INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UNIVERSITY OF CALIFORNIA, SAN DIEGO

The Analysis and Design of Mismatch-Shaping Digital-to-Analog Converters

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

Doctor of Philosophy

in Electrical and Computer Engineering

by

Jared Eugene Welz

Committee in charge:

Professor Ian Galton, Chair
Professor William Hodgkiss
Professor Bhaskar Rao
Professor Paul Siegel
Professor Patrick Fitzsimmons

2002

UMI Number: 3055804

UMI®

UMI Microform 3055804

Copyright 2002 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

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

Copyright ©

Jared Eugene Welz. 2002

All rights reserved.

To Shirlene Miyake

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Ian Galton, for his guidance, support, patience, and most importantly, friendship throughout this whole Ph.D odyssey. As his "Padawan learner", I was able to realize a potential that, at the onset of the journey, was apparent only in my midichlorian count. I'll always remember the wisdom of his Jedi lessons, including: "try not! Do or do not...there is no try!" I'd also like to thank his family—Kerry, Riley, and Mitchell—for "bringing balance to the Force."

I would like to thank Professor William Hodgkiss, Professor Paul Siegel, Professor Patrick Fitzsimmons, and Professor Bhaskar Rao for serving on my doctoral committee. I would also like to thank Karol Previte, Carolyn Kuttner, Jim Thomas, and the rest of the rebel alliance in the ECE Department for their support.

I'd like to thank my biological parents, Mary Joan and Ed Welz, for believing in their little Gungan. Specifically, I thank my pa for filling out my undergraduate college applications and providing me with the opportunities that made this degree possible and yet were never accessible to him.

I'd also like to thank my parents-in-law, Lois and Richard Miyake, for their love, friendship, and support during my years at UCSD. I am grateful for how they took care of me and my babies and for providing me the much-needed refuge from the dark side of graduate school.

I would like to thank my friends in the rebel alliance, many of which I consider family, for being there for me when I have needed them and for understanding my struggles with the dark side as a Padawan learner at UCSD. This collection of friends includes the Balloonheads: Brian "Luke Skywalker" Ehler, John "Chewbacca" Guest, and Leland "Yoda" Jay. This collection also includes Shelly Ehler,

Michelle Guest, GiGi Carrano, Alison and Jeff Toda, Stephanie and Peter Wu, Ashley and Thomas Sohn, Hena Borneo, Shirley Wang, Diane Wong, Joey and Bill Malohn...May the Force be with you, always.

I wish to thank the current and former members of the ISPiG Jedi Council for their friendship and wisdom—Eric "EZ-E" Fogleman, Asaf "Saf-dogg" Fishov, Sheng "The Man" Ye, Sudhakar "Monsoon" Pamarti, Ash "The Swami Salami" Swaminathan, Eric "The Goose" Siragusa, Alan "Fish Boy" Lewis, Bill Huff, and Henrik Jensen. Particularly, I'd to like express my gratitude to the Monsoon: much of this work was inspired by our conversations that ranged from the meaning of statistical independence to yo-yo tricks. Also, I'd like to thank the venerable Jedi Master EZ-E for mentoring me during the toughest times of my graduate school experience. His support during these times enabled me to overcome my struggles with the dark side that nearly thwarted the completion of this dissertation.

I would like to thank my babies on the moon of Endor—Pooka (the Wampa), Noodle (my bestest buddy in the whole wide world), Jumar, Cricket, Pumpkin, Zippity, Kiwi, Lima, Mila, Mochi, and Chia --whose love is more powerful then any Death Star.

Finally, I'd like to thank my R2 unit, Shirlene Miyake, for helping me make "the Kessel run in less than twelve parsecs." Her love inspired every result in this dissertation while her brilliance helped guide me through each obstacle in my Jedi training. More importantly, she showed me that there is so much more to life then going "into Toshi Station to pick up some power converters."

The four chapters that compose this dissertation are intended to be published as separate papers. Chapter 1 has been submitted for review with the *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing* and covers

material presented at the *International Symposium on Circuits and Systems* in May 2002. Chapter 2 appeared in the November 2001 issue of the *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing.* Chapter 3 and 4 are in preparation for submission to the *IEEE Transactions on Information Theory.* Chapter 4 includes material that was presented at the *International Conference on Acoustics. Speech. and Signal Processing* in May 2001. This work was supported by the Office of Naval Research under Grant N00014-98-1-0830.

VITA

1993	Bachelor of Science. University of California. Irvine
1994	Master of Science. University of California. Los Angeles
1994-1995	Assistant Network Engineer AirTouch International (now Vodafone AirTouch)
1995-1996	RF Engineer Pacific Bell Wireless (now Cingular Wireless)
1996-1997	Advanced Technologies Engineer Los Angeles Cellular (now AT&T Wireless)
1997-2002	Graduate Student Researcher Department of Electrical and Computer Engineering. University of California. San Diego
2002	Doctor of Philosophy. University of California. San Diego

ABSTRACT OF DISSERTATION

The Analysis and Design of

Mismatch-Shaping Digital-to-Analog Converters

by

Jared Eugene Welz

Doctor of Philosophy in Electrical and Computer Engineering

(Communications Theory and Systems)

University of California. San Diego. 2002

Professor Ian Galton. Chair

Multi-bit digital-to-analog converters (DACs) are often constructed by combining several 1-bit DACs in parallel. In such a DAC. mismatches among the 1-bit DACs cause its output to be a nonlinear function of its input. This error is modeled as an additive noise source called the *DAC noise*. The DAC noise limits the attainable resolution of the multi-bit DAC and. if not addressed. prohibits its use in high-performance applications. *Mismatch-shaping* DACs mitigate this problem by suppressing the DAC noise power in the data signal's frequency band so that most of it can be removed by frequency-selective filters. These DACs facilitate multi-bit delta-sigma ($\Delta\Sigma$) modulation and have thus become widely used in high-performance $\Delta\Sigma$ data converters.

However, the theoretical analyses of mismatch-shaping DACs have been limited. For most architectures, the analysis is limited to proving that the DAC noise power spectral density (PSD) is zero at some frequency. Typically, this analysis pertains only to the specific architecture and does not provide a reasonable estimate of the signal-band power of the DAC noise. Consequently, engineers usually rely on simulations to predict their DAC's performance, which can be misleading.

This dissertation provides a unifying theory for mismatch-shaping DACs and furthers the development and analysis of an architecture called the *tree-structured DAC*. The unifying theory, which is given in Chapter 1, is in the form of necessary and sufficient conditions for a multi-bit DAC to be a mismatch-shaping DAC. These conditions are used to analyze and compare several well-known mismatch-shaping DACs. Chapter 2 presents different implementations of the tree-structured DAC that give rise to performance and complexity trade-offs. One such implementation, the dithered first-order low-pass tree-structured DAC, is analyzed in Chapter 3. In this chapter, expressions for the DAC noise PSD and signal-band power are derived and used to obtain an achievable power bound for the DAC noise. In Chapter 4, the DAC noise PSD expression from Chapter 3 is used to develop the theoretical DAC noise PSD for the tree-structured DAC in a $\Delta\Sigma$ modulator application.

Chapter 1

Necessary and Sufficient Conditions for Mismatch Shaping in Multi-Bit Digital-to-Analog Converters

Jared Welz, Ian Galton

Abstract—Multi-bit DACs are often constructed by combining several 1-bit DACs of equal or different weights in parallel. In such DACs, component mismatches give rise to signal dependent error that can be viewed as additive *DAC noise*. In some cases these DACs use dynamic element matching techniques to decorrelate the DAC mismatch noise from the input sequence and suppress its power in certain frequency bands. Such DACs are referred to as mismatch-shaping DACs and have been used widely as enabling components in state-of-the-art delta-sigma data converters. Several different mismatch-shaping DAC topologies have been presented, but theoretical analyses have been scarce and no general unifying theory has been presented in the previously published literature. This paper presents such a unifying theory in the form of necessary and sufficient conditions for a multi-bit DAC to be a mismatch-shaping DAC, and applies the conditions to evaluate the DAC noise generated by several of the previously published mismatch-shaping DACs, and qualitatively compare their behavior.

I. INTRODUCTION

MOST multi-bit digital-to-analog converters (DACs) consist of multiple 1-bit DACs. In each case, the digital input sequence is decomposed into multiple 1-bit sequences each of which drives a 1-bit DAC. Each 1-bit DAC generates one of two analog output levels depending upon whether its input bit is high or low. The

outputs of the 1-bit DACs are summed to form the output of the multi-bit DAC. The primary differences among the various multi-bit DAC architectures reside in how the multi-bit input sequence is mapped to the multiple 1-bit DAC input sequences, and how the output levels of the 1-bit DACs are scaled relative to each other.

In practice, component mismatches inevitably introduced during circuit fabrication, most notably mismatches among nominally identical unit capacitors or current sources, cause the 1-bit DAC output levels to deviate from their ideal values. The resulting error can be modeled, without approximation, as additive error and is referred to as *DAC noise*. In present VLSI technology, the values of nominally identical components can rarely be matched to better than a standard deviation of 0.1%. In Nyquist-rate DACs, i.e., DACs that convert digital signals with a pass-band from zero up to half their sample-rate, this translates into DAC noise that limits the achievable signal-to-noise-and-distortion ratio (SINAD) to less than 70 dB. Moreover, without some form of dither or other randomization technique, the DAC noise is a deterministic, nonlinear function of the input sequence so it contains harmonic distortion which can be problematic in many applications.

Dynamic element matching (DEM) techniques can be applied to multi-bit DACs both to suppress the power of the DAC mismatch noise in specific frequency bands and to eliminate the harmonic distortion. Such multi-bit DACs are referred to as *mismatch-shaping DACs*. They are particularly useful in applications that require high precision within relatively narrow frequency bands. As such, in recent years they have become widely used in high-performance delta-sigma ($\Delta\Sigma$) data converters.

Although numerous mismatch-shaping DAC architectures have been developed, published mathematical analyses of these DACs have been limited and disjoint to

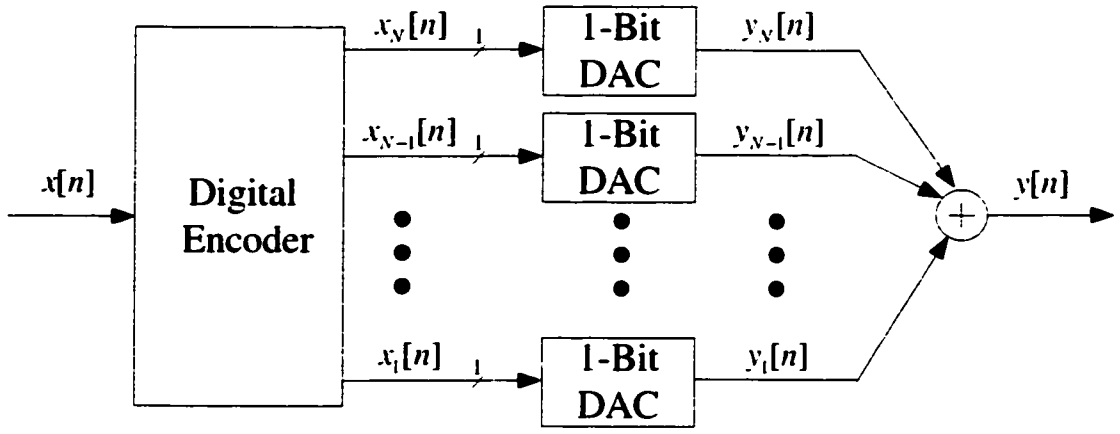


Figure 1.1: The general multi-bit DAC.

date. Most analyses have been individually tailored to specific architectures, and in most cases simulations have been relied upon to determine the characteristics of the DAC noise, which can be misleading. Consequently, there is no unifying theory that applies to multi-bit DACs in general. This lack of theory has made it difficult to compare the merits of the different mismatch-shaping DAC architectures, and likely has impeded the development of new mismatch-shaping DAC architectures.

This paper provides a unifying theory in the form of necessary and sufficient conditions for a general multi-bit DAC to be a mismatch-shaping DAC. Unlike previous analyses [1] the conditions do not rely on properties of the component mismatches. The utility of the conditions is demonstrated by using them to analyze and qualitatively compare most of the widely used mismatch-shaping DAC architectures published to date: first-order, lowpass implementations of the vector feedback [2], data-weighted averaging (DWA) [3],[4], butterfly shuffler [5], tree structured [6], and partitioned DWA [7] DACs.

II. THE GENERAL MULTI-BIT DAC

The general multi-bit DAC shown in Figure 1.1 consists of a digital encoder and a

bank of N 1-bit DACs. The output of the ith 1-bit DAC is given by

$$y_i[n] = \begin{cases} \frac{\Delta_i}{2} + e_{h_i}, & \text{if } x_i[n] \text{ is high:} \\ -\frac{\Delta_i}{2} + e_{l_i}, & \text{if } x_i[n] \text{ is low:} \end{cases} \tag{1}$$

where Δ_i is the nominal step size of the ith 1-bit DAC, and e_{h_i} and e_{l_i} are its high and low errors, respectively. In many applications, the 1-bit DAC errors result from component mismatches introduced during fabrication of the 1-bit DACs. As such, they are modeled here as arbitrary constants. The digital encoder output is a vector, $\vec{x}[n]$, of N 1-bit sequences, $x_1[n], \ldots, x_N[n]$. The value of each 1-bit sequence is taken to be 1/2 when it is high and -1/2 when it is low. Ideally, a DAC's output is a scaled version of its input. To ensure that the multi-bit DAC approaches this ideal behavior when the 1-bit DAC errors approach zero, the digital encoder determines its output sequences under the following restriction:

$$\sum_{i=1}^{N} \Delta_i \cdot x_i[n] = \Delta_D \cdot x[n], \tag{2}$$

where Δ_D is the nominal smallest step size of the multi-bit DAC. Thus, if the 1-bit DAC errors were all zero, (1) and (2) imply that the DAC output would be given by

$$y[n] = \Delta_D x[n]. \tag{3}$$

However, in practice the 1-bit DAC errors are nonzero, and, as a result, the multi-bit DAC output is a nonlinear function of the multi-bit DAC input. The error from this nonlinearity can be written as additive error:

$$y[n] = \Delta_D x[n] + \tilde{e}[n]. \tag{4}$$

The error sequence $\tilde{e}[n]$ often contains a constant offset and scaled version of the input; therefore, it is convenient to write (4) as

$$y[n] = \alpha x[n] + \beta + e[n], \tag{5}$$

where α and β are constants, and $e[n]$ is called the *DAC noise*. In a well-designed system, the DAC noise is a zero mean sequence that is uncorrelated from the multi-bit DAC input, and the constants α and β depend only on the 1-bit DAC errors.

Mismatch-shaping DACs are designed such that the digital encoder has several possible output vector values, $\vec{x}[n]$, that satisfy (2) for most DAC input values. For example, in a multi-bit DAC in which all the 1-bit DACs have the same nominal step size, a nominal output value of zero is obtained for any output vector with an equal number of high and low bit values. By exploiting this flexibility, the DAC noise can be tailored so that its PSD has desired properties regardless of the values of the 1-bit DAC errors. Therefore, a multi-bit DAC is said to *produce DAC noise with a given set of PSD properties* if, for any collection of 1-bit DAC errors, there exist constants α and β, and a sequence $e[n]$ with the given set of PSD properties such that $y[n] = \alpha x[n] + \beta + e[n]$.

Various DAC noise PSD properties can be obtained by mismatch-shaping DACs. In some DACs, the digital encoder operates such that the DAC noise is white; *i.e.*, its PSD is constant with respect to frequency. In such DACs, the power of the white noise depends upon the 1-bit DAC errors (*e.g.*, larger 1-bit DAC errors tend to increase the power of the DAC noise), but the DAC noise is white for any choice of the 1-bit DAC errors. In other DACs, the digital encoder operates such that the DAC noise PSD is continuous with a value of zero at zero frequency: $\omega = 0$. In such cases, the power of the DAC noise tends to reside predominantly at high frequencies. Again, the overall power of the DAC noise depends upon the 1-bit DAC errors, but the zero at $\omega = 0$ and the weighting of the PSD toward high frequencies occurs for any choice of 1-bit DAC errors. Various other DACs are possible that achieve different DAC noise properties. In each case, specific properties (*e.g.*, zero

location) of the DAC noise PSD are preserved regardless of the 1-bit DAC errors.

III. THE CONDITION FOR MISMATCH SHAPING

The theorem below presents a necessary and sufficient condition for the general multi-bit DAC to produce DAC noise with a given set of PSD properties.

Theorem: The multi-bit DAC in Figure 1.1 produces DAC noise with a given set of PSD properties if and only if there exist $N-1$ sequences $\phi_1[n], \ldots, \phi_{N-1}[n]$ such that:

(a) each digital encoder output is given by

$$x_i[n] = m_i x[n] + \sum_{j=1}^{N-1} d_{i,j} \cdot \phi_j[n]. \tag{6}$$

where $d_{i,j}$ and m_i are constants, and

(b) for any selection of the $N-1$ constants $c_1, \ldots, c_{N-1}$, there exist two constants a and b, and a sequence $\varepsilon[n]$ with the given set of PSD properties such that

$$\sum_{j=1}^{N-1} c_j \cdot \phi_j[n] = ax[n] + b + \varepsilon[n]. \tag{7}$$

Proof: Because $x_i[n]$ is interpreted as 1/2 when high and -1/2 when low, (1) can be written as

$$y_i[n] = \xi_i x_i[n] + \gamma_i. \tag{8}$$

where $\xi_i \equiv \Delta_i - (e_{h_i} - e_{l_i})$ and $\gamma_i \equiv (e_{h_i} + e_{l_i})/2$. Given $y[n] = \sum_{i=1}^{N} y_i[n]$, (8) implies that

$$y[n] = \sum_{i=1}^{N} \xi_i x_i[n] + \beta_o. \tag{9}$$

where $\beta_o \equiv \sum_{i=1}^{N} \gamma_i$.

Sufficiency: Assume that the $N-1$ sequences, $\phi_1[n], \ldots, \phi_{N-1}[n]$, exist and satisfy (a) and (b) in the theorem. Substituting (6) into (9) gives

$$y[n] = \underbrace{\left(\sum_{i=1}^{N} \xi_i m_i\right)}_{\equiv \alpha_o} x[n] + \sum_{j=1}^{N-1} \underbrace{\left(\sum_{k=1}^{N} \xi_k d_{k,j}\right)}_{\equiv c_j} \phi_j[n] + \beta_o. \tag{10}$$

Condition (b) implies that the second summation in (10) can be decomposed as in (7). Thus, substituting (7) into (10) gives

$$y[n] = \underbrace{(a + \alpha_o)}_{\equiv \alpha} x[n] + \underbrace{b + \beta_o}_{\equiv \beta} + \varepsilon[n]. \tag{11}$$

where $\varepsilon[n]$ has the given set of PSD properties, so the multi-bit DAC produces DAC noise with the given set of PSD properties.

Necessity: To reduce wordiness, all "linear combinations" discussed hereafter are assumed to have constant coefficients. Let $\phi_N[n] = \sum_{i=1}^{N} (\Delta_i/\Delta_D)\, x_i[n]$, which, by (2), implies that $\phi_N[n] = x[n]$, and let $\phi_1[n], \ldots, \phi_{N-1}[n]$ be any collection of $N-1$ linear combinations of the digital encoder outputs subject to the constraint that these linear combinations and the one that generates $\phi_N[n]$ are linearly independent. Then there exists an invertible $N \times N$ matrix A with values $a_{j,k}$, where j is the row number and k is the column number, with $a_{N,k} \equiv \Delta_k/\Delta_D$ such that $\vec{\phi}[n] = A\vec{x}[n]$, and, for each j,

$$\phi_j[n] = \sum_{k=1}^{N} a_{j,k} \cdot x_k[n]. \tag{12}$$

Let D, whose value in its ith row and jth column is denoted $d_{i,j}$, be the inverse matrix of A. This implies that $\vec{x}[n] = D\vec{\phi}[n]$ and, for each i,

$$x_i[n] = \sum_{j=1}^{N} d_{i,j} \cdot \phi_j[n]. \tag{13}$$

With $m_i \equiv d_{i,N}$, (13) is identical to (6) because $\phi_N[n] = x[n]$. Therefore, the $N-1$ sequences $\phi_1[n], \ldots, \phi_{N-1}[n]$ satisfy condition (a) in the theorem.

To show that the $N-1$ sequences satisfy condition (b) in the theorem, assume the multi-bit DAC produces DAC noise with the given set of PSD properties. In (9), ξ_i and β_0 are arbitrary constants because each DAC error is an arbitrary constant. Thus, by assumption, *for any* selection of the constants $\xi_1, \ldots, \xi_N$, and β_0, there exist constants a and b, and a sequence $\varepsilon[n]$ with the given set of PSD properties such that

$$\sum_{i=1}^{N} \xi_i x_i[n] + \beta_0 = ax[n] + b + \varepsilon[n]. \tag{14}$$

It follows from (12) that

$$\sum_{j=1}^{N-1} c_j \cdot \phi_j[n] = \sum_{i=1}^{N} \underbrace{\left(\sum_{j=1}^{N-1} c_j \cdot a_{j,i} \right)}_{\equiv d_i} x_i[n]. \tag{15}$$

for any selection of $N-1$ constants, $c_1, \ldots, c_{N-1}$. Since (14) is satisfied for any selection of ξ_i and β_0, suppose $\xi_i = d_i$ for each i, and $\beta_0 = 0$. In this case, the left-hand side of (14) is the same as the right-hand side of (15), which implies (7). Thus, the $N-1$ sequences satisfy condition (b) in the theorem.

■

Therefore, in mismatch-shaping DACs, there are $N-1$ underlying sequences that, given the DAC input, determine the digital encoder outputs and, when linearly combined, produce a sequence that has the same form as the DAC output, *i.e.*, $ax[n] + b + \varepsilon[n]$, where the gain and offset depend on the coefficients in this linear combination, and the sequence $\varepsilon[n]$ has the same PSD properties as the DAC noise.

In efficient mismatch-shaping DACs, none of the $N-1$ underlying sequences are constant for all DAC input sequences. To verify this assertion, suppose one of the digital encoder outputs were a linear combination of the other digital encoder

outputs plus an offset: *i.e.*, for some j,

$$x_j[n] = \sum_{\substack{i=1 \\ i \neq j}}^{N} d_i \cdot x_i[n] + d_0. \tag{16}$$

where each d_i is a constant. As a consequence of this linear dependence, an *equivalent* multi-bit DAC could be implemented using fewer than N 1-bit DACs: the only difference between the original and equivalent implementations would result from the 1-bit DAC errors in each. For example, if $x_j[n]$ were given by (16), then the jth 1-bit DAC could be removed by changing the nominal step sizes of the other 1-bit DACs according to the following: for $i \neq j$, $\Delta_i^{new} = \Delta_i^{old} + d_i \Delta_j^{old}$.

The theorem can be used to show that the DAC noise from a given architecture has certain PSD properties. However, the corollary presented next is more convenient for this application.

Corollary 1: Given the multi-bit DAC in Figure 1.1, let $\phi_1[n], \ldots, \phi_{N-1}[n]$, and $\psi[n]$ be sequences formed by taking N linearly independent, linear combinations of the digital encoder outputs with $\psi[n] = \sum_{i=1}^{N} (\Delta_i / \Delta_D) x_i[n]$. Then, the multi-bit DAC in Figure 1.1 produces DAC noise with a given set of PSD properties if and only if, for any selection of the $N-1$ constants $c_1, \ldots, c_{N-1}$, there exist two constants a and b, and a sequence $\varepsilon[n]$ with the given set of PSD properties such that

$$\sum_{j=1}^{N-1} c_j \cdot \phi_j[n] = ax[n] + b + \varepsilon[n]. \tag{17}$$

Proof: The proof follows directly from that of the theorem as the $N-1$ sequences in the corollary are formed the same way as in the proof of the theorem.

■

Therefore, to show that the DAC noise PSD from a given multi-bit DAC has a certain property, derive the $N-1$ sequences, $\phi_1[n], \ldots, \phi_{N-1}[n]$, as described in the corollary and show that any linear combination of these sequences can be written as in (17). The $N-1$ sequences in the corollary result from linear combinations of the digital encoder outputs, and there are many possible choices for these sequences. However, for a given multi-bit DAC, these sequences can often be chosen to minimize the effort required to show they satisfy (17). Several examples of this application are presented in the following section.

The following corollary is more convenient than the theorem or the first corollary for proving that the DAC noise from a given architecture *does not* have certain PSD properties.

Corollary 2: The multi-bit DAC in Figure 1.1 produces DAC noise with a given set of PSD properties if and only if, for any selection of N constants, $d_1, \ldots, d_N$, there exist constants a and b, and a sequence $\varepsilon[n]$ with the given set of PSD properties such that

$$\sum_{i=1}^{N} d_i \cdot x_i[n] = ax[n] + b + \varepsilon[n]. \tag{18}$$

Proof: By definition, the multi-bit DAC produces DAC noise with the given set of PSD properties, if and only if, for any selection of the 1-bit DAC errors, there exist two constants α and β, and a sequence $e[n]$ with the given set of PSD properties such that $y[n] = \alpha x[n] + \beta + e[n]$. The relationship between $y[n]$ and the 1-bit DAC errors is manifest in (9) as each constant ξ_i and β_o are functions of the 1-bit DAC errors. Because any value of ξ_i and β_o can be obtained by an appropriate choice of the 1-bit DAC errors, the multi-bit DAC produces DAC noise with the given set of

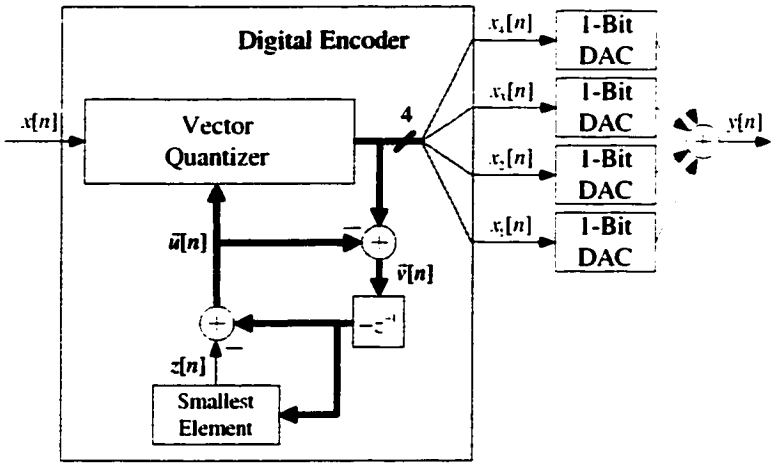


Figure 1.2: A first-order, lowpass vector feedback DAC.

PSD properties if and only if, for any selection of the constants $\xi_1, \ldots, \xi_N$, and β_o, there exist two constants α and β, and a sequence $e[n]$ with the given set of PSD properties such that

$$\sum_{i=1}^{N} \xi_i x_i[n] + \beta_o = \alpha x[n] + \beta + e[n]. \tag{19}$$

With $d_i \equiv \xi_i$, $b \equiv \beta - \beta_o$, $a \equiv \alpha$, and $\varepsilon[n] \equiv e[n]$, (19) is equivalent to (18).

■

Therefore, to show that the DAC noise does not have the given PSD properties, it is sufficient to find a linear combination of the digital encoder outputs that cannot be expressed as in (18). An example of this application is also shown in the following section.

IV. ARCHITECTURE ANALYSIS

The theorem and corollaries presented in the previous section are used in this section to analyze and compare several of the previously published multi-bit DAC architectures. Specifically, vector feedback, data weighted averaging, butterfly shuffler, tree structured, and partitioned data weighted averaging DAC architectures are considered.

VECTOR FEEDBACK

A 5-level (*i.e.*, $N = 4$) example of the vector feedback DAC is shown in Figure 1.2 [2]. Its input range is $\{-N/2, -N/2+1, \ldots, N/2\}$. Its 1-bit DACs all have the same nominal step size (*i.e.*, $\Delta_i = \Delta_D$ for each i). The digital encoder consists of a *vector quantizer*, a *smallest-element* block, two vector adders, and a vector unit delay. The vector $\vec{u}[n]$ consists of N elements, the ith of which is associated with the ith output bit of the digital encoder. At each sample time, n, the vector quantizer determines the $x[n] + N/2$ largest elements of $\vec{u}[n]$, and sets the associated output bits of the digital encoder high. It sets the remaining output bits low. The digital encoder calculates each element of $\vec{u}[n]$ as

$$u_i[n] = -v_i[n-1] - z[n]. \tag{20}$$

where

$$v_i[n] = x_i[n] - u_i[n]. \tag{21}$$

and $z[n] = \min_i\{-v_i[n-1]\}$, *i.e.*, it is equal to the smallest element of $-\vec{v}[n-1]$.

To show that the feedback system within the digital encoder is stable, it is sufficient to show that $u_i[n]$ and $v_i[n]$ are bounded sequences for each value of i. Suppose that at some sample time, n_0, the largest element of $\vec{u}[n_0]$ has a value of $P \geq 1$. It follows from (20) that $u_i[n] \geq 0$ for each i and one element of $\vec{u}[n]$ equals zero for each n. The operation of the vector quantizer implies that $x_j[n_0] - x_i[n_0] = 1$ only when $u_j[n_0] \geq u_i[n_0]$. So (21) implies that

$$|v_i[n_0] - v_j[n_0]| \leq \max\{|u_i[n_0] - u_j[n_0]|, 1\} \leq P. \tag{22}$$

It follows from (20) that $u_j[n_0+1] - u_i[n_0+1] = v_i[n_0] - v_j[n_0]$, and since one element of $\vec{u}[n_0+1]$ is zero, (22) implies that $u_i[n_0+1] \leq P$ for each i. By induction, $u_i[n]$ must be a bounded sequence for each i, and, therefore, (21) implies that $v_i[n]$ must also be a bounded sequence for each i.

To apply Corollary 1. let

$$\phi_j[n] \equiv x_{j+1}[n] - x_j[n] \tag{23}$$

for $j = 1, \ldots, N-1$. Because all the 1-bit DACs have the same nominal step size. $\psi[n]$. as defined in the statement of Corollary 1. is given by

$$\psi[n] = x_1[n] + \cdots + x_N[n]. \tag{24}$$

To show that (23) and (24) are linearly independent combinations of the digital encoder output sequences as required by the corollary. it is sufficient to show that. when $x_1[n], \ldots, x_N[n]$ are linearly independent sequences. the expression

$$\sum_{j=1}^{N-1} c_j \cdot \phi_j[n] + c_N \cdot \psi[n] = 0. \tag{25}$$

where $c_1, \ldots, c_N$ are constants. is only satisfied with $c_j = 0$ for each j. Substituting (23) and (24) into (25) gives

$$\sum_{j=1}^{N} (c_{j-1} - c_j + c_N)\, x_j[n] + c_N \cdot x_N[n] = 0. \tag{26}$$

where c_0 is *defined* to be zero. With linearly independent digital encoder outputs. (26) implies that $c_j - c_{j-1} = c_N$ for $j = 1, \ldots, N-1$. and $c_{N-1} = -c_N$. Solving this difference equation gives $c_j = j \cdot c_N$ for $j = 1, \ldots, N-1$. Since $c_{N-1} = (N-1)\, c_N$ and $c_{N-1} = -c_N$ both hold. it follows that $c_N = 0$. Therefore. $c_j = 0$ for each j.

It is next shown that the choice of $\phi_j[n]$ given by (23) satisfies (17) with $a = 0$. $b = 0$. and an $\varepsilon[n]$ whose PSD is zero at $\omega = 0$. By virtue of Corollary 1. this implies that the PSD of the DAC noise also has a zero at $\omega = 0$. and. therefore. that the vector feedback DAC shown in Figure 1.2 is a first-order mismatch-shaping DAC.

Substituting (20) into (21) gives $x_i[n] = v_i[n] - v_i[n-1] - z[n]$. With (23) this implies

$$\phi_j[n] = v_{j+1}[n] - v_{j+1}[n-1] - v_j[n] + v_j[n-1].$$

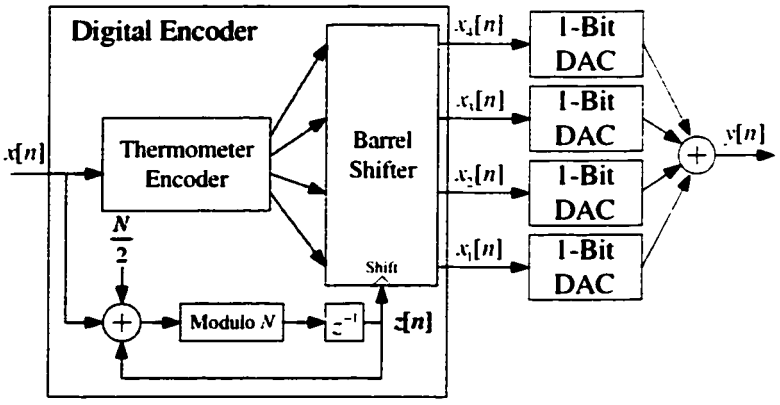


Figure 1.3: The DWA DAC.

Therefore,

$$\left|\sum_{m=0}^{n}\phi_j[m]\right| = |v_{j+1}[n] - v_j[n] - v_{j+1}[0] + v_j[0]|\,. \tag{27}$$

The partial sum in (27) is bounded for all n because $v_i[n]$ is a bounded sequence for each value of i. As shown in the Appendix, this implies that the PSD of $\phi_j[n]$ is zero at $\omega = 0$. It is also shown in the Appendix that any linear combination of such sequences has a PSD equal to zero at $\omega = 0$. Therefore, by Corollary 1, the DAC noise has this property too.

DWA

A 5-level example of the DWA DAC is shown in Figure 1.3 [3], [4]. Like the vector feedback DAC, its input range is $\{-N/2, -N/2 + 1, \ldots, N/2\}$, and all of its 1-bit DACs have the same nominal step size. The digital encoder consists of a thermometer encoder and a barrel shifter. Additionally, it consists of a modulo-N block, a unit delay, and an adder that constitute a *modulo-N accumulator*. At each sample time, n, the thermometer encoder, whose N outputs are binary sequences, selects its bottom $x[n] + N/2$ outputs high and its remaining outputs low. The modulo-N accumulator output, $z[n]$, controls the operation of the barrel shifter as follows: with its inputs and outputs labeled 1 to N from bottom to top, the barrel

shifter, at sample time n, routes input i to output $1 + (z[n] + i - 1) \bmod N$. Thus, the digital encoder outputs are generated by performing a modulo-N shift of the thermometer encoder outputs.

The values of $z[n]$ and $x[n]$ determine the digital encoder outputs at sample time n, and $z[n+1] = (x[n] + z[n] + N/2) \bmod N$. If $z[n] < z[n+1]$, then the high digital encoder outputs at time n are those numbered $z[n]+1, z[n]+2, \ldots, z[n+1]$, and the remaining outputs are low. However, if $z[n] > z[n+1]$, then the low digital encoder outputs at time n are those numbered $z[n+1]+1, z[n+1]+2, \ldots, z[n]$, and the remaining outputs are high. If $z[n+1] = z[n]$, then $x[n] = \pm N/2$, and all of the digital encoder outputs are either high or low at time n. Therefore, at each sample time, n, there is a contiguous segment of either high or low outputs of the digital encoder, and $z[n]$ and $z[n+1]$ determine the segment's starting and ending points.

To analyze the DAC noise using Corollary 1, let $\phi_j[n] \equiv x_{j+1}[n] - x_j[n]$ for $j = 1, \ldots, N-1$. As previously shown, $\psi[n]$, as defined in the corollary, is given by (24), and the N linear combinations that generate $\phi_1[n], \ldots, \phi_{N-1}[n]$, and $\psi[n]$ are linearly independent as required by the corollary.

As in the previous analysis, to show that the DAC noise PSD is zero at $\omega = 0$, it is sufficient to show that the partial sum of $\phi_j[n]$ is a bounded sequence. To show this, note that the $N-1$ sequences $\phi_1[n], \ldots, \phi_{N-1}[n]$ detect the edges—*i.e.*, starting and ending points—of the contiguous segment of high or low digital encoder outputs. If $x[n] = \pm N/2$, there are no edges to detect and $\phi_j[n] = 0$ for each j. However, if $x[n] \neq \pm N/2$, $\phi_j[n]$ is nonzero only when j corresponds to an edge of the contiguous segment:

$$\phi_j[n] = \begin{cases} -1. & \text{if } j = z[n]: \\ 1. & \text{if } j = z[n+1]: \\ 0. & \text{otherwise.} \end{cases} \tag{28}$$

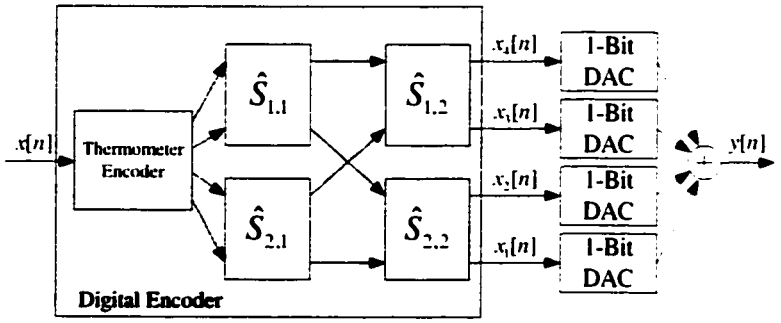


Figure 1.4: The butterfly shuffler DAC.

This implies that the nonzero samples of $\phi_j[n]$ alternate between -1 and 1, and the partial sum of $\phi_j[n]$ is a bounded sequence:

$$\left| \sum_{m=0}^{n} \phi_j[m] \right| \leq 1. \tag{29}$$

Therefore, the DAC noise PSD is also zero at $\omega = 0$.

BUTTERFLY SHUFFLER

An example 5-level butterfly shuffler DAC is shown in Figure 1.4 [5]. Like the previously analyzed DACs, its input range is $\{-N/2, -N/2+1, \ldots, N/2\}$, and all of its 1-bit DACs have the same nominal step size. Unlike the previously analyzed DACs, the butterfly shuffler DAC requires that N be a power of 2; *i.e.*, $N = 2^b$, where b is a positive integer. The digital encoder consists of a thermometer encoder and N *swapper cells*, which are labeled $\hat{S}_{l,m}$ and positioned in a matrix with $l = 1, \ldots, 2^{b-1}$, and $m = 1, \ldots, b$, corresponding to the row and column numbers, respectively. The input and output sequences of each swapper cell are 1-bit sequences; the values of each are taken to be 1/2 and -1/2 at sample times when the sequence is high and low, respectively. At each sample time, n, each swapper cell determines its outputs by routing its inputs either straight through or swapped. The thermometer encoder, whose operation is described in the previous sub-section, is not a necessary component as it can be replaced by any encoder that

has N 1-bit outputs and ensures that exactly $x[n] + N/2$ of its outputs are high at each sample time, n.

