}$$

Figure 2.7: A Highly-segmented DEM DAC.

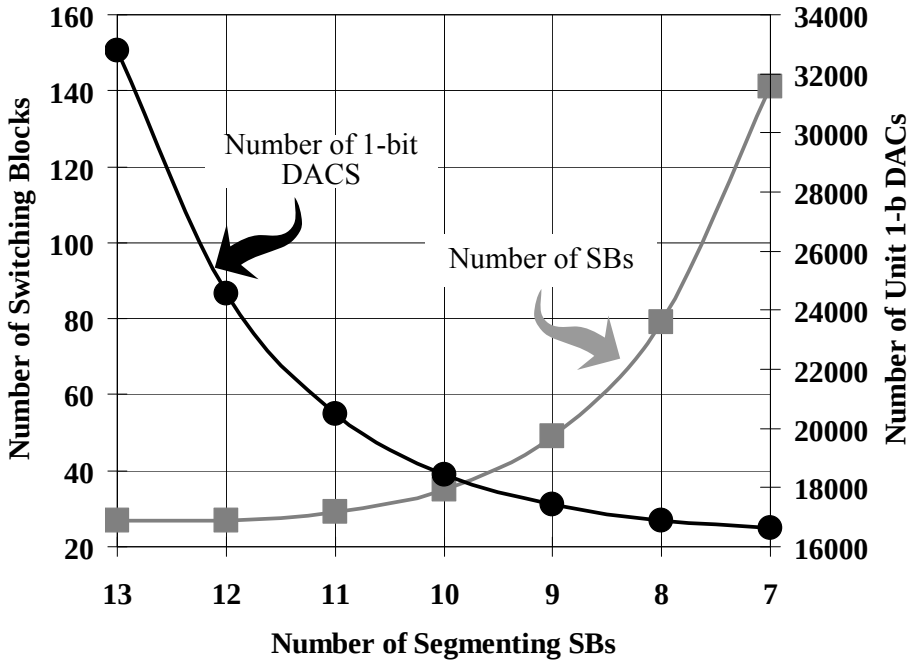


Figure 2.8: Trade-off in the design of the segmented DEM encoder.

VIII. REFERENCES

1 K. Chan, I. Galton, "A 14b 100MS/s DAC with Fully Segmented Dynamic ElementMatching," *IEEE International Solid-State Circuits Conference*, pp. 582-583, 675, February, 2006.

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

3 A. Fishov, et. al., "Segmented Mismatch-Shaping D/A Conversion," *in Proc. IEEE Symp. Circuits and Systems*, 2002.

4 I. Galton, "Spectral Shaping of Circuit Errors in Digital-to-Analog Converters," *IEEE Journal of Solid State Circuits,* vol. 44, no. 10, pp. 808-817, Oct, 1997.

5 K. Chan, J. Zhu, I. Galton, "A 150MS/s 14-bit Segmented DEM DAC with Greater than 83dB of SFDR Across the Nyquist band," *Symposium on VLSI Circuits, Digest of Technical Papers*, June 14, 2007

Chapter 3

Dynamic A 14-b 100-Ms/s DAC with Fully Segmented Dynamic Element Matching

I. Introduction

A 14-b 100-MS/s Nyquist-rate DAC is demonstrated that achieves high linearity across its Nyquist band enabled by a dynamic element matching (DEM) technique called fully segmented DEM (FSDEM). In contrast to previous high-resolution DACs wherein DEM, trimming, calibration, or special layout techniques are applied just to a subset of large current steering DAC elements [1-4], harmonic distortion from pulse-shape and timing mismatches among DAC elements is avoided because FSDEM manipulates all the DAC elements simultaneously. This relaxes several design and layout constraints on the current steering circuits. Unlike previous DEM techniques, the complexity of FSDEM does not grow exponentially with resolution, so it makes involving all the DAC elements in the DEM algorithm practical.