Let $\hat{x}_{2l-1,m}[n]$ and $\hat{x}_{2l,m}[n]$ denote the top and bottom inputs of $\hat{S}_{l,m}$, respectively. Using $\hat{S}_{1,1}$ in Figure 1.4 as an example,

$$\hat{x}_{1,2}[n] = \frac{1}{2}\left(\hat{x}_{1,1}[n] + \hat{x}_{2,1}[n] + \hat{s}_{1,1}[n]\right). \tag{30}$$

and

$$\hat{x}_{3,2}[n] = \frac{1}{2}\left(\hat{x}_{1,1}[n] + \hat{x}_{2,1}[n] - \hat{s}_{1,1}[n]\right). \tag{31}$$

where $\hat{s}_{1,1}[n]$ is called a *swapper sequence*. It is generated within $\hat{S}_{1,1}$ and is restricted to the values $\{-1, 0, 1\}$. Thus, each swapper cell $\hat{S}_{l,m}$ uses its swapper sequence, $\hat{s}_{l,m}[n]$, as in (30) and (31) to determine its outputs. In the first-order butterfly shuffler DAC, each swapper cell alternates between swapping and not swapping so that

$$\left|\sum_{k=0}^{n} \hat{s}_{l,m}[k]\right| \leq 1. \tag{32}$$

which, as shown in the Appendix, implies that the PSD of each swapper sequence is zero at $\omega = 0$.

Generalizing (30) and (31) to the other swapper cells in Figure 1.4, the top digital encoder output in the figure can be written as

$$x_4[n] = \frac{1}{4}\left(\sum_{k=1}^{4} \hat{x}_{k,1}[n] + \hat{s}_{1,1}[n] + \hat{s}_{2,1}[n]\right) + \frac{1}{2}\hat{s}_{1,2}[n]. \tag{33}$$

Since $x[n] + N/2$ of the thermometer encoder outputs are high at time n, it follows that $\sum_{k=1}^{4} \hat{x}_{k,1}[n] = x[n]$. This and (33) imply

$$x_4[n] = \frac{1}{4}\left(x[n] + \hat{s}_{1,1}[n] + \hat{s}_{2,1}[n]\right) + \frac{1}{2}\hat{s}_{1,2}[n]. \tag{34}$$

Therefore, the top digital encoder output is a linear combination of $x[n]$ and the swapper sequences. It follows by similar reasoning that this holds for every digital

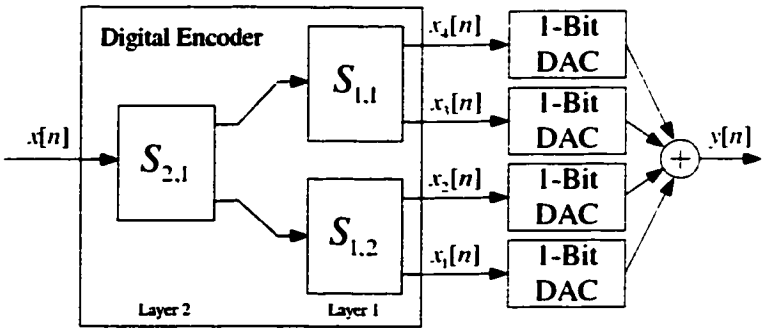


Figure 1.5: The tree structured DAC.

encoder output, and in general with $N = 2^b$.

$$x_i[n] = x[n]/N + \sum_{l=1}^{2^{b-1}} \sum_{m=1}^{b} c_{l,m}^{(i)} \cdot s_{l,m}[n]. \tag{35}$$

where each $c_{l,m}^{(i)}$ is a constant that is either $\pm 1/2^{b-m+1}$ or 0.

To apply Corollary 1, once again let $\phi_j[n] \equiv x_{j+1}[n] - x_j[n]$ for $j = 1, \ldots, N-1$. As previously shown, because all the 1-bit DACs have the same nominal step size, these $N-1$ sequences satisfy the linear independence condition of the corollary. It follows from (32) and (35) that, for each j, $\phi_j[n]$ is a linear combination of swapper sequences whose PSDs are zero at $\omega = 0$, which, as shown in the Appendix, implies that the PSD of $\phi_j[n]$ is zero at $\omega = 0$. Therefore, the DAC noise PSD is also zero at $\omega = 0$.

TREE STRUCTURED

An example 5-level tree structured DAC is shown in Figure 1.5 [6]. Like the previously analyzed DACs, its input range is $\{-N/2, -N/2+1, \ldots, N/2\}$, and all of its 1-bit DACs have the same nominal step size. This DAC requires that N be a power of two. The digital encoder consists of *switching blocks*, which are labeled $S_{k,r}$, where $k = 1, \ldots, b$, denotes the layer number, and $r = 1, \ldots, 2^{b-k}$, denotes the depth in the layer. If the input to $S_{k,r}$ is denoted $x_{k,r}[n]$ and each sequence $x_i[n]$ is

also denoted $x_{0,i}[n]$. the switching blocks are interconnected such that the top and bottom outputs of $S_{k,r}$ are $x_{k-1,2r-1}[n]$ and $x_{k-1,2r}[n]$. respectively. The outputs of $S_{k,r}$ are given by

$$x_{k-1,2r-1}[n] = \tfrac{1}{2}\left(x_{k,r}[n] + s_{k,r}[n]\right). \tag{36}$$

and

$$x_{k-1,2r}[n] = \tfrac{1}{2}\left(x_{k,r}[n] - s_{k,r}[n]\right). \tag{37}$$

where $s_{k,r}[n]$ is called the *switching sequence* and is generated within $S_{k,r}$.

Analogously to the butterfly shuffler DAC. the switching blocks in the first-order tree structured DAC ensure that

$$\left|\sum_{m=0}^{n} s_{k,r}[m]\right| \leq 1. \tag{38}$$

which. as shown in the Appendix. implies that the PSD of $s_{k,r}[n]$ is zero at $\omega = 0$. By recursively solving the switching block outputs in (36) and (37) as functions of the switching sequences and the DAC input $x[n]$. it follows that

$$x_i[n] = x[n]/N + \sum_{k=1}^{b}\sum_{r=1}^{2^{b-k}} d_{k,r}^{(i)} \cdot s_{k,r}[n]. \tag{39}$$

where each $d_{k,r}^{(i)}$ is a constant that is either $\pm 1/2^k$ or 0.

Once again. Corollary 1 can be applied by using the $N-1$ sequences $\phi_j[n] \equiv x_{j+1}[n] - x_j[n]$ for $j = 1, \ldots, N-1$. As previously shown. the $N-1$ sequences satisfy the linear independence condition in the corollary. The PSD of each $\phi_j[n]$ sequence is zero at $\omega = 0$ because. from (39). each sequence results from a linear combination of switching sequences whose PSDs are zero at $\omega = 0$. Therefore. the DAC noise PSD is also zero at $\omega = 0$.

QUALITATIVE COMPARISONS

Comparisons among mismatch-shaping DACs can be made using the necessary and sufficient condition presented in the theorem. One comparison can be made

concerning how easily each of the four previously analyzed DACs combat harmonic distortion in its DAC noise. In the butterfly shuffler and tree structured DACs, the DAC noise is a linear combination of shaped sequences—*i.e.*, swapper and switching sequences—that are generated within their digital encoders. Therefore, as shown in the Appendix, if these shaped sequences have bounded PSDs, then their DAC noise PSDs are also bounded and thus do not contain spurious tones. This can be accomplished by incorporating randomness in the shaped sequences to prevent any tonal behavior. The relative ease for which this is accomplished is shown in [8] where pseudorandom sequences are employed by the switching blocks in both first- and second-order, lowpass tree structured DACs to eliminate harmonic distortion in the DAC noise.

However, the vector feedback and DWA DACs obtain DAC noise with the given set of PSD properties without explicitly generating sequences with these properties. This indirect approach for spectrally shaping the DAC noise makes it more difficult to eliminate or reduce spurious tones. To remove spurious tones in the vector feedback DAC, randomness must somehow be incorporated into the vector quantizer's operation, but, to the knowledge of the authors, no such vector quantizer has been demonstrated to date. To remove or reduce spurious tones in the DWA DAC, its architecture must be changed. Most variants of the DWA DAC are designed to reduce, relative to the DWA DAC, the harmonic distortion in the DAC noise. Examples of such DWA variants are presented in [7],[9], and [10]. To successfully reduce harmonic distortion, each of these published first-order architectures requires that the multi-bit DAC input includes a random component—*e.g.*, the quantization noise from a $\Delta\Sigma$ modulator. This is not required in the previously mentioned first-order, tree structured DAC whose DAC noise PSD is bounded regardless of

the DAC input [11].

Another comparison can be made concerning the ease for which a mismatch-shaping DAC obtains *higher-order*—*i.e.*, greater than first-order—spectral shaping of the DAC noise. Such DACs are desirable because the DAC noise in a higher-order DAC usually has less signal-band power. This comparison does not include DWA because it is inherently a first-order DAC. The theorem states that, given the DAC input, $N-1$ sequences are required to generate the digital encoder outputs in a mismatch-shaping DAC. However, with $N=2^b$, where b is a positive integer, the butterfly shuffler DAC requires $b \cdot N/2$ swapper sequences, which, for $b>1$, are more than necessary as $b \cdot N/2 > N-1$. Additionally, as b increases, the number of extra sequences utilized by the DAC grows at a faster rate than an exponential function. Each swapper sequence depends on its swapper cell input, which depends on the DAC input. This dependence and the extra swapper sequences makes it difficult to ensure that each swapper sequence has the desired PSD properties in higher-order implementations.

For example, to implement a second-order, lowpass butterfly shuffler DAC, it follows from [8] that each swapper sequence must satisfy the following:

$$\left| \sum_{j=0}^{n} \sum_{k=0}^{j} \hat{s}_{l,m}[k] \right| \leq B. \tag{40}$$

where B is a constant. Because the value of each swapper cell output is either -1/2 or 1/2 at each sample time, n, it follows that

$$\hat{s}_{l,m}[n] = \begin{cases} \pm 1. & \text{if } \hat{x}_{2l-1,m}[n] \neq \hat{x}_{2l,m}[n]: \\ 0. & \text{if } \hat{x}_{2l-1,m}[n] = \hat{x}_{2l,m}[n]. \end{cases} \tag{41}$$

Therefore, if the N inputs to the column-one swapper cells are thermometer encoded as in Figure 1.4, then the column-one swapper sequences are restricted as follows:

$$\hat{s}_{l,1}[n] = \begin{cases} \pm 1. & \text{if } x[n] = N/2 - (2l-1): \\ 0. & \text{otherwise.} \end{cases} \tag{42}$$

At each sample time, at most one of the 2^{b-1} swapper sequences in the first column is nonzero; the choice of which is determined by the DAC input. As b increases, this dependence on the DAC input makes it more difficult for these swapper sequences to satisfy (40) and has prohibited the implementation of the second-order, lowpass butterfly shuffler DAC.

However, the vector feedback and tree structured DACs process N and $N-1$ internal sequences, respectively, to generate their digital encoder outputs. Because, for $b > 2$, these DACs process fewer internal sequences than the butterfly shuffler DAC, their internal sequences and DAC noise have less dependence on the DAC input, which enables the implementation of higher-order DACs. For example, in the tree structured DAC, the layer that directly processes the DAC input, layer b, only has one switching block as opposed to the 2^{b-1} swapper cells in the first column of the butterfly shuffler DAC. For the switching blocks presented in [8], the switching sequence is restricted as follows:

$$s_{k,r}[n] = \begin{cases} \pm 1, & \text{if } x_{k,r}[n] + 2^{k-1} \text{ is odd;} \\ 0, & \text{if } x_{k,r}[n] + 2^{k-1} \text{ is even.} \end{cases} \tag{43}$$

Therefore, the switching sequence in layer b depends only on the parity of the DAC input, which is much less restrictive than the dependence exhibited by the column-one swapper sequences shown in (42). Examples of second-order lowpass implementations of the vector-feedback and tree structured DACs are presented in [12] and [13], respectively.

PARTITIONED DWA

The partitioned DWA (P-DWA) DAC, shown in Figure 1.6, was designed to not only suppress the DAC noise power near $\omega = 0$, but to reduce, in comparison to the DWA DAC, the spurious tones in the DAC noise [7]. Its input range is

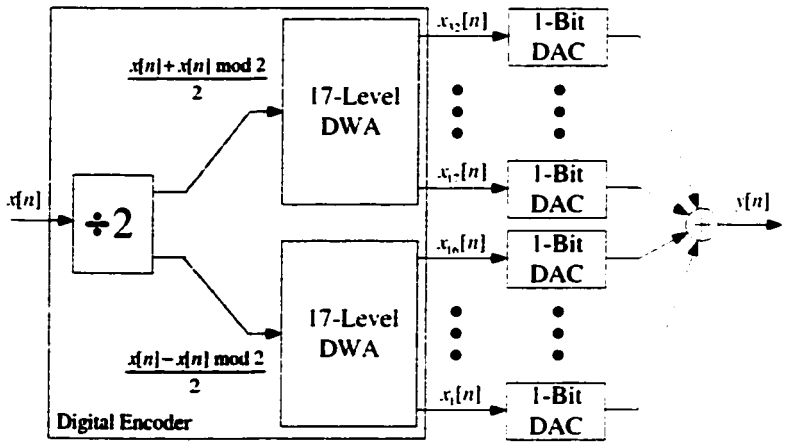


Figure 1.6: The partitioned DWA DAC.

$\{-16, -15, \ldots, 16\}$. All of its 1-bit DACs have the same nominal step size. The digital encoder consists of two 17-level DWA digital encoders and a *divide-by-two* block. The top output of the divide-by-two block is $x[n]/2$ rounded up to the nearest integer (*i.e.*, $\lceil x[n]/2 \rceil$), and the bottom output is $x[n]/2$ rounded down to the nearest integer (*i.e.*, $\lfloor x[n]/2 \rfloor$).

Corollary 2 is applied next to show that the DAC noise PSD is *not* zero at $\omega = 0$. Since the difference between the outputs of the divide-by-two block is one when $x[n]$ is odd and zero otherwise, it follows that

$$\sum_{i=17}^{32} x_i[n] - \sum_{j=1}^{16} x_j[n] = x[n] \bmod 2. \tag{44}$$

By Corollary 2, if the above linear combination *cannot* be written as $ax[n]+b+\varepsilon[n]$, where a and b are constants and the PSD of $\varepsilon[n]$ is zero at $\omega = 0$, then the DAC noise PSD is *not* zero at $\omega = 0$. Therefore, from (44), it is sufficient to show that, for some $x[n]$, the PSD of the sequence

$$\varepsilon[n] \equiv (x[n] \bmod 2) - (ax[n] + b)\,. \tag{45}$$

is not zero at $\omega = 0$ for any choice of the constants a and b. Since

$$x[n] = 2\left\lfloor \frac{x[n]}{2} \right\rfloor + (x[n] \bmod 2)\,. \tag{46}$$

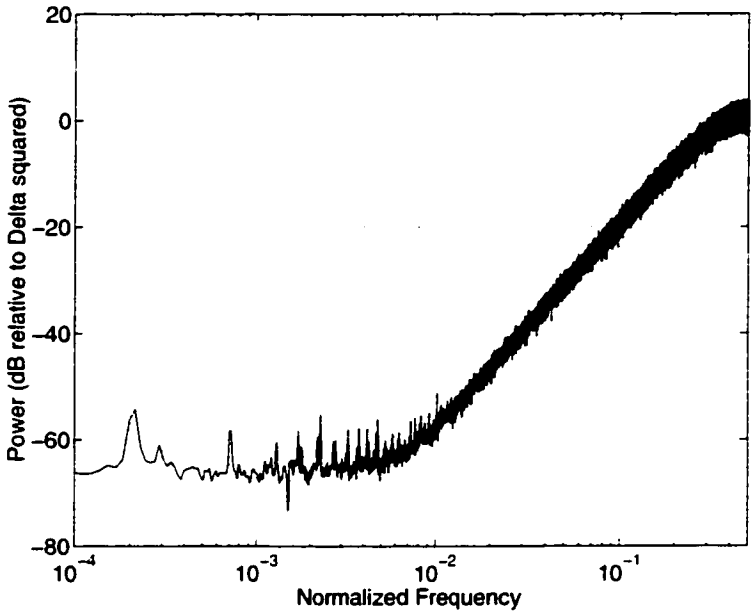


Figure 1.7: Output noise PSD from a simulation of a 2nd-order, analog $\Delta\Sigma$ modulator using the partitioned DWA DAC.

(45) can be written as

$$\varepsilon[n] = (1-a)\,(x[n] \bmod 2) - \left(2a\left\lfloor \frac{x[n]}{2} \right\rfloor + b\right). \tag{47}$$

The DAC input, $x[n]$, can be chosen so that the sequences $2\,(x[n] \bmod 2) - 1$ and $\lfloor x[n]/2 \rfloor$ are white and uncorrelated. For this $x[n]$, (47) implies that the PSD of $\varepsilon[n]$ is not zero at $\omega = 0$ for any selection of the constants a and b. Therefore, by Corollary 2, the DAC noise PSD is also not zero at $\omega = 0$.

Figure 1.7 displays the output noise PSD from a behavioral simulation of a second-order, analog $\Delta\Sigma$ modulator that employs the P-DWA DAC. The $\Delta\Sigma$ modulator input was a -1dB (relative to full scale) sinusoid with frequency $0.0015 f_s$, where f_s is the sample rate. The PSD units are dB relative to Δ^2, where Δ is the step size of the analog-to-digital converter within the $\Delta\Sigma$ modulator. The frequency axis is normalized with respect to the sample rate. The 1-bit DAC errors were chosen as independent Gaussian random variables with a standard deviation of 1% of the 1-bit DAC's nominal step size.

The output noise in the simulation includes the DAC noise and quantization noise. The simulation shows that, as a result of the DAC noise, the output noise PSD is not zero at $\omega = 0$. However, the simulation suggests that, compared to conventional DWA, the DAC noise in this implementation has less harmonic distortion. The reduced harmonic distortion is a result of the randomness in $x[n]$, which causes $(x[n] \bmod 2)/2$ to act as an additive and subtractive dither sequence that, as shown in Figure 1.6, is fed into top and bottom DWA DACs, respectively.

V. CONCLUSION

Necessary and sufficient conditions for mismatch shaping with a general multi-bit DAC have been presented, proved, and discussed. For the DAC noise to have certain PSD properties, the conditions show that there must be $N - 1$ underlying sequences in the general multi-bit DAC that, when linearly combined, produce a sequence that consists of an offset, a scaled version of the multi-bit DAC input, and another sequence that has the given PSD properties. As example applications, the conditions have been used to show that the DAC noise PSDs of four widely-used lowpass DACs are zero at $\omega = 0$ and that the DAC noise PSD of another lowpass DAC is not zero at $\omega = 0$. Additionally, the theory has been used to compare the case for which several DACs combat spurious tones in their DAC noise and obtain higher-order shaped DAC noise.

APPENDIX

Two lemmas are presented below that supplement the analyses in Section IV. The first lemma proves that if a sequence $\gamma[n]$ has a partial sum that is a bounded sequence, then the PSD of $\gamma[n]$ is zero at $\omega = 0$. The second lemma proves an inequality for PSDs that is used to show that an arbitrary linear combination of

sequences whose PSDs are zero or bounded at a given frequency gives rise to a sequence whose PSD is also zero or bounded, respectively, at that frequency. It is assumed throughout that the PSDs exist for all sequences considered.

Lemma 1: Let $\gamma[n]$ be a sequence whose partial sum is bounded in magnitude by a constant B for all n: *i.e.*,

$$\left| \sum_{m=0}^{n} \gamma[m] \right| \leq B, \tag{48}$$

for all n. Then, the PSD of $\gamma[n]$ (if it exists) is zero at $\omega = 0$.

Proof: As proved in [14], the PSD of $\gamma[n]$ is given by

$$S_{\gamma,\gamma}\left(e^{j\omega}\right) = \lim_{M\to\infty} \frac{1}{M} E\left\{ \left| \Gamma_M\left(e^{j\omega}\right) \right|^2 \right\}. \tag{49}$$

where $E\{\cdot\}$ is the expectation operator, and $\Gamma_M\left(e^{j\omega}\right)$ is the M-point Fourier transform of $\gamma[n]$:

$$\Gamma_M\left(e^{j\omega}\right) = \sum_{n=0}^{M-1} \gamma[n] e^{-j\omega n}. \tag{50}$$

Evaluating the PSD at $\omega = 0$ gives

$$S_{\gamma,\gamma}\left(e^{j0}\right) = \lim_{M\to\infty} \frac{1}{M} E\left\{ \left| \sum_{n=0}^{M-1} \gamma_i[n] \right|^2 \right\}. \tag{51}$$

However, from (48), the partial sum of $\gamma[n]$ in the above expression is bounded in magnitude by B: therefore,

$$S_{\gamma,\gamma}\left(e^{j0}\right) \leq \lim_{M\to\infty} \frac{B^2}{M} = 0. \tag{52}$$

Because $S_{\gamma,\gamma}\left(e^{j\omega}\right)$ is nonnegative for all ω, (52) implies that $S_{\gamma,\gamma}\left(e^{j0}\right) = 0$.

■

Lemma 2: If $S_{x,x}\left(e^{j\omega}\right)$ and $S_{y,y}\left(e^{j\omega}\right)$ are the PSDs of $x[n]$ and $y[n]$, respectively, and $z[n] = x[n] + y[n]$, then

$$S_{z,z}\left(e^{j\omega}\right) \leq 2\left(S_{x,x}\left(e^{j\omega}\right) + S_{y,y}\left(e^{j\omega}\right)\right). \tag{53}$$

where $S_{z,z}\left(e^{j\omega}\right)$ is the PSD of $z[n]$.

Proof: Let $X_M\left(e^{j\omega}\right)$, $Y_M\left(e^{j\omega}\right)$, and $Z_M\left(e^{j\omega}\right)$ be the M-point Fourier transforms of $x[n]$, $y[n]$, and $z[n]$, respectively—*i.e.*,

$$X_M\left(e^{j\omega}\right) = \sum_{n=0}^{M-1} x[n]e^{-j\omega n}. \tag{54}$$

and likewise for the Fourier transforms of $y[n]$ and $z[n]$. The Cauchy-Schwartz inequality implies that

$$|a+b|^2 \leq 2\left(|a|^2 + |b|^2\right). \tag{55}$$

where a and b are complex numbers. Therefore, it follows from the linearity of the Fourier Transform that, for every ω,

$$\left|Z_M\left(e^{j\omega}\right)\right|^2 \leq 2\left(\left|X_M\left(e^{j\omega}\right)\right|^2 + \left|Y_M\left(e^{j\omega}\right)\right|^2\right). \tag{56}$$

As shown in [14],

$$S_{z,z}\left(e^{j\omega}\right) = \lim_{M\to\infty} \frac{1}{M} E\left\{\left|Z_M\left(e^{j\omega}\right)\right|^2\right\}. \tag{57}$$

and likewise for PSDs of $x[n]$ and $y[n]$, where $E\{\cdot\}$ is the expectation operator. Therefore, (56), (57), and the linearity of the expectation operator imply (53).

■

Therefore, it follows from (53) that if, at some frequency ω_o, $S_{x,x}\left(e^{j\omega_o}\right) = 0$, and $S_{y,y}\left(e^{j\omega_o}\right) = 0$, then $S_{z,z}\left(e^{j\omega_o}\right) = 0$ because the PSD is always nonnegative. Thus, the sum of two sequences whose PSDs are zero at some frequency gives rise to a sequence whose PSD is also zero at that frequency. Additionally, if the PSDs of $x[n]$ and $y[n]$ are bounded functions—there exists a constant B such that $S_{x,x}\left(e^{j\omega}\right) \leq B$, and $S_{y,y}\left(e^{j\omega}\right) \leq B$ for all ω—then (53) implies that the PSD of $z[n]$ is also a bounded function: $S_{z,z}\left(e^{j\omega}\right) \leq 4B$. Therefore, by mathematical induction, any linear combination of sequences whose PSDs are zero or bounded at

a given frequency give rise to another sequence whose PSD is also zero or bounded, respectively, at that frequency.

CHAPTER ACKNOWLEDGMENT

The text of Chapter 1 consists of material that has been submitted for publication as a Regular Paper in the *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*. The dissertation author was the primary researcher. Ian Galton supervised the research which forms the basis of the chapter.

REFERENCES

1. L. Hernández. "A model of mismatch-shaping D/A conversion for linearized DAC architectures." *IEEEc Trans. on Circuits and Systems—I: Fundamental Theory and Applications*. vol. 45. no. 10. pp. 1068-1076. Oct. 1998.

2. R. Schreier. B. Zhang. "Noise-shaped multi-bit D/A converter employing unit elements." *Electronics Letters*. vol. 31. no. 20. pp. 1712-1713. Sept. 28. 1995.

3. M. J. Story. "Digital to analogue converter adapted to select input sources based on a preselected algorithm once per cycle of a sampling signal." U.S. Patent No. 5.138.317. Aug. 11. 1992.

4. R. T. Baird. T. S. Fiez. "Linearity enhancement of multi-bit $\Delta\Sigma$ A/D and D/A converters using data weighted averaging." *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*. vol. 42. no. 12. pp. 753-762. Dec. 1995.

5. R. W. Adams. T. W. Kwan. "Data-directed scrambler for multi-bit noise shaping D/A converters." U.S. Patent No. 5.404.142. Apr. 4. 1995.

6. I. Galton. "Spectral shaping of circuit errors in digital-to-analog converters." *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*. vol. 44. no. 10. pp. 808-817. Oct. 1997.

7. K. Vleugels. S. Rabii. B.A. Wooley. "A 2.5 V sigma-delta modulator for broadband communications applications." *IEEE Journal of Solid-State Circuits*. vol. 36. no. 12. pp. 1887-1899. Dec. 2001.

8. J. Welz. I. Galton. E. Fogleman. "Simplified logic for first-order and second-order

mismatch-shaping digital-to-analog converters." *IEEE Transaction on Circuits and Systems—II: Analog and Digital Signal Processing.* vol. 48. no. 11. Nov. 2001.

9. I. Fujimori. L. Longo. A. Hairapetian. K. Seiyama. S. Kosic. J. Cao. S. Chan. "A 90dB SNR. 2.5 MHz output-rate ADC using cascaded multibit delta-sigma modulation at 8x oversampling ratio." *IEEE Journal of Solid-State Circuits.* vol. 35. no. 12. pp. 1820-1828. Dec. 2000.

10. R. Radke. A. Eshraghi. T. Fiez. "A spurious-free delta-sigma DAC using rotated data weighted averaging." *Proceedings of the 1999 IEEE Custom Integrated Circuits Conference.* pp. 125-128. May. 1999.

11. J. Welz. I. Galton. "The mismatch-noise PSD from a tree-structured DAC in a second-order delta-sigma modulator with a midscale input." *Proceedings of the IEEE International Conference on Acoustics. Speech. and Signal Processing.* vol. 4. May 7-11. 2001.

12. A. Yasuda. H. Tanimoto. T. Iida. "A third-order $\Delta\Sigma$ modulator using second-order noise-shaping dynamic element matching." *IEEE J. Solid-State Circuits.* vol. 33. no. 12. pp. 1879-1886. Dec. 1998.

13. E. Fogleman. J. Welz. I. Galton. "An audio ADC delta-sigma modulator with 100dB SINAD and 102dB DR using a second-order mismatch-shaping DAC." *IEEE Journal of Solid State Circuits.* vol. 36. no. 3. pp. 339-48. March 2001.

14. S. Haykin. *Adaptive Filter Theory.* Prentice Hall. New Jersey. 1996.

Chapter 2

Simplified Logic for First-Order and Second-Order Mismatch-Shaping Digital-to-Analog Converters

Jared Welz, Ian Galton, Eric Fogleman

Abstract—Mismatch-shaping DACs have become widely used in high-performance delta-sigma data converters because they facilitate delta-sigma modulators with multibit quantization. Relative to single-bit quantization, multibit quantization significantly relaxes the analog circuit performance necessary to achieve a given level of data converter precision, but significant digital logic is required to perform the mismatch shaping. In modern VLSI processes optimized for digital circuitry, this tends to be a good tradeoff in terms of both area and power consumption. It is nonetheless desirable to minimize the digital complexity as much as possible. Moreover, in delta-sigma ADCs the mismatch-shaping logic is in the feedback path of the delta-sigma modulator, so it is essential to maintain a sufficiently small propagation delay through the mismatch-shaping logic. This paper presents and analyzes several variations of the switching blocks within a tree-structured mismatch-shaping DAC that result in the most hardware-efficient first-order and second-order mismatch-shaping DAC implementations yet known to the authors. The variations presented allow designers to trade off complexity for propagation-delay reduction so as to tailor designs to specific applications.

I. INTRODUCTION

IN $\Delta\Sigma$ data converters, both $\Delta\Sigma$ analog-to-digital converters (ADCs) and $\Delta\Sigma$ digital-to-analog converters (DACs), coarse quantization is used in conjunction

with quantization-noise shaping and filtering to achieve high-precision data conversion. In both cases, coarse DACs are required. Unlike the error introduced by the coarse quantization, the error introduced by at least one of the coarse DACs in a $\Delta\Sigma$ data converter is not attenuated inside the data converter's signal band. In switched-capacitor implementations, most of the DAC error arises from static capacitor mismatches, which give rise to step-size mismatches in the multibit DACs. The resulting step-size mismatches are memoryless functions of the DAC's input, so the DAC can be viewed as an ideal DAC followed by a memoryless nonlinear function. The nonlinearity tends to fold out-of-band quantization noise into the signal band thereby limiting the overall accuracy of the data converter.

To avoid this problem, many $\Delta\Sigma$ data converters employ 1-bit quantization. With 1-bit quantization, the coarse DAC is implemented by a 1-bit DAC. Since a 1-bit DAC only generates two levels, it only has one step, and so it is inherently linear. However, with 1-bit quantization in the $\Delta\Sigma$ modulator, quantization-noise shaping must be limited to maintain the $\Delta\Sigma$ modulator's stability. Additionally, the power of the quantization noise in the 1-bit $\Delta\Sigma$ modulator exceeds that of its input, so $\Delta\Sigma$ data converters with 1-bit quantization are extremely sensitive to any nonlinearity or timing error, such as op-amp slewing or clock jitter, which can fold this quantization noise into the signal band.

To avoid these problems, multibit *mismatch-shaping* DACs have been developed [1]-[52]. In these DACs, digital logic is used to scramble the DAC capacitor or current-source connections in such a fashion that the error introduced by the device mismatches, referred to as *DAC noise*, is suppressed within the data converter's signal band. For lowpass mismatch-shaping DACs, the DAC noise is suppressed near dc so that its power spectral density (PSD) is shaped like the magnitude response of

a first-order, or in some cases, second-order highpass filter. The five main classes of mismatch-shaping DACs include individual-level averaging (ILA) [11]-[12], vector feedback [13]-[16], data-weighted averaging (DWA) [17]-[31], butterfly shuffler [32]-[37], and tree-structured [38]-[44]. The criteria used to compare these DACs include complexity, propagation delay, spurious-tone avoidance, and the order, or degree, of DAC noise suppression.

In [40], a tree-structured mismatch-shaping DAC is introduced that has led to the most efficient implementations of dithered first- and second-order mismatch-shaping DACs known to the authors [43], [44]. Moreover, the first-order tree-structured DAC is the only one for which dither is known to completely eliminate spurious tones in its DAC noise [46]. This paper furthers the development of this DAC by presenting new implementations of its digital logic that are more hardware efficient and have less propagation delay than those presented in [40]. The digital logic is first partitioned into functional blocks, one of which determines the shape of the DAC noise's PSD and another that is responsible for the digital logic's propagation delay. The hardware for the digital logic is presented through interchangeable variations of these functional blocks so that the DAC can be tailored to meet varying specifications for signal-band DAC noise power, propagation delay, and complexity. Efficient first and second-order mismatch-shaping logic are presented and the resulting DAC noise from each is analyzed to show it has the desired spectral shape. Additionally, medium-speed and high-speed implementations of the DAC are presented that offer a tradeoff between propagation delay and complexity.

This paper is divided into six sections. Section II reviews the tree-structured DAC and presents the functional partitioning of its digital logic. Additionally, this section presents an example application of a 5-bit, second-order ADC $\Delta\Sigma$ modulator

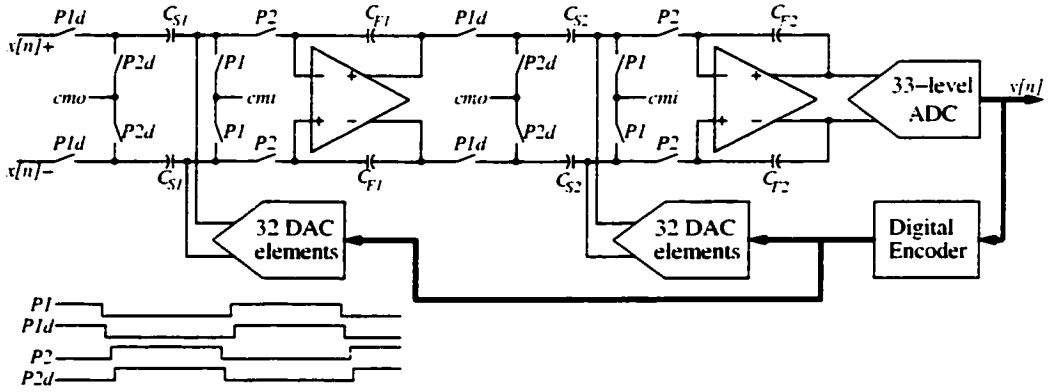


Figure 2.1: An example second-order, 33-level, lowpass analog $\Delta\Sigma$ modulator realized with switched capacitors.

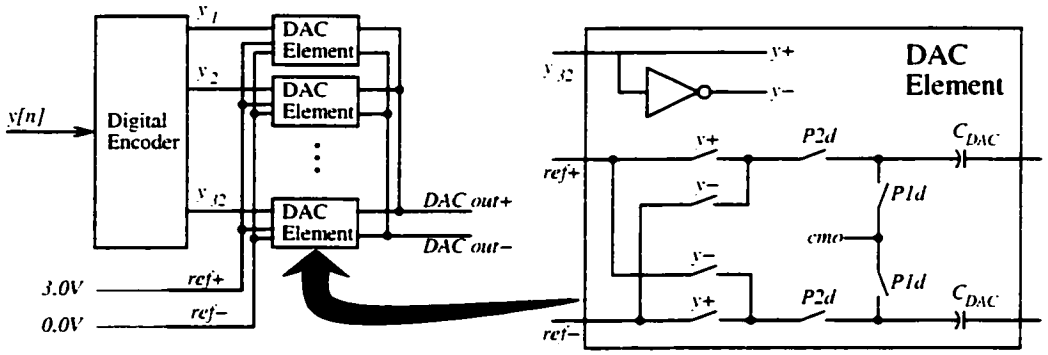


Figure 2.2: A 33-level mismatch-shaping DAC with switched capacitor DAC elements.

that is used throughout the paper to illustrate the DAC's performance and complexity. Section III presents and analyzes the first-order and second-order mismatch-shaping logic, while Section IV presents the medium-speed and high-speed implementations of the DAC. Section V presents a hardware comparison between the different tree-structured DAC implementations and other mismatch-shaping DACs presented in literature.

II. THE TREE-STRUCTURED DAC

THE $\Delta\Sigma$ MODULATOR APPLICATION

The 5-bit ADC $\Delta\Sigma$ modulator presented in [43] is shown in Figure 2.1. It consists of two delayed switched-capacitor integrators, a 33-level flash ADC, and

two 33-level DACs. As shown in Figures 2.1 and 2.2, each 33-level DAC consists of a bank of 32 *DAC elements* and a shared digital encoder whose outputs, $y_i[n]$ $(i = 1, \ldots, 32)$, are 1-bit sequences. Each DAC element can be viewed as a 1-bit DAC whose analog output is a charge packet applied to the summing node of an integrator. A DAC element is said to be "selected high" when its input is high; otherwise it is said to be "selected low". For convenience, the output of the ADC, $y[n]$, is interpreted as an integer between 0 and 32. For each ADC output sample, the digital encoder chooses which $y[n]$ of the DAC elements to select high and which $(32 - y[n])$ of the DAC elements to select low. In other words, if $y_i[n]$ is interpreted numerically as one when high and zero when low, the DAC encoder ensures that $y[n] = y_1[n] + \cdots + y_{32}[n]$.

Mismatches among the capacitor values of the DAC elements cause the output of each multibit DAC to be a nonlinear function of its input. The resulting nonlinear error is represented, without approximation, as an additive noise source referred to as *DAC noise*. As shown in Figure 2.1, an output from one of the DACs is added to the $\Delta\Sigma$ modulator's input. Thus, the $\Delta\Sigma$ modulator does not attenuate any of the signal-band noise power from this DAC. However, the digital encoder can select the DAC elements such that most of the DAC noise power resides outside of the signal band.

To demonstrate the improvements that are realized by mismatch shaping, the DAC presented in [44] was tested with and without the mismatch shaping. The input for each test was a 1.5kHz, -1dB (relative to full scale) sinusoid. With mismatch shaping, the resultant signal-to-noise-and-distortion ratio (SINAD) was 100dB, whereas without mismatch shaping, the resultant SINAD was 64dB. In general, the tradeoff for the improved performance is the additional hardware and

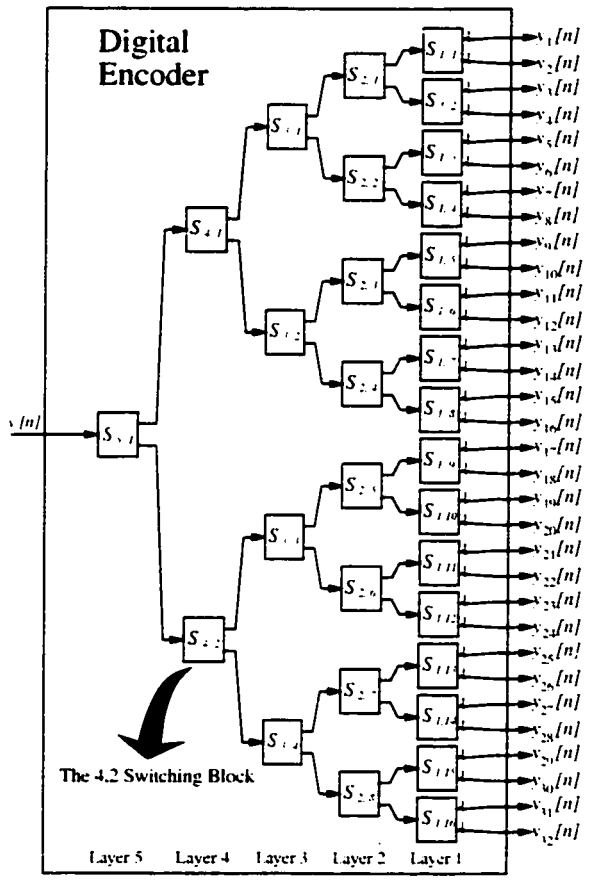


Figure 2.3: The 33-level tree-structured digital encoder.

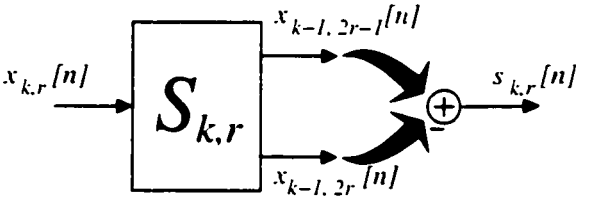


Figure 2.4: The switching block $S_{k,r}$.

propagation delay incurred by the digital encoder. However, the propagation delay of the digital encoder only affects the design of high-speed $\Delta\Sigma$ data converters. Examples of commercially available data converters that employ mismatch-shaping DACs to a similar advantage are presented in [47]-[52].

THE TREE-STRUCTURED DIGITAL ENCODER

The architecture for a 33-level, tree-structured digital encoder is shown in Figure

2.3. The nodes of this digital encoder are called *switching blocks*. Each switching block is labeled $S_{k,r}$, where k and r represent the switching block's layer number and position within the layer, respectively. Each switching block $S_{k,r}$ has a single input, which is denoted $x_{k,r}[n]$, and two outputs. If each digital encoder output sequence $y_i[n]$ is also denoted $x_{0,i}[n]$, then the switching blocks are interconnected such that the top output of $S_{k,r}$ is $x_{k-1,2r-1}[n]$ and the bottom output is $x_{k-1,2r}[n]$. The *switching sequence* $s_{k,r}[n]$ is defined as the difference between the top and bottom output sequences of $S_{k,r}$:

$$s_{k,r}[n] = x_{k-1,2r-1}[n] - x_{k-1,2r}[n]. \tag{1}$$

Figure 2.4 illustrates the input and output sequences of $S_{k,r}$ along with the relationship between its switching sequence and output sequences.

As shown in [40], the DAC noise is a linear combination of the switching sequences. In general, for a DAC of the type shown in Figure 2.3 with 2^b DAC elements, the output can be written as

$$u[n] = \gamma y[n] + \beta + e[n], \tag{2}$$

where

$$e[n] = \sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} \Delta_{k,r} s_{k,r}[n]. \tag{3}$$

and γ, β, and $\Delta_{k,r}$ are constants that are functions of the inevitable, static errors that result from process variations during VLSI circuit fabrication.

Therefore, if the switching sequences are all uncorrelated and share the same characteristics in their PSDs (*e.g.*, first-order highpass shaping), the DAC noise also possesses these characteristics. The problem of shaping the PSD of the DAC noise reduces to the problem of creating switching sequences with the desired spectral

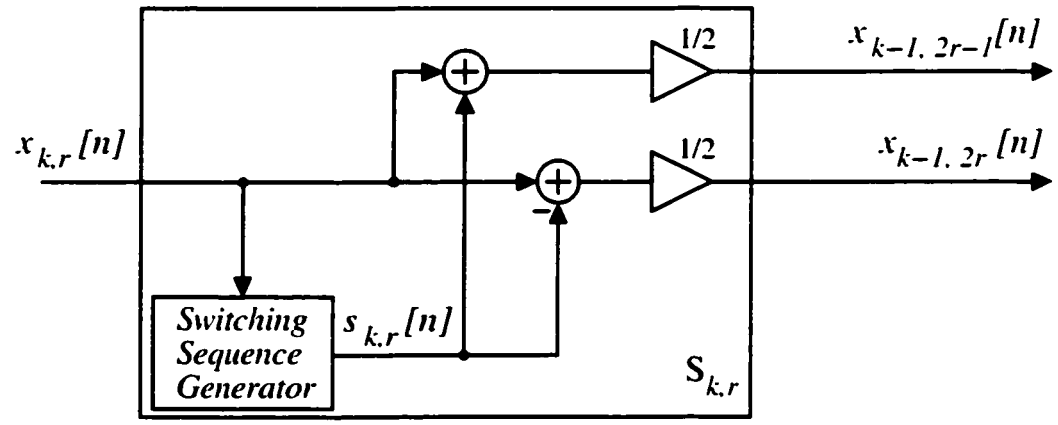


Figure 2.5: The signal processing performed in the switching block.

shaping. Unfortunately, this problem is complicated by the constraints on the switching sequence described next.

CONSTRAINTS ON THE SWITCHING SEQUENCE

The switching sequence is generated within the switching block to obtain the desired spectral properties of the DAC noise. However, the switching sequences must be constrained to satisfy restrictions inherent to the digital encoder. As previously described, each of the digital encoder's outputs, $y_i[n]$ $(i = 1, \ldots, 32)$, is limited to the set $\{0, 1\}$ and their sum must equal the DAC input: $y[n] = y_1[n] + \cdots + y_{32}[n]$. It is shown in [40] that these conditions are met if each switching block satisfies the following two-part *Number Conservation Rule:* the two outputs of each switching block must be in the range $\{0, 1, \cdots, 2^{k-1}\}$ where k is the layer number, and their sum must equal the input to the switching block:

$$x_{k-1,2r-1}[n] + x_{k-1,2r}[n] = x_{k,r}[n]. \tag{4}$$

From (1) and (4), the input/output relationships of switching block $S_{k,r}$ are

$$x_{k-1,2r-1}[n] = \frac{1}{2}(x_{k,r}[n] + s_{k,r}[n]), \qquad \text{and} \qquad x_{k-1,2r}[n] = \frac{1}{2}(x_{k,r}[n] - s_{k,r}[n]). \tag{5}$$

The above expressions are implemented by the block diagram shown in Figure 2.5.

It can be shown that the number conservation rule is satisfied by each switching block $S_{k,r}$ if

$$s_{k,r}[n] = \begin{cases} 0. & \text{if } x_{k,r}[n] \text{ is even:} \\ \pm 1. & \text{if } x_{k,r}[n] \text{ is odd.} \end{cases} \tag{6}$$

This is more restrictive than necessary: however, it significantly simplifies the switching block's hardware. In Figure 2.5 this restriction is reflected by the switching sequence generator's dependence on the input sequence $x_{k,r}[n]$.

IMPLEMENTATION OF THE SWITCHING BLOCK

The switching sequence $s_{k,r}[n]$ is a ternary sequence, and so it can be represented as two single-bit sequences. It follows from (6) that the magnitude of the switching sequence is entirely determined by the input to the switching block, so the switching block can only control the sign of the switching sequence. To separate the magnitude and sign of the switching sequence, let $o_{k,r}[n]$ and $q_{k,r}[n]$ represent $s_{k,r}[n]$ as

$$s_{k,r}[n] = \begin{cases} 0. & \text{if } o_{k,r}[n] = 0; \\ 1. & \text{if } o_{k,r}[n] = 1.\ q_{k,r}[n] = 1; \\ -1. & \text{if } o_{k,r}[n] = 1.\ q_{k,r}[n] = 0; \end{cases} \tag{7}$$

where

$$o_{k,r}[n] = \begin{cases} 1. & \text{if } x_{k,r}[n] \text{ is odd;} \\ 0. & \text{if } x_{k,r}[n] \text{ is even.} \end{cases} \tag{8}$$

The sequence $q_{k,r}[n]$ represents the sign of $s_{k,r}[n]$. It is chosen by the switching block to ensure the switching sequence is appropriately shaped as described in Section III. The $o_{k,r}[n]$ sequence is referred to as the *parity sequence* and represents the magnitude of $s_{k,r}[n]$.

Figure 2.6 displays a convenient functional partitioning of the switching block. The Parity Logic (PL) determines the parity of the switching block's input and generates the parity sequence $o_{k,r}[n]$. The Sequencing Logic (SL) produces the sign sequence $q_{k,r}[n]$ and is responsible for the spectral shaping of the switching sequence. The combination of the sequencing logic and parity logic constitute the

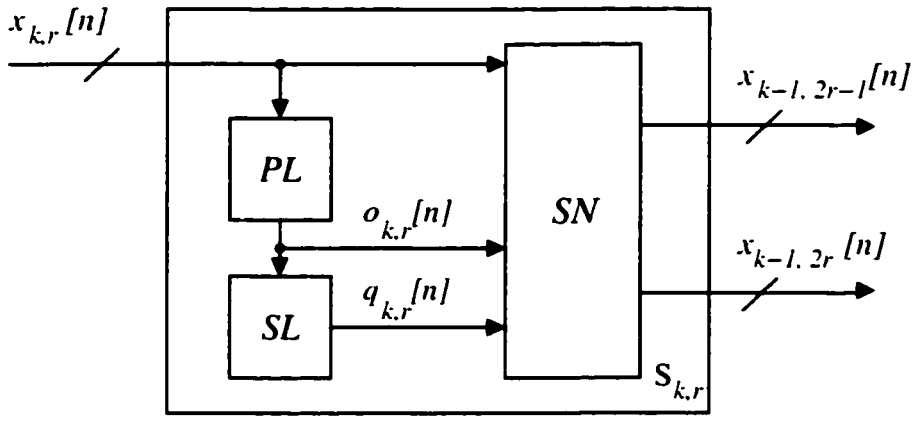


Figure 2.6: A functional partitioning of the switching block where SN is the *Splitting Network*. PL is the *Parity Logic*, and SL is the *Sequencing Logic*.

switching sequence generator shown in Figure 2.5. Given $x_{k,r}[n]$ and the binary representation of $s_{k,r}[n]$ (*i.e.*, $o_{k,r}[n]$ and $q_{k,r}[n]$), the role of the Splitting Network (SN) is to perform the arithmetic operations shown in Figure 2.5 that generate the switching block's two output sequences.

III. LOWPASS SEQUENCING LOGIC

HIGHPASS SWITCHING SEQUENCES

In lowpass mismatch-shaping DACs, the signal band is near dc, so the mismatch-shaping logic is designed such that most of the DAC noise power resides at higher frequencies. In other words, in a lowpass mismatch-shaping DAC, the PSD of the DAC noise resembles the magnitude response of a discrete-time highpass filter. Sequences of this type are called *highpass sequences*. Thus, the sequencing logic blocks in a lowpass tree-structured DAC create highpass switching sequences.

To meaningfully characterize the spectral properties of the highpass switching sequences, a quantitative definition of an Lth-order highpass switching sequence is required. In a $\Delta\Sigma$ modulator with a quantization-noise transfer function that contains zeros only at dc, the order of the $\Delta\Sigma$ modulator corresponds to the number of

dc zeros. Let *quantization noise* denote the component of the $\Delta\Sigma$ modulator output arising from the errors induced by quantization. In an Lth-order lowpass $\Delta\Sigma$ modulator, the quantization noise is commonly called Lth-order highpass noise. A key property of this highpass noise is that it can be processed by L cascaded accumulators such that the values in the accumulators remain bounded. The L dc poles from the accumulators "cancel" the L dc zeros in the noise transfer function. However, if one more accumulator were cascaded, its output would become unbounded regardless of the accumulators' initial values.

In contrast to the quantization noise, the switching sequence, as a result of its constraints in (6), cannot be generated by filtering a causal, bounded sequence by a system with L dc zeros. So the concept of the switching sequence's order is vague without a more applicable definition. By defining the highpass order of a switching sequence using the accumulator property described above, a transfer function is associated with this sequence, and the desired properties of its PSD are implied.

Definition: Let $\alpha_L[n]$ be the "Lth-sum sequence" of $s_{k,r}[n]$:

$$\alpha_L[n] \equiv \overbrace{\sum_{n_1=0}^{n-1}\sum_{n_2=0}^{n_1-1}\cdots\sum_{n_L=0}^{n_{L-1}-1}}^{L\text{ Summations}} s_{k,r}[n_L]. \tag{9}$$

The sequence $s_{k,r}[n]$ is an Lth-order highpass switching sequence if its Lth-sum sequence is a *bounded sequence*—*i.e.*, there exists a number $K < \infty$ such that $|\alpha_L[n]| < K$ for all n—, and its $(L+1)$st-sum sequence is an unbounded sequence.

If $s_{k,r}[n]$ is an Lth-order highpass switching sequence, then it can be shown that the slope of its PSD is $20L$ dB/decade near dc provided the PSD of $\alpha_L[n]$ is continuous and nonzero in a neighborhood of dc. This definition provides a means to create switching sequences that are Lth-order highpass shaped and conform to

(6).

FIRST-ORDER LOWPASS SEQUENCING LOGIC

To produce a switching sequence $s_{k,r}[n]$ that is a first-order highpass switching sequence, the switching block ensures that its partial sum, $\alpha_1[n]$, is a bounded sequence. Suppose the input to switching block $S_{k,r}$ is always odd and thus, from (6), $s_{k,r}[n] = \pm 1$ for all n. One method for ensuring that $\alpha_1[n]$ is a bounded sequence is by choosing $s_{k,r}[n]$ to be the alternating sequence: $s_{k,r}[n] = (-1)^n = \cos(\pi n)$. With this switching sequence, the resulting partial sum sequence is bounded in magnitude by 1:

$$|\alpha_1[n]| = \left| \sum_{m=0}^{n-1} s_{k,r}[m] \right| \leq 1. \tag{10}$$

and the resulting switching sequence is a single tone of normalized frequency $\omega = \pi$. For many applications, it is desirable to have DAC noise and thus switching sequences that do not contain any tones.

One way to eliminate tones in this scenario and yet obtain a first-order highpass switching sequence is to construct $s_{k,r}[n]$ by randomly choosing between the following two types of *symbols*: "$1, -1$" and "$-1, 1$". When n is even (*i.e.*, $n = 2m$), one of the two symbols is chosen randomly by a fair coin toss, and the chosen symbol is placed in the switching sequence. With this construction, the switching sequence can be written as $s_{k,r}[2m] = \pm 1$ and $s_{k,r}[2m+1] = -s_{k,r}[2m]$. The *alternating property*—*i.e.*, $s_{k,r}[2m+1] = -s_{k,r}[2m]$—ensures that the partial sum sequence satisfies (10), while the random symbol type selection prevents $s_{k,r}[n]$ from containing any periodicities. Therefore, the resulting switching sequence is a first-order highpass switching sequence that does not contain tones.

This method of using symbols to construct the switching sequence can be generalized to include even inputs to the switching block. When the switching block's

input is even, it follows from (6) that the switching block has no choice but to force the switching sequence to be zero. To include potential zero runs in the switching sequence, the two symbols described above are generalized to be

$$1,\underbrace{0,\cdots,0,}_{\substack{\text{Until next}\\ \text{odd } x_{k,r}[n]}}-1,\underbrace{0,\cdots,0}_{\substack{\text{Until next}\\ \text{odd } x_{k,r}[n]}} \quad \text{and} \quad -1,\underbrace{0,\cdots,0,}_{\substack{\text{Until next}\\ \text{odd } x_{k,r}[n]}}1,\underbrace{0,\cdots,0}_{\substack{\text{Until next}\\ \text{odd } x_{k,r}[n]}}. \tag{11}$$

Each symbol begins in the switching sequence with a nonzero value that corresponds to an odd switching block input. The only other nonzero *element* within a symbol has the alternate sign of the first element. For a switching sequence $s_{k,r}[n]$ composed of these symbols, this alternating property ensures that its partial sum satisfies (10), which implies that the resulting switching sequence is a first-order highpass switching sequence. Additionally, by randomly choosing between the two symbol types, the resulting switching sequence cannot contain tones.

As an example, consider the following segment of the input sequence to the switching block $S_{k,r}$:

$$x_{k,r}[n] = \ldots.1,2,2,1,0,1,1,2,\ldots$$

where the segment starts with the value "1" and ends with the value "2". The parity sequence $o_{k,r}[n]$ for this input is

$$o_{k,r}[n] = \ldots.1,0,0,1,0,1,1,0,\ldots$$

The parity sequence $o_{k,r}[n]$ dictates the magnitude of the switching sequence $s_{k,r}[n]$: therefore, the zeros in the parity sequence correspond to zeros in the switching sequence. Given this parity sequence, the symbols "1, 0, 0, −1, 0" and "−1, 1, 0" are used to construct the switching sequence

$$s_{k,r}[n] = \ldots.1,0,0,-1,0,-1,1,0,\ldots$$

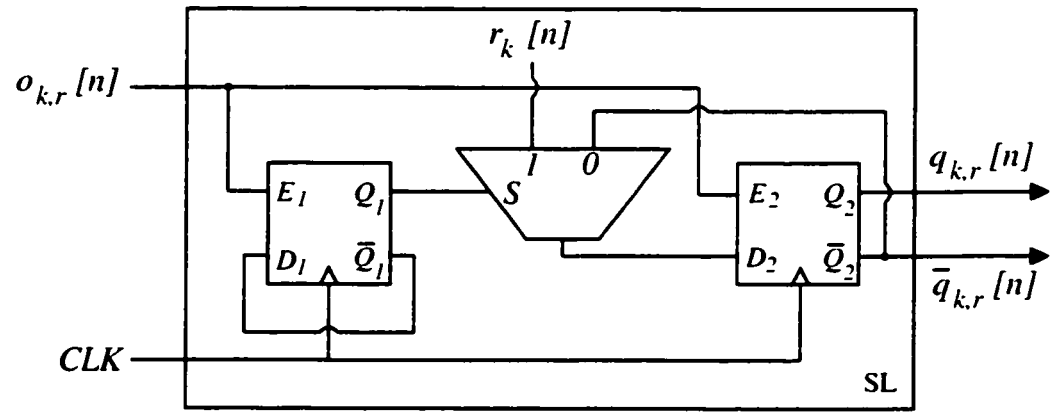


Figure 2.7: The first-order lowpass sequencing logic with dither.

The choice of the symbol "1.0.0.−1.0" over "−1.0.0.1.0" and "−1.1.0" over "1.−1.0" is arbitrary as any combination of these symbols ensure that $|\alpha_1[n]| \leq 1$. In this example, the resulting partial sum sequence is

$$\alpha_1[n] = \ldots.0.1.1.1.0.0.-1.0.\ldots$$

Figure 2.7 displays an example of sequencing logic that generates these symbols in the switching sequence. This sequencing logic contains two D flip-flops and a 2:1 multiplexer. Additionally, a pseudorandom sequence $r_k[n]$ is used to select between the two symbol types and is generated by logic that is not shown in the figure.

Each symbol type from (11) must be further decomposed into two "halves" to describe how the sequencing logic in Figure 2.7 generates the desired switching sequence. The first half of the symbol—*i.e.*, the first "$\pm 1.0.\cdots.0$"—is called the *head* of the symbol, and the second half is called the *tail*. The four states of the D flip-flops correspond to the two symbol types in $s_{k,r}[n]$ and the two segments, head and tail, of the symbol. The bit in the leftmost flip-flop represents the value of $|\alpha_1[n]|$. Since $|\alpha_1[n+1]| = 1$ when $s_{k,r}[n]$ is an element of a symbol's head, and $\alpha_1[n+1] = 0$ when $s_{k,r}[n]$ is an element of a symbol's tail, the bit in the leftmost flip-flop tracks whether $s_{k,r}[n]$ is an element of the head or the tail of a symbol. The rightmost flip-flop contains the $q_{k,r}[n]$ sequence that dictates the symbol types.

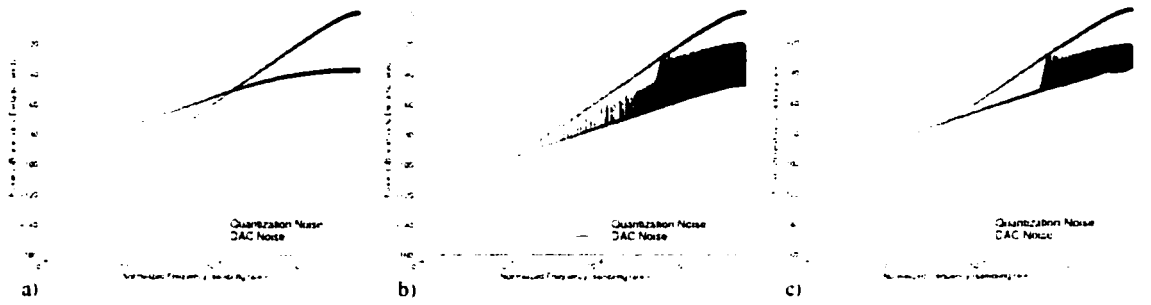


Figure 2.8: DAC noise and quantization noise from a simulation of a 5-bit $\Delta\Sigma$ modulator with the first-order lowpass sequencing logic and dither in a) all of switching blocks. b) none of the switching blocks. and c) the switching blocks in layers 3. 4. and 5.

The symbol types are chosen randomly according to the pseudorandom sequence $r_k[n]$ so that there are no tones in the switching sequence. This pseudorandom sequence is called the *dither* sequence. and a switching block that uses a dither sequence to select its symbols types is called a *dithered* switching block. Ideally. the dither sequence is a sequence of bits that are uniformly distributed and independent. In this implementation. each switching block in a given layer shares the same dither sequence.

Undithered switching blocks may also be utilized to reduce hardware complexity and potentially decrease signal-band DAC noise power. In an undithered switching block. the same symbol type is used throughout the switching sequence. and the sequencing logic can be reduced to a single D flip-flop. The resulting switching sequence can contain tones that lower the noise floor of its PSD relative to the dithered case. This reduced noise floor can give rise to less signal-band DAC noise power. However. the resulting spurious tones in the DAC noise can be prohibitive for a given application. To optimize this tradeoff. some combination of dithered and undithered switching blocks may be employed.

Figure 2.8 displays the PSDs of the DAC noise and quantization noise from behavioral simulations of the 5-bit $\Delta\Sigma$ modulator that was introduced in Section II.

The units of the PSDs are dB relative to Δ^2, where Δ is the step size of the ADC. The capacitor errors in the DAC banks were modeled as independent Gaussian random variables with standard deviations of 1% of their nominal value. This is *not* equivalent to "1% matching" which implies that adjacent capacitors in a given IC are matched within 1%. The input to the $\Delta\Sigma$ modulator was a -1dB (relative to full-scale), 1.5kHz ($\approx 0.0005 f_s$) sinusoid. To illustrate the effects of dither, a dither sequence was applied to selected switching blocks in the simulated $\Delta\Sigma$ modulator. The noise PSDs in Figure 2.8 illustrate how the dither sequences either eliminate or reduce spurious tones in the DAC noise depending on which switching blocks are dithered.

The total hardware required for the sequencing logic in a $(2^b + 1)$-level digital encoder depends on how many switching blocks are dithered. When all switching blocks are dithered, $2 \cdot (2^b - 1)$ D flip-flops, $2^b - 1$ multiplexers, and b pseudorandom sequences are required. When none of the switching blocks are dithered, $2^b - 1$ D flip-flops are required. For the implementation of the $\Delta\Sigma$ modulator in [43], the multiplexer in the sequencing logic is realized by three NAND gates, and the pseudorandom sequences are constructed using a pseudorandom number generator with 28 D flip-flops and 7 XOR gates. The total hardware required for the sequencing logic (not including the pseudorandom number generator) in the digital encoder presented in [43] is 62 D flip-flops and 93 NAND gates.

SECOND-ORDER LOWPASS SEQUENCING LOGIC

The first-order lowpass sequencing logic generates a first-order highpass switching sequence regardless of the values in the switching block's input sequence. However, the restrictions on $s_{k,r}[n]$ given by (6) prevent an analogous claim for the second-order lowpass sequencing logic. For $s_{k,r}[n]$ to be a second-order highpass

switching sequence, the switching block attempts to bound the magnitude of its *double sum sequence* $\alpha_2[n]$ by a constant $K < \infty$ for all n:

$$|\alpha_2[n]| = \left| \sum_{l=0}^{n-1} \alpha_1[l] \right| = \left| \sum_{l=0}^{n-1} \sum_{m=0}^{l-1} s_{k,r}[m] \right| < K.$$

Because the parity of $x_{k,r}[n]$ dictates when $s_{k,r}[n]$ is zero, the sequence $|\alpha_2[n]|$ can be made arbitrarily large by applying the appropriate $x_{k,r}[n]$. For example, suppose $x_{k,r}[0] = 1$, and $x_{k,r}[n] = 0$ for all $n > 0$. Given $\alpha_1[0] = \alpha_2[0] = 0$, then $|\alpha_1[n]| = 1$ and $|\alpha_2[n]| = n - 1$ for all $n > 0$. However, if $x_{k,r}[n]$ is odd with some regularity (as is the case when the DAC is used in a $\Delta\Sigma$ modulator), a switching sequence can be constructed whose double sum is a bounded sequence, thereby giving rise to second-order highpass shaped DAC noise.

One method for creating such a switching sequence is to again use symbols of the form in (11), but with the symbol type chosen to minimize the magnitude of the double sum sequence, $\alpha_2[n]$. In this case, the magnitude of $\alpha_1[n]$ is bounded by one, so the switching sequence is at least a first-order highpass switching sequence. At any time n within a symbol's head, $|\alpha_1[n+1]| = 1$, and it follows that

$$\alpha_2[n+2] = \alpha_2[n+1] + \alpha_1[n+1] = \alpha_2[n+1] \pm 1. \tag{12}$$

Thus, $\alpha_2[n+1]$ increments or decrements by one at each sample time within a symbol's head. However, at any time n within a symbol's tail, $\alpha_1[n+1] = 0$ and $\alpha_2[n+2] = \alpha_2[n+1]$. It follows that the symbol's type and the length of its head determine the values in $\alpha_2[n]$: if a symbol starts at time n and its head's length is N_o samples, it can be shown using induction that

$$\alpha_2[n + N_o + 1] = \alpha_2[n] + (\pm N_o)\,. \tag{13}$$

where the sign of N_o is determined by the symbol type. To minimize the value of $|\alpha_2[n + N_o + 1]|$, the sign of N_o in the above expression should be the opposite of the sign of $\alpha_2[n]$.

To construct such a switching sequence, each switching block ideally calculates $\alpha_2[n]$ with which it selects between the two symbol types. However, when implemented with finite register sizes, the switching block can only estimate $\alpha_2[n]$. This estimate, which is denoted $\hat{\alpha}_2[n]$, has a maximum $M_{max} \geq 0$ and minimum $M_{min} \leq 0$ which are determined by the number of states in a finite-state machine. Therefore, the estimate $\hat{\alpha}_2[n]$ equals $\alpha_2[n]$ only as long as $\alpha_2[n]$ does not exceed the estimate's range (*i.e.*, $M_{min} \leq \alpha_2[n] \leq M_{max}$). The switching block uses the sign of $\hat{\alpha}_2[n]$ to determine the symbol types. To approximate $\alpha_2[n]$, the sequence $\hat{\alpha}_2[n]$ *saturates* when it reaches M_{max} or M_{min}:

$$\hat{\alpha}_2[n+1] = \begin{cases} \hat{\alpha}_2[n] + \alpha_1[n], & \text{if } M_{min} \leq \hat{\alpha}_2[n] + \alpha_1[n] \leq M_{max}; \\ M_{max}, & \text{if } \hat{\alpha}_2[n] + \alpha_1[n] > M_{max}; \\ M_{min}, & \text{if } \hat{\alpha}_2[n] + \alpha_1[n] < M_{min}. \end{cases} \tag{14}$$

The effect of saturation in the above equation can be represented as an accumulated additive error:

$$\hat{\alpha}_2[n+1] = \hat{\alpha}_2[n] + \alpha_1[n] + \varepsilon[n+1]. \tag{15}$$

where $\varepsilon[n]$ is called the *saturation error*. The behavior of the saturation error determines whether the switching sequence is a second-order highpass switching sequence. Since $\alpha_1[n]$ is constrained to the set $\{-1, 0, 1\}$, it follows that $\varepsilon[n]$ is also constrained to this set. Let $N = \min(M_{max}, |M_{min}|)$. For $\varepsilon[n]$ to be nonzero, there must be a run of at least N zeros in the parity sequence $o_{k,r}[n]$. Thus, $x_{k,r}[n]$ must be even for N consecutive samples to cause any saturation error. If the switching block's input is odd at least once within every N-length segment, the saturation error is always zero. From (15), it follows that

$$\hat{\alpha}_2[n] = \varepsilon[n] + \sum_{j=0}^{n-1} (\alpha_1[j] + \varepsilon[j]).$$

The sequence $\alpha_2[n]$ is the partial sum of $\alpha_1[n]$; thus, it follows that

$$\alpha_2[n] = \hat{\alpha}_2[n] - \sum_{j=0}^{n} \varepsilon[j]. \tag{16}$$

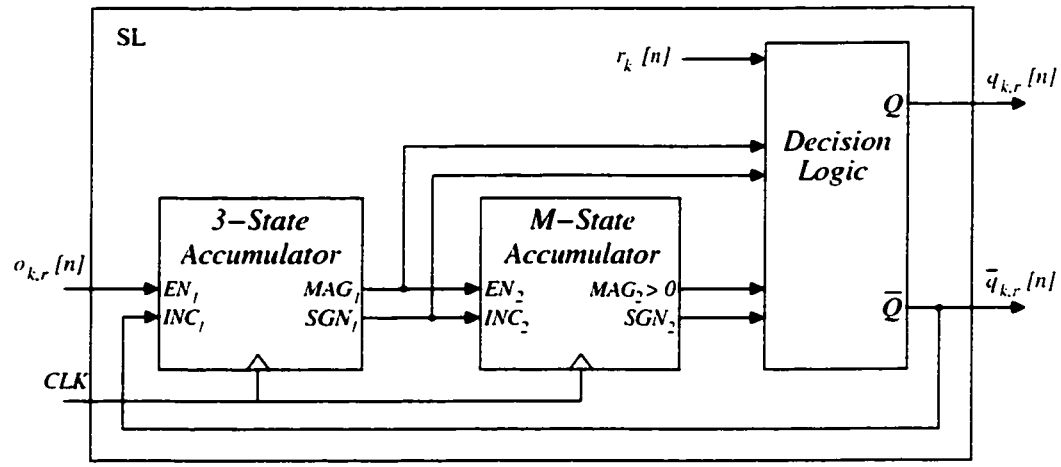


Figure 2.9: The second-order, lowpass sequencing logic with dither.

Because $\hat{\alpha}_2[n]$ is a bounded sequence, $\alpha_2[n]$ is a bounded sequence if and only if the partial sum of $\varepsilon[n]$ is a bounded sequence. Therefore, $s_{k,r}[n]$ is a second-order highpass switching sequence if and only if the partial sum of $\varepsilon[n]$ is a bounded sequence.

The second-order lowpass sequencing logic is shown in Figure 2.9. The 3-state accumulator produces $-\alpha_1[n]$ and the M-state accumulator produces $-\hat{\alpha}_2[n]$. Therefore, the sign of the value in the M-state accumulator is used to choose the symbol types. However, when the M-state accumulator's value and hence $\hat{\alpha}_2[n]$ is zero, the dither sequence $r_k[n]$ is used to choose the symbol type randomly as a means of reducing the spurious tones in $s_{k,r}[n]$. The 3-state accumulator tracks the intrasymbol information for the switching sequence: $|\alpha_1[n+1]| = 1$ when $s_{k,r}[n]$ is an element of the symbol's head, and $\alpha_1[n+1] = 0$ when $s_{k,r}[n]$ is an element of the symbol's tail. When $s_{k,r}[n]$ is in the head of a symbol, the sign of the 3-state accumulator's value is the sign of the tail's first element.

The following is a more detailed description of each element in Figure 2.9:

1. 3-State Accumulator: A state machine that implements an accumulator

restricted to the following three states: $\{-1.0.1\}$.

- EN_1 and INC_1 control the state transitions of the 3-state accumulator as follows:

$$I_1[n+1] = \begin{cases} I_1[n]. & \text{if } EN_1 = 0: \\ I_1[n]+1. & \text{if } EN_1.INC_1 = 1: \\ I_1[n]-1. & \text{otherwise:} \end{cases} \tag{17}$$

  where $I_1[n]$ is the accumulator's state at time n. The feedback prevents $I_1[n]$ from incrementing or decrementing beyond its three states.

- $MAG_1 = |I_1[n]|$ is the magnitude of the accumulator.

- SGN_1 represents the sign of $I_1[n]$:

$$SGN_1 = \begin{cases} 1. & \text{if } I_1[n] > 0: \\ 0. & \text{if } I_1[n] < 0: \\ \text{don't care.} & \text{if } I_1[n] = 0. \end{cases}$$

2. M-State Accumulator: A state machine that implements a saturating accumulator restricted to the M integers in the set $\{-\lceil (M-1)/2 \rceil, \ldots, \lfloor (M-1)/2 \rfloor\}$.

- EN_2 and INC_2 control the state transitions of the M-state logic as follows:

$$I_2[n+1] = \begin{cases} I_2[n]. & \text{if } EN_2 = 0: \\ \min\left(I_2[n]+1, N_{max}\right). & \text{if } EN_2.\ INC_2 = 1: \\ \max\left(I_2[n]-1, N_{min}\right). & \text{otherwise:} \end{cases}$$

  where $I_2[n]$ is the accumulator's state at time n, $N_{min} = -\lceil (M-1)/2 \rceil$, and $N_{max} = \lfloor (M-1)/2 \rfloor$.

- "$MAG_2 > 0$" is high when $|I_2[n]| > 0$ and low when $I_2[n] = 0$.

- SGN_2 represents the sign of $I_2[n]$ and is analogous to SGN_1 in the 3-State Accumulator.

3. Decision Logic: Combinational logic that generates Q and its complement, $\overline{Q}$, as follows:

$$Q = \begin{cases} SGN_1. & \text{if } MAG_1 = 1: \\ SGN_2. & \text{if } MAG_1 = 0.\ \text{"}MAG_2 > 0\text{"} = 1: \\ r_k[n]. & \text{otherwise:} \end{cases} \tag{18}$$

where $r_k[n]$ is a pseudorandom sequence that approximates a sequence of bits that are uniformly distributed and independent.

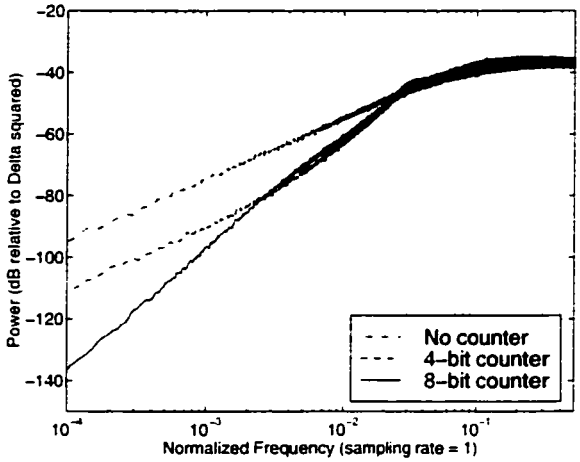


Figure 2.10: DAC noise from a simulation of an analog $\Delta\Sigma$ modulator with the second-order, lowpass sequencing logic with dither and varying counter sizes for the M-state logic.

Figure 2.10 displays DAC noise PSDs from behavioral simulations of the 5-bit $\Delta\Sigma$ modulator presented in Section II with the second-order lowpass sequencing logic. Except for the sequencing logic, all other characteristics of these simulations were the same as those for the first-order lowpass case described previously. Various M-state accumulators were implemented with counters of different sizes. For smaller values of M, the saturation error contributes more power to the DAC noise. In the limit when "No Counter" is used (*i.e.*, when $M = 1$ and $I_2[n] = 0$ for all n), the sequencing logic reduces to the first-order lowpass sequencing logic. When the M-state accumulator is implemented with a 4-bit counter, the power of the signal-band DAC noise decreases relative to the "No Counter" noise, but the saturation error prevents the DAC noise from being second-order highpass shaped. However, with the M-state accumulator realized by an 8-bit counter, the DAC noise in Figure 2.10 has the spectral shape of a second-order highpass sequence.

The additional hardware required to implement the second-order sequencing logic relative to the first-order sequencing logic includes the decision logic, which can be implemented by two 2:1 multiplexers and an inverter, and the M-state

accumulator. If $M = 2^{b_o}$ and the M-state accumulator is implemented with a b_o-bit up/down counter, then SGN_2 is the MSB of the counter and "$MAG_2 > 0$" can be realized by an OR gate with a fan-in of $(b_o - 1)$-bits. The second-order sequencing logic for the implementation of this switching block in [44] uses a 4-bit counter and requires 25 total gates and flip-flops.

IV. SPLITTING NETWORK AND PARITY LOGIC

In an ADC $\Delta\Sigma$ modulator as in Figure 2.1, the delay of the feedback DAC must be small enough so that its output is available well before the next $\Delta\Sigma$ modulator input is clocked in. Therefore, the delay introduced by the switching blocks can limit the maximum sample-rate of the ADC $\Delta\Sigma$ modulator. The sequencing logic blocks presented in Section III do not contribute to the switching block's propagation delay because their outputs can be set before their next input is available. However, the splitting network and parity logic do cause propagation delay.

If the input to the switching block were a binary encoded number, the splitting network could be implemented with binary adders as shown in Figure 2.5, and the parity logic would require no hardware as the input's parity bit would be its LSB. However, the propagation delay introduced by the adders could be significant. In this section, splitting networks are presented that avoid using conventional adders by employing alternative coding schemes for the switching blocks' input and output sequences. Without conventional adders, these splitting networks tend to introduce less propagation delay. The two splitting networks in this section constitute the medium-speed and high-speed switching blocks that offer a tradeoff between complexity and propagation delay. Additionally, efficient implementations of the parity logic blocks are presented for both switching block types.

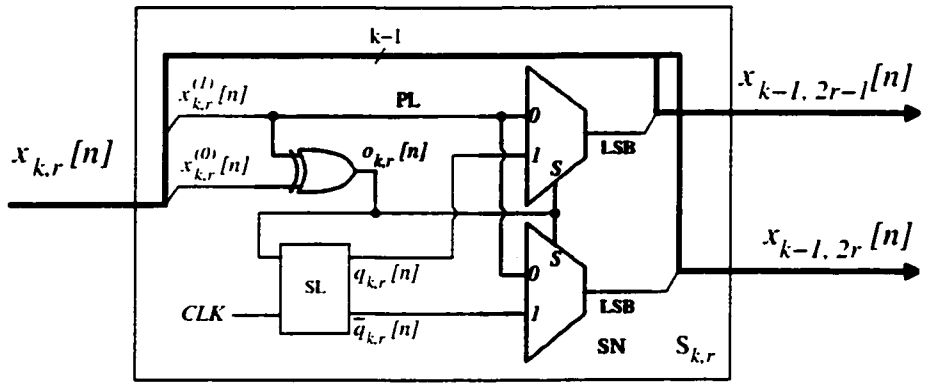


Figure 2.11: The medium-speed switching block.

THE MEDIUM-SPEED SWITCHING BLOCK

Figure 2.11 displays the medium-speed switching block. The parity logic consists of an XOR gate and the splitting network consists of two 2:1 multiplexers. In this section, the sequence "$x_{k,r}[n]$" represents both the input of $S_{k,r}$ and its numerical value; the appropriate representation can be determined by its context. The switching block employs "extra-LSB encoding" of its input and output sequences. Motivated by [39] and detailed in [42], the extra-LSB code of $x_{k,r}[n]$ consists of $k+1$ bits that are denoted $x_{k,r}^{(i)}[n]$ $(i = 0, \ldots, k)$, each of which take on a value of one or zero. The numerical value of $x_{k,r}[n]$ is interpreted as

$$x_{k,r}[n] = \sum_{i=1}^{k} 2^{i-1} x_{k,r}^{(i)}[n] + x_{k,r}^{(0)}[n]. \tag{19}$$

Thus, the extra-LSB code contains two LSBs, $x_{k,r}^{(0)}[n]$ and $x_{k,r}^{(1)}[n]$, both with unity weighting. A conventional unsigned binary encoded number can be converted to an extra-LSB encoded number by appending the 0th bit and setting it low.