II. Fully Segmented DEM

As shown in Figure 3.1, the DAC contains 28 1-bit *DAC elements* each of which consists of a switch driver and a weighted current steering (CS) cell. During the *n*th sample interval, $nT_s \le t < (n+1)T_s$, the differential output current from the *r*th DAC element is either $A_r\Delta_{pos\,r}(t-nT_s)/2$ or $-A_r\Delta_{neg\,r}(t-nT_s)/2$ depending on $x_r[n]$, where A_r is the CS cell *weighting factor* (i.e., $A_1 = A_2 = 8192$, $A_3 = A_4 = 4096$, etc.), and $\Delta_{pos\,r}(t)$ and

$\Delta_{negr}(t)$ are return-to-zero (RZ) pulses that are zero for $t < 0$ and $t \geq T_s$. The RZ pulses are modeled as $\Delta_{posr}(t) = \Delta(t) + e_{posr}(t)$ and $\Delta_{negr}(t) = \Delta(t) + e_{negr}(t)$, where $\Delta(t)$ is the *ideal LSB DAC pulse* and $e_{posr}(t)$ and $e_{negr}(t)$ are *DAC element mismatch pulses* caused by DAC element pulse shape, timing, and amplitude errors.

The purpose of the FSDEM encoder is to cause the error introduced by the DAC element mismatch pulses to be uncorrelated from the desired signal thereby avoiding harmonic distortion. For each DAC input sample, $x[n]$, where $x[n]$ is an integer between −8192 and 8192, there are multiple sets of 28 DAC element input bits that would yield the correct overall DAC output pulse in the absence of DAC element mismatch pulses. At each sample time, the FSDEM encoder chooses one of these sets pseudo-randomly. The functional details are shown in Figure 3.2. The FSDEM encoder consists of a tree of digital *switching blocks* that each use a pseudo-random sequence, $s_{k,r}[n]$, called a *switching sequence*, to calculate its two output sequences. By design, the switching sequences are uncorrelated with $x[n]$ and each other, and are white.

It can be shown that the differential output current of the DAC for $nT_s \leq t < (n+1)T_s$ is

$$I_{out}(t) = \alpha(t - nT_s)x[n] + \beta(t - nT_s) + e(n, t - nT_s),$$

where $\alpha(t)$, $\beta(t)$, and $e(n, t)$ are RZ pulses that depend on the DAC element mismatch pulses. Both $\alpha(t)$ and $\beta(t)$ are signal-independent, so they do not introduce noise or harmonic distortion. The $e(n, t)$ term has the form

$$e(n,t) = \Delta_{1,1}(t)s_{1,1}[n] + \sum_{i=2}^{14}\left(\Delta_{i,1}(t)s_{i,1}[n] + \Delta_{1,i}(t)s_{1,i}[n]\right),$$

where the $\Delta_{k,r}(t)$ terms are signal-independent RZ pulses that depend only on the DAC element mismatch pulses. Hence, $e(n,t)$ is an RZ pulse whose shape during each sample interval depends on the switching sequences. The train of these pulses in $I_{out}(t)$ is called DAC noise. The above-mentioned switching sequence properties ensure that the DAC noise is uncorrelated with the signal, and that the nth DAC noise pulse is uncorrelated with the mth DAC noise pulse for every $n \neq m$.

Consequently, DAC element pulse shape, timing, and amplitude mismatches and the glitches they cause do not introduce harmonic distortion. This eases several circuit design issues because linearity no longer depends on good component matching and low glitch power; mismatch becomes a secondary concern so the DAC elements can be optimized for low parasitic capacitance and high output impedance to improve linearity at high frequencies. It also obviates the need for glitch reduction techniques such as having a single high-linearity RZ switch circuit following the summed CS cell outputs [5]. Instead, each DAC element in the present design contains its own RZ switch driver. Furthermore, as indicated in Figure 3.1, the switch drivers are not even scaled linearly with the CS cell weights which saved circuit area. Without FSDEM, this alone would have seriously degraded linearity. Despite these benefits there is a drawback to FSDEM: twice the number of CS cells are required relative to a 14-bit binary-weighted DAC. Hence, FSDEM is most appropriate for applications in which maximizing linearity is more critical than minimizing current consumption.