With this coding technique, the arithmetic performed by the splitting network only modifies the two LSBs of $x_{k,r}[n]$. As described in Section II, the switching sequence $s_{k,r}[n]$ is nonzero only when $x_{k,r}[n]$ is odd. It follows from (19) that whenever $x_{k,r}[n]$ is odd, one of its LSBs is one and the other is zero. Thus, the splitting network adds ± 1 to $x_{k,r}[n]$ when only one of its LSBs is high, which

implies that the carry bit can never propagate beyond the two LSBs. When $x_{k,r}[n]$ is odd, the splitting network adds one to $x_{k,r}[n]$ by setting both of its LSBs high, or subtracts one from $x_{k,r}[n]$ by setting both of its LSBs low. Since the sequences $x_{k,r}[n] + s_{k,r}[n]$ and $x_{k,r}[n] - s_{k,r}[n]$ are always even valued, both LSBs of these sequences are equal. The splitting network performs the divide-by-two operation by right shifting the $k-1$ MSBs of $x_{k,r}[n]$ and using one of the LSBs of $x_{k,r}[n] \pm s_{k,r}[n]$ as the second LSB of each output.

The two LSBs of $x_{k,r}[n]$ determine its parity. The value of $x_{k,r}[n]$ is odd only when one of its LSBs is one and the other is zero; otherwise, it is even. Therefore, the parity logic implements

$$o_{k,r}[n] = x_{k,r}^{(0)}[n] \oplus x_{k,r}^{(1)}[n].$$

where $\oplus$ represents the XOR operation.

The hardware in each switching block is independent of its location in the digital encoder; therefore, the $(2^b + 1)$-level digital encoder requires $2 \cdot (2^b - 1)$ 2:1 multiplexers for its splitting networks and $2^b - 1$ XOR gates for its parity logic. The efficiency of this implementation increases as the number of bits are increased because the complexity of each switching block does not depend on the bit width — *i.e.*, number of bits — of its input. The medium-speed switching block is used in the 33-level digital encoder presented in [43] wherein the two multiplexers of each splitting network are realized by five NAND gates. The splitting networks and parity logic blocks for this implementation require a total of 186 logic gates. For the $\Delta\Sigma$ADC shown in Figure 2.1, additional hardware is required to convert the thermometer coded output of the flash ADC to an extra-LSB code.

The delay performance for the digital encoder is determined by the digital encoder's *critical path*, which is defined as the longest path that an input bit must

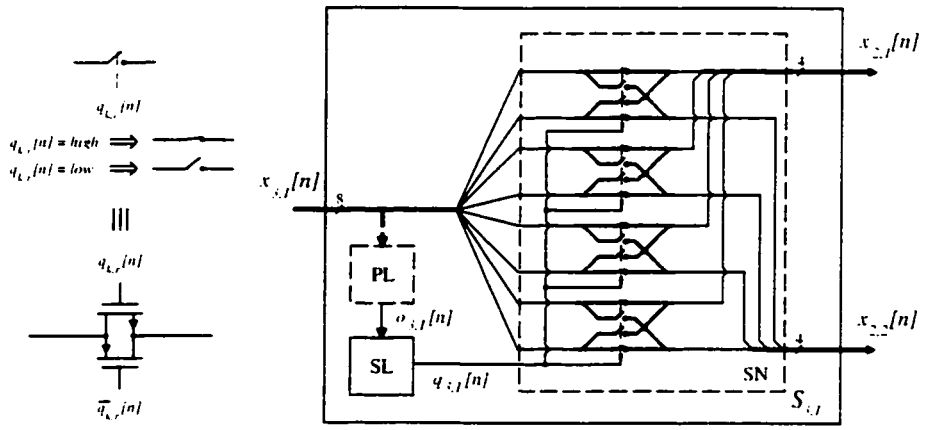


Figure 2.12: The splitting network for a high-speed switching block and the CMOS implementation of a transmission gate.

traverse in a given clock period to set an output bit. Within the medium-speed switching block, the longest path from its input to its outputs consists of an XOR gate and a 2:1 multiplexer. Therefore, the critical path of the $(2^b + 1)$-level digital encoder consists of b XOR gates and b 2:1 multiplexers. HSpice 0.5μm CMOS simulations of the 33-level digital encoder presented in [43] showed that this digital encoder has a delay of approximately 5.7 ns. This does not include the propagation delay of the thermometer-to-binary conversion performed in the $\Delta\Sigma$ADC's digital common-mode rejection flash ADC.

THE HIGH-SPEED SWITCHING BLOCK

Figure 2.12 displays an example high-speed switching block whose splitting network consists entirely of switches implemented by CMOS transmission gates. In this architecture, the parity logic does not physically reside within the switching block. The parity sequences are generated by an XOR tree as shown in Figure 2.13. The high-speed switching block employs thermometer encoding of its input and output sequences. The sequence $x_{k,r}[n]$ is thermometer encoded if it has 2^k

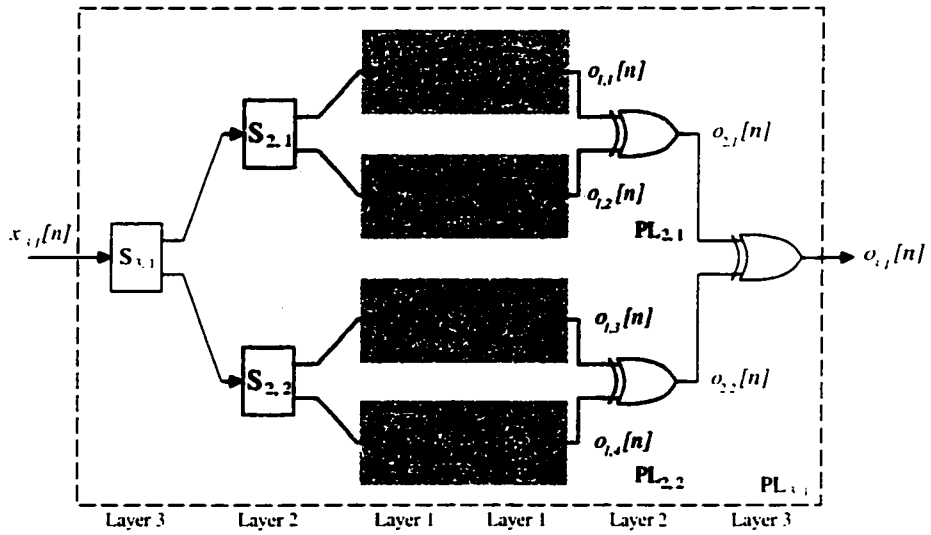


Figure 2.13: The parity logic for the high-speed switching block.

bits ($x_{k,r}^{(i)}[n]$, $i = 1, \ldots, 2^k$) that are assigned as follows:

$$x_{k,r}^{(i)}[n] = \begin{cases} 1, & \text{if } i \leq x_{k,r}[n]; \\ 0, & \text{else.} \end{cases} \tag{20}$$

Thus, with thermometer encoding,

$$x_{k,r}[n] = \sum_{i=1}^{2^k} x_{k,r}^{(i)}[n].$$

With thermometer encoding, the splitting network performs the desired arithmetic by routing the odd indexed bits of $x_{k,r}[n]$ to one output and the even indexed bits of $x_{k,r}[n]$ to the other output, or vice versa, depending upon $q_{k,r}[n]$. It can be shown that the numerical values of the sequences that comprise the even indexed bits and odd indexed bits of $x_{k,r}[n]$ are

$$\sum_{i=1}^{2^{k-1}} x_{k,r}^{(2i)}[n] = \left\lfloor \frac{x_{k,r}[n]}{2} \right\rfloor, \quad \text{and} \quad \sum_{i=1}^{2^{k-1}} x_{k,r}^{(2i-1)}[n] = \left\lceil \frac{x_{k,r}[n]}{2} \right\rceil;$$

respectively. Because $s_{k,r}[n]$ is limited to the set $\{-1, 0, 1\}$, it follows that

$$x_{k-1,2r-1}[n] = \begin{cases} \left\lfloor \frac{x_{k,r}[n]}{2} \right\rfloor, & \text{if } q_{k,r}[n] = 0; \\ \left\lceil \frac{x_{k,r}[n]}{2} \right\rceil, & \text{if } q_{k,r}[n] = 1; \end{cases} \tag{21}$$

and

$$x_{k-1,2r}[n] = \begin{cases} \left\lceil \frac{x_{k,r}[n]}{2} \right\rceil, & \text{if } q_{k,r}[n] = 0; \\ \left\lfloor \frac{x_{k,r}[n]}{2} \right\rfloor, & \text{if } q_{k,r}[n] = 1. \end{cases} \tag{22}$$

Therefore, by routing the input's even and odd indexed bits to separate outputs based on $q_{k,r}[n]$, the splitting network realizes the arithmetic in (21) and (22). Moreover, by preserving the order of these bits, the splitting network ensures its outputs are thermometer encoded.

Since the splitting network does not rely on $o_{k,r}[n]$ to route the bits of $x_{k,r}[n]$, the current sample of $o_{k,r}[n]$ can be determined after the outputs of the digital encoder are set. The parity logic block in this section exploits this flexibility to minimize its hardware. The number of gates required to directly determine the parity of a thermometer encoded number is proportional to its bit width. However, using the XOR tree as shown in Figure 2.13, each parity logic block accounts for only one XOR gate.

The XOR tree is a consequence of the functional relationship between the outputs of a switching block and its input. From (4), the values of the output sequences of a switching block must add to the value of the input. Thus, the parity of $x_{k,r}[n]$ can be determined by the parities of $x_{k-1,2r-1}[n]$ and $x_{k-1,2r}[n]$:

$$o_{k,r}[n] = o_{k-1,2r-1}[n] \oplus o_{k-1,2r}[n]. \tag{23}$$

The outputs of each switching block in layer one are 1-bit sequences. This implies that $x_{0,r}[n] = o_{0,r}[n]$. By recursively implementing (23), the XOR tree generates the parity bits for each switching block.

The hardware counts in the medium-speed and high-speed switching blocks differ only in their splitting networks. With the high-speed switching block, the number of transmission gates in the splitting network depends on the bit width

of the switching block's input. However, the number of transmission gates per layer is independent of the layer number: each bit of a switching block's input is processed by two transmission gates—one on and one off—and the total number of input bits is constant for each layer. Thus, with the high-speed switching block, the $(2^b + 1)$-level digital encoder requires $b \cdot 2^{b+1}$ transmission gates for its splitting networks and $2^b - 1$ XOR gates for its parity logic. A 33-level implementation of this digital encoder for the $\Delta\Sigma$ADC shown in Figure 2.1 requires 320 transmission gates for its splitting network and 31 XOR gates for its parity logic. If the input to the digital encoder were a binary encoded number, as in the case of a $\Delta\Sigma$DAC, a binary-to-thermometer encoder would also be required to implement this digital encoder.

The high-speed switching block tends to have less propagation delay than the medium-speed switching block because the parity logic in the high-speed switching block does not contribute to its delay. As previously mentioned, the sequencing logic does not require the current sample of $o_{k,r}[n]$ to produce $q_{k,r}[n]$. Therefore, $q_{k,r}[n]$ can be calculated and used to set the transmission gates before $y[n]$ is clocked into the digital encoder. Additionally, the XOR tree processes the output bits of the digital encoder and does not contribute to the digital encoder's critical path. Therefore, the critical path of the $(2^b + 1)$-level digital encoder, which is experienced by each of its input bits, consists of b preset transmission gates. HSpice 0.5μm CMOS simulations of a 33-level digital encoder with high-speed switching blocks showed that this digital encoder has a delay of approximately 1.1 ns, which is approximately a 5-times improvement over a 33-level digital encoder with medium-speed switching blocks. This delay does not include the propagation delay of a binary-to-thermometer encoder that would be required in a $\Delta\Sigma$DAC. An implementation

Tree-Structure		Data-Weighted Averaging		Butterfly Shuffler[32]
Med. Speed	High Speed	Barrel Shifter[28]	Binary/Therm[18]	
5-bit T/B encoder	320 T-gates	5-bit T/B encoder	5-bit T/B encoder	320 T-gates
93 MUXes	31 MUXes	320 T-gates	2 5-bit adders	160 XORs
31 XORs	31 XORs	5-bit adder	62 XNORs	80 INVs
31 DFFs	31 DFFs	5 DFFs	36 DFFs	80 DFFs

Table 2.1: Estimated hardware requirements for undithered mismatch-shaping DAC encoders for use within a 5-bit ΔΣADC.

5-bit Tree-Structure		Data-Weighted Averaging		5-bit Butterfly Shuffler[32]
Med. Speed	High Speed	3-bit Rotational[26]	5-bit BiDWA[21]	
5-bit T/B encoder	320 T-gates	3-bit T/B encoder	5-bit T/B encoder	320 T-gates
155 MUXes	93 MUXes	6×1024-bit ROM	320 T-gates	320 XORs
31 XORs	31 XORs	6 DFFs	15 MUXes	80 INVs
62 DFFs	62 DFFs	1 random bit	2 5-bit adders	80 MUXes
5 random bits	5 random bits		10 DFFs	160 DFFs
				5 random bits

Table 2.2: Estimated hardware requirements for mismatch-shaping DAC encoders with dither or other harmonic distortion compensation for use within a ΔΣADC of specified size.

that uses the high-speed switching blocks for its minimal delay is presented in [45].

V. HARDWARE COMPARISON FOR VARIOUS MISMATCH-SHAPING DACS

To compare the hardware complexity of the tree-structured mismatch-shaping DAC encoders presented here to other implementations, Tables 2.1 and 2.2 give estimated hardware requirements for mismatch-shaping DAC encoders appropriate for use in a ΔΣADC. When possible, the DAC encoder hardware is estimated for an implementation in the 5-bit ΔΣADC shown in Figure 2.1. In both tables, the abbreviations "INV", "MUX", "XOR", and "XNOR" stand for inverter, 2:1 multiplexer, exclusive-or, and exclusive-nor, respectively. The abbreviation "T-gate" denotes a two-transistor CMOS transmission gate, and the abbreviation "T/B

encoder" denotes a thermometer-to-binary encoder. A D flip-flop, denoted "DFF", is assumed to have true and complemented outputs available; the D flip-flop with enable, shown in Figure 2.7, is implemented using a D flip-flop and a 2:1 multiplexer.

The mismatch-shaping DAC encoders shown in Table 2.1 provide no hardware to eliminate or reduce spurious tones and the hardware differences are not as pronounced. However, when extra hardware is utilized to combat harmonic distortion, Table 2.2 shows that both the Bidirectional DWA (BiDWA) and tree-structured DAC encoders contain the least hardware. The BiDWA DAC encoder requires minimal hardware because it depends entirely on the randomness of its input to reduce tones in its resulting DAC noise. Any dc input to a $(2^b + 1)$-level BiDWA DAC, besides the trivial inputs of 0 and 2^b, causes its DAC noise to be tonal. On the other hand, the dithered tree-structured DAC has been mathematically proven to produce no tones in its DAC noise [46]. In the Butterfly Shuffler architecture, it is assumed that the logic driving the swapper cells is implemented as the sequencing logic for the tree-structured DAC and only one random bit is used for each *column* in the swapper-cell matrix. For the second-order tree-structured DAC, the hardware difference becomes more pronounced as the 5-bit implementation presented in [44] requires only 988 gates while the 3-bit second-order architecture presented in [15] requires 3500 gates.

VI. CONCLUSION

This paper has presented various implementations of the tree-structured DAC. First-order and second-order lowpass sequencing logic have been presented that provide a tradeoff between DAC-noise power and hardware complexity. High-speed and medium-speed implementations of the splitting network and parity logic have been presented that offer a tradeoff between the digital encoder's propagation delay

and hardware complexity. By appropriately choosing between medium-speed, high-speed, first-order dithered or non-dithered, or second-order implementations, the tree-structured DAC can be optimized for hardware complexity, propagation delay, signal-band DAC-noise power, and DAC-noise harmonic distortion.

CHAPTER ACKNOWLEDGMENT

The text of Chapter 2 appeared as a Regular Paper in the *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 48, no. 11, Nov. 2001. The dissertation author was the primary researcher. Ian Galton supervised the research which forms the basis of this paper. Eric Fogleman contributed to the presentation of the sequencing logic and the hardware comparisons.

REFERENCES

1. W. Redman-White, D. J. L. Bourner, "Improved dynamic linearity in multi-level Σ-Δ converters by spectral dispersion of D/A distortion products," *Proc. ECCTD'89 European Conf. Circuit Theory and Design*, Brighton, U.K., pp. 205-208, Sept. 5-8, 1989.

2. R. K. Henderson, O. Nys, "Dynamic element matching techniques with arbitrary noise shaping function," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 293-296, May 1996.

3. O. Nys, R. K. Henderson, "An analysis of dynamic element matching techniques in sigma-delta modulation," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 231-234, May 1996.

4. M. Adams, C. Toumazou, "A novel architecture for reducing the sensitivity of multibit sigma-delta ADCs to DAC nonlinearity," *Proceedings of the IEEE Symposium on Circuits and Systems*, pp.17-20, May 1995.

5. A. Keady, C. Lyden, "Comparison of mismatch error shaping in multibit over-sampled converters," *Electronic Letters*, vol. 36, no. 6, pp. 506-508, March 19, 1998.

6. L. Hernandez, "Binary weighted D/A converters with mismatch-shaping," *Electronic Letters*, vol. 33, no. 24, pp. 2006-2008, Nov. 20, 1997.

7. L. Hernández. "A model of mismatch-shaping D/A conversion for linearized DAC architectures." *IEEE Trans. on Circuits and Systems – I: Fundamental Theory and Applications*. vol. 45. no. 10. pp. 1068-1076. Oct. 1998.

8. J. Steensgaard. U.-K. Moon. G. Temes. "Mismatch-shaping serial digital-to-analog converter." *Proceedings of the IEEE International Symposium on Circuits and Systems*. vol. 2. pp. 5-8. May 1999.

9. D. Scholnik. J. Coleman. "Vector delta-sigma modulation with integral shaping of hardware-mismatch errors." *Proceedings of the IEEE International Symposium on Circuits and Systems*. vol. 5. pp. 677-680. May 2000.

10. J. Steensgaard. "High-resolution mismatch-shaping digital-to-analog converters." *Proceedings of the IEEE International Symposium on Circuits and Systems*. vol. 1. pp. 516-519. May 2001.

11. B. H. Leung. S. Sutarja. "Multi-bit sigma-delta A/D converter incorporating a novel class of dynamic element matching techniques." *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*. vol. 39. no. 1. pp. 35-51. Jan. 1992.

12. F. Chen. B. H. Leung. "A high resolution multibit sigma-delta modulator with individual level averaging." *IEEE J. Solid-State Circuits*. vol. SC-30. no. 4. pp. 453-460. April 1995.

13. R. Schreier. B. Zhang. "Noise-shaped multi-bit D/A converter employing unit elements." *Electronics Letters*. vol. 31. no. 20. pp. 1712-1713. Sept. 28. 1995.

14. H. Lin. J. Barreiro da Silva. B. Zhang. R. Schreier. "Multi-bit DAC with noise-shaped element mismatch." *IEEE International Symposium on Circuits and Systems. Circuits and Systems*. pp. 235-238. May 1996.

15. A. Yasuda. H. Tanimoto. T. Iida. "A third-order $\Delta\Sigma$ modulator using second-order noise-shaping dynamic element matching." *IEEE J. Solid-State Circuits*. vol. 33. no. 12. pp. 1879-1886. Dec. 1998.

16. Z. Czarnul. K. Oda. T. Iida. "A straightforward design of mismatch-shaped multi-bit $\Delta\Sigma$ D/A Systems." *Proceedings of the IEEE International Symposium on Circuits and Systems*. vol. 5. pp. 717-720. May 2000.

17. M. J. Story. "Digital to analogue converter adapted to select input sources based on a preselected algorithm once per cycle of a sampling signal." U.S. Patent No. 5.138.317. Aug. 11. 1992.

18. H. Spence Jackson. "Circuit and method for cancelling nonlinearity error asso-

ciated with component value mismatches in a data converter." U.S. Patent No. 5,221,926. June 22, 1993.

19. R. T. Baird, T. S. Fiez, "Linearity enhancement of multi-bit ΔΣ A/D and D/A converters using data weighted averaging," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 12, pp. 753-762, Dec. 1995.

20. T. Hamasaki, Y. Shinohara, H. Tersawa, K. Ochiai, M. Hiraoka, H. Hanayama, "A 3-V, 22-mW multibit current-mode ΔΣ DAC with 100 dB dynamic range," *IEEE J. Solid-State Circuits*, vol. 31, no. 12, pp. 1888-1894, Dec. 1996.

21. O. Nys, R. K. Henderson, "A 19-bit low-power multi-bit sigma-delta ADC based on data weighted averaging," *IEEE J. Solid-State Circuits*, vol. 32, pp.933-942, July 1997.

22. I. Fujimori, T. Sugimoto, "A 1.5 V, 4.1 mW dual-channel audio delta-sigma D/A converter," *IEEE J. Solid-State Circuits*, vol 33, no. 12, pp. 1863-1870, Dec. 1998.

23. I. Fujimori, A. Nogi, T. Sugimoto, "A multi-bit delta-sigma audio DAC with 120-dB dynamic range," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 8, pp. 1066-1073, Aug. 2000

24. I. Fujimori, L. Longo, A. Hairapetian, K. Seiyama, S. Kosic, J. Cao, S. Chan, "A 90dB SNR, 2.5 MHz output-rate ADC using cascaded multibit delta-sigma modulation at 8x oversampling ratio," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 12, pp. 1820-1828, Dec. 2000.

25. K. Chen, T. Kuo, "An improved technique for reducing baseband tones in sigma-delta modulators employing data weighted averaging algorithm without adding dither," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no.1, pp. 63-68, Jan. 1999.

26. R. Radke, A. Eshraghi, T. Fiez, "A spurious-free delta-sigma DAC using rotated data weighted averaging," *Proceedings of the 1999 IEEE Custom Integrated Circuits Conference*, pp. 125-128, May, 1999.

27. D. Cini, C. Samori, A. Lacaita, "Double-index averaging: a novel technique for dynamic element matching in Δ-Σ A/D converters," *IEEE Trans. on Ciruits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 4, pp. 353-358, Apr. 1999.

28. Y. Geerts, M. Steyaert, W. Sansen, "A high-performance multibit ΔΣ CMOS ADC," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 12, pp. 1829-1840.

Dec. 2000.

29. X. Gong, E. Gaalaas, M. Alexander, D. Hester, E. Walburger, J. Bian, "A 120dB multi-bit SC audio DAC with second-order noise shaping," *IEEE ISSCC Dig. of Tech. Papers*, pp. 344-345, Feb. 2000.

30. M. Vadipour, "Techniques for preventing tonal behavior of data weighted averaging algorithm in Δ-Σ modulators," *IEEE Trans. on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. 47, no. 11, pp. 1137-1144, Nov. 2000.

31. K. Vleugels, S. Rabii, B. Wooley, "A 2.5V broadband multi-bit $\Delta\Sigma$ modulator with 95dB dynamic range," *IEEE ISSCC Dig. of Tech. Papers*, pp. 50-51, Feb. 2001.

32. R. W. Adams, T. W. Kwan, "Data-directed scrambler for multi-bit noise shaping D/A converters," U.S. Patent No. 5,404,142, Apr. 4, 1995.

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

34. R. Adams, K. Nguyen, K. Sweetland, "A 113-dB SNR oversampling DAC with segmented noise-shaped scrambling," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 1871-1878, Dec. 1998.

35. T. Brooks, D. Robertson, D. Kelly, A. DelMuro, S. Harston, "A 16b sigma-delta pipeline ADC with 2.5MHz output data rate," IEEE ISSCC Digest of Technical Papers, pp. 209-210, Feb. 1997.

36. T. Brooks, D. Robertson, D. Kelly, A. Del Muro, S. Harston, "A cascaded sigma-delta pipeline A/D converter with 1.25 MHz Signal Bandwidth and 89 dB SNR," *IEEE J. Solid-State Circuits*, vol. 32, no. 12, pp. 1896-1906, Dec. 1997.

37. P. Ferguson, X. Haurie, G. Temes, "A highly linear low-power 10-bit DAC for GSM," *IEEE 2000 Custom Integrated Circuits Conference*, pp. 261-264, May 2000.

38. I. Galton, "Noise-shaping D/A converters for delta sigma modulation," *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 1996.

39. A. Keady, C. Lyden, "Tree structure for mismatch noise-shaping multibit DAC," *Electronic Letters*, vol. 33, no. 17, pp. 1431-1432, Aug. 1997.

40. I. Galton. "Spectral shaping of circuit errors in digital-to-analog converters." *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing.* vol. 44. no. 10. pp. 808-817. Oct. 1997.

41. I. Galton. "Spectral shaping of circuit errors in digital-to-analog converters. " U.S. Patent No. 5.684.482. Nov 4. 1997.

42. H.T. Jensen. I. Galton. "A reduced-complexity mismatch-shaping DAC for delta-sigma data converters." *Proceedings of the IEEE International Symposium on Circuits and Systems.* pp. 504-7: May 31-June 3. 1998.

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

44. E. Fogleman. J. Welz. I. Galton. "An audio ADC delta-sigma modulator with 100dB SINAD and 102dB DR using a second-order mismatch-shaping DAC." *IEEE Journal of Solid State Circuits.* vol. 36. no. 3. pp. 339-48. March 2001.

45. J. Grilo. I. Galton. K. Wang. R. Montemayor. "A 12 mW ADC delta-sigma modulator with 80dB of dynamic range integrated in a single-chip bluetooth transceiver." *IEEE 2001 Custom Integrated Circuits Conference.* pp. 23-26. May 2001.

46. J. Welz. I. Galton. "The mismatch-noise PSD from a tree-structured DAC in a second-order delta-sigma modulator with a midscale input." *Proceedings of the IEEE International Conference on Acoustics. Speech. and Signal Processing.* vol. 4. May 7-11. 2001.

47. AK2700: "High precision. high speed ADC: 16-Bits. 2.5MSPS." Product Data Sheet. Asahi Kasei Microsystems: Mar. 2000.

48. AK2710: "High speed DAC w/16-bits resolution at 2.5MSPS." Product Data Sheet. Asahi Kasei Microsystems: Dec. 1999.

49. AD9260: "High-speed oversampling CMOS ADC with 16-Bit resolution at a 2.5 MHz output word rate." Product Data Sheet. Analog Devices. Inc.: Rev. B. 2000.

50. AD1853: "Stereo. 24-bit. 192 kHz. multibit $\Delta\Sigma$ DAC." Product Data Sheet. Analog Devices. Inc.: Rev. A. 1999.

51. PCM1737: "24-bit. 192kHz sampling enhanced multi-level. delta-sigma. audio digital-to-analog converter." Product Data Sheet. Burr-Brown Corporation: March 2000.

52. CS4396: "24-bit. 192kHz D/A converter for digital audio." Product Data Sheet. Cirrus Logic. Inc.: July 1999.

Chapter 3

The PSD of the DAC Noise in the Dithered First-Order Tree-Structured Digital-to-Analog Converter

Jared Welz, Ian Galton

Abstract—The performance of a multi-bit digital-to-analog converter (DAC) is usually limited by how closely its analog components can be matched when fabricated. When a multi-bit DAC is constructed by combining several 1-bit DACs in parallel, the static mismatches among its 1-bit DACs cause its output to be a nonlinear function of its input. The resulting error, called *DAC noise*, limits the DAC's attainable signal-to-noise ratio (SNR). *Mismatch-shaping DACs* mitigate this problem by exploiting built-in redundancy to suppress most of the DAC noise power in the data signal's frequency band. Simulations are usually relied upon to estimate DAC noise power and behavior, which can be misleading because the DAC noise depends on the DAC input. This paper presents a mathematical analysis of the DAC noise PSD in the dithered first-order low-pass tree-structured DAC. This DAC ensures that its DAC noise has a spectral null at dc by generating digital, dc-free sequences using the same techniques that have been developed for line codes. The derived expression for the DAC noise PSD depends on the statistics of these sequences and is used to show various properties of the DAC noise. Specifically, an attainable bound is derived for the signal-band DAC noise power that is independent of the DAC input and can be used as an estimate for the DAC noise power.

I. INTRODUCTION

MULTI-BIT DACs are often constructed by combining several 1-bit DACs in parallel. The multi-bit DAC input is converted to the 1-bit sequences that

drive these 1-bit DACs, and their outputs are summed to obtain an analog version of this input. Multi-bit DACs of this type differ in the number and nominal step sizes of the 1-bit DACs and how the 1-bit DAC inputs are generated.

The key problem with these multi-bit DACs results from the static mismatches among the 1-bit DACs, which are inevitable in their fabrication in today's VLSI technology. Ideally, the multi-bit DAC output is a scaled version of the input; however, these mismatches cause the output to be a memoryless, nonlinear function of the input. The error resulting from this nonlinear function is modeled as an additive noise source called *DAC noise*. The DAC noise limits the effective resolution of these DACs and can contain spurious tones. Both of these symptoms prohibit the use of this DAC in many applications.

A popular method for mitigating this problem is to suppress the DAC noise in some frequency band, called the *signal band*, that encompasses the data signal's spectrum so that most of the out-of-band DAC noise power can be removed by frequency-selective filters. DACs that use this technique are called mismatch-shaping or dynamic element matching DACs [1]-[6]. Each mismatch-shaping DAC employs redundant 1-bit DACs so that, for most values of the mismatch-shaping DAC input, there are several ways to select which 1-bit DAC inputs are high and which are low. This freedom of choice is exploited to modulate the DAC noise so that most of its power resides outside of the signal band. These DACs have facilitated multi-bit $\Delta\Sigma$ modulation [7]-[9] for data conversion and consequently are enabling components in most of today's high-performance $\Delta\Sigma$ data converters [10]-[16].

To date, most of the DAC noise theoretical analyses have been limited to showing that the DAC noise PSD vanishes at some frequency. Most other analyses,

like that for the spurious tones in the DAC noise, have been based on simulations, which can be misleading. Moreover, the values of the signal-band DAC noise power have been estimated using simplified models that assume the DAC noise is independent of the DAC input, which is never true. Moreover, the DAC noise in many implementations is correlated to the DAC input and contains spurious tones.

This paper presents a theoretical analysis of the DAC noise in two versions of the dithered first-order low-pass tree-structured DAC [6], [15]-[19]. The DAC noise in this DAC is a linear combination of digital sequences, called *switching sequences*, that are generated within the DAC. In the analysis of both versions of this DAC, expressions for the DAC noise PSDs are derived as functions of the switching sequence statistics and 1-bit DAC mismatches. These PSD expressions are used to produce a bound of the signal-band DAC noise power for each version of the DAC. Each bound is independent of the multi-bit DAC input and can be used as a worst-case estimate in the design of data converters that employ these DACs. Moreover, each bound is shown to be tight as there is a DAC implementation and input that achieve it.

This paper is divided into four sections and an appendix. Section II reviews the operation of the dithered first-order low-pass tree-structured DACs. This section shows how line coding techniques are used to ensure that the DAC noise PSD has a spectral null at dc. Section III presents and discusses the expressions for the switching sequence PSD and signal band power. Section IV presents and discusses the DAC noise signal-band power bound for the two tree-structured DAC implementations. The Appendix presents the majority of mathematics that form the basis of this paper including the DAC noise PSD expressions for both DAC implementations.

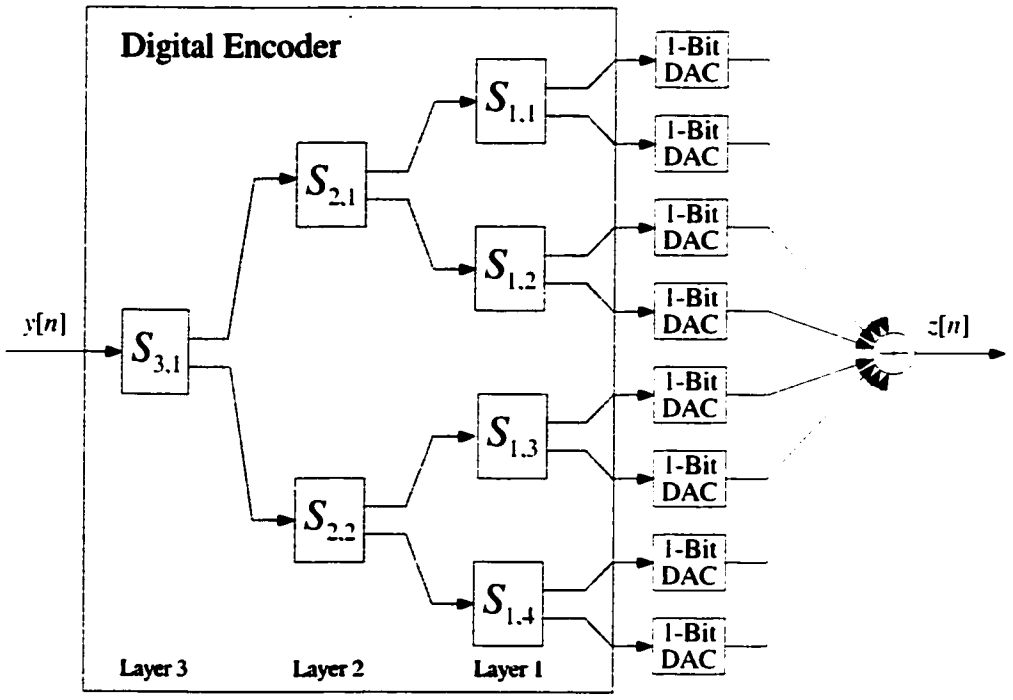


Figure 3.1: A 9-level tree-structured DAC.

II. THE TREE-STRUCTURED DAC

An example 9-level tree-structured DAC is shown in Figure 3.1. In general, the $(2^b + 1)$-level tree-structured DAC, where b is a positive integer, consists of a bank of 2^b 1-bit DACs and a *digital encoder*. The output of the *i-th* 1-bit DAC is given by

$$y_i[n] = \begin{cases} \frac{\Delta_D}{2} + e_{h_i}, & \text{if } x_i[n] \text{ is high;} \\ -\frac{\Delta_D}{2} + e_{l_i}, & \text{if } x_i[n] \text{ is low;} \end{cases} \quad (1)$$

where Δ_D is the nominal smallest step size of the tree-structured DAC, and e_{h_i} and e_{l_i} are the 1-bit DAC's high and low errors, respectively. The 1-bit DAC errors result from inevitable inaccuracies in the fabrication of the 1-bit DACs and are taken to be arbitrary constants. The digital encoder consists of b layers of *switching blocks*, labeled $S_{k,r}$, where $k = 1, \ldots, b$ is the layer number, and $r = 1, \ldots, 2^{b-k}$, is the depth within the layer. The switching blocks are described in more detail later in this section. The input to the digital encoder, $y[n]$, is a sequence whose range is

$\{-2^{b-1}.....2^{b-1}\}$. Typically, $y[n]$ consists of a noise component whose power can be spread across all frequencies and a data signal whose power is confined to the radial frequencies in the interval $(-\pi/OSR, \pi/OSR)$, where OSR is the *oversampling ratio*. The digital encoder outputs, $x_i[n]$ for $i = 1.....2^b$, are 1-bit sequences whose values are taken to be -1/2 at sample times when low and 1/2 at sample times when high.

Ideally, the DAC output is a scaled version of the DAC input: $z[n] = \Delta_D y[n]$. To ensure that the DAC approaches this ideal behavior when the 1-bit DAC errors approach zero, the digital encoder outputs must satisfy the following:

$$x_1[n] + \ldots + x_{2^b}[n] = y[n]. \tag{2}$$

This holds for any multi-bit DAC that is constructed by combining 2^b 1-bit DACs of the same nominal step size with a digital encoder as shown in Figure 3.1. For each value of $y[n]$ except $\pm 2^{b-1}$, there are several possible ways to choose which digital encoder outputs are high and which are low under the constraint that (2) is satisfied. For example, if $y[n] = 0$, (2) is satisfied when the number of digital encoder outputs that are high equals the number of outputs that are low. This inherent redundancy is exploited by the mismatch-shaping DAC to manipulate its DAC noise. In the tree-structured DAC, the processing of the switching blocks, as described next, makes this relationship between the choices of digital encoder and the DAC noise manifest.

Let $x_{k,r}[n]$ denote the input to $S_{k,r}$. With $x_i[n]$ also denoted $x_{0,i}[n]$, the switching blocks are interconnected so that top and bottom outputs of $S_{k,r}$ are $x_{k-1,2r-1}[n]$ and $x_{k-1,2r}[n]$, respectively. To ensure that (2) is satisfied and that $x_i[n] = \pm 1/2$, it is sufficient, as proven in [6], that each switching block satisfies the following two-part *Number Conservation Rule:* the two outputs of each switching block must

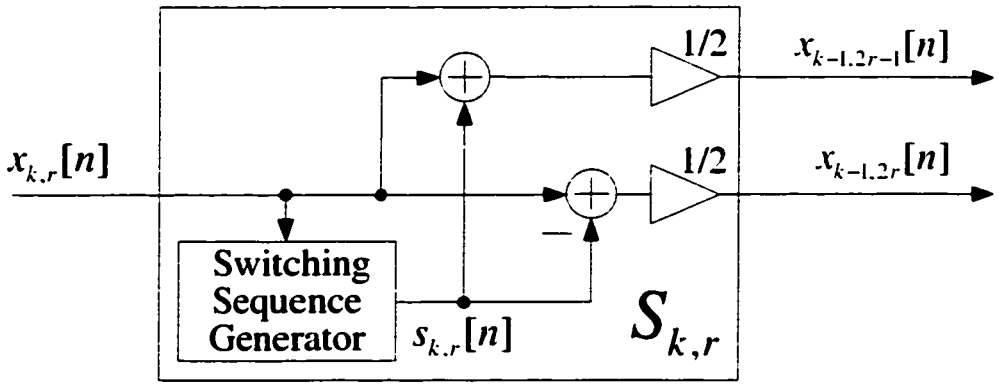


Figure 3.2: The signal processing performed by the switching block.

be in the range $\{-2^{k-2}, \ldots, 2^{k-2}\}$ where k is the layer number, and their sum must equal the input to the switching block:

$$x_{k-1,2r-1}[n] + x_{k-1,2r}[n] = x_{k,r}[n]. \tag{3}$$

This rule is satisfied using the switching block architecture shown in Figure 3.2, which consists of a *switching sequence generator*, an adder, a subtracter, and two divide-by-two elements. Figure 3.2 indicates that

$$x_{k-1,2r-1}[n] = \tfrac{1}{2}\left(x_{k,r}[n] + s_{k,r}[n]\right). \tag{4}$$

and

$$x_{k-1,2r}[n] = \tfrac{1}{2}\left(x_{k,r}[n] - s_{k,r}[n]\right). \tag{5}$$

where $s_{k,r}[n]$ is called the *switching sequence*. To motivate the description of the switching sequence generator, the relationship between the switching sequences and the DAC noise is shown next.

As proven in [6], the DAC output can be written as

$$z[n] = \alpha y[n] + \beta + e[n]. \tag{6}$$

where $y[n]$ is the DAC input, α and β are constants that are functions of the 1-bit

DAC errors, and $e[n]$, called the *DAC noise*, is given by

$$e[n] = \sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} \Delta_{k,r} s_{k,r}[n]. \tag{7}$$

where

$$\Delta_{k,r} = \frac{1}{2^k} \sum_{i=(r-1)2^k+1}^{(r-1)2^k+2^{k-1}} [(e_{h_i} - e_{l_i}) - (e_{h_{i+2^{k-1}}} - e_{l_{i+2^{k-1}}})]. \tag{8}$$

Thus, the DAC noise is a linear combination of the switching sequences. The different possible values of the switching sequences represent the different choices for how the digital encoder selects its output values so that (2) is satisfied. As shown next, the switching sequence generators choose their switching sequences to manipulate the behavior of the DAC noise.

In the dithered, first-order low-pass tree-structured DAC, the switching sequence generator in $S_{k,r}$ selects its switching sequence under the following constraints:

1. It satisfies the following:

$$s_{k,r}[n] = \begin{cases} \pm 1, & \text{if } o_{k,r}[n] = 1; \\ 0, & \text{if } o_{k,r}[n] = 0; \end{cases} \tag{9}$$

   where $o_{k,r}[n]$, called the *parity sequence* of $S_{k,r}$, is 1 when $x_{k,r}[n] + 2^{k-1}$ is odd and 0 otherwise;

2. It is a dc-free sequence;

3. It contains no tones for any choice of the switching block input.

Condition 1 ensures that the switching block satisfies the range requirement of the Number Conservation Rule, while Conditions 2 and 3 ensure that the DAC noise PSD has both of these "desired" properties. With these constraints, the switching sequence can be viewed as a pseudoternary code for its respective parity sequence. Since the switching sequence generator is a finite-state machine, it follows from [20]

that if the parity sequence consists of independent and identically distributed (i.i.d.) bits, then $s_{k,r}[n]$ is a dc-free sequence if and only if its *running digital sum*, given by

$$RDS_{k,r}(m) \equiv \sum_{n=0}^{m} s_{k,r}[n]. \tag{10}$$

takes on only a finite number of values for all m. However, the parity sequence is not necessarily a sequence of i.i.d. bits, but, as shown in the Appendix, this necessary and sufficient condition holds in general.

A common line code that satisfies the first two constraints is the bipolar code [21] (where $o_{k,r}[n]$ represents the data and $s_{k,r}[n]$ is the code). With this code, the nonzero switching sequence values always alternate between 1 and -1; thus, $RDS_{k,r}(m)$ takes on only two values. The *undithered switching block* presented in [17] generates this switching sequence. However, if $o_{k,r}[n] = 1$ for all n, then $s_{k,r}[n] = \sin(\pi n)$, which implies that the bipolar code does not satisfy the third constraint.

To satisfy all three conditions, the switching sequence is constructed by concatenating two types of *symbols*:

$$\text{Type 1: } 1,\underbrace{0,\cdots,0,}_{\substack{\text{Until next}\\ o_{k,r}[n]=1}}-1,\underbrace{0,\cdots,0}_{\substack{\text{Until next}\\ o_{k,r}[n]=1}}; \tag{11}$$

and

$$\text{Type 2: } -1,\underbrace{0,\cdots,0,}_{\substack{\text{Until next}\\ o_{k,r}[n]=1}}1,\underbrace{0,\cdots,0}_{\substack{\text{Until next}\\ o_{k,r}[n]=1}}; \tag{12}$$

the choice of which is made randomly by an approximated *fair coin toss*. Using such symbols to generate $s_{k,r}[n]$ ensures that $RDS_{k,r}(m) \in \{-1, 0, 1\}$; thus, $s_{k,r}[n]$ has a spectral null at dc.

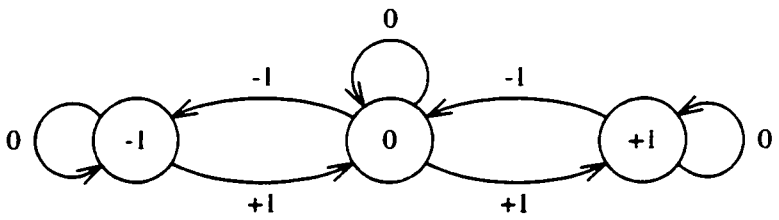


Figure 3.3: The FSTD for the switching sequence code where the state corresponds to the value of $RDS_{k,r}(m)$.

Figure 3.3 shows the finite state transition diagram (FSTD) for the switching sequence generator where the states correspond to the values of $RDS_{k,r}(m)$ and the edge labels are the outputs that occur with the associated changes of state. The state of the switching sequence generator changes only at sample times when the parity sequence is 1. Moreover, it transitions from 0 to 1 at sample times when a Type 1 symbol begins in the switching sequence, and it transitions from 0 to -1 at sample times when a Type 2 symbol begins in the switching sequence; the choice of which, as previously described, is random. Example implementations of this switching sequence generator using 2 D flip-flops are presented in [15] and [17].

If a symbol starts at the "present" sample time n_0, the random symbol type selection implies that, regardless of the parity sequence, $o_{k,r}[n]$, the present and future samples of the switching sequence are uncorrelated from the past samples:

$$E\{s_{k,r}[n_0 - l]s_{k,r}[n_0 + m]\} = 0. \tag{13}$$

for all $m \geq 0$ and $l > 0$. Since every other nonzero sample of $s_{k,r}[n]$ is the start of a new symbol, this implies that the switching sequence does not contain tones regardless of its respective parity sequence.

The choice of the symbol type in $s_{k,r}[n]$ is made randomly with the 1-bit *dither sequence* $d_{k,r}[n]$. The dither sequence approximates a sequence of uniformly distributed, i.i.d. bits whose values are taken to be 1/2 at sample times when high

and -1/2 at sample times when low. If a symbol starts at sample time n_0, then that symbol is a Type 1 symbol if $d_{k,r}[n_0] = 1/2$, and it is a Type 2 symbol if $d_{k,r}[n_0] = -1/2$. For (13) to hold, it is sufficient that $d_{k,r}[n]$ be independent of $x_{k,r}[n]$. Therefore, each switching block in the same layer k can share the same dither sequence—*i.e.*, $d_{k,r}[n] \equiv d_k[n]$ for each layer k. Implementations utilizing this dithering scheme require only b dither sequences, which are realized by pseudorandom sequence generators as demonstrated in [15]. As shown in Section IV, a much tighter DAC noise power bound is obtained when an independent dither sequence is employed by each switching block; however, this implementation requires $2^b - 1$ dither sequences.

III. SWITCHING SEQUENCE SPECTRUM

As reviewed in the previous section, the DAC noise in the tree-structured DAC is a linear combination of the switching sequences. Thus, the DAC noise PSD is a function of the switching sequence PSDs and cross spectra. This section presents and discusses an expression for the switching sequence PSD and signal-band power. The switching sequence cross spectrum is addressed in the Appendix. First, some intuition behind the switching sequence PSD and its derivation is provided along with some required terminology.

The dependence of the switching sequence on the parity sequence in (9) prevents a conventional analysis of its PSD. If $o_{k,r}[n]$ were a sequence of i.i.d. bits, then $s_{k,r}[n]$ could be written as a function of the Markov chain $RDS_{k,r}(m)$, and techniques such as those presented in [22] could be used to analyze the PSD. If $o_{k,r}[n]$ were periodic, then $s_{k,r}[n]$ would be a cyclostationary sequence, and its PSD could be determined by the commonly known techniques (*e.g.*, see [23]) that were introduced in [24]. However, in general, $o_{k,r}[n]$ is neither periodic nor a sequence of i.i.d. bits, so a new

technique must be developed to determine the PSD of $s_{k,r}[n]$.

The technique presented in this paper relies on the randomness in the symbol type selection. As a consequence of this randomness, samples of $s_{k,r}[n]$ that are in different symbols are orthogonal—*i.e.*, if n_0 and n_1 are sample times such that $s_{k,r}[n_0]$ and $s_{k,r}[n_1]$ are in different symbols, then

$$E\{s_{k,r}[n_0]s_{k,r}[n_1]\} = 0. \tag{14}$$

Therefore, the PSD of $s_{k,r}[n]$ depends only on the correlation between samples of $s_{k,r}[n]$ that are within the same symbol. These intrasymbol correlation statistics are conveniently described using the terminology presented next.

Let the symbols described in (11) and (12) be divided into two "halves" where the first $\pm 1.0\ldots.0$ segment is called the *head* of the symbol, and the second such segment is called the *tail* of the symbol. The *head length* of a symbol is defined to be the number of samples of $s_{k,r}[n]$ that constitute the head of that symbol. Let the *head-length process*, $H_{k,r}$, be the random process that represents the head lengths of symbols in $s_{k,r}[n]$; thus, $H_{k,r}[m]$ is the number of samples in the head of the *m-th* symbol in $s_{k,r}[n]$. The definitions of *tail length* and the *tail-length process*, $T_{k,r}$, are analogous to those for the head length and head-length process, respectively.

Theorem 1: The PSD of $s_{k,r}[n]$ is

$$S_{k,r}\left(e^{j\omega}\right) = 2\sigma_{k,r}^2 E\left\{\sin^2\left(\frac{\omega H_{k,r}[m]}{2}\right)\right\}. \tag{15}$$

where $\sigma_{k,r}^2 \equiv E\{s_{k,r}^2[n]\}$, and the signal-band power of $s_{k,r}[n]$ is

$$P_{k,r}\left(OSR\right) = \frac{\sigma_{k,r}^2 E\left\{1 - \operatorname{sinc}\left(\frac{H_{k,r}[m]}{OSR}\right)\right\}}{OSR}. \tag{16}$$

where $\operatorname{sinc}(x) \equiv \sin(\pi x)/(\pi x)$.

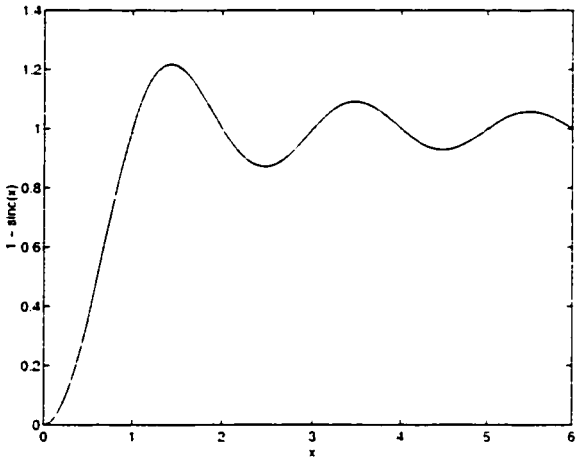


Figure 3.4: The function $1 - \mathrm{sinc}(x)$.

Proof: Presented in the Appendix.

■

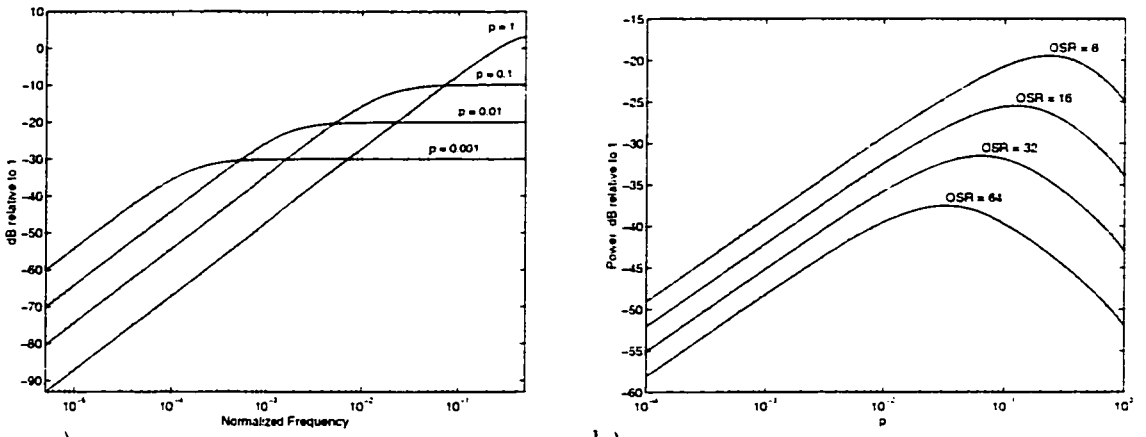
Some properties of the above switching sequence PSD can be discerned even though it depends on the switching sequence head-length statistics and variance. For example, it is shown next that this PSD has a continuous derivative, which implies that the switching sequence cannot contain tones. Let $\phi_{H_{k,r}}(\omega) \equiv E\{e^{j\omega H_{k,r}[m]}\}$, which implies that $\mathrm{Re}\{\phi_{H_{k,r}}(\omega)\} = E\{\cos(\omega H_{k,r}[m])\}$, where $\mathrm{Re}\{\cdot\}$ is the real-part operator. Therefore, the switching sequence can be written as

$$S_{k,r}(e^{j\omega}) = \sigma_{k,r}^2 \left(1 - \mathrm{Re}\{\phi_{H_{k,r}}(\omega)\}\right). \tag{17}$$

Provided $E\{H_{k,r}[m]\} < \infty$, then $\phi_{H_{k,r}}(\omega)$ has a continuous derivative [25] as it is the characteristic function of $H_{k,r}[m]$. Therefore, it follows from (17) that $S_{k,r}(e^{j\omega})$ also has this property in this case. However, if $E\{H_{k,r}[m]\} = \infty$, then $\sigma_{k,r}^2 = 0$ and thus $S_{k,r}(e^{j\omega}) = 0$ because, as proven in Lemma A1 in the Appendix,

$$\sigma_{k,r}^2 = \frac{2}{E\{H_{k,r}[m]\} + E\{T_{k,r}[m]\}}. \tag{18}$$

Therefore, $S_{k,r}(e^{j\omega})$ has a continuous derivative in this case too. By the same reasoning, the real part of the cross spectrum of two switching sequences, as given



a) b)

Figure 3.5: The a) PSD and b) signal-band power of $s_{k,r}[n]$ given its input parity sequence is an i.i.d. Bernoulli sequence with $p = P(o_{k,r}[n] = 1)$.

in Theorem A1 in the Appendix, also has a continuous derivative. This implies that the DAC noise PSD also has this property and thus contains no spurious tones

Properties of the switching sequence signal-band power can also be derived using (16). Shown in Figure 3.4 is a portion of the function that is the argument of the expectation operator in (16). This figure suggests the desired property that the switching sequence power, and thus the DAC noise power, can be made arbitrarily small by increasing the oversampling ratio. Additionally, for a fixed OSR, (16) and (18) imply that the signal-band switching sequence power can be decreased by sufficiently decreasing or increasing, respectively, the head lengths of symbols in $s_{k,r}[n]$. This suggests that, as proven in the next section, there is an upper bound for the switching sequence signal-band power.

Consider the following simplified scenario. Let $o_{k,r}[n]$ be a sequence of i.i.d. Bernoulli trials with $p \equiv P(o_{k,r}[n] = 1)$, and $q \equiv P(o_{k,r}[n] = 0) = 1 - p$. The desired switching sequence statistics are then

$$\sigma_{k,r}^2 = P(o_{k,r}[n] = 1) = p. \tag{19}$$

and

$$P\left(H_{k,r}[m]=h\right)=q^{h-1}p. \tag{20}$$

Substituting (19) and (20) into (15) gives the following switching sequence PSD:

$$S_{k,r}\left(e^{j\omega}\right)=\frac{2p\left(1+q\right)\sin^2\left(\omega/2\right)}{p^2+4q\sin^2\left(\omega/2\right)}. \tag{21}$$

Figure 3.5a shows the switching sequence PSD given above for varying values of p. Additionally, Figure 3.5b shows the switching sequence signal-band power for varying values of p and OSR. Figure 3.5b shows that that, for this simplified parity sequence, the signal-band power of the switching sequence is bounded as a function of OSR. As shown in the next section, this is true in general.

IV. DAC NOISE POWER BOUND

A key part of the proof of the DAC noise power bound is the derivation of the switching sequence power bound, which is provided next.

Theorem 2: The signal-band power of $s_{k,r}[n]$ is bounded as follows:

$$P_{k,r}\left(OSR\right)\leq\frac{2}{OSR\left(OSR+1\right)}. \tag{22}$$

and the bound is achieved if and only if $H_{k,r}[m]=OSR$ and $T_{k,r}[m]=1$ almost surely (a.s.) (*i.e.*, with probability one).

Proof: Since the tail length of every symbol is at least 1 sample, it follows from (18) that

$$\sigma_{k,r}^2\leq\frac{2}{E\left\{H_{k,r}[m]\right\}+1}. \tag{23}$$

Additionally, for any positive integer H, Lemma A2 in the Appendix provides

$$1-\operatorname{sinc}\left(\frac{H}{OSR}\right)\leq\frac{H+1}{OSR+1}. \tag{24}$$

where equality is obtained if and only if $H = OSR$. Substituting (23) and (24) into (16) proves (22). and equality is obtained if and only if $P\left(H_{k,r}[m] = OSR\right)$ and $P\left(T_{k,r}[m] = 1\right)$ are both 1.

■

Because the DAC noise is a linear combination of the switching sequences as shown in (7). Theorem 2 implies that a DAC noise power bound could be obtained as a function of the oversampling ratio and the switching sequence coefficients ($\Delta_{k,r}$ for all k and r). However. in practical circuits. the values of these coefficients are not known. and the DAC noise power is typically estimated as a function of the oversampling ratio and matching statistics of the 1-bit DACs. Thus. to obtain a more useful result. the DAC noise power bounds presented in this paper are functions of the matching of the 1-bit DACs and not the $\Delta_{k,r}$ coefficients. Before the bounds are presented. some additional definitions are required concerning the matching characteristics of the 1-bit DACs.

Denote $e_{h_i} - e_{l_i}$ as the *step-size error* of the *i-th* 1-bit DAC. Let the *relative step-size error* of the *i-th* 1-bit DAC be defined as

$$\delta_i \equiv (e_{h_i} - e_{l_i}) - \frac{1}{2^b}\sum_{j=1}^{2^b}(e_{h_j} - e_{l_j}). \tag{25}$$

Thus. δ_i is the difference between the step size of the *i-th* 1-bit DAC. Δ_i. and the sample average of all the 1-bit DACs. $\bar{\Delta}_D \equiv \frac{1}{2^b}\sum_{j=1}^{2^b}\Delta_j$. Let the sample variance of the step-size errors be denoted

$$\bar{\sigma}_\delta^2 \equiv \frac{1}{2^b}\sum_{i=1}^{2^b}\delta_i^2. \tag{26}$$

As shown next. the DAC noise PSD is bounded by a function of the oversampling ratio and the sample variance given above.

Theorem 3: If a dither sequence is shared by all the switching blocks in each layer, the DAC noise power is bounded as follows:

$$D_{OSR} \leq \frac{4^b \bar{\sigma}_\delta^2}{2 \cdot OSR\,(OSR+1)}. \tag{27}$$

and when $\bar{\sigma}_\delta^2 \neq 0$, this bound is achieved if and only if the following two conditions hold:

1. $H_{1,r}[m] = OSR$ and $T_{1,r}[m] = 1$ a.s. for each $r \in \{1, \ldots, 2^{b-1}\}$:
2. There exists a constant $\hat{\delta}$ such that $\delta_{2j-1} = \hat{\delta}$ and $\delta_{2j} = -\hat{\delta}$ for each $j \in \{1, \ldots, 2^{b-1}\}$.

Moreover, if a unique dither sequence is used in each switching block, then the DAC noise power is bounded as follows:

$$D_{OSR} \leq \frac{2^b \bar{\sigma}_\delta^2}{OSR\,(OSR+1)}. \tag{28}$$

and when $\bar{\sigma}_\delta^2 \neq 0$, the bound is achieved if and only if or the first condition from the previous case holds and the second condition is relaxed to be the following:

$2.'$ $\delta_{2j-1} = -\delta_{2j}$ for each $j \in \{1, \ldots, 2^{b-1}\}$.

Proof: Presented in the Appendix.

■

Theorem 3 implies that, for either dithering scenario, the DAC noise power bound is achieved if the relative mismatch errors satisfy Condition 2, the state of the switching sequence generators in layer one are reset to 0 at sample time $n = 0$, and the DAC input is given by

$$y[n] = \begin{cases} 0, & \text{if } n \bmod (OSR+1) = 0,\ OSR; \\ 2^{b-1}, & \text{otherwise.} \end{cases} \tag{29}$$

In this scenario, $H_{1,r}[m] = OSR$ and $T_{1,r}[m] = 1$ for each $r = 1, \ldots, 2^{b-1}$ and all $m > 0$, which satisfies Condition 1 in the theorem.

The DAC noise power bound is larger in the case where a dither sequence is shared by switching blocks in the same layer because the switching sequences can be correlated in this case. If a symbol starts in $s_{k,r_1}[n]$ and $s_{k,r_2}[n]$ $(r_1 \neq r_2)$ at the same sample time, then the type of each symbol is chosen by the same dither sequence because $d_{k,r_1}[n] = d_{k,r_2}[n] \equiv d_k[n]$. Therefore, these symbols are the same type, and this event gives rise to correlation between the two switching sequences. Although correlation between switching sequences can increase or decrease the DAC noise power, it increases the DAC noise power bound. By using an independent dither sequence in each switching block, a smaller DAC noise power bound is obtained at the cost of additional hardware.

Theorem 3 can be used to discern a guideline concerning the circuit layout of the tree-structured DAC. To achieve either power bound, $\delta_{2j-1} = -\delta_{2j}$ for $j = 1, \ldots, b$. Therefore, to minimize either bound, the DAC should be layed out to optimize the matching between the $(2j-1)$-*st* and $(2j)$-*th* 1-bit DACs. Typically, this is achieved by placing these 1-bit DACs as close as possible to each other or, if possible, interlacing the components of these 1-bit DACs on the integrated circuit. This guideline is in conflict to the often-used practice of the common centroid layout where the goal is to optimize matching amongst all the 1-bit DACs.

The DAC noise power bound can be used for noise budgeting in the design of circuits, such as $\Delta\Sigma$ data converters, that employ the first-order tree-structured DAC. The worst-case matching among 1-bit DACs is often characterized by the maximum or "3σ" relative mismatch error. This maximum error is typically given as a percent, denoted here as $100\xi\%$, of the sample average of the step sizes, $\bar{\Delta}_D$. This implies that $|\delta_i| \leq \xi\bar{\Delta}_D \approx \xi\Delta_D$, which, with (26), leads to $\bar{\sigma}_\delta \leq \xi\Delta_D$. Substituting

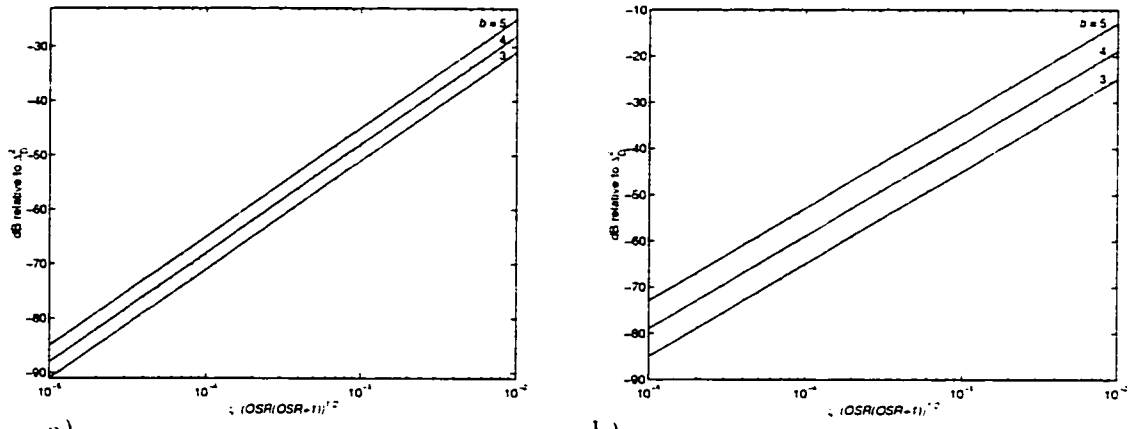


Figure 3.6: DAC noise power bound relative to Δ_D^2 as a function percent mismatch and oversampling ratio with a unique dither sequence used in a) each switching block and b) each layer.

this inequality into (27) and (28) gives

$$\frac{D_{OSR}}{\Delta_D^2} \leq \frac{4^b}{2}\left(\frac{\xi}{\sqrt{OSR(OSR+1)}}\right)^2. \tag{30}$$

and

$$\frac{D_{OSR}}{\Delta_D^2} \leq 2^b\left(\frac{\xi}{\sqrt{OSR(OSR+1)}}\right)^2. \tag{31}$$

respectively. These upper bounds are shown as functions of $\xi/\sqrt{OSR(OSR+1)}$ for $b = 3, 4, 5$ in Figure 3.6. Thus, the size of the tree-structured DAC (*i.e.*, b), the oversampling ratio, the worst-case matching percent, and the dithering scheme can be chosen using (30) and (31) to ensure the DAC noise power is less than the value budgeted to it in a given application.

V. CONCLUSION

Expressions for the switching sequence spectrum and signal-band power in the dithered first-order low-pass tree-structured DAC have been derived. These expressions have been used to obtain an attainable bound on the signal-band DAC noise power for both versions of this DAC. Necessary and sufficient conditions have been given for the bound to be achieved in each case. Additionally, it has been shown

that by using an independent dither sequence in each switching block as opposed to each layer, the DAC noise power bound is smaller and achieved under less stringent conditions on the mismatch errors. Therefore, this dithering scheme is better suited in applications where the bound is used as an estimate for the DAC noise power. It has also been shown that, regardless of the dither scheme, the switching sequence PSD has a continuous derivative, which implies that the DAC noise in both implementations is void of spurious tones.

APPENDIX

The following material provides most of the mathematics to support the theory that is presented in this paper. It is tacitly assumed throughout that all spectral densities considered exist and all sequences are ergodic.

Proposition: Suppose 1) that s[n] is the output of a finite sequential state machine driven by an input sequence which takes on a finite number of values for all n, and 2) that s[n] has a PSD. Then, s[n] has a spectral null at dc if and only if its its running digital sum, $RDS(m) \equiv \sum_{n=0}^{m} s[n]$, takes on a finite number of values for all m.

Proof: First, suppose that $RDS(m)$ takes on a finite number of values for all m. This implies $RDS(m)$ is a *bounded sequence*; *i.e.*, there exists a constant B such that $|RDS(m)| \leq B$ for all m. Therefore, Lemma 1 in [26], which is a generalization of Lemma 1 in [27] (the proof in this lemma does not require that the underlying probability measure be a Markov measure) proves that $s[n]$ has a spectral null at dc. This proof is repeated next because [26] is currently not published. The PSD of $s[n]$ is given by

$$S\left(e^{j\omega}\right) = \lim_{M\to\infty} \frac{1}{M} E\left\{\left|S_M\left(e^{j\omega}\right)\right|^2\right\}. \tag{32}$$

where $E\{\cdot\}$ is the expectation operator, and $S_M\left(e^{j\omega}\right)$ is the M-point Fourier transform of $s[n]$:

$$S_M\left(e^{j\omega}\right) = \sum_{n=0}^{M-1} s[n]e^{-j\omega n}. \tag{33}$$

Evaluating the PSD at $\omega = 0$ gives

$$S\left(e^{j0}\right) = \lim_{M\to\infty} \frac{1}{M} E\left\{\left|\sum_{n=0}^{M-1} s[n]\right|^2\right\}. \tag{34}$$

However, since $RDS(m)$ takes on a finite number of values for all m, there exists a constant B such that $|RDS(m)| \leq B$ for all m. This and (34) indicates

$$S\left(e^{j0}\right) \leq \lim_{M\to\infty} \frac{B^2}{M} = 0. \tag{35}$$

Because $S\left(e^{j\omega}\right)$ is nonnegative for all ω, (35) implies that $S\left(e^{j0}\right) = 0$.

Suppose $s[n]$ has a spectral null at dc. Let z_n represent the *state* of the finite-state sequential machine at time n. If the machine input is an i.i.d. sequence, then it follows from [20] that there exists a complex-valued function $\phi(\cdot)$ such that

$$s[n] = \phi(z_{n+1}) - \phi(z_n). \tag{36}$$

However, the machine input, $o[n]$, is not necessarily an i.i.d. sequence. Regardless, any sequence $o[n]$ can be a sample path of an i.i.d. sequence, so (36) must hold in general. Therefore, $RDS(m) = \phi(z_{m+1}) - \phi(z_0)$, which implies that $RDS(m)$ can take on only a finite number of values for all m.

■

Notation and Definitions: Given the layer number k, let $s_{k,r_1}[n] \equiv s_1[n]$ and $s_{k,r_2}[n] \equiv s_2[n]$. Two symbols in the switching sequences $s_1[n]$ are $s_2[n]$ are called *joint symbols* if they start at the same sample time. Let $H_1[m]$ and $H_2[m]$ represent the head lengths of the *m-th* symbols in $s_1[n]$ and $s_2[n]$, respectively. Let $\hat{H}_1[m]$ and $\hat{H}_2[m]$ be the head lengths of the *m-th* joint symbols in $s_1[n]$ and $s_2[n]$, respectively.

Theorem A1. Switching Sequence Cross Spectrum: Given $s_1[n]$ and $s_2[n]$ employ the same dither sequence, the real part of the cross spectrum of $s_1[n]$ and $s_2[n]$ is given by

$$S_{s_1,s_2}\left(e^{j\omega}\right) = \sigma_1\sigma_2\sqrt{\rho_1\rho_2}E\left\{\sin^2\left(\frac{\omega\hat{H}_1[m]}{2}\right) + \sin^2\left(\frac{\omega\hat{H}_2[m]}{2}\right) - \sin^2\left(\frac{\omega(\hat{H}_1[m]-\hat{H}_2[m])}{2}\right)\right\}. \tag{37}$$

where σ_1 and σ_2 are the standard deviations of $s_1[n]$ and $s_2[n]$, respectively, and ρ_1 and ρ_2 are the probabilities that symbols in $s_1[n]$ and $s_2[n]$, respectively, are joint.

Proof: For $\lambda = 1, 2$, let $w_{\lambda,i}[n]$ be a window sequence that equals one when $s_\lambda[n]$ is an element of the *i-th* joint symbol and zero otherwise. Additionally, let $w_{\lambda,0}[n] \equiv 1 - \sum_{i=1}^{\infty} w_{\lambda,i}[n]$. Therefore, each switching sequence can be written as

$$s_\lambda[n] = \sum_{i=0}^{\infty} w_{\lambda,i}[n]s_\lambda[n]. \tag{38}$$

For $i \neq l$, $w_{1,i}[n]s_1[n]$ and $w_{2,l}[n]s_2[n]$ are orthogonal because the signs of the nonzero values in each sequence are determined by independent, uniform dither sequences. By the same reasoning, $w_{1,0}[n]s_1[n]$ is orthogonal of $w_{2,l}[n]s_2[n]$ for every l, and $w_{1,i}s_1[n]$ is orthogonal of $w_{2,0}[n]s_2[n]$ for every i. In other words,

$$E_x\left\{w_{1,i}[n]s_1[n]w_{2,l}[m]s_2[m]\right\} = 0. \tag{39}$$

for any n, m, when either i or l is zero, or $i \neq l$, where $E_x\{\cdot\}$ is the conditional expectation operator given the switching block inputs (*i.e.*, $E_x\{\cdot\}$ only averages over the possible symbol type choices).

The cross spectrum is derived below by taking the expected value of a time-averaged estimate. Let $\hat{N}_1$ and $\hat{N}_2$ be the number of samples of $s_1[n]$ and $s_2[n]$, respectively, that include the first N joint symbols. Let $N_s = max\{\hat{N}_1, \hat{N}_2\}$. The time-averaged cross spectrum estimate can be written as

$$P_N\left(e^{j\omega}\right) = \frac{1}{N_s}\left(\sum_{n=0}^{N_s-1} s_1[n]e^{-j\omega n}\right)\left(\sum_{m=0}^{N_s-1} s_2[m]e^{j\omega m}\right). \tag{40}$$

Since only N joint symbols are included in this spectrum estimate, it follows that

$$P_N\left(e^{j\omega}\right) = \frac{1}{N_s}\left(\sum_{n=0}^{N_s-1}\sum_{i=0}^{N} w_{1,i}[n]s_1[n]e^{-j\omega n}\right)\left(\sum_{m=0}^{N_s-1}\sum_{l=0}^{N} w_{2,l}[m]s_2[m]e^{j\omega m}\right). \quad (41)$$

Let $S_N\left(e^{j\omega}\right) = E_x\{P_N\left(e^{j\omega}\right)\}$, which, upon rearranging the sums in (41), can be written as

$$S_N\left(e^{j\omega}\right) = \frac{1}{N_s}E_x\left\{\left(\sum_{i=0}^{N}\sum_{n=0}^{N_s-1} w_{1,i}[n]s_1[n]e^{-j\omega n}\right)\left(\sum_{l=0}^{N}\sum_{m=0}^{N_s-1} w_{2,l}[m]s_2[m]e^{j\omega m}\right)\right\}. \quad (42)$$

From (39), the cross terms, with respect to window indices, in the above expectation are all zero (*i.e.*, the terms where $i \neq l$). Moreover, any term in (42) that includes an index of $i = 0$ or $l = 0$ is also zero. Therefore, (42) can be simplified to

$$S_N\left(e^{j\omega}\right) = \frac{1}{N_s}\sum_{i=1}^{N} E_x\left\{\left(\sum_{n=0}^{N_s-1} w_{1,i}[n]s_1[n]e^{-j\omega n}\right)\left(\sum_{m=0}^{N_s-1} w_{2,i}[m]s_2[m]e^{j\omega m}\right)\right\}. \quad (43)$$

Let $N[i]$ denote the sample time of the start of the *i-th* joint symbol ($i \leq N$), and $d[i]$ be the dither sequence sample that chooses the symbol type of the *i-th* joint symbol. The sequences $w_{1,i}[n]s_1[n]$ and $w_{2,i}[m]s_2[m]$ (for $i,\ j > 0$) are nonzero for only two samples (*i.e.*, the first element of the head and tail of the symbol), and so

$$\sum_{n=0}^{N_s-1} w_{1,i}[n]s_1[n]e^{-j\omega n} = 2d[i]e^{-j\omega N[i]}\left(1 - e^{-j\omega \dot{H}_1[i]}\right). \quad (44)$$

and

$$\sum_{m=0}^{N_s-1} w_{2,i}[m]s_2[m]e^{j\omega m} = 2d[i]e^{j\omega N[i]}\left(1 - e^{j\omega \dot{H}_2[i]}\right). \quad (45)$$

Substituting (44) and (45) into (43), gives

$$S_N\left(e^{j\omega}\right) = \frac{1}{N_s}\sum_{i=1}^{N} E_x\left\{4d^2[i]\left(1 - e^{-j\omega \dot{H}_1[i]}\right)\left(1 - e^{j\omega \dot{H}_2[i]}\right)\right\}. \quad (46)$$

However, $4d^2[i] = 1$ for each i, which implies that there is no randomness with respect to the dither sequence in the above argument of the expectation operator:

thus.

$$S_N\left(e^{j\omega}\right) = \frac{1}{N_s}\sum_{i=1}^{N}\left(1 - e^{-j\omega\hat{H}_1[i]}\right)\left(1 - e^{j\omega\hat{H}_2[i]}\right). \tag{47}$$

Let $\hat{S}_N\left(e^{j\omega}\right)$ be the real part of $S_N\left(e^{j\omega}\right)$: it follows from the linearity of the real part operator that

$$\hat{S}_N\left(e^{j\omega}\right) = \frac{2}{N_s}\sum_{i=1}^{N}\sin^2\left(\frac{\omega\hat{H}_1[i]}{2}\right) + \sin^2\left(\frac{\omega\hat{H}_2[i]}{2}\right) - \sin^2\left(\frac{\omega(\hat{H}_1[i]-\hat{H}_2[i])}{2}\right). \tag{48}$$

Let N_1 and N_2 be the total number of symbols in $s_1[n]$ and $s_2[m]$ up to and including the Nth joint symbol. Because a switching sequence is nonzero (± 1) only twice within a symbol. the time-averaged estimate of the variance of $s_1[n]$ and $s_2[n]$ is

$$\bar{\sigma}_1^2 \equiv \frac{1}{N_s}\sum_{n=0}^{N_1-1} s_1^2[n] = \frac{2N_1}{N_s}. \tag{49}$$

and

$$\bar{\sigma}_2^2 \equiv \frac{1}{N_s}\sum_{n=0}^{N_2-1} s_2^2[n] = \frac{2N_2}{N_s}. \tag{50}$$

respectively. Additionally. after N joint symbols. the fraction of symbols in $s_1[n]$ and $s_2[n]$ that are joint is given by

$$\bar{\rho}_1 \equiv \frac{N}{N_1}. \tag{51}$$

and

$$\bar{\rho}_2 \equiv \frac{N}{N_2}. \tag{52}$$

respectively. Thus. (49). (50). (51). and (52) is substituted into (48) to give

$$\hat{S}_N\left(e^{j\omega}\right) = \bar{\sigma}_1\bar{\sigma}_2\sqrt{\bar{\rho}_1\bar{\rho}_2}\frac{1}{N}\sum_{i=1}^{N}\sin^2\left(\frac{\omega\hat{H}_1[i]}{2}\right) + \sin^2\left(\frac{\omega\hat{H}_2[i]}{2}\right) - \sin^2\left(\frac{\omega(\hat{H}_1[i]-\hat{H}_2[i])}{2}\right). \tag{53}$$

With $E_N\{\cdot\}$ defined as the time-averaged expectation operator. (53) becomes

$$\hat{S}_N\left(e^{j\omega}\right) = \bar{\sigma}_1\bar{\sigma}_2\sqrt{\bar{\rho}_1\bar{\rho}_2}E_N\left\{\sin^2\left(\frac{\omega\hat{H}_1[m]}{2}\right) + \sin^2\left(\frac{\omega\hat{H}_2[m]}{2}\right) - \sin^2\left(\frac{\omega(\hat{H}_1[m]-\hat{H}_2[m])}{2}\right)\right\}. \tag{54}$$

Under the ergodicity assumption, the time averages in (54) converge to ensemble averages as $N \to \infty$. Therefore, with $S_{s_1,s_2}\left(e^{j\omega}\right) = \lim_{N\to\infty} \hat{S}_N\left(e^{j\omega}\right)$, (37) follows from (54).