III. Circuit Details and Measurement Results

A simplified circuit diagram of one of the 28 DAC elements is shown in Figure

3.3. It consists of a switch driver and a CS cell. The switch driver consists of a flip-flop to retime the $x_r[n]$ bit and NAND gates to generate the CS cell switch signals. The CS cell is a pMOS cascode current source with current steering switches. The non-overlapping two-phase clock signal and NAND gate circuitry are designed to achieve a low current-steering crossover point. The RZ design ensures signal-independent switch driver activity and DAC element supply current. The remaining dominant sources of high-frequency CS cell nonlinearity are nonlinear capacitance associated with the common source node of the switches, and nonlinear, finite current source output impedance [1]. The CS cell circuits were designed and laid out to minimize these sources of nonlinearity without regard to matching.

The DAC is 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 output was differentially coupled to a spectrum analyzer through a wideband transformer for testing. The measured spurious-free-dynamic range (SFDR) of the DAC versus input frequency is shown Figure 3.4, and representative PSD plots with and without FSDEM enabled are shown in Figure 3.5. With FSDEM enabled, the worst-case SFDR values across the Nyquist bands are 74.4 dB and 78.9 dB for sample-rates of 100 MS/s and 70 MS/s, respectively. As expected, and in contrast to other high-

resolution DACs, the measured SFDR values show little dependence on signal frequency. With FSDEM disabled, these values drop to less than 54 dB which is expected given the lack of attention paid to DAC element matching. A full performance summary and die photograph are shown in Figure. 3.6 and Figure 3.7.

IV. Chapter Acknowledgements

The Text of this chapter appeared as a paper in *IEEE International Solid-State Circuits Conference*, pp. 582-583, 675, February, 2006. The dissertation author was the primary researcher. Ian Galton supervised the research that forms the basis of the chapter.

K. Chan, I. Galton, "A 14b 100MS/s DAC with Fully Segmented Dynamic Element-Matching," *IEEE International Solid-State Circuits Conference*, pp. 582-583, 675, February, 2006.

V. FIGURES

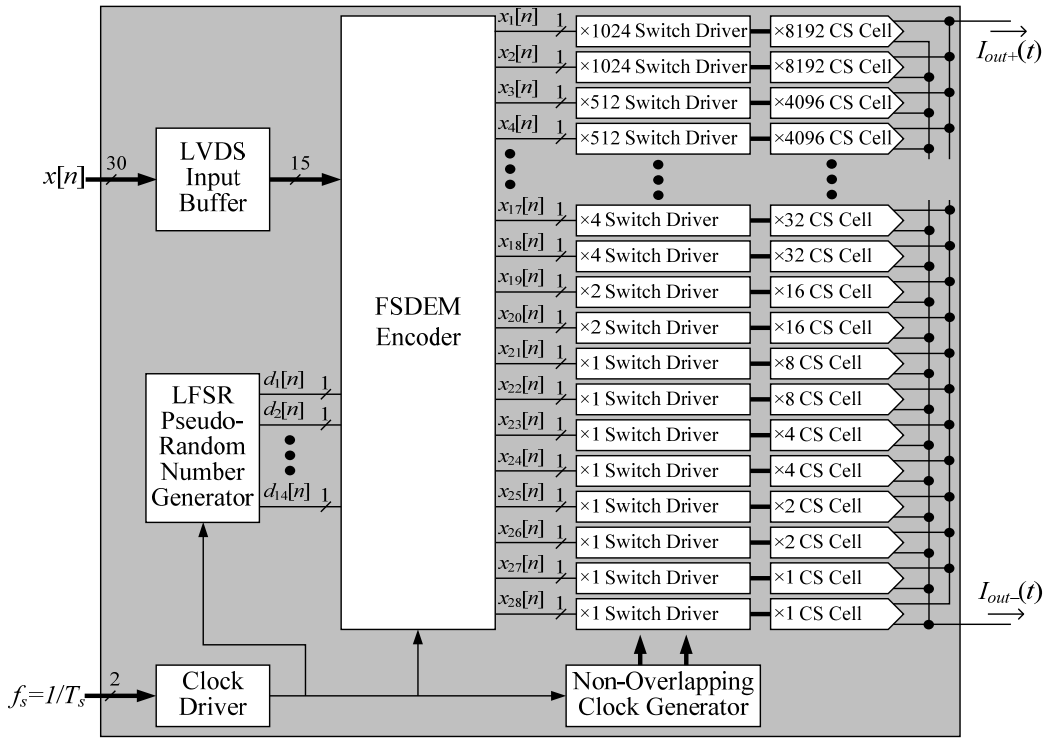


Figure 3.1: Block diagram of integrated circuits.