■

Corollary A1. Cross Spectrum Area: Given an oversampling ratio of OSR, and $s_1[n]$ and $s_2[n]$ employ the same dither sequence, the signal-band area of the real part of the cross spectrum of $s_1[n]$ and $s_2[n]$ is given by

$$A_{OSR} = \frac{\sigma_1\sigma_2\sqrt{\rho_1\rho_2}}{2 \cdot OSR} E\left\{1 - \text{sinc}\left(\frac{\hat{H}_1[m]}{OSR}\right) - \text{sinc}\left(\frac{\hat{H}_2[m]}{OSR}\right) + \text{sinc}\left(\frac{\hat{H}_1[m] - \hat{H}_2[m]}{OSR}\right)\right\}. \quad (55)$$

Proof: Given Theorem A1, the cross spectrum area is

$$A_{OSR} \equiv \frac{1}{2\pi} \int_{\frac{-\pi}{OSR}}^{\frac{\pi}{OSR}} S_{s_1,s_2}\left(e^{j\omega}\right) d\omega. \quad (56)$$

Because the argument of the expectation operator in (37) consists of bounded functions, Fubini's Theorem [28] implies that the integral and expected value, implied in (56), can be swapped. Thus, (55) results upon evaluating this integral.

■

Theorem 1. Switching Sequence PSD and Signal-Band Power: See Section III for the theorem statement

Proof: With $s_1[n] = s_2[n] = s_{k,r}[n]$, $\sigma_1\sigma_2 = \sigma_{k,r}^2$, and since every symbol in the same switching sequence starts at the same sample time, $\rho_1 = \rho_2 = 1$ and $\hat{H}_1[m] = \hat{H}_2[m] = H_{k,r}[m]$. Substituting these values into (37) and (55) leads to (15) and (16), respectively.

■

Theorem A2. DAC Noise PSD: Given each switching block in the same layer shares a dither sequence, the DAC noise PSD is given by

$$D\left(e^{j\omega}\right) = \sum_{k=1}^{b}\left(\sum_{r=1}^{2^{b-k}} \Delta_{k,r}^2 S_{k,r}\left(e^{j\omega}\right) + 2\Delta_{k,r}\left(\sum_{\hat{r}=1}^{r-1} \Delta_{k,\hat{r}} S_{k,r,\hat{r}}\left(e^{j\omega}\right)\right)\right). \tag{57}$$

where $S_{k,r}\left(e^{j\omega}\right)$ is the switching sequence PSD for $s_{k,r}[n]$ as given by (15), and $S_{k,r,\hat{r}}\left(e^{j\omega}\right)$ is the real part of the cross spectrum of $s_{k,r}[n]$ and $s_{k,\hat{r}}[n]$ as given by (37). Moreover, if a unique dither sequence is used in each switching block, the DAC noise PSD is

$$D\left(e^{j\omega}\right) = \sum_{k=1}^{b}\sum_{r=1}^{2^{b-k}} \Delta_{k,r}^2 S_{k,r}\left(e^{j\omega}\right). \tag{58}$$

Proof: First, assume that switching blocks in the same layer share a dither sequence. Because switching sequences in different layers employ independent dither sequences, these switching sequences are uncorrelated and thus have no cross spectrum. Therefore, only the cross spectrum from switching sequences in the same layer contribute to the DAC noise power.

Let $u_{k,r}[n]$ be the sequence

$$u_{k,r}[n] = \sum_{\hat{r}=1}^{r} \Delta_{k,\hat{r}} s_{k,\hat{r}}[n]. \tag{59}$$

To apply mathematical induction, suppose for some $r_0 = 1, \ldots, 2^{b-k} - 1$, that the PSD of $u_{k,r_0}[n]$ is

$$U_{k,r_0}\left(e^{j\omega}\right) = \sum_{r=1}^{r_0} \Delta_{k,r}^2 S_{k,r}\left(e^{j\omega}\right) + 2\Delta_{k,r}\left(\sum_{\hat{r}=1}^{r_0-1} \Delta_{k,\hat{r}} S_{k,r,\hat{r}}\left(e^{j\omega}\right)\right). \tag{60}$$

The PSD of $u_{k,r_0+1}[n]$ can be written as

$$U_{k,r_0+1}\left(e^{j\omega}\right) = U_{k,r_0}\left(e^{j\omega}\right) + \Delta_{k,r_0+1}^2 S_{k,r_0+1}\left(e^{j\omega}\right) + 2\Delta_{k,r_0+1} C_{k,r_0}\left(e^{j\omega}\right). \tag{61}$$

where $C_{k,r_0}\left(e^{j\omega}\right)$ is the real part of the cross spectrum of $u_{k,r_0}[n]$ and $s_{k,r_0+1}[n]$, which, given (59), is calculated to be

$$C_{k,r_0}\left(e^{j\omega}\right) = \sum_{\hat{r}=1}^{r_0} \Delta_{k,\hat{r}} S_{k,\hat{r},r_0+1}\left(e^{j\omega}\right). \tag{62}$$

Substituting (60) and (62) into (61) gives

$$U_{k,r_0+1}\left(e^{j\omega}\right) = \sum_{r=1}^{r_0+1} \Delta_{k,r}^2 S_{k,r}\left(e^{j\omega}\right) + 2\Delta_{k,r}\left(\sum_{\hat{r}=1}^{r_0} \Delta_{k,\hat{r}} S_{k,r,\hat{r}}\left(e^{j\omega}\right)\right). \tag{63}$$

Therefore, it follows from mathematical induction that (60) holds for each r_0. Since the switching sequences in different layers are uncorrelated, it follows from (7) and (59) that

$$D\left(e^{j\omega}\right) = \sum_{k=1}^{b} U_{k,2^{b-k}}\left(e^{j\omega}\right). \tag{64}$$

Substituting (60) (with $r_0 = 2^{b-k}$) into (64) gives (57).

When an independent dither sequence is employed by each switching block, all of the switching sequences are uncorrelated, which implies that $S_{k,r,\hat{r}}\left(e^{j\omega}\right) = 0$ for all ω, k, and $\hat{r} \neq r$. Substituting this into (57) leads to (58).

■

Corollary A2. DAC Noise Signal-Band Power: Given an independent dither is shared by all the switching blocks in each layer, the signal-band DAC noise power is

$$D_{OSR} = \sum_{k=1}^{b}\left(\sum_{r=1}^{2^{b-k}} \Delta_{k,r}^2 P_{k,r}\left(OSR\right) + 2\Delta_{k,r}\left(\sum_{\hat{r}=1}^{r-1} \Delta_{k,\hat{r}} A_{k,r,\hat{r}}\left(OSR\right)\right)\right). \tag{65}$$

where $P_{k,r}\left(OSR\right)$ is the signal-band power of $s_{k,r}[n]$ (as in (16)) and $A_{k,r,\hat{r}}\left(OSR\right)$ is the signal-band area of the cross spectrum of $s_{k,r}[n]$ and $s_{k,\hat{r}}[n]$ (as in (55)). If a unique dither is used in each switching block, then the signal-band DAC noise power is

$$D_{OSR} = \sum_{k=1}^{b}\sum_{r=1}^{2^{b-k}} \Delta_{k,r}^2 P_{k,r}\left(OSR\right). \tag{66}$$

Proof: The proof follows directly from Corollary A1. Theorem 1. Theorem A2. and the linearity of the integral.

■

Lemma A1: The switching sequence variance is

$$\sigma_{k.r}^2 = \frac{2}{E\{H_{k.r}[m]\} + E\{T_{k.r}[m]\}}. \tag{67}$$

Proof: Let M_s be the number of samples in the first M symbols $s_{k.r}[n]$. Given the ergodicity assumption. it follows that

$$\sigma_{k.r}^2 = \lim_{M\to\infty} \frac{1}{M_s} \sum_{l=0}^{M_s} s_{k.r}^2[l]. \tag{68}$$

Since $s_{k.r}^2[n] = 1$ twice within every symbol. (68) can be simplified to

$$\sigma_{k.r}^2 = \lim_{M\to\infty} \frac{2M}{M_s}. \tag{69}$$

Additionally. the ergodicity assumption implies

$$E\{H_{k.r}[m] + T_{k.r}[m]\} = \lim_{M\to\infty} \frac{1}{M} \sum_{i=1}^{M} H_{k.r}[i] + T_{k.r}[i]. \tag{70}$$

However. $\sum_{i=1}^{M} H_{k.r}[i] + T_{k.r}[i]$ is the total number of samples comprising the first M symbols. *i.e.*. M_s. This implies that

$$E\{H_{k.r}[m] + T_{k.r}[m]\} = \lim_{M\to\infty} \frac{M_s}{M}. \tag{71}$$

This and (69) imply (67).

■

Lemma A2: Given H and OSR are positive integers.

$$1 - \text{sinc}\left(\frac{H}{OSR}\right) \leq \frac{H+1}{OSR+1}. \tag{72}$$

where equality is obtained if and only if $H = OSR$.

Proof: This lemma requires three *claims*, which are stated and proven next. These claims concern the following function:

$$f_\gamma(x) \equiv \frac{1 - \text{sinc}(x)}{x + \gamma}. \tag{73}$$

where γ is a constant in the interval $(0, 1)$ and $x > 0$.

Claim 1: For $x \geq 2$.

$$f_\gamma(x) < \frac{1}{2}. \tag{74}$$

Proof: Let

$$f(x) \equiv \frac{1 - \text{sinc}(x)}{x}. \tag{75}$$

which, since $\gamma > 0$, implies that $f(x) > f_\gamma(x)$ for all $x > 0$. The derivative of $f(x)$ is evaluated to be

$$f'(x) = \frac{2\cos\left(\frac{\pi x}{2}\right)\left[\text{sinc}\left(\frac{x}{2}\right) - \cos\left(\frac{\pi x}{2}\right)\right]}{x^2}. \tag{76}$$

The above derivative is zero for the following values of $x \geq 2$: (1) $x = (2l + 1)$, where l is a positive integer, and (2) $x > 0$ such that

$$\text{sinc}\left(\frac{x}{2}\right) = \cos\left(\frac{\pi x}{2}\right). \tag{77}$$

For the first case,

$$f(x) = \frac{1}{(2l+1)} < \frac{1}{2}. \tag{78}$$

For the second case,

$$f(x) = \frac{\sin^2\left(\frac{\pi x}{2}\right)}{x} \leq \frac{1}{x}. \tag{79}$$

where $x > 0$ satisfies

$$\tan\left(\frac{\pi x}{2}\right) = \frac{\pi x}{2}. \tag{80}$$

However, the smallest $x > 0$ that satisfies (80) is greater than 2. Thus, for this x, (79) implies that $f(x) < 1/2$. This, (78), and the First Derivative Theorem [29] imply that all the local maxima of $f(x)$ for $x \geq 2$ have values that are less then $1/2$. Since $f(2) = 1/2$, this implies that $f(x) \leq 1/2$ for all $x \geq 2$, and since $f_\gamma(x) < f(x)$, this also implies (74).

■

Claim 2: For $x \in (0, 2]$, the function $f_\gamma(x)$ has one local maximum, which is its global maximum for $x > 0$.

Proof: The derivative of $f_\gamma(x)$ is evaluated to be

$$f'_\gamma(x) = \frac{(2x+\gamma)\operatorname{sinc}(x) - (x+\gamma)\cos(\pi x) - x}{x(x+\gamma)^2}. \tag{81}$$

which can be written as

$$f'_\gamma(x) = \frac{4x\cos^2\left(\frac{\pi x}{2}\right) h\left(\frac{\pi x}{2}\right) + \gamma\cos(\pi x)\, h(\pi x)}{\pi x^2 (x+\gamma)^2}. \tag{82}$$

where

$$h(y) \equiv \tan(y) - y. \tag{83}$$

Given the properties of the tangent function, it follows that $h(y) > 0$ for $y \in (0, \pi/2)$, and $h(y) < 0$ for $y \in ((2l-1)\pi/2, \pi l]$, where l is any positive integer. This and the properties of the cosine imply that $f'_\gamma(x) > 0$ for $x \in (0, 1]$, and $f'_\gamma(x) < 0$ for $x \in [3/2, 2]$. Thus, the First Derivative Theorem implies that there are no local

maximums for $f_{\gamma}(x)$ in the intervals $(0, 1]$ and $[3/2, 2]$, and there is at least one local maximum in the interval $(1, 3/2)$.

Using (81), the expression $f'_{\gamma}(x) = 0$ can be simplified to

$$\underbrace{2(x+\gamma)\cos^2\left(\frac{\pi x}{2}\right) - (2x+\gamma)\,\mathrm{sinc}(x)}_{\equiv g(x)} = \gamma. \tag{84}$$

The derivative of $g(x)$ is solved to be

$$g'(x) = \left(-\pi(x+\gamma) + \frac{\gamma}{\pi x^2}\right)\sin(\pi x) + 2\left(1+\frac{\gamma}{x}\right)\sin^2\left(\frac{\pi x}{2}\right) - \frac{\gamma}{x}. \tag{85}$$

For all $x \in (1, 3/2)$ and $\gamma \in (0, 1)$, it follows that $\pi(x+\gamma) > \gamma/(\pi x^2)$, $\sin^2(\pi x/2) > 1/2$, and

$$\frac{\frac{\gamma}{x}}{2\left(1+\frac{\gamma}{x}\right)} > \frac{1}{5}. \tag{86}$$

This and (85) imply that $g'(x) > 0$ for all values of $x \in (1, 3/2)$, and $g(x)$ is a strictly increasing function in this range [29]. This implies that there is at most one value of $x \in (1, 3/2)$ that satisfies (84), and thus, by the First Derivative Theorem, there is at most one local maximum for $f_{\gamma}(x)$ in this range.

This and the previous arguments imply that $f_{\gamma}(x)$ has exactly one local maximum for $x \in (0, 2]$, and because $f_{\gamma}(1) = 1/(1+\gamma) > 1/2$, Claim 1 implies that this local maximum is the global maximum of this function for $x > 0$.

■

Claim 3: The global maximum of $f_{\gamma}(x)$ is achieved for a single value of x in the interval $(1, 1+\gamma)$.

Proof: It follows from Claim 2 that, in order to prove this claim, it is sufficient to prove that

$$f_{\gamma}(1) > f_{\gamma}(1+\gamma). \tag{87}$$

Using the trigonometric identity for the sine of a sum, it follows that

$$\operatorname{sinc}(1+\gamma) = -\frac{\gamma \operatorname{sinc}(\gamma)}{1+\gamma}. \tag{88}$$

This implies

$$f_{\gamma}(1+\gamma) = \frac{1+\frac{\gamma \operatorname{sinc}(\gamma)}{1+\gamma}}{1+2\gamma}. \tag{89}$$

which can be written as

$$f_{\gamma}(1+\gamma) = \left(\frac{1+\gamma(1+\operatorname{sinc}(\gamma))}{1+2\gamma}\right)\frac{1}{1+\gamma}. \tag{90}$$

Since $f_{\gamma}(1) = 1/(1+\gamma)$, and $\operatorname{sinc}(\gamma) < 1$ for all $\gamma \in (0,1)$, (90) implies that $f_{\gamma}(1+\gamma) < f_{\gamma}(1)$.

■

Fix the value of $OSR > 1$, and consider the function $f_{\frac{1}{OSR}}\left(\frac{H}{OSR}\right)$, where H is a positive integer. Since, $f_{\frac{1}{OSR}}\left(\frac{OSR}{OSR}\right) = OSR/(OSR+1) > 1/2$. Claim 1 implies that the maximum of this function occurs for some value of $H < 2 \cdot OSR$. Moreover, Claim 2 and Claim 3 imply that the maximum of this function is achieved at either $H = OSR$ or $H = OSR + 1$. However, substituting $\gamma = 1/OSR$ into (87) indicates that $f_{\frac{1}{OSR}}\left(\frac{OSR}{OSR}\right) > f_{\frac{1}{OSR}}\left(\frac{OSR+1}{OSR}\right)$. Therefore, the global maximum of $f_{\frac{1}{OSR}}\left(\frac{H}{OSR}\right)$ is $OSR/(OSR+1)$, which implies (72), and it is achieved only when $H = OSR$.

■

Notation and Definitions: Let $\vec{\nu}_{k,r}$ be a 2^b-length column vector whose *i-th* component is defined to be

$$\nu_{k,r,i} \equiv \begin{cases} \left(\frac{1}{2}\right)^{k/2}. & \text{if } (r-1)2^k < i \le (r-1)2^k + 2^{k-1}; \\ -\left(\frac{1}{2}\right)^{k/2}. & \text{if } (r-1)2^k + 2^{k-1} < i \le r2^k; \\ 0. & \text{otherwise.} \end{cases} \tag{91}$$

Moreover, let $\vec{\delta}$ be the 2^b-length column vectors whose *i-th* component is δ_i.

Lemma A3: Given $c_{k,r}$ is a nonnegative constant for each k and r.

$$\sum_{k=1}^{b}\sum_{r=1}^{2^{b-k}} c_{k,r}\Delta_{k,r}^2 \le \max_{k,r}\{2^{b-k}c_{k,r}\}\bar{\sigma}_{\delta}^2. \tag{92}$$

and equality is obtained if and only if

$$\vec{\delta} = \sum_{(k,r)\in K} b_{k,r}\vec{\nu}_{k,r}. \tag{93}$$

where each $b_{k,r}$ is a constant, and

$$K \equiv \left\{(k,r) \mid 2^{b-k}c_{k,r} = \max_{k,r}\{2^{b-k}c_{k,r}\}\right\}. \tag{94}$$

Proof: It follows from the definitions of $\Delta_{k,r}$, γ_i, and $\vec{\nu}_{k,r}$ as given in (8), (25) and (91), respectively, that

$$\Delta_{k,r} = \frac{\vec{\nu}_{k,r}^T\vec{\delta}}{2^{k/2}} = \frac{\vec{\delta}^T\vec{\nu}_{k,r}}{2^{k/2}}. \tag{95}$$

This and the distributive and associative properties of matrices imply that the left-hand side of (92) can be written as

$$\sum_{k=1}^{b}\sum_{r=1}^{2^{b-k}} c_{k,r}\Delta_{k,r}^2 = \vec{\delta}^T \underbrace{\left(\sum_{k=1}^{b}\sum_{r=1}^{2^{b-k}} \frac{c_{k,r}}{2^k}\vec{\nu}_{k,r}\vec{\nu}_{k,r}^T\right)}_{\equiv D}\vec{\delta}. \tag{96}$$

Given $(k_1,r_1) \neq (k_2,r_2)$ (and each are plausible layer numbers and depths), (91) implies that $\nu_{k_1,r_1,i}$ is a constant function of i for all values of i where $\nu_{k_2,r_2,i} \neq 0$. This implies that $\vec{\nu}_{k_1,r_1}^T\vec{\nu}_{k_2,r_2} = 0$ because the set of nonzero values of $\vec{\nu}_{k_2,r_2}$ consists of an equal number of values that are $(1/2)^{k_2/2}$ and $-(1/2)^{k_2/2}$. Moreover, (91) implies that $\vec{\nu}_{k,r}^T\vec{\nu}_{k,r} = 1$ for each k and r. Therefore, the $2^b - 1$ vectors, $\vec{\nu}_{k,r}$ for all k and r, that compose the matrix D are orthonormal. This implies that the expression for the matrix D in (96) is the spectral decomposition of the matrix [30].

and each vector $\vec{\nu}_{k,r}$ is eigenvector of this matrix with an associated eigenvalue of $\lambda_{k,r}$ which is given by

$$\lambda_{k,r} = \frac{c_{k,r}}{2^k}. \tag{97}$$

Since D is a symmetric matrix, the Rayleigh-Ritz Theorem [30] implies that the quadratic expression on the right-hand side of (96) is bounded above by $\lambda_{max}\vec{\delta}^T\vec{\delta}$, where λ_{max} is the maximum eigenvalue of D. This and (97) imply that

$$\sum_{k=1}^{b}\sum_{r=1}^{2^{b-k}} c_{k,r}\Delta_{k,r}^2 \leq \max_{k,r}\left\{\frac{c_{k,r}}{2^k}\right\}\vec{\delta}^T\vec{\delta}. \tag{98}$$

which, given $2^b\bar{\sigma}_\delta^2 = \vec{\delta}^T\vec{\delta}$, proves (92). Additionally, the Rayleigh-Ritz Theorem states that the bound is achieved if and only if $\vec{\delta}$ is a linear combination of the eigenvectors whose associated eigenvalues are equal to λ_{max} as given in (93).

■

Lemma A4: The real-part of the signal-band area of the cross-spectrum of the sequences $\Delta_{k,r_1}s_{k,r_1}[n]$ and $\Delta_{k,r_2}s_{k,r_2}[n]$ is bounded as follows:

$$\Delta_{k,r_1}\Delta_{k,r_2}A_{k,r_1,r_2}(OSR) \leq \frac{\Delta_{k,r_1}^2 P_{k,r_1}(OSR) + \Delta_{k,r_2}^2 P_{k,r_2}(OSR)}{2}. \tag{99}$$

where equality is achieved if and only if $\Delta_{k,r_1} = \Delta_{k,r_2} = 0$ or $s_{k,r_1}[n] = s_{k,r_2}[n]$ a.s. and $\Delta_{k,r_1} = \Delta_{k,r_2}$.

Proof: Let $w[n] = \Delta_{k,r_1}s_{k,r_1}[n] - \Delta_{k,r_2}s_{k,r_2}[n]$. By computing the PSD of $w[n]$ and integrating it across the range of the signal band, the power of this sequence is found to be

$$P_w(OSR) = \Delta_{k,r_1}^2 P_{k,r_1}(OSR) + \Delta_{k,r_2}^2 P_{k,r_2}(OSR) - 2\Delta_{k,r_1}\Delta_{k,r_1}A_{k,r_1,r_2}(OSR). \tag{100}$$

Since $P_w(OSR) \geq 0$, (99) follows from (100).

The bound is trivially achieved if $\Delta_{k,r_1} = \Delta_{k,r_2} = 0$; therefore, assume that this does not hold for the remainder of the proof. If $\Delta_{k,r_1} s_{k,r_1}[n] = \Delta_{k,r_2} s_{k,r_2}[n]$ a.s., then $w[n] = 0$ a.s. Therefore, $P_w(OSR) = 0$ in this case, and, upon substituting this into (100), equality is obtained in (99). Because $s_{k,r_1}[n]$ and $s_{k,r_2}[n]$ are both constrained to the range $\{-1, 0, 1\}$, $\Delta_{k,r_1} s_{k,r_1}[n] = \Delta_{k,r_2} s_{k,r_2}[n]$ a.s. if and only if $\Delta_{k,r_1} = \pm\Delta_{k,r_2}$ and $s_{k,r_1}[n] = \pm s_{k,r_2}[n]$ a.s. However, two switching sequences are only correlated when a symbol in each starts at the sample time, and in such cases, the switching sequences have positive correlation. Therefore, $s_{k,r_1}[n] \neq -s_{k,r_2}[n]$ a.s., which implies that $\Delta_{k,r_1} s_{k,r_1}[n] = \Delta_{k,r_2} s_{k,r_2}[n]$ a.s. if and only if $\Delta_{k,r_1} = \Delta_{k,r_2}$ and $s_{k,r_1}[n] = s_{k,r_2}[n]$ a.s.

If $\Delta_{k,r_1} \neq \Delta_{k,r_2}$ and $s_{k,r_1}[n] = s_{k,r_2}[n]$ a.s., then $w[n] = (\Delta_{k,r_1} - \Delta_{k,r_2})\, s_{k,r_1}[n]$ a.s., and $P_w(OSR) > 0$. This and (100) imply equality is not achieved in (99) in this case.

Suppose $s_{k,r_1}[n] \neq s_{k,r_2}[n]$ a.s. Recall the notation used in Theorem A1 and that $\tilde{H}_1[m]$ represents the head length of the *m-th* joint symbol in $s_{k,r_1}[n]$. Let $\hat{H}_1[m]$ be the head length of the *m-th non-joint* symbol in $s_{k,r_1}[n]$. By averaging the joint and non-joint symbols, it follows from (15) that the PSD of $s_{k,r_1}[n]$ can be written as

$$S_{k,r_1}\left(e^{j\omega}\right) = 2\sigma_1^2\rho_1 E\left\{\sin^2\left(\frac{\omega\tilde{H}_1[m]}{2}\right)\right\} + 2\sigma_1^2(1-\rho_1)\, E\left\{\sin^2\left(\frac{\omega\hat{H}_1[m]}{2}\right)\right\}. \quad (101)$$

Furthermore, consider the analogous definition and result for $s_{k,r_2}[n]$.

Suppose, for purpose of contradiction, that $P_w(OSR) = 0$. The PSD of $w[n]$ is

$$S_w\left(e^{j\omega}\right) = \Delta_{k,r_1}^2 S_{k,r_1}\left(e^{j\omega}\right) + \Delta_{k,r_2}^2 S_{k,r_2}\left(e^{j\omega}\right) - 2\Delta_{k,r_1}\Delta_{k,r_2} S_{k,r_1,r_2}\left(e^{j\omega}\right). \quad (102)$$

where $S_{k,r_1,r_2}\left(e^{j\omega}\right)$ is the real part of the cross spectrum of $s_{k,r_1}[n]$ and $s_{k,r_2}[n]$ as given in (37). Since $S_w\left(e^{j\omega}\right)$ is continuous, $w[n]$ has no signal-band power if

and only if $S_w\left(e^{j\omega}\right) = 0$ for all $\omega \in (-\pi/OSR, \pi/OSR)$. Therefore, the second derivative of $S_w\left(e^{j\omega}\right)$ is zero at $\omega = 0$. However, it follows from (37), (101), and Fatou's Lemma [28] that

$$\lim_{\omega \to 0} \frac{S_w\left(e^{j\omega}\right)}{\omega^2} \geq \frac{1}{2}\left(\Delta_{k,r_1}^2 \sigma_1^2 (1-\rho_1) + \Delta_{k,r_2}^2 \sigma_2^2 (1-\rho_2)\right). \tag{103}$$

Since $s_{k,r_1}[n] \neq s_{k,r_2}[n]$ a.s., there is a finite probability that symbols in both switching sequences are not joint: *i.e.*, $\rho_1 < 1$ and $\rho_2 < 1$. This and (103) imply that the second derivative of $S_w\left(e^{j\omega}\right)$, if it exists, is greater than 0 which is a contradiction. Therefore, $P_w(OSR) > 0$ in this case, which implies that equality is not obtained in (99).

■

Theorem 3. DAC Noise Power Bound: See Section IV for the theorem statement.

Proof: Consider the case where an independent dither sequence is used only for each layer of the DAC. Substituting the inequality in (99) into (65) indicates

$$\begin{aligned} D_{OSR} \leq & \sum_{k=1}^{b} \left(\sum_{r=1}^{2^{b-k}} \Delta_{k,r}^2 P_{k,r}(OSR) \right) \\ & + \sum_{r_1=1}^{2^{b-k}} \sum_{r_2=1}^{r_1-1} \left(\Delta_{k,r_1}^2 P_{k,r_1}(OSR) + \Delta_{k,r_2}^2 P_{k,r_2}(OSR) \right). \end{aligned} \tag{104}$$

Simplifying (104) gives

$$D_{OSR} \leq \sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} 2^{b-k} \Delta_{k,r}^2 P_{k,r}(OSR). \tag{105}$$

Substituting the power bound in (22) into (105) leads to

$$D_{OSR} \leq \frac{1}{OSR\,(OSR+1)} \sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} 2^{b-k+1} \Delta_{k,r}^2. \tag{106}$$

Applying Lemma A3 with $c_{k,r} \equiv 2^{b-k+1}$, the inequality in (92) is substituted into (106) to give

$$D_{OSR} \leq \frac{1}{OSR\,(OSR+1)} \max_{k,r}\left\{2 \cdot 4^{b-k}\right\} \bar{\sigma}_{\delta}^{2}. \tag{107}$$

Since $\max_{k,r}\left\{2 \cdot 4^{b-k}\right\} = 4^{b}/2$, (27) follows from (107).

Now, consider the case where an independent dither sequence is used in each switching block. Substituting the power bound from (22) into (66) gives

$$D_{OSR} \leq \frac{1}{OSR\,(OSR+1)} \sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} 2 \cdot \Delta_{k,r}^{2}. \tag{108}$$

Applying Lemma A3 again but with $c_{k,r} \equiv 2$, the inequality in (92) is substituted into (108) to give

$$D_{OSR} \leq \frac{\bar{\sigma}_{\delta}^{2}}{OSR\,(OSR+1)} \max_{k,r}\left\{2^{b-k+1}\right\}. \tag{109}$$

Since $\max_{k,r}\left\{2^{b-k+1}\right\} = 2^{b}$, (28) follows from (108) and (109).

For both dithering schemes, $\max_{k,r}\{2^{b-k} c_{k,r}\}$ is achieved with $k = 1$. Thus, Lemma A3 implies that the relative mismatch error vector, $\vec{\delta}$, achieves equality in this case if and only if it is a linear combination of the vectors $\vec{v}_{1,r}$ for $r = 1, \ldots, 2^{b-1}$. From (91), such a vector is characterized by having $\delta_{2j} = -\delta_{2j-1}$ for $j = 1, \ldots, 2^{b-1}$. With these relative mismatch errors, (8) implies that, for $k > 1$, $\Delta_{k,r} = 0$ for each r. In this case, the DAC noise is solely a linear combination of switching sequences in the first layer.

From Theorem 2, the signal-band power of $s_{k,r}[n]$ is maximized only when $H_{k,r}[m] = OSR$ and $T_{k,r}[m] = 1$ a.s. In order for each switching sequence in layer k_0 to satisfy this condition, each parity sequence in this layer must almost surely be a deterministic function of the DAC input and thus not dependent on a dither sequence. For this to hold, $s_{k,r}[n] = 0$ a.s. for each $k > k_0$ and r, and $s_{k,r}[n]$ is

almost surely not a deterministic sequence for each $k < k_0$ and r. Moreover, since OSR is assumed to be greater than 1, this condition holds only if $s_{k_0,r_1}[n] = s_{k_0,r_2}[n]$ a.s. for each r_1 and r_2.

The inequality given in (28) depends only on the inequalities in Lemma A3 and Theorem 2. Therefore, it follows from the previous arguments that equality is obtained in (28) if and only if $\delta_{2j} = -\delta_{2j-1}$ for each $j = 1, \ldots, 2^{b-1}$, and $H_{1,r}[m] = OSR$ and $T_{1,r}[m] = 1$ a.s. for each r.

The inequality in (27) also depends on that in Lemma A4. As previously discussed, if $H_{1,r}[m] = OSR$ and $T_{1,r}[m] = 1$ a.s. for each r, then $s_{1,r_1}[n] = s_{1,r_2}[n]$ a.s. for each r_1 and r_2. Therefore, given this holds, equality is achieved in (99) for every $r_1 \neq r_2$ if and only if there exists a constant $\hat{\delta}$ such that $\Delta_{1,r} = \hat{\delta}$ for each $r = 1, \ldots, 2^{b-1}$. Given this condition holds, (8) implies that

$$\delta_{2j} - \delta_{2j-1} = 2\hat{\delta}. \tag{110}$$

If, in addition, $\delta_{2j} = -\delta_{2j-1}$ as required to achieve the inequality in (92), then (110) implies that $\delta_{2j} = -\delta_{2j-1} = \hat{\delta}$ for each j. Therefore, the bound in (27) is achieved if and only if this condition holds and $H_{1,r}[m] = OSR$ and $T_{1,r}[m] = 1$ a.s. for each r.

■

CHAPTER ACKNOWLEDGMENT

The text of Chapter 3 is to be submitted, in part or in full, for publication as a Regular Paper in the *IEEE Transactions on Information Theory*. The dissertation author was the primary researcher. Ian Galton supervised the research which forms the basis of this paper.

REFERENCES

1. B. H. Leung. S. Sutarja. "Multi-bit sigma-delta A/D converter incorporating a novel class of dynamic element matching techniques." *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing.* vol. 39. no. 1. pp. 35-51. Jan. 1992.

2. M. J. Story. "Digital to analogue converter adapted to select input sources based on a preselected algorithm once per cycle of a sampling signal." U.S. Patent No. 5.138.317. Aug. 11. 1992.

3. R. T. Baird. T. S. Fiez. "Linearity enhancement of multi-bit $\Delta\Sigma$ A/D and D/A converters using data weighted averaging." *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing.* vol. 42. no. 12. pp. 753-762. Dec. 1995.

4. R. Schreier. B. Zhang. "Noise-shaped multi-bit D/A converter employing unit elements." *Electronics Letters.* vol. 31. no. 20. pp. 1712-1713. Sept. 28. 1995.

5. R. W. Adams. T. W. Kwan. "Data-directed scrambler for multi-bit noise shaping D/A converters." U.S. Patent No. 5.404.142. Apr. 4. 1995.

6. I. Galton. "Spectral shaping of circuit errors in digital-to-analog converters." *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing.* vol. 44. no. 10. pp. 808-817. Oct. 1997.

7. W. Chou. R.M. Gray. "Dithering and its effects on sigma-delta and multistage sigma-delta modulation." *IEEE Transactions on Information Theory.* vol.37. no.3. part .1. pp.500-13. May 1991.

8. N. He. F. Kuhlmann. A. Buzo. "Multiloop sigma-delta quantization." *IEEE Transactions on Information Theory.* vol.38. no.3. pp.1015-28. May 1992.

9. I. Galton. "Granular quantization noise in a class of delta-sigma modulators." *IEEE Transactions on Information Theory.* vol.40. no.3. pp.848-59. May 1994.

10. T. W. Kwan. R. W. Adams. R. Libert. "A stereo multibit sigma delta DAC with asynchronous master-clock interface." *IEEE Journal of Solid-State Circuits.* vol. 31. no. 12. pp. 1881-1887. Dec. 1996.

11. R. Adams. K. Nguyen. K. Sweetland. "A 113-dB SNR oversampling DAC with segmented noise-shaped scrambling." *IEEE J. Solid-State Circuits.* vol. 33. no. 12. pp. 1871-1878. Dec. 1998.

12. T. Brooks. D. Robertson. D. Kelly. A. Del Muro. S. Harston. "A cascaded sigma-

delta pipeline A/D converter with 1.25 MHz signal bandwidth and 89 dB SNR." *IEEE J. Solid-State Circuits*, vol. 32, no. 12, pp. 1896-1906, Dec. 1997.

13. A. Yasuda, H. Tanimoto, T. Iida, "A third-order $\Delta\Sigma$ modulator using second-order noise-shaping dynamic element matching," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 1879-1886, Dec. 1998.

14. I. Fujimori, L. Longo, A. Hairapetian, K. Seiyama, S. Kosic, J. Cao, S. Chan, "A 90dB SNR, 2.5 MHz output-rate ADC using cascaded multibit delta-sigma modulation at 8x oversampling ratio," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 12, pp. 1820-1828, Dec. 2000.

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

16. E. Fogleman, J. Welz, I. Galton, "An audio ADC delta-sigma modulator with 100dB SINAD and 102dB DR using a second-order mismatch-shaping DAC," *IEEE Journal of Solid State Circuits*, vol. 36, no. 3, pp. 339-48, March 2001.

17. J. Welz, I. Galton, E. Fogleman, "Simplified logic for first-order and second-order mismatch-shaping digital-to-analog converters," *IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 48, no. 11, Nov. 2001.

18. J. Welz, I. Galton, "The mismatch-noise PSD from a tree-structured DAC in a second-order delta-sigma modulator with a midscale input," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2625-2628, May 7-11, 2001.

19. J. Grilo, I. Galton, K. Wang, R. Montemayor, "A 12-mW ADC delta-sigma modulator with 80 dB of dynamic range integrated in a single-chip Bluetooth transceiver," *IEEE Journal of Solid-State Circuits*, vol. 37, no.3 , pp. 271-278, March 2002.

20. G.L. Pierobon, "Codes for zero spectral density at zero frequency," *IEEE Transactions on Information Theory*, vol. IT-30, pp. 435-439, Mar. 1984.

21. H. Kobayashi, "A survey of coding schemes for transmission or recording of digital data," *IEEE Transactions on Communications*, vol. COM-19, pp.1087-1099, Dec. 1971.

22. G. Bilardi, R. Padovani, G. Pierobon, "Spectral analysis of functions of Markov chains with applications," *IEEE Transactions on Communications*, vol. COM-31, no. 7, pp. 853-861, July 1983.

23. A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill. 1991.

24. W.R. Bennett. "Statistics of regenerative digital transmission." *Bell Syst. Tech. Journal*. vol. 37. pp. 1501-1542. Nov. 1958.

25. R. Durrett. *Probability: Theory and Examples*. New York: Duxbury Press. 1996.

26. J. Welz. I. Galton. "Necessary and sufficient conditions for mismatch shaping in multi-bit DACs." under review in *IEEE Transactions on Circuits and Systems II —Analog and Digital Signal Processing*.

27. B.H. Marcus. P.H. Siegel. "On codes with spectral nulls at rational submultiples of the symbol frequency." *IEEE Transactions on Information Theory*. vol. IT-33. pp. 557-568. July 1987.

28. G.B. Folland. *Real Analysis: Modern Techniques and Their Applications*. New York: John Wiley and Sons. 1999.

29. G. Thomas. R. Finney. *Calculus and Analytic Geometry*. Massachusetts: Addison-Wesley. 1988.

30. R. Horn. C. Johnson. *Matrix Analysis*. New York: Cambridge University Press. 1985.

Chapter 4

The PSD of the First-Order Tree-Structured DAC in a Second-Order ADC Delta-Sigma Modulator with a Midscale Input

Jared Welz, Ian Galton

Abstract—A popular method of creating a multi-bit digital-to-analog converter (DAC) is to combine several 1-bit DACs in parallel. Ideally, the output of such a DAC is a scaled version of its input; however, static mismatches among its 1-bit DACs cause its output to be a nonlinear function of its input. The resulting error, called *DAC noise*, limits the DAC's attainable signal-to-noise-and-distortion ratio (SINAD) and thus the effective resolution of the DAC. *Mismatch-shaping DACs* mitigate this problem by suppressing the DAC noise power in a frequency band that is inhabited by most of the data signal's power. Most of today's high-performance delta-sigma ($\Delta\Sigma$) data converters employ these DACs along with frequency-selective filters to enhance the DAC's effective resolution. However, little is understood about the DAC noise in mismatch-shaping DACs, especially when the DAC is used in a $\Delta\Sigma$ data converter. Simulations are usually relied upon to estimate the characteristics of the DAC noise, such as the signal-band power. Such simulations can be misleading because the DAC noise depends on the DAC input. This paper presents an analysis of the dithered first-order low-pass tree-structured DAC in a second-order analog $\Delta\Sigma$ modulator with a midscale constant input. Specifically, the analysis develops a theoretical DAC noise power spectral density (PSD) that compares well with behavioral simulations. This $\Delta\Sigma$ modulator was chosen as it has been used in record-setting analog-to-digital converters (ADCs). The midscale constant input was chosen because the DAC noise is the most noticeable in this case. Additionally, simulations and experimental results demonstrated that the DAC noise performance in this case was worse than that for sinusoidal inputs.

I. INTRODUCTION

MULTI-BIT DACs are often constructed by combining several 1-bit DACs in parallel. The multi-bit DAC input is converted to the 1-bit sequences that drive the 1-bit DACs, and the outputs of these DACs are summed to obtain an analog value, which, ideally, is a scaled version of the multi-bit DAC input. However, static mismatches among the 1-bit DACs, which are inevitable in modern VLSI technology, cause the output to be a memoryless, nonlinear function of the input. The signal-dependent portion of the resulting error is modeled, without approximation, as an additive noise source called the *DAC noise*. If not addressed, the DAC noise prohibits the use of this multi-bit DAC in most high-performance applications.

Mismatch-shaping (or *dynamic element matching*) DACs [1]-[6] use spectral shaping techniques to mitigate this problem. A mismatch-shaping DAC exploits redundancy in its 1-bit DACs to shape the PSD of the DAC noise so that most of its power resides outside of the *signal band*—*i.e.*, the range of frequencies that contain most of the data signal's power. Frequency-selective filters are then typically applied to remove most of the DAC noise power while preserve most of the data signal's power. The improved SINAD translates to an increase in the effective resolution of the DAC. The noise shaping techniques employed by mismatch-shaping DACs make them ideal for use in $\Delta\Sigma$ modulators [7]-[9]. Consequently, mismatch-shaping DACs have become enabling components in most of today's high-performance $\Delta\Sigma$ data converters [10]-[16].

To date, the theoretical analyses of mismatch-shaping DACs in literature have been limited, especially for DACs in $\Delta\Sigma$ modulator applications. Most of the mathematical theory developed for mismatch-shaping DACs has been used to show that

the DAC noise in a given architecture has a spectral null at some frequency. Theoretical estimates of the signal-band DAC noise power are usually obtained with a simplified model that assumes the DAC noise is independent of the DAC input. However, the DAC noise depends on the DAC input in all mismatch-shaping DACs, and in most cases, the DAC noise is correlated to the DAC input and thus can contain spurious tones. Although the DAC noise PSD has been determined for trivial inputs to a mismatch-shaping DAC [17], simulations have been relied upon to estimate the behavior of DAC noise in $\Delta\Sigma$ modulator applications, which can be misleading. Moreover, explanations and design rules that accompany such simulations are usually not well substantiated.

This paper presents a theoretical analysis of a first-order tree-structured DAC [6], [15], [16], [18]-[21] in a second-order ADC $\Delta\Sigma$ modulator with a midscale constant input. Specifically, a DAC noise PSD curve is generated using the PSD expression derived in [21] and statistics of the DAC input that are obtained with a $\Delta\Sigma$ modulator model. The constant midscale input was chosen because, without a data signal, the DAC noise is especially conspicuous. Additionally, it was witnessed in both behavioral simulations and circuit tests that the midscale input gave rise to more signal-band DAC noise power than that obtained by any sinusoidal input, which is the typical input used to test the performance of the $\Delta\Sigma$ modulator. Thus, the values of the DAC noise power that are presented in this paper can be used as favorable estimates in the design of the $\Delta\Sigma$ modulator.

This paper is divided into four sections and an Appendix. Section II reviews the operation of the $\Delta\Sigma$ modulator and tree-structured DAC. Section III presents and discusses the DAC noise PSD and signal-band power values and compares them with those obtained in behavioral simulations. Section IV presents the $\Delta\Sigma$ modu-

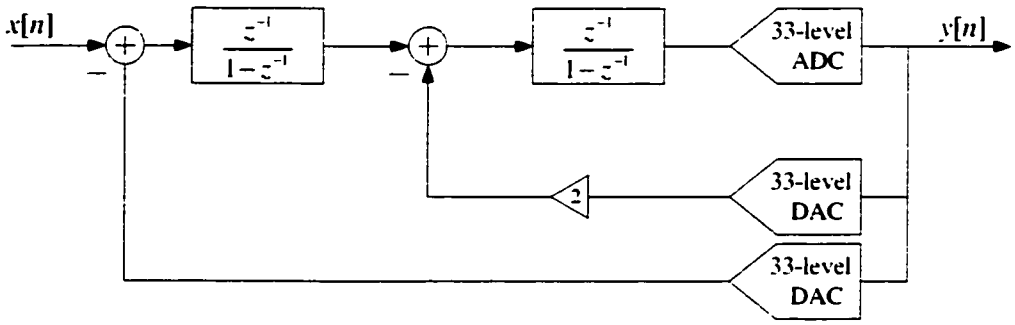


Figure 4.1: A 5-bit, second-order, ADC $\Delta\Sigma$ modulator.

lator model and the statistics that give rise to the DAC noise PSD given in Section III. This section also discusses how the $\Delta\Sigma$ modulator model is used to obtain those statistics. Section IV concludes the paper. The Appendix presents the mathematics that use the $\Delta\Sigma$ modulator model to obtain the DAC noise PSD.

II. THE $\Delta\Sigma$ MODULATOR APPLICATION

THE SECOND-ORDER $\Delta\Sigma$ MODULATOR

Figure 4.1 shows the 33-level, second-order low-pass ADC $\Delta\Sigma$ modulator that is analyzed in this paper. It consists of an ADC, two multi-bit DACs, two delayed accumulators, two subtracters, and a gain element. Two implementations of this $\Delta\Sigma$ modulator are presented in [12] and [15], both of which gave rise to record-setting data converters. The real-valued input sequence, $x[n]$, results from sampling a continuous-time data signal at a rate of 8-times and 64-times its Nyquist rate in the implementations presented in [12] and [15], respectively. The ratio of the sampling rate to the Nyquist rate is called the *oversampling ratio* and is denoted OSR. Therefore, most of the data signal's power is constrained near dc in the interval $(-\pi/OSR, \pi/OSR)$, which is called the *signal band*. The analysis in this paper, however, assumes that the $\Delta\Sigma$ modulator input is the constant midscale value: *i.e.*, $x[n] = 0$.

The $\Delta\Sigma$ modulator is used to obtain high-resolution data conversion by using spectral shaping techniques on the error that results from the coarse ADC. Let the *quantization error*, denoted $\varepsilon[n]$, be the difference between the output and input of the ADC quantizer, and the *quantization noise* be the component of the $\Delta\Sigma$ modulator output that results from the quantization error. If all components are ideal, the transfer function from the input to output of the $\Delta\Sigma$ modulator, called the *signal transfer function*, is given by the Z-transform z^{-2}. On the other hand, the transfer function from the ADC quantizer to the output, called the *noise transfer function*, is given by $\left(1 - z^{-1}\right)^2$. Thus, the $\Delta\Sigma$ modulator acts as an all-pass filter for its input and a second-order high-pass filter for its quantization error. Since the signal band is near dc, most of the power of the quantization noise resides outside of the signal band where it can subsequently be removed by digital filters. However, noise from the DAC in the outside feedback loop of the $\Delta\Sigma$ modulator in Figure 4.1 is injected into the signal path and is not high-pass filtered. Consequently, this DAC noise often limits the attainable SINAD and hence resolution of the $\Delta\Sigma$ modulator.

THE TREE-STRUCTURED DAC

An example 9-level tree-structured DAC is shown in Figure 4.2. In general, the $(2^b + 1)$-level tree-structured DAC consists of a bank of 2^b 1-bit DACs and a *digital encoder*. The output of the *i-th* 1-bit DAC is

$$y_i[n] = \begin{cases} \frac{\Delta_D}{2} + e_{h_i}, & \text{if } x_i[n] \text{ is high;} \\ -\frac{\Delta_D}{2} + e_{l_i}, & \text{if } x_i[n] \text{ is low;} \end{cases} \tag{1}$$

where Δ_D is the nominal step size of the tree-structured DAC, and e_{h_i} and e_{l_i} are the 1-bit DAC's high and low errors, respectively. The 1-bit DAC errors result from inevitable errors that occur in the fabrication of the 1-bit DACs and are assumed to be time-invariant random variables. The digital encoder consists of b layers of

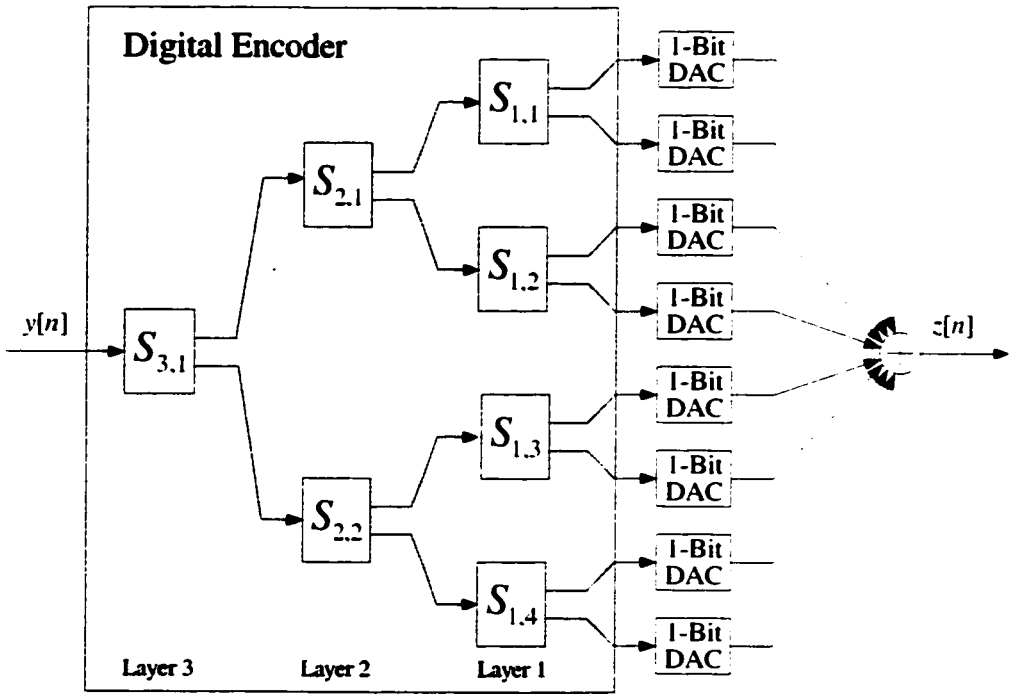


Figure 4.2: A 9-level tree-structured DAC.

switching blocks, which are labeled $S_{k,r}$, where $k = 1, \ldots, b$ is the layer number, and $r = 1, \ldots, 2^{b-k}$, is the depth within the layer. The input to $S_{k,r}$, which is denoted $x_{k,r}[n]$, is constrained to be in the range $\{-2^{k-1}, \ldots, 2^{k-1}\}$. The digital encoder outputs, $x_i[n]$, are 1-bit sequences whose values are taken to be -1/2 at sample times when low and 1/2 at sample times when high. With $x_i[n]$ also denoted $x_{0,i}[n]$, the switching blocks are interconnected so that top and bottom outputs of $S_{k,r}$ are $x_{k-1,2r-1}[n]$ and $x_{k-1,2r}[n]$, respectively.

Figure 4.3 shows the operation of the switching block. As illustrated in the figure, the outputs of $S_{k,r}$ are given by

$$x_{k-1,2r-1}[n] = \tfrac{1}{2}\left(x_{k,r}[n] + s_{k,r}[n]\right). \tag{2}$$

and

$$x_{k-1,2r}[n] = \tfrac{1}{2}\left(x_{k,r}[n] - s_{k,r}[n]\right). \tag{3}$$

where $s_{k,r}[n]$ is called the *switching sequence* and is generated within $S_{k,r}$. Let

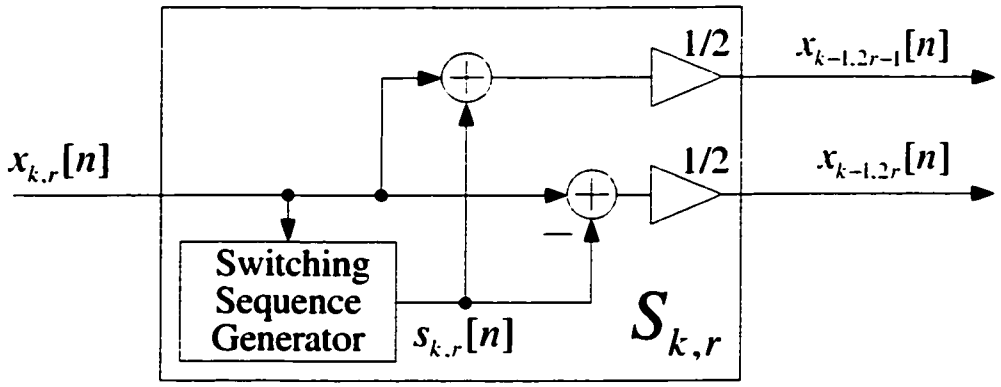


Figure 4.3: The signal processing performed by the switching block.

$o_{k,r}[n]$ be the *parity sequence* of $S_{k,r}$, which is defined to be 1 when $x_{k,r}[n] + 2^{k-1}$ is odd and 0 otherwise. To ensure each switching block output is in the required range, the switching sequence is restricted as follows:

$$s_{k,r}[n] = \begin{cases} \pm 1, & \text{if } o_{k,r}[n] = 1; \\ 0, & \text{if } o_{k,r}[n] = 0. \end{cases} \tag{4}$$

As shown in [6], the DAC output can be written as

$$z[n] = \alpha y[n] + \beta + e[n]. \tag{5}$$

where $y[n]$ is the DAC input, α and β are constants that are functions of the 1-bit DAC errors, and $e[n]$ is the *DAC noise*, which is given by

$$e[n] = \sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} \Delta_{k,r} s_{k,r}[n]. \tag{6}$$

where

$$\Delta_{k,r} = \frac{1}{2^k} \sum_{i=(r-1)2^k+1}^{(r-1)2^k+2^{k-1}} [(e_{h_i} - e_{l_i}) - (e_{h_{i+2^{k-1}}} - e_{l_{i+2^{k-1}}})]. \tag{7}$$

Therefore, (6) implies that the switching sequences can be tailored to manipulate the PSD of the DAC noise.

As detailed in [21], the switching sequences in the dithered first-order low-pass tree-structured DAC are coded to ensure that the DAC noise PSD vanishes at dc and

has no spurious tones. This is accomplished by constructing $s_{k,r}[n]$ by concatenating the following two types of *symbols*:

$$\text{Type 1: } 1.\underbrace{0.\cdots.0.}_{\substack{\text{Until next}\\ o_{k,r}[n]=1}}-1.\underbrace{0.\cdots.0}_{\substack{\text{Until next}\\ o_{k,r}[n]=1}}: \tag{8}$$

and

$$\text{Type 2: } -1.\underbrace{0.\cdots.0.}_{\substack{\text{Until next}\\ o_{k,r}[n]=1}}1.\underbrace{0.\cdots.0}_{\substack{\text{Until next}\\ o_{k,r}[n]=1}}. \tag{9}$$

The choice of each symbol type is made "randomly" using the 1-bit dither sequence $d_{k,r}[n]$. The dither sequence approximates a sequence of independent and identically distributed (i.i.d.) bits that have a uniform distribution and are also independent of $x_{k,r}[n]$. If a symbol starts in $s_{k,r}[n]$ at sample time n_0, then that symbol is a Type 1 symbol if $d_{k,r}[n_0]$ is high, and it is a Type 2 symbol if $d_{k,r}[n_0]$ is low. For the implementation analyzed in this paper and presented in [15], all switching blocks in a given layer share the same dither sequence. Thus, b dither sequences are required in this case, which are realized with pseudorandom sequence generators.

III. THE DAC NOISE PSD

As shown in [21], the DAC noise PSD is a function of the statistics of the switching sequence symbols. These statistics are expressed using the definitions presented next. Let the first *half*—*i.e.*, the first $\pm 1.0.\ldots.0$ segment of a symbol—be called the *head* of the symbol, and the second half be called the *tail* of a symbol. The *head length* of a symbol denotes the number of samples in the head of that symbol. Let $H_{k,r}$ be the *head-length process* for $s_{k,r}[n]$; thus, $H_{k,r}[m]$ is the number of samples in the head of the *m-th* symbol in $s_{k,r}[n]$. The definitions of the *tail length* and *tail-length process*, $T_{k,r}$, follow analogously.

The DAC noise PSD and signal-band power expressions used in this section require the following two assumptions:

1. The switching sequence cross-spectra are negligible:
2. The head-length distributions and variances for switching sequences in the same layer are the same: *i.e.*, $\sigma^2_{k,r_1} = \sigma^2_{k,r_2}$, and $H_{k,r_1}[m]$ and $H_{k,r_2}[m]$ have the same distribution for each $r_1,\ r_2 = 1, \ldots, 2^{b-k}$.

These assumptions are justified later in this section. Given both assumptions hold, $H_k[m] \equiv H_{k,1}[m]$, and $\sigma^2_k \equiv \sigma^2_{k,1}$, it follows from [21] that the DAC noise PSD can be written as

$$\hat{D}\left(e^{j\omega}\right) \equiv \sum_{k=1}^{b} \Lambda_k S_k\left(e^{j\omega}\right). \tag{10}$$

where $S_k\left(e^{j\omega}\right)$ is the layer-k switching sequence PSD given by

$$S_k\left(e^{j\omega}\right) = 2\sigma^2_k E\left\{\sin^2\left(\frac{\omega H_k[m]}{2}\right)\right\}. \tag{11}$$

and

$$\Lambda_k \equiv \sum_{r=1}^{2^{b-k}} \Delta^2_{k,r}. \tag{12}$$

Furthermore, the DAC noise signal-band power can be written as

$$\hat{D}_{OSR} \equiv \sum_{k=1}^{b} \Lambda_k P_k\left(OSR\right). \tag{13}$$

where $P_k\left(OSR\right)$ is the layer-k switching sequence signal-band power as given by

$$P_k(OSR) = \frac{\sigma^2_k E\left\{1 - \mathrm{sinc}\left(\frac{H_k[m]}{OSR}\right)\right\}}{OSR}. \tag{14}$$

It follows from (10) and (13) that the DAC noise PSD and signal-band power are linear combinations of the switching sequence PSDs and signal-band powers, respectively. By examining these PSDs and signal-band powers, some insight can be gained into the behavior of the DAC noise.

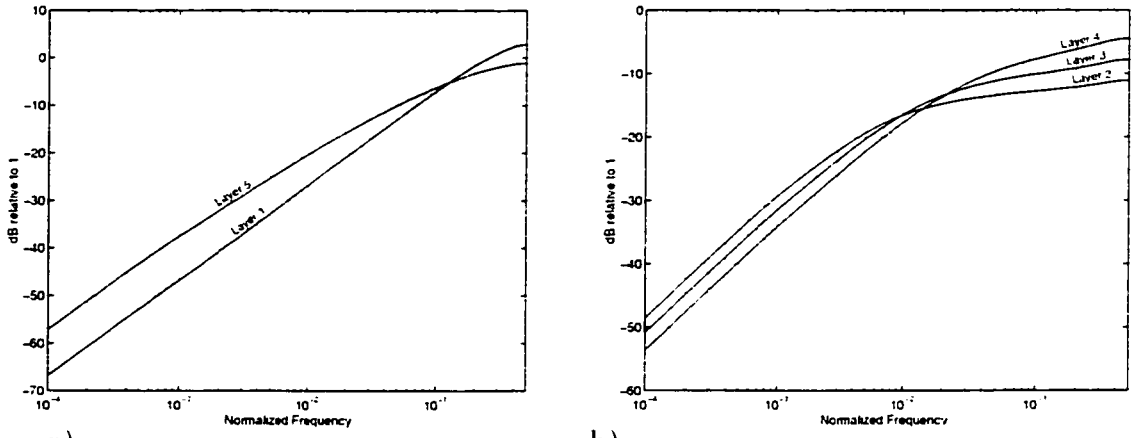


Figure 4.4: The switching sequence PSDs for a) layers 1 and 5, and b) layers 2, 3 and 4 as estimated using the $\Delta\Sigma$ modulator model.

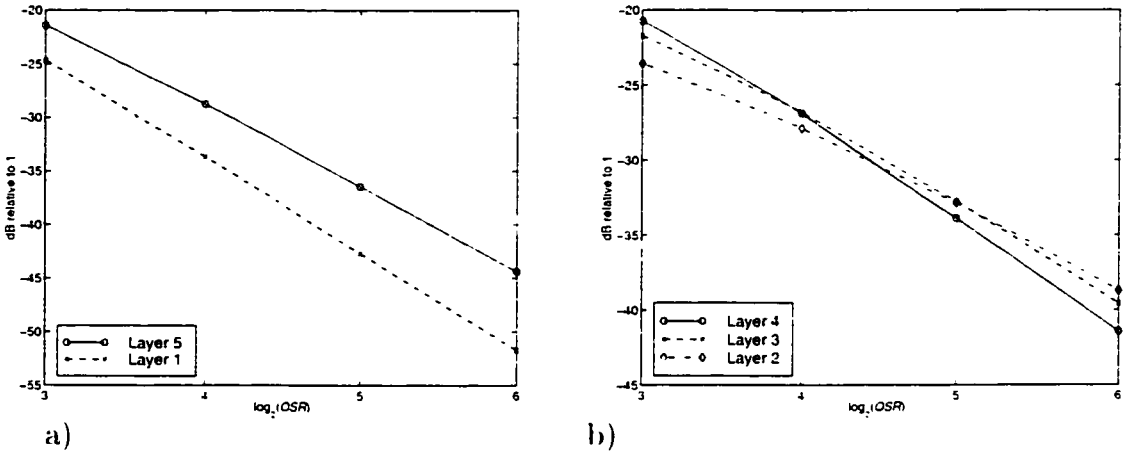


Figure 4.5: The switching sequence signal-band power for a) layers 1 and 5, and b) layers 2, 3 and 4 as estimated using the $\Delta\Sigma$ modulator model.

Shown in Figures 4.4 and 4.5 are the theoretical switching sequence PSDs and signal-band powers, respectively, for a 33-level DAC in the $\Delta\Sigma$ modulator in Figure 4.1. These results are obtained using the switching sequence statistics provided in the next section. As shown in Figure 4.4, all switching sequence PSDs have the same "high-pass" shape, but with different 3dB bandwidths. This gives rise to the varying signal-band powers shown in Figure 4.5.

Design guidelines can be extrapolated from these figures. If the goal is to minimize the DAC noise power for a small signal input to the $\Delta\Sigma$ modulator,

Figure 4.5 can be used to decide how the 1-bit DACs should be layed out. For example, suppose the oversampling ratio is 64 (as in the $\Delta\Sigma$ modulator presented in [15]). Figure 4.5 shows that the switching sequences in layer 2 have the most signal-band power. Therefore, to minimize these switching sequences' contribution to the DAC noise power, (13) implies that the 1-bit DACs should be layed out so that Λ_2 is minimized. Given (7) and (12), this is accomplished by optimizing the matching between the *i-th* and $(i+2)$-*nd* 1-bit DACs for $i = 1, \ldots, 2^b - 2$. Typically, this is achieved by placing these 1-bit DACs as close as possible to each other or, if possible, interlacing the components of these 1-bit DACs on the integrated circuit.

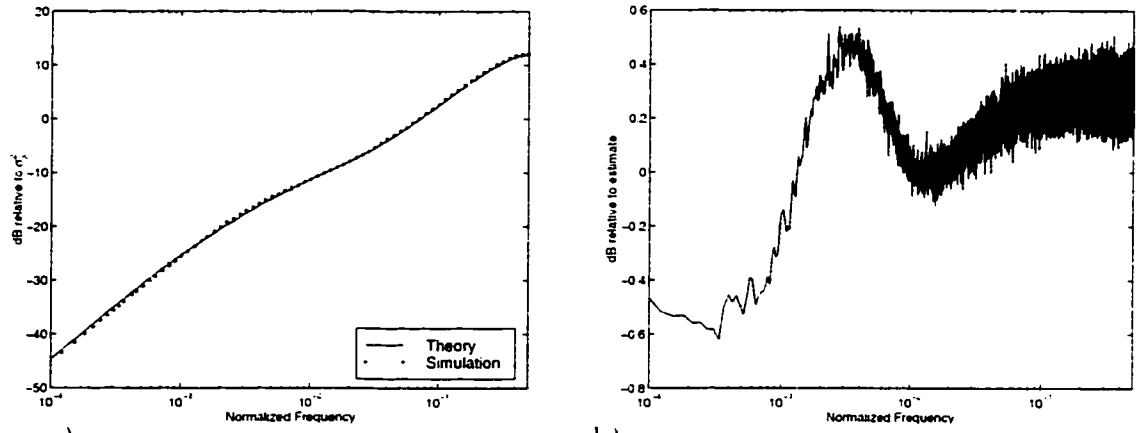
In [19], DAC noise PSDs from theory and simulation are compared for a specific collection of 1-bit DAC errors. However, in this paper, the *average* DAC noise PSDs are compared—*i.e.*, average with respect to 1-bit DAC errors. The average DAC noise PSD and signal-band power can be calculated by assuming statistics for the 1-bit DAC *step-size errors*: $e_{h_i} - e_{l_i}$ for $i = 1, \ldots, 2^b$. If the step-size errors are taken to be i.i.d. random variables with standard deviations denoted σ_δ, it follows from (7) and (12) that $E\{\Lambda_k\} = 2^b\sigma_\delta^2/4^k$. Substituting this into (10) gives the following average DAC noise PSD:

$$\bar{D}\left(e^{j\omega}\right) \equiv 2^b\sigma_\delta^2 \sum_{k=1}^{b} \frac{S_k\left(e^{j\omega}\right)}{4^k}. \tag{15}$$

and substituting it into and (13) gives the following average DAC noise signal-band power:

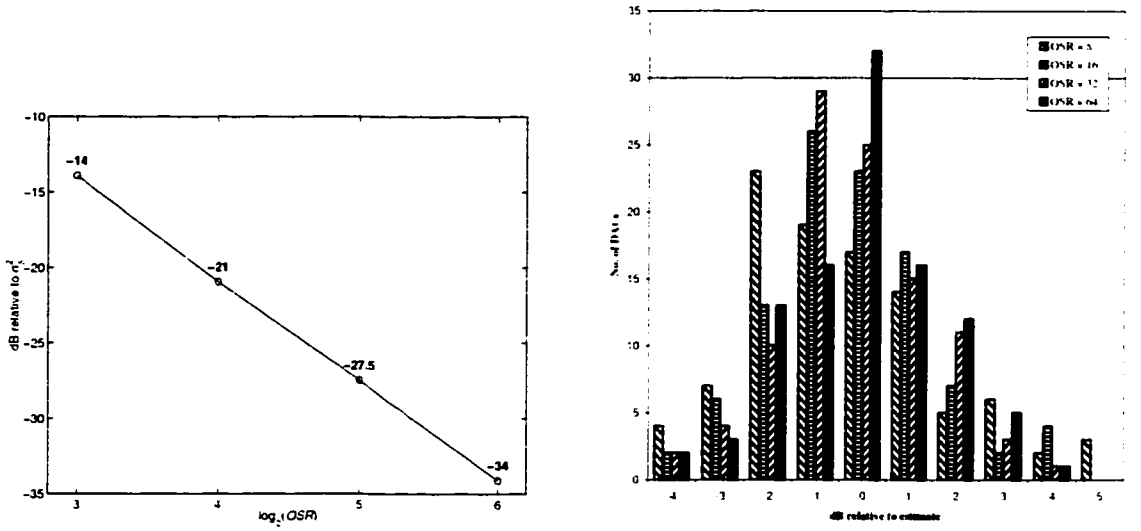
$$\bar{D}_{OSR} \equiv 2^b\sigma_\delta^2 \sum_{k=1}^{b} \frac{P_k\left(OSR\right)}{4^k}. \tag{16}$$

Figures 4.6 and 4.7 show and compare the average DAC noise PSDs and signal-band powers, respectively, obtained from theory and simulation of the 33-level DAC in the $\Delta\Sigma$ modulator in Figure 4.1. One hundred simulations of the $\Delta\Sigma$ modulator were performed, each with a different pair of mismatch-shaping DACs. The 1-bit



a) b)

Figure 4.6: The a) average DAC noise PSD from simulation and theory, and b) the dB difference between the average DAC noise PSD from simulation and theory.



a) b)

Figure 4.7: The a) average DAC noise signal-band powers from theory and b) histograms showing the distribution of the DAC noise signal-band powers relative to those from theory.

DAC errors for each simulation were modeled as independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and standard deviations of 0.3% of $\Delta_D/2$. This corresponds to reasonable matching precision by the standards of present-day switched-capacitor CMOS circuit technology. The input to the $\Delta\Sigma$ modulator was an i.i.d. sequence of Gaussian random variables with zero mean and standard deviations of 0.1% of the nominal step size of the ADC. This

sequence models the front-end kT/C analog noise in the ADC $\Delta\Sigma$ modulator presented in [15]. All other components of the $\Delta\Sigma$ modulator were modeled as ideal. The average DAC noise PSD from these simulations resulted by averaging the one hundred DAC noise PSDs obtained in these simulations.

Figure 4.6 shows how well the average DAC noise PSD from theory matches that from simulation. However, this does not illustrate how the DAC noise PSD in each simulation varied from the theoretical estimate. To accomplish this, Figure 4.7 provides histograms that show the distribution of the DAC noise signal-band powers, relative to those from theory, obtained in the one-hundred simulations.

The average DAC noise PSD is obtained by assuming statistics concerning the 1-bit DAC step-size errors that cannot be justified. As discussed in [22]-[24], the mismatches among devices on an integrated circuit (IC) typically exhibit correlation. Moreover, there can be correlation among the 1-bit step size errors in different realizations of the multi-bit DAC unless the 1-bit DACs are "shuffled" – *i.e.*, their placement on the IC randomized – between realizations, which is not practical.

However, the first two assumptions provided at the beginning of this section are reasonable. The first assumption is justified by the fact that, in the given $\Delta\Sigma$ modulator with a midscale input, the DAC input, $y[n]$, is usually in the range $\{-1, 0, 1\}$. Since an independent dither sequence is used in each layer, only switching sequences in the same layer can be correlated. Moreover, correlation occurs only when symbols start at the same sample time in switching sequences in the same layer. However, when $y[n] \in \{-1, 0, 1\}$, switching sequences in layers $k > 1$ are uncorrelated because at most one switching sequence in each layer is nonzero. For layer 1 in this case, at most one switching sequence is *zero*. However, because switching sequences in this layer are nonzero so often, their head lengths are typically small and consequently,

their cross spectra, as detailed in [21], contributes little or nothing to the DAC noise signal-band power.

The second assumption is justified by the symmetry of the tree-structured DAC and the behavior of the DAC input. The symmetry of the dither sequence statistics, switching sequence symbol types (as given in (8) and (9)), and switching block input/output expressions (as given in (2) and (3)) imply that the tree-structured DAC does nothing to differentiate between switching block inputs, and thus, switching sequences in the same layer. Moreover, the DAC input, in the case of the $\Delta\Sigma$ modulator in Figure 4.1, consists mostly of random quantization noise that does little or nothing to differentiate between switching sequences in the same layer.

IV. THE SWITCHING SEQUENCE STATISTICS

As shown in the previous section, the DAC noise PSD depends on the head-length distributions and variances of the switching sequences. This section presents an estimate of these statistics using a simplified model of the $\Delta\Sigma$ modulator in Figure 4.1. This model and its assumptions are discussed next. The switching sequence statistics are then presented with a description of how they are derived.

The development of the $\Delta\Sigma$ modulator model is motivated by the complexities of an ADC $\Delta\Sigma$ modulator and the switching sequences' dependence on the $\Delta\Sigma$ modulator output. It follows from (8) and (9) that a head of length h occurs in $s_{k,r}[n]$ when there is a run of $h-1$ zeros $o_{k,r}[n]$. Thus, the head-length distribution for symbols in $s_{k,r}[n]$ depends on the multivariate distribution of the parity sequence, $o_{k,r}[n]$. Furthermore, the multivariate distribution of $o_{k,r}[n]$ depends on that of the DAC input, which is the $\Delta\Sigma$ modulator output. Given the $\Delta\Sigma$ modulator has a constant midscale input, its output consists mainly of the quantization noise, but it also includes noise from both DACs and the analog circuits. These noise sources

excessively complicate the multivariate characteristics of the $\Delta\Sigma$ modulator output and thus the parity sequences. However, a sufficiently accurate estimate of the switching sequence statistics can be obtained by excluding these noise sources.

Based on the $\Delta\Sigma$ modulator in Figure 4.1, the $\Delta\Sigma$ modulator model consists of the following assumptions:

1. All components of the $\Delta\Sigma$ modulator and its input are ideal, and there is no analog circuit noise;

2. The quantization error, $\varepsilon[n]$, is pairwise independent and uniformly distributed across the LSB interval, $(-\Delta_A/2, \Delta_A/2)$, where Δ_A is the ADC step size.

The first assumption is justified by the fact that, because the ADC quantizer is coarse in a $\Delta\Sigma$ modulator, the quantization error typically has much more total power, across all frequencies, than any other noise source in the $\Delta\Sigma$ modulator. The second assumption is justified by the analyses in [7]-[9] where it is shown that the quantization error in a second-order ADC $\Delta\Sigma$ modulator asymptotically has these properties. In [7] and [8], this result is a consequence of the inevitable i.i.d. front-end analog circuit noise in this ADC $\Delta\Sigma$ modulator, whereas in [9], it is a consequence of the inevitable irrational dc offset at the input to this ADC $\Delta\Sigma$ modulator. Thus, the $\Delta\Sigma$ modulator model is contradictory because the second assumption requires characteristics of the ADC $\Delta\Sigma$ modulator that are voided by the first assumption. Regardless, it gives rise to the following switching sequence statistics that, as shown later in this section, match well with those obtained in behavioral simulations.

Theorem 1: Given the $\Delta\Sigma$ modulator model output is the input to the tree-

structured DAC. the switching sequence variance is

$$\sigma_k^2 = \begin{cases} 1 - \left(\frac{1}{2}\right)^b. & \text{for } k = 1: \\ \left(\frac{1}{2}\right)^{b-k+1}. & \text{otherwise:} \end{cases} \tag{17}$$

and the probability that a head length is h samples is

$$P\left(H_k[m] = h\right) = \begin{cases} 2^{b-k+2} P_k\left(h, 0\right). & \text{for } k > 1: \\ 1 - \frac{2^{b-1}}{2^b-1}\left(P\left(1\right) - P\left(2\right)\right). & \text{for } k = 1,\ h = 1: \\ \frac{2^{b+1}}{2^b-1}\left(P\left(h-1\right) - 2P\left(h\right)\right). & \text{otherwise:} \end{cases} \tag{18}$$

where

$$P\left(l\right) = \begin{cases} \frac{1}{4l}\left(\frac{1}{2}\left(\left(\frac{1}{2}\right)^{\left\lfloor\frac{l+1}{2}\right\rfloor} + \left(\frac{1}{2}\right)^{\left\lceil\frac{l+1}{2}\right\rceil}\right)\right)^{b-1}. & \text{for } l \text{ odd:} \\ \frac{l}{4(l-1)(l+1)}\left(\frac{1}{2}\left(\left(\frac{1}{2}\right)^{\left\lfloor\frac{l+1}{2}\right\rfloor} + \left(\frac{1}{2}\right)^{\left\lceil\frac{l+1}{2}\right\rceil}\right)\right)^{b-1}. & \text{for } l \text{ even:} \end{cases} \tag{19}$$

and $P_k\left(h, m\right)$ is the function that satisfies

$$P_{k-1}\left(h, m\right) = \sum_{i=m}^{\left\lfloor\frac{h-1}{2}\right\rfloor} \left(\frac{1}{2}\right)^{i+3} \left(\binom{i}{m} + 2\binom{i-1}{m-1}\right) P_k\left(h, i\right). \tag{20}$$

where $\binom{a}{b}$ is the combination function with $\binom{-1}{-1}$ *defined* to be 1. and

$$P_b\left(h, m\right) = \begin{cases} 0. & m > \frac{h-1}{2}: \\ \frac{(m+1)^2}{h(h+1)(h+2)} + \frac{m^2}{h(h-1)(h-2)}. & h > 2,\ m < \frac{h-1}{2}: \\ \frac{(m+1)^2}{h(h+1)(h+2)}. & \text{otherwise.} \end{cases} \tag{21}$$

Proof: The proof is provided in the Appendix

■

Although Theorem 1 does not provide closed form expressions for most of the head-length probabilities. the recursive expressions provided can be evaluated using programming software. Using such techniques. several of the head-length probabilities from Theorem 1 are shown in Figure 4.8. This figure includes the first 15 head-length probabilities for switching sequences in the 33-level tree-structured DAC. For

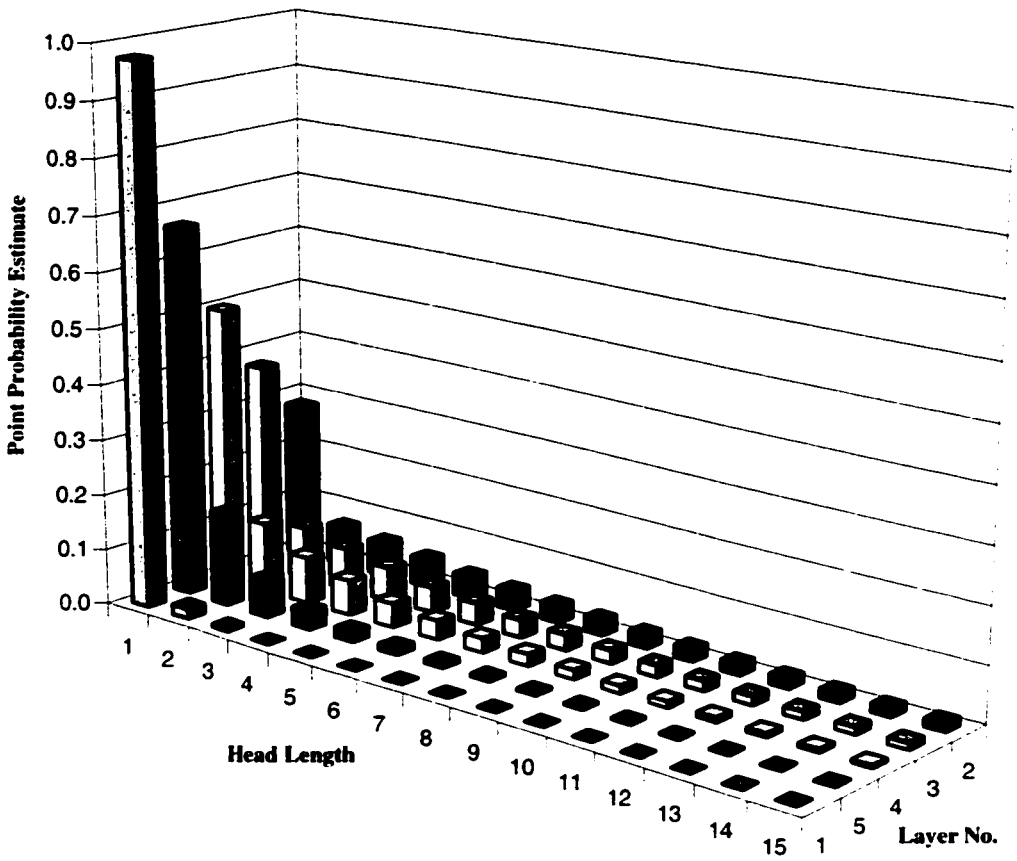
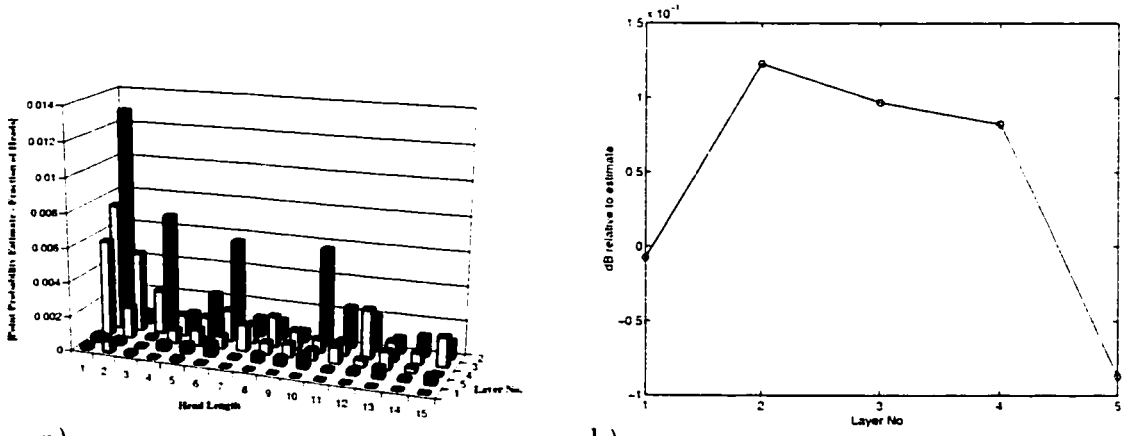


Figure 4.8: Head-length probabilities for lengths 1 to 15 estimated using the $\Delta\Sigma$ modulator model.



a) b)

Figure 4.9: The switching sequence time-averaged statistics obtained from behavioral simulations: a) the magnitude difference between the head-length probabilities from behavioral simulations and those estimated using the $\Delta\Sigma$ modulator model and b) switching sequence variances relative to the respective variances estimated using the $\Delta\Sigma$ modulator model.

the remaining values. each probability point function continues to asymptotically

decrease towards zero, the rate of which depends on the layer number. With the exception of the first layer, the smaller the layer number, the slower the head-length probability function decays to zero. Figure 4.9 shows how these probabilities and the switching sequence variances in (17) vary with those obtained with from the behavioral simulations described in the previous section. It follows from Figure 4.9a that, with the exception of the first layer, the difference between head-length probabilities obtained by theory and simulation becomes more pronounced as the layer number is decreased. On the other hand, the variances from each match well regardless of the layer number as shown in Figure 4.9b.