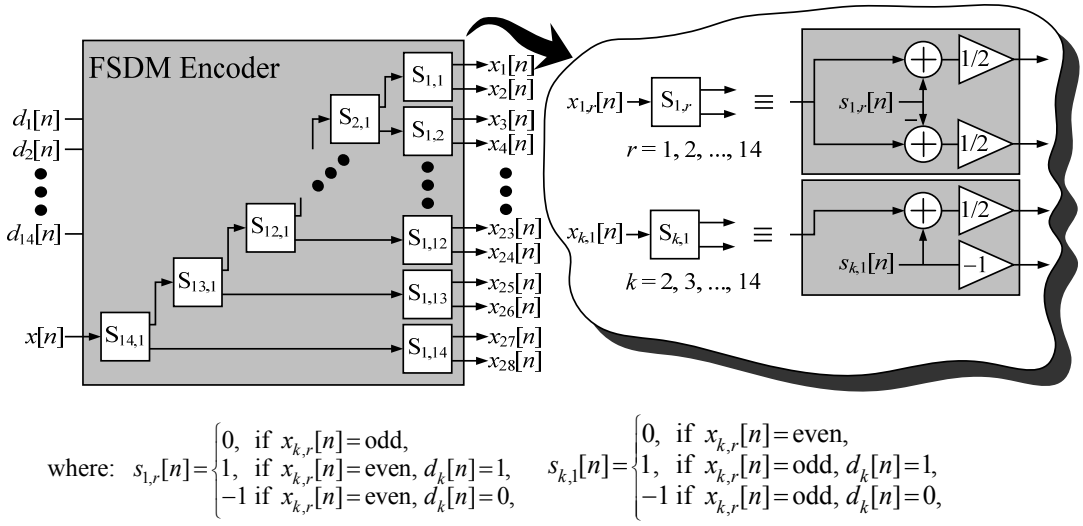


Figure 3.2: Signal processing details of FSDEM encoder.

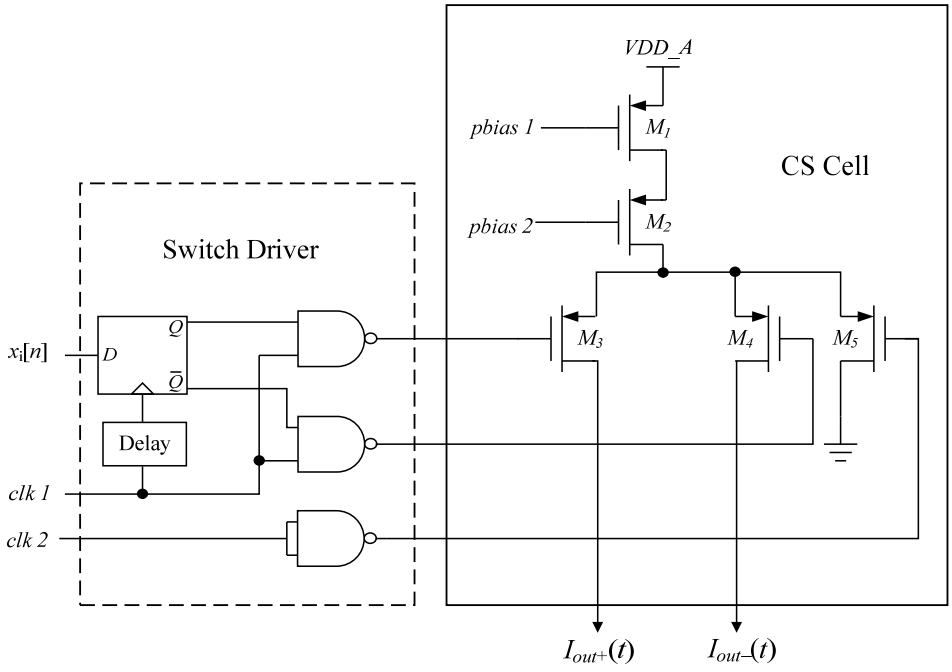


Figure 3.3: DAC element circuit details.