The remainder of this section describes how the $\Delta\Sigma$ modulator model is used to obtain the variances and head-length distributions given in Theorem 1. It is assumed hereafter that the input to the tree-structured DAC is the output of the $\Delta\Sigma$ modulator model, and, for convenience, the ADC step size is 1 (*i.e.*, $\Delta_A = 1$). The next theorem shows how the quantization error dictates the behavior of the $\Delta\Sigma$ modulator model.

Theorem 2: For any integer m, the quantization error can be written as,

$$\varepsilon[n+m] = \tfrac{1}{2} - \langle m\,(\varepsilon[n-1] - \varepsilon[n]) + \tfrac{1}{2} - \varepsilon[n] \rangle\,. \tag{22}$$

where $\langle\cdot\rangle$ is the modulo-1 operator.

Proof: The proof is provided in the Appendix.

■

Thus, (22) implies that two samples of quantization error completely determine the entire sequence, and since

$$y[n] = \varepsilon[n] - 2\varepsilon[n-1] + \varepsilon[n-2]. \tag{23}$$

two samples of quantization error completely determine the $\Delta\Sigma$ modulator output. The statistics of two samples of quantization error are given by the second assumption of the $\Delta\Sigma$ modulator model. Therefore, these statistics, (22), and (23) can be used to determine the multivariate distribution of $y[n]$. However, it is not necessary to obtain the complete multivariate distribution of $y[n]$: it is only necessary to obtain probability functions that can be used to derive the switching sequence statistics.

Such probability functions are derived by exploiting the characteristics of the switching block inputs as given in the following lemma.

Lemma 1: Each switching block input, $x_{k,r}[n]$, has the following properties:

a) Range Property: $x_{k,r}[n]$ is restricted to the set $\{-1, 0, 1\}$ for all n;

b) Alternating Property: the nonzero samples of $x_{k,r}[n]$ alternate between 1 and -1;

c) Symmetry Property: given $\vec{x}_{k,r}[n]$ is the M-length vector whose components are given by $x_{k,r}[n], \ldots, x_{k,r}[n+M-1]$.

$$P\left(\vec{x}_{k,r}[n] = \vec{x}_M\right) = P\left(\vec{x}_{k,r}[n] = -\vec{x}_M\right). \tag{24}$$

where $\vec{x}_M$ is a real-valued M-length vector.

Proof: The proof is provided in the Appendix.

■

As discussed next, Lemma 1 implies that the switching sequence variances and head-length probabilities can be obtained with the following two probability functions:

$$P_k(l) \equiv P\left(x_{k,r}[n+i] = (-1)^i \text{ for } i = 0, \ldots, l-1\right). \tag{25}$$

and

$$P_k(h,m) \equiv P(x_{k,r}[n]=1,\ x_{k,r}[n+h]=-1,\ \exists\ m \text{ values of } i \in \{1,\ldots,h-1\} \text{ such that } x_{k,r}[n+i]=1). \tag{26}$$

These probability functions were chosen because the operation of the switching block enables $P_{k-1}(h,m)$ and $P_{k-1}(l)$ to be determined as functions of $P_k(h,m)$ and $P_k(l)$, respectively.

Recall that the parity sequence $o_{k,r}[n]$ determines the magnitude of $s_{k,r}[n]$ and hence its symbols' head lengths. Given the Restriction Property in Lemma 1 and the definition of $o_{k,r}[n]$,

$$o_{k,r}[n] = \begin{cases} |x_{k,r}[n]|, & \text{for } k>1: \\ 1-|x_{k,r}[n]|, & \text{for } k=1: \end{cases} \tag{27}$$

Furthermore, the Symmetry Property implies that $P(|x_{k,r}[n]|=1) = 2P_k(1)$. Therefore, this and (27) imply that the switching sequence variance, σ_k^2, can be determined using $P_k(l)$ as follows

$$\sigma_k^2 = P(o_{k,r}[n]=1) = \begin{cases} P(|x_{k,r}[n]|=1) = 2P_k(1), & \text{for } k>1: \\ P(x_{k,r}[n]=0) = 1-2P_k(1), & \text{for } k=1. \end{cases} \tag{28}$$

Given Lemma 1 and (27), the head-length probability functions are the following conditional probabilities:

$$P(H_k[m]=h) = \begin{cases} \frac{P(|x_{k,r}[n]|=1;\ x_{k,r}[n+i]=0 \text{ for } i=1,\ldots,h-1;\ |x_{k,r}[n+h]|=1)}{P(|x_{k,r}[n]|=1)}, & \text{for } k>1: \\ \frac{P(x_{k,r}[n]=0;\ |x_{k,r}[n+i]|=1 \text{ for } i=1,\ldots,h-1;\ x_{k,r}[n+h]=0)}{P(x_{k,r}[n]=0)}, & \text{for } k=1: \end{cases} \tag{29}$$

where, for $h=1$, both $P(|x_{k,r}[n+i]|=1 \text{ for } i=1,\ldots,h-1)$ and $P(x_{k,r}[n+i]=0 \text{ for } i=1,\ldots,h-1)$ are *defined* to be one. As shown in the proof of Theorem 1 in the Appendix, the properties given by Lemma 1 and the Law of Total Probability are used to determine the following layer-1 head-length probabilities:

$$P(H_1[m]=h) = \begin{cases} \frac{2(P_1(h-1)-2P_1(h))}{1-2P_1(1)}, & \text{for } h>1: \\ 1-\frac{2(P_1(1)-P_1(2))}{1-2P_1(1)}, & \text{for } h=1: \end{cases} \tag{30}$$

The Alternating Property and (26) imply that

$$P_k(h,0) = P\left(x_{k,r}[n] = 1:\ x_{k,r}[n+i] = 0 \text{ for } i = 1,\ldots,h-1:\ x_{k,r}[n+h] = -1\right). \tag{31}$$

because, if $x_{k,r}[n+i] = -1$ for some $i = 1,\ldots,h-1$, then there must be another value of i in this range such that $x_{k,r}[n+i] = 1$. Thus, for $k > 1$, it follows from the Symmetry Property, (25), (26), (29), and (31) that

$$P(H_k[m] = h) = \frac{P_k(h,0)}{P_k(1)}. \tag{32}$$

The derivations of the two probability functions are left to the Appendix, but the following is a brief description of the steps taken in these derivations. First, the initial values of these two probability functions, *i.e.*, $P_h(l)$ and $P_h(h,m)$, are determined by the statistics of the $\Delta\Sigma$ modulator model output. The switching block's operation is then used to determine $P_{k-1}(l)$ as a function of $P_k(l)$ and $P_{k-1}(h,m)$ as a function of $P_k(h,i)$ for $i \geq m$. A closed form expression is given for $P_1(l) \equiv P(l)$ in (19), and $P_k(1)$ is evaluated to give (17). However, a closed form expression was not obtained for $P_k(h,m)$: it is expressed using the difference equations in (20) with the initial conditions given in (21).

V. CONCLUSION

This paper has presented the analysis of the DAC noise PSD and signal-band power for a tree-structured DAC in a second-order ADC $\Delta\Sigma$ modulator with a midscale constant input. Specifically, this paper has provided the first theoretical DAC noise PSD for a mismatch-shaping DAC in a $\Delta\Sigma$ modulator application. A simplified model for the $\Delta\Sigma$ modulator has been constructed to obtain the switching sequence statistics that are required to produce the DAC noise PSD. With these

statistics, a theoretical DAC noise PSD curve has been produced and shown to compare well that obtained in behavioral simulations.

APPENDIX

This appendix provides the mathematics that form the basis of this paper. Before the results are given, some essential definitions and assumptions are provided. First, it is assumed throughout that the described $\Delta\Sigma$ modulator model is driving the dithered, $(2^b + 1)$-level, first-order low-pass tree-structured DAC. Let $z_{k,r}[n]$ represent the state of $s_{k,r}[n]$ as follows

$$z_{k,r}[n] = \begin{cases} 1. & \text{if } s_{k,r}[n] \text{ is in the head of a symbol:} \\ -1. & \text{otherwise.} \end{cases} \tag{33}$$

Assume that, at sample time n_0, $\{z_{k,r}[n_0] : k = 1, \ldots, b;\ r = 1, \ldots, 2^{b-k}\}$ is a collection of uniform, independent random variables. This assumption represents the uncertainty in the switching sequence states that results from driving the DAC with a noisy input for sample times prior to n_0.

Let $R_n[m]$ be the *double sum* of the $\Delta\Sigma$ modulator output as follows

$$R_n[m] \equiv \sum_{l=n}^{m+n-1} \sum_{i=n}^{l} y[i]. \tag{34}$$

where $R_n[0]$ is *defined* to be zero. Finally, let $\vec{y}[n]$ be the M-length vector whose components are given by $y[n], \ldots, y[n + M - 1]$. Define the vectors $\vec{x}_{k,r}[n]$, $\vec{s}_{k,r}[n]$, and $\vec{R}_n[m]$ analogously. It is tacitly assumed throughout that the quantization error is strictly bounded in magnitude by 1/2 (*i.e.*, $|\varepsilon[n]| < 1/2$) because the event $|\varepsilon[n]| = 1/2$ occurs with zero probability under the assumptions of the $\Delta\Sigma$ modulator model. Note that, because Theorem 1 depends on Theorem 2, it is presented after Theorem 2 in this Appendix.

Theorem 2: See Section IV for the theorem statement.

Proof: First, apply mathematical induction for $m \geq -1$. For $m = -1$ and 0, (22) is satisfied because the modulo-1 of any number between 0 and 1 is simply that number. Suppose (22) holds for all $m \leq m_0$, where $m_0 \geq 0$, and let $m = m_0 + 1$. Because there are no noise sources besides the quantization error, and the input is a midscale constant, the $\Delta\Sigma$ modulator output is the quantization noise:

$$y[n] = \varepsilon[n] - 2\varepsilon[n-1] + \varepsilon[n-2]. \tag{35}$$

Since $\langle y[n] \rangle = 0$, (35) implies that

$$\varepsilon[n] = \tfrac{1}{2} - \left\langle -2\varepsilon[n-1] + \varepsilon[n-2] + \tfrac{1}{2} \right\rangle. \tag{36}$$

Upon evaluating (36) at sample time $n+m$, it follows that

$$\varepsilon[n+m] = \tfrac{1}{2} - \left\langle -2\varepsilon[n+m-1] + \varepsilon[n+m-2] + \tfrac{1}{2} \right\rangle. \tag{37}$$

Under the induction hypothesis, $\varepsilon[n+m-1]$ and $\varepsilon[n+m-2]$ can be evaluated using (22), which, upon substituting this into (37) gives

$$\begin{aligned} \varepsilon[n+m] = \tfrac{1}{2} - \Big\langle 2\left\langle (m-1)\left(\varepsilon[n-1] - \varepsilon[n]\right) + \tfrac{1}{2} - \varepsilon[n] \right\rangle \\ - \left\langle (m-2)\left(\varepsilon[n-1] - \varepsilon[n]\right) + \tfrac{1}{2} - \varepsilon[n] \right\rangle \Big\rangle. \end{aligned} \tag{38}$$

Removing the modulo-1 operators within the argument of the "larger" modulo-1 operator in (38) gives

$$\varepsilon[n+m] = \tfrac{1}{2} - \Big\langle 2(m-1)\left(\varepsilon[n-1] - \varepsilon[n]\right) - 2\varepsilon[n] - (m-2)\left(\varepsilon[n-1] - \varepsilon[n]\right) - \tfrac{1}{2} + \varepsilon[n] \Big\rangle. \tag{39}$$

Simplifying (39) gives (22), which implies, by mathematical induction, that (22) holds for all $m \geq -1$.

Now, let $m \leq 0$. First, upon evaluation, (22) is satisfied with $m = 0$ and -1. To apply mathematical induction, suppose (22) is satisfied for all $m \leq m_0$, and let $m = m_0 - 1$. Since $\langle y[n] \rangle = 0$, (35) implies

$$\varepsilon[n-2] = \tfrac{1}{2} - \langle -2\varepsilon[n-1] + \varepsilon[n] + \tfrac{1}{2} \rangle. \tag{40}$$

Evaluating the above expression at sample time $n+m+2$ gives

$$\varepsilon[n+m] = \tfrac{1}{2} - \langle -2\varepsilon[n+m+1] + \varepsilon[n+m+2] + \tfrac{1}{2} \rangle. \tag{41}$$

Using the induction hypothesis, $\varepsilon[n+m+1]$ and $\varepsilon[n+m+2]$ can be evaluated as functions of $\varepsilon[n]$ and $\varepsilon[n-1]$ using (22). Upon performing this substitution and simplifying as in the previous induction proof, (22) follows and thus holds for all m by mathematical induction.

■

Corollary A1:

$$R_n[m] = \Big\lfloor m\left(\varepsilon[n-2] - \varepsilon[n-1]\right) + \tfrac{1}{2} - \varepsilon[n-1] \Big\rfloor. \tag{42}$$

where $\lfloor \cdot \rfloor$ is the function that rounds down to the nearest integer.

Proof: Applying (35) to the definition of $R_n[m]$ gives

$$R_n[m] = \varepsilon[m+n-1] - \varepsilon[n-1] + m\left(\varepsilon[n-2] - \varepsilon[n-1]\right). \tag{43}$$

From (22), $\varepsilon[m+n-1]$ can be written as

$$\varepsilon[m+n-1] = \tfrac{1}{2} - \big\langle m\left(\varepsilon[n-2] - \varepsilon[n-1]\right) + \tfrac{1}{2} - \varepsilon[n-1] \big\rangle. \tag{44}$$

Substituting (44) into (43) and applying the identity $\lfloor x \rfloor = x - \langle x \rangle$ validates (42).

■

Lemma 1: See Section IV for the lemma statement

Proof: To improve this proof's readability, it is divided into four claims. The first two apply to the $\Delta\Sigma$ modulator output, $y[n] \equiv x_{b,1}[n]$, and the last two apply to switching sequence inputs in the other layers.

Claim 1: The $\Delta\Sigma$ modulator output satisfies both the Range and Alternating Properties.

Proof: Since $|\varepsilon[n-2]-\varepsilon[n-1]|<1$, and $0<1/2-\varepsilon[n-1]<1$, it follows from (42) that $|R_n[m]|\leq m$. Since $y[n]=R_n[1]$, this implies that $y[n]$ has the Range Property. Suppose, at sample time n_0, $y[n_0]=\pm 1$. Let n_0+m $(m>0)$ be the next sample time where $y[n]$ is nonzero. This and (34) imply that $R_{n_0}[m+1]=\pm(m+1)+y[n_0+m]$. Since $|y[n_0+m]|=1$ and $|R_{n_0}[m+1]|\leq m+1$, this implies that $y[n_0+m]=\mp 1$, and $y[n]$ has the Alternating Property.

■

Claim 2: The $\Delta\Sigma$ modulator output satisfies the Symmetry Property: *i.e.*, given an M-length vector $\vec{y}_M$,

$$P\left(\vec{y}[n]=\vec{y}_M\right)=P\left(\vec{y}[n]=-\vec{y}_M\right). \tag{45}$$

Proof: Given $\vec{y}_M$ whose elements are denoted $y_1,\ldots,y_M$, let $R_m\equiv\sum_{l=1}^{m}\sum_{i=1}^{l}y_i$, and let $\vec{R}_M$ be the M-length vector whose elements are $R_1,\ldots,R_M$. Therefore, $\vec{y}[n]=\vec{y}_M$ if and only if $\vec{R}_n[m]=\vec{R}_M$, and to prove (45), it is sufficient to show that

$$P\left(\vec{R}_n[m]=\vec{R}_M\right)=P\left(\vec{R}_n[m]=-\vec{R}_M\right). \tag{46}$$

By assumption, $\varepsilon[n-1]=\varepsilon_1$ and $\varepsilon[n-2]=\varepsilon_2$, where ε_1 and ε_2 are independent random variables that are uniformly distributed across the interval $(-1/2,1/2)$. From (42), ε_1 and ε_2 determine the values of $R_n[m]$ for all $m\geq 0$. To better represent the dependence of $R_n[m]$ on ε_1 and ε_2, let

$$f_m(\varepsilon_1,\varepsilon_2)\equiv m(\varepsilon_2-\varepsilon_1)+\tfrac{1}{2}-\varepsilon_1. \tag{47}$$

Thus, (42) implies that $R_n[m] = \lfloor f_m(\varepsilon_1, \varepsilon_2) \rfloor$, and

$$P\left(\vec{R}_n[m] = \vec{R}_M\right) = P\left(\lfloor f_m(\varepsilon_1, \varepsilon_2) \rfloor = R_m, \text{ for } m = 1, \ldots, M\right). \tag{48}$$

Because ε_1 and ε_2 are independent and uniformly distributed across the symmetric interval $(-1/2, 1/2)$, $(-\varepsilon_1, -\varepsilon_2)$ has the same distribution as $(\varepsilon_1, \varepsilon_2)$, which implies that

$$\begin{aligned} P(\lfloor f_m(-\varepsilon_1, -\varepsilon_2) \rfloor = R_m, \text{ for } m = 1, \ldots, M) \\ = P\left(\lfloor f_m(\varepsilon_1, \varepsilon_2) \rfloor = R_m, \text{ for } m = 1, \ldots, M\right). \end{aligned} \tag{49}$$

Using (47), the function $f_m(-\varepsilon_1, -\varepsilon_2)$ can be simplified to

$$f_m(-\varepsilon_1, -\varepsilon_2) = 1 - f_m(\varepsilon_1, \varepsilon_2). \tag{50}$$

Since $\lfloor x + 1 \rfloor = \lfloor x \rfloor + 1$, and $\lfloor -x \rfloor = -\lceil x \rceil$, it follows from (50) that

$$\lfloor f_m(-\varepsilon_1, -\varepsilon_2) \rfloor = 1 - \lceil f_m(\varepsilon_1, \varepsilon_2) \rceil. \tag{51}$$

This implies that

$$\lfloor f_m(-\varepsilon_1, -\varepsilon_2) \rfloor = \begin{cases} 1 - f_m(\varepsilon_1, \varepsilon_2), & \text{if } f_m(\varepsilon_1, \varepsilon_2) \text{ is an integer;} \\ -\lfloor f_m(\varepsilon_1, \varepsilon_2) \rfloor, & \text{otherwise.} \end{cases} \tag{52}$$

Because $f_m(\varepsilon_1, \varepsilon_2)$ is a continuous function of random variables with continuous distributions, the probability that $f_m(\varepsilon_1, \varepsilon_2)$ is an integer is zero, which implies

$$P\left(\lfloor f_m(-\varepsilon_1, -\varepsilon_2) \rfloor = -\lfloor f_m(\varepsilon_1, \varepsilon_2) \rfloor \text{ for every } m\right) = 1. \tag{53}$$

Thus, $\lfloor f_m(-\varepsilon_1, -\varepsilon_2) \rfloor = -\lfloor f_m(\varepsilon_1, \varepsilon_2) \rfloor$ almost surely, and

$$\begin{aligned} P(\lfloor f_m(-\varepsilon_1, -\varepsilon_2) \rfloor = R_m, \text{ for } m = 1, \ldots, M) \\ = P\left(\lfloor f_m(\varepsilon_1, \varepsilon_2) \rfloor = -R_m, \text{ for } m = 1, \ldots, M\right). \end{aligned} \tag{54}$$

The probability function equalities in (49) and (54) imply (46).

■

Claim 3: Each switching block input satisfies the Range and Alternating Properties.

Proof: This is proved by induction on the layer number k. For $k = b$, the switching block input is the output of the $\Delta\Sigma$ modulator model, which, by Claim 1, satisfy this claim's assertion. Suppose this assertion holds for $k > 1$ and consider the switching block $S_{k,r}$, for some $r = 1, \ldots, 2^{b-k}$. By the symmetry of the switching block's operation, it is sufficient to prove that the top output of $S_{k,r}$ satisfies the claim's assertion. Since $s_{k,r}[n]$ and $x_{k,r}[n]$ are limited to $\{-1, 0, 1\}$, it follows from (2) that $x_{k-1,2r-1}[n]$ is also in this range because it is restricted to be an integer. Therefore, $x_{k-1,2r-1}[n]$ has the Range Property.

From (4) and the induction hypothesis, $s_{k,r}[n]$ alternates between being in the head and tail of a symbol at sample times when $|x_{k,r}[n]| = 1$. Therefore, at these sample times, $z_{k,r}[n]$ alternates between 1 and -1. Since $z_{k,r}[n]$ and $x_{k,r}[n]$ (by the induction hypothesis) both alternate between 1 and -1 at sample times when $x_{k,r}[n]$ is nonzero, either $x_{k,r}[n] = z_{k,r}[n]o_{k,r}[n]$, or $x_{k,r}[n] = -z_{k,r}[n]o_{k,r}[n]$ for all n. Suppose a symbol of length S samples starts in $s_{k,r}[n]$ at sample time n_0. For $l = 0, \ldots, S-1$, (2) implies

$$x_{k-1,2r-1}[n_0 + l] = \begin{cases} x_{k,r}[n_0 + l], & \text{if } d_{k,r}[n_0] = x_{k,r}[n_0]/2; \\ 0, & \text{otherwise.} \end{cases} \tag{55}$$

If $x_{k,r}[n] = z_{k,r}[n]o_{k,r}[n]$, then the two nonzero samples in $x_{k,r}[n_0], \ldots, x_{k,r}[n_0 + S - 1]$ alternate from 1 to -1. Otherwise, the two nonzero samples in this segment alternate from -1 to 1. This holds for every symbol in $s_{k,r}[n]$, and since $s_{k,r}[n]$ consist entirely of concatenated symbols, (55) implies that the nonzero samples of $x_{k-1,2r-1}[n]$ alternate between 1 and -1. Therefore, $x_{k-1,2r-1}[n]$ satisfies the Alternating Property, and by mathematical induction, the claim's assertion holds for all k.

■

Claim 4: Each switching block input satisfies the Symmetry Property: *i.e.*,

$$P\left(\vec{x}_{k,r}[n] = \vec{x}_M\right) = P\left(\vec{x}_{k,r}[n] = -\vec{x}_M\right). \tag{56}$$

for any M-length vector $\vec{x}_M$.

Proof: This is proved with induction on the layer number k. By Claim 2, (56) holds for $x_{b,1}[n] = y[n]$. Suppose (56) holds for each switching block in layer $k > 1$. Since the signs of the nonzero values in each switching sequence are determined by a dither sequence, it follows that

$$P\left(\vec{s}_{k,r}[n] = \vec{s}_M\right) = P\left(\vec{s}_{k,r}[n] = -\vec{s}_M\right). \tag{57}$$

for any M-length vector s_M. It follows from (2), (3), (57), and the induction hypothesis that the Symmetry Property holds for each switching sequence in layer $k-1$. Thus, by mathematical induction, this property holds for all switching block inputs.

■

Combining Claims 3 and 4 prove the lemma.

■

Lemma A1: Given two positive integers l and m,

$$R_n[l+m] > \left(R_n[l] - 1\right)\left(1 + \tfrac{m}{l}\right). \tag{58}$$

and

$$R_n[l+m] < \left(R_n[l] + 1\right)\left(1 + \tfrac{m}{l+1}\right). \tag{59}$$

for every n.

Proof: Let $\varepsilon[n-1] = 1/2 - \gamma_1$ and $\varepsilon[n-2] = 1/2 - \gamma_2$, where, by assumption, γ_1 and γ_2 are independent random variables that are uniformly distributed across the

interval (0, 1). Given (42), this implies that

$$R_n[i] = \lfloor i(\gamma_1 - \gamma_2) + \gamma_1 \rfloor. \tag{60}$$

Because the floor function rounds down to the nearest integer, (60) implies

$$R_n[i] \leq i(\gamma_1 - \gamma_2) + \gamma_1 < R_n[i] + 1. \tag{61}$$

which gives

$$\frac{(i+1)\gamma_1 - (R_n[i]+1)}{i} < \gamma_2 \leq \frac{(i+1)\gamma_1 - R_n[i]}{i}. \tag{62}$$

for all $i > 0$. Therefore, if $i = m + l$ is applied to the left-hand side of (62), and $i = l$ is applied to the right-hand side of (62), it follows that

$$\frac{(l+m+1)\gamma_1 - (R_n[l+m]+1)}{l+m} < \frac{(l+1)\gamma_1 - R_n[l]}{l}. \tag{63}$$

which, given $\gamma_1 < 1$, can be simplified to

$$(l+m)R_n[l] - l(R_n[l+m]+1) < m\gamma_1 < m. \tag{64}$$

Upon simplifying (64), (58) follows.

For the second inequality in this lemma, note that (61) also gives the following inequality:

$$\frac{i\gamma_2 + R_n[i]}{i+1} \leq \gamma_1 < \frac{i\gamma_2 + R_n[i] + 1}{i+1}. \tag{65}$$

for all $i > 0$. Therefore, if $i = m + l$ is applied to the left-hand side of (65), and $i = l$ is applied to the right-hand side of (65), it follows that

$$\frac{(l+m)\gamma_2 + R_n[l+m]}{l+m+1} < \frac{l\gamma_2 + R_n[l]+1}{l+1}. \tag{66}$$

which can be simplified to

$$(l+1)R_n[l+m] < (R_n[l]+1)(l+m+1) - m\gamma_2. \tag{67}$$

Because $\gamma_2 > 0$, (67) implies

$$(l+1)\,R_n[l+m] < (R_n[l]+1)\,(l+m+1)\,. \tag{68}$$

Dividing both sides of (68) by $l+1$ gives (59).

■

Notation: Let Z_L be a segment composed of $L \geq 0$ zeros, and let S_N ($N \geq 1$) denote the following segment of $N+1$ samples:

$$S_N \equiv \underbrace{1,0,\ldots,0,}_{N \text{ samples}} -1; \tag{69}$$

where only the number of zeros in S_N vary as a function of N.

Lemma A2: If a segment of $y[n]$ is given by

$$y[n] = \ldots, S_{N_1}, Z_L, S_{N_2}, \ldots, \tag{70}$$

with $N_1 > 1$, then $L = 0$ and $|N_2 - N_1| \leq 1$.

Proof: Without loss of generality, assume that the sequence starts at time n. The double of sum of the segment in (70) is given by

$$R_n[i] = \begin{cases} i, & \text{for } i = 1,\ldots,N_1; \\ N_1, & \text{for } i = N_1+1,\ldots,N_1+L+1; \\ i-L-1, & \text{for } i = N_1+L+2,\ldots,N_1+L+N_2+1; \\ N_1+N_2, & \text{for } i = N_1+L+N_2+2. \end{cases} \tag{71}$$

It is shown first that if $N_1 > 1$, then $L = 0$. With $l = N_1$ and $m = L+1$, it follows from (71) that $R_n[l] = R_n[l+m] = N_1$, and (58) implies

$$N_1 > (N_1 - 1)\left(1 + \tfrac{N_1}{L+1}\right). \tag{72}$$

Given $N_1 > 1$, then (72) implies that

$$L < \frac{N_1}{N_1 - 1} - 1 < 1. \tag{73}$$

Since L is a nonnegative integer, (73) implies that $L = 0$ whenever $N_1 > 1$.

Next it is shown that $|N_2 - N_1| \leq 1$ when $N_1 > 1$. Let $l_1 = N_1 + L + 1$, $m_1 = N_2$, $l_2 = N_1$, and $m_2 = L + N_2 + 2$. This and (71) imply $R_n[l_1] = R_n[l_2] = N_1$, and $R_n[l_1 + m_1] = R_n[l_2 + m_2] = N_1 + N_2$. Using the l_1 and m_1 values with (59), and the l_2 and m_2 values with (58), it follows that

$$\left(1 + \tfrac{L+N_2+2}{N_1}\right)(N_1 - 1) < N_1 + N_2 < (N_1 + 1)\left(1 + \tfrac{N_2}{N_1+L+2}\right). \tag{74}$$

With $N_1 > 1$, it has already been shown that $L = 0$ and so (74) can be simplified to

$$\frac{N_1 + N_2 + 2}{N_1}(N_1 - 1) < N_1 + N_2 < (N_1 + 1)\frac{N_1 + N_2 + 2}{N_1 + 2}. \tag{75}$$

Upon simplifying, the lower bound in (75) gives $N_1 - N_2 < 2$, and the upper bound gives $N_1 - N_2 > -2$, which implies $|N_1 - N_2| \leq 1$ since N_1 and N_2 are integers.

■

Lemma A3: Given the integers $K, N_1, N_2, N_3 \geq 1$ and $L \geq 0$, let a segment of $y[n]$ be given by

$$y[n] = \ldots . S_{N_1} . \underbrace{S_{N_2} \ldots . S_{N_2}}_{K \text{ segments}} . Z_L . S_{N_3} \ldots . \tag{76}$$

If $N_1 \neq N_2$ and either $N_1 > 1$ or $N_2 > 1$, then $L = 0$ and either $N_3 = N_1$ or $N_3 = N_2$.

Proof: Assume, without loss of generality, that the segment given by (76) begins at sample time n. The double sum of this segment is

$$R_n[i] = \begin{cases} i. & \text{for } 1 \leq i \leq N_1: \\ i - 1 - \left\lfloor \frac{i - N_1 - 1}{N_2 + 1} \right\rfloor. & \text{for } 0 \leq i - N_1 \leq (N_2 + 1)K + 1: \\ N_1 + N_2 K. & \text{for } 0 \leq i - [N_1 + 1 + (N_2 + 1)K] \leq L: \\ i - (L + K + 1). & \text{for } 0 \leq i - [N_1 + 1 + (N_2 + 1)K + L] \leq N_3: \\ N_1 + N_2 K + N_3. & \text{for } i = N_1 + (N_2 + 1)K + L + N_3 + 2. \end{cases} \tag{77}$$

First, it is shown that if $N_1 \neq N_2$ and either $N_1 > 1$ or $N_2 > 1$, then $L = 0$. If $N_2 > 1$, it follows from Lemma A2 that $L = 0$. Suppose $N_2 = 1$, which implies that $N_1 = 2$ by Lemma A2 and the hypothesis. To apply Lemma A1, let $l = N_1 = 2$, and $m = 1 + K(N_2 + 1) + L = 1 + 2K + L$. Using (77), this implies $R_n[l] = N_1 = 2$, and $R_n[l + m] = N_1 + N_2K = 2 + K$, and (58) gives

$$K + 2 > 1 + \frac{1 + 2K + L}{2}. \tag{78}$$

Simplifying (78) gives $L < 1$, which, since L is a nonnegative integer, implies $L = 0$ for any $K \geq 1$.

Next it is shown that given $N_1 \neq N_2$ and either $N_1 > 1$ or $N_2 > 1$, then either $N_3 = N_2$ or $N_3 = N_1$. Suppose $N_3 \neq N_2$. Lemma A2 implies that $|N_3 - N_2| = 1$ and $|N_1 - N_2| = 1$. To apply Lemma A1, let $l_1 = N_1 + 1 + K(N_2 + 1)$, $m_1 = L + N_3$, $l_2 = N_1$, and $m_2 = (N_2 + 1)K + L + N_3 + 2$. With these values, (77) implies that $R_n[l_1] = N_1 + KN_2$, $R_n[l_2] = N_1$, and $R_n[l_1 + m_1] = R_n[l_2 + m_2] = N_1 + KN_2 + N_3$. Substituting the first set of values (*i.e.*, l_1, etc.) into (59) and the second set of values into (58) gives

$$(N_1 - 1)\left(1 + \frac{(N_2+1)K + L + N_3 + 2}{N_1}\right) < N_1 + KN_2 + N_3 < (N_1 + KN_2 + 1)\left(1 + \frac{L + N_3}{N_1 + 2 + K(N_2+1)}\right). \tag{79}$$

Since, as previously shown, $L = 0$ under the given assumptions, the upper bound of (79) can be simplified to

$$N_3(K + 1) < N_1 + 2 + K(N_2 + 1). \tag{80}$$

while the lower bound can be simplified to

$$N_3 + K(N_2 - 1) > (N_1 - 2)(K + 1). \tag{81}$$

Since $N_2 \leq N_1 + 1$, (80) implies that $N_3 < N_1 + 2$ for any $K > 0$. Moreover, since $N_2 - 1 \geq N_3$, (81) implies that $N_3 > N_1 - 2$. Therefore, $|N_3 - N_1| \leq 1$.

Since $|N_2 - N_1| = |N_2 - N_3| = 1$. this implies $N_3 = N_1$. Because $N_3 = N_1$ when $N_3 \neq N_2$. either $N_3 = N_2$ or $N_3 = N_1$ under the given assumptions.

■

Theorem A1. Characterization of the ΔΣ Modulator Output: There exists a nonnegative integer N such that $y[n]$ either consists entirely of the concatenation of the segments S_N and S_{N+1} or $-S_N$ and $-S_{N+1}$.

Proof: Lemma A3 implies that if S_N and S_{N+1} constitute a segment of $y[n]$. then there are no zero runs between such segments and all segments that follow are of the same type. The Symmetry Property in Lemma 1 implies that the same result holds for the segments $-S_N$ and $-S_{N+1}$.

■

Definition: A segment S of the ΔΣ modulator output is called a *Type A* segment if there exists a subsegment in S given by $\ldots -1.0 \ldots$. otherwise. the segment is called a *Type B* segment.

Lemma A4: For m odd. $R_n[m]$ is greater than or equal to $(m+1)/2$ if and only if $y[n] = 1$. and the segment $y[n] \ldots . y[n+m-1]$ is a Type B segment.

Proof: (Necessity) Suppose $y[n] = 1$ and the segment $y[n] \ldots . y[n+m-1]$ is a Type B segment. Because a negative one is always immediately followed by a one in this segment. it follows that

$$R_n[l] = \begin{cases} R_n[l-1]. & \text{if } y[n+l-1] = -1: \\ R_n[l-1] + 1. & \text{otherwise:} \end{cases} \tag{82}$$

for $0 < l \leq m$. This implies that $R_n[l] = l - N_{-1}[l]$. where $N_{-1}[l]$ is the number of negative ones in the segment $y[n] \ldots . y[n+l-1]$. Given m is odd. Theorem A1

implies that there are at most $(m-1)/2$ negative ones in the segment $y[n], \ldots, y[n+m-1]$. Therefore, $R_n[m] \geq (m+1)/2$.

(Sufficiency) Now, suppose $R_n[m] \geq (m+1)/2$. Since $R_n[m] > 0$, $y[n] \neq -1$. Recall that $R_n[m]$ is given by (60) as a function of the two independent and uniformly distributed (across the interval (0, 1)) random variables γ_1 and γ_2. If $y[n] = R_n[1] = 0$, it follows that $2\gamma_1 - \gamma_2 < 1$, which implies $\gamma_1 < (1+\gamma_2)/2$. Inserting this inequality into (60) gives $R_n[m] < (m+1)/2$. Therefore, $y[n]$ is neither -1 nor 0, which implies that $y[n] = 1$ by Lemma 1.

Suppose the segment $y[n], \ldots, y[n+m-1]$ is a Type A segment. Theorem A1 implies then that every one in this segment is immediately followed by a negative one, which implies that

$$R_n[l] = \begin{cases} R_n[l-1]+1, & \text{if } y[n+l-1] = 1; \\ R_n[l-1], & \text{otherwise;} \end{cases}$$

for $0 < l \leq m$. Thus, $R_n[l] = N_1[l]$, where $N_1[l]$ is the number of ones in the segment $y[n], \ldots, y[n+l-1]$. Given $y[n], \ldots, y[n+m-1]$ is a Type A segment, there are at most $(m-1)/2$ ones in this segment. This implies that $R_n[m] \leq (m-1)/2$, which contradicts the assumption that $R_n[m] \geq (m+1)/2$. Therefore, the segment $y[n], \ldots, y[n+m-1]$ must be a Type B segment.

■

Theorem A2: Given m is a nonnegative integer, and h is a positive integer, it follows that

$$P_b(h, m) = \begin{cases} 0, & m > \frac{h-1}{2}; \\ \frac{(m+1)^2}{h(h+1)(h+2)} + \frac{m^2}{h(h-1)(h-2)}, & h > 2,\ m < \frac{h-1}{2}; \\ \frac{(m+1)^2}{h(h+1)(h+2)}, & \