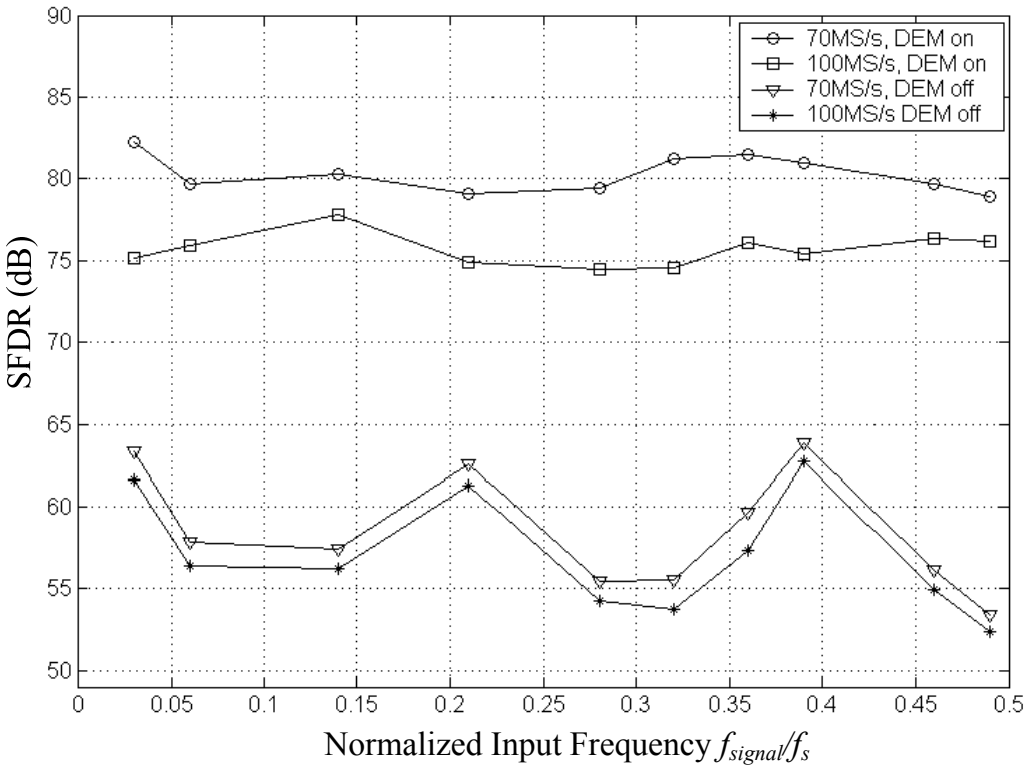


Figure 3.4: Measured SFDR versus frequency

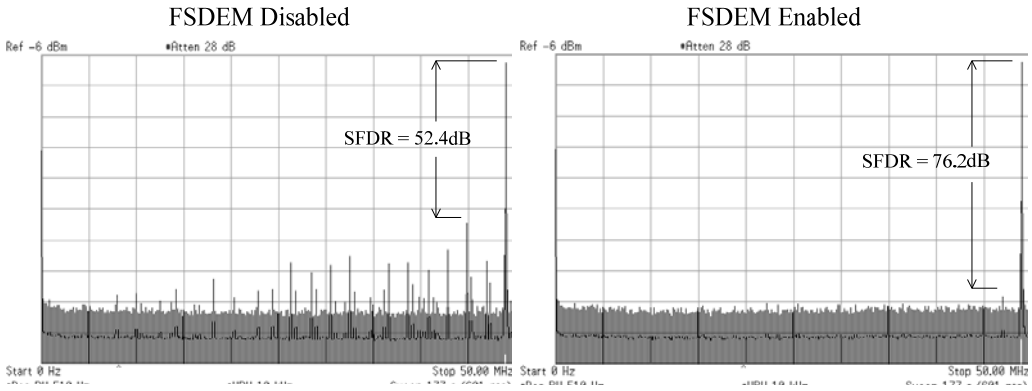


Figure 3.5: Representative measured PSD plots.

Technology	TSMC 0.18 μm CMOS
Update Rate	100 MS/s
Package	QFN 64 with exposed paddle
Single-Tone SFDR @ 70 MS/s, 0 dBFS	78.9 to 82.3 dB across the Nyquist
Single-Tone SFDR @ 100 MS/s, 0 dBFS	74.4 to 77.8 dB across the Nyquist
Two-Tone SFDR @ 100 MS/s, −6 dBFS $f_{signal\ 1}$=13.97 MHz, $f_{signal\ 2}$=14.94 MHz	82.1 dB
Two-Tone SFDR @ 100 MS/s, −6 dBFS $f_{signal\ 1}$=27.98 MHz, $f_{signal\ 2}$=28.95 MHz	82.8 dB
Two-Tone SFDR @ 100 MS/s, −6 dBFS $f_{signal\ 1}$=45.99 MHz, $f_{signal\ 2}$=46.97 MHz	80.6 dB
Full-Scale Current	16 mA
Supply Voltages	Analog: 1.8 V, Digital: 2.3 V
Current Consumption @100 MS/s	Analog:30 mA, Digital: 53.5 V
Area (including bond pads)	4.8 mm x 2.4 mm
Active Area	3.18 mm^2

Figure 3.6: Performance table.

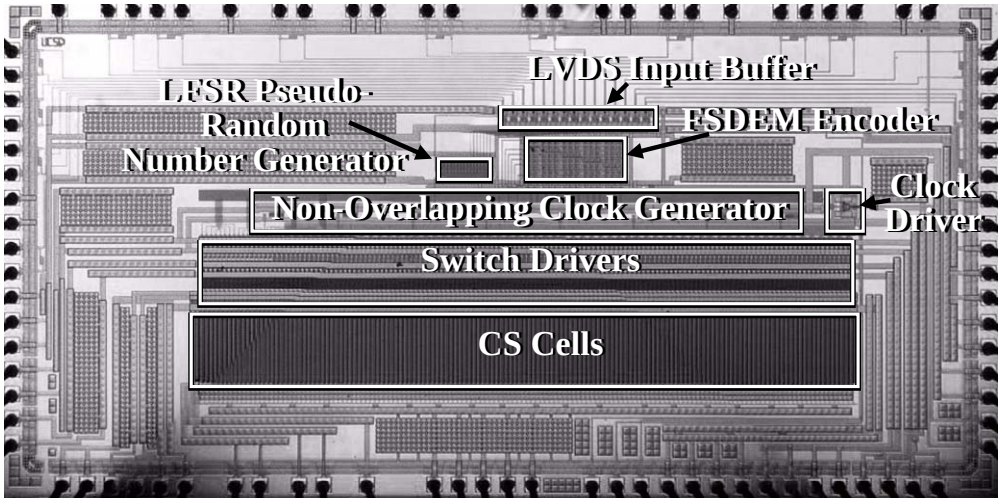


Figure 3.7: Die photograph.

VI. REFERENCES

1. W. Schofield, et. al., "A 16b 400MS/s DAC with <-80dBc IMD to 300MHz and <-160dBm/Hz Noise Power Spectral Density," *IEEE ISSCC Dig. Tech. Papers*, 2003.

2. B. Jewett, et. al., "A 1.2 GS/s 15b DAC for Precision Signal Generation," *IEEE ISSCC Dig. Tech. Papers*, 2005.

3. J. Hyde, et. al., "A 300-MS/s 14-bit Digital-to-Analog Converter in Logic CMOS," *IEEE J. Solid-State Circuits*, May 2003.

4. G. A. M. Van-Der-Plas, et. al., "A 14-bit Intrinsic Accuracy Q^2 Random Walk CMOS DAC," *IEEE J. Solid-State Circuits*, Dec 1999.

5. Q. Huang, et. al., "A 200MS/s 14b 97mW DAC in 0.18μm CMOS," *ISSCC Dig. Tech. Papers*, 2004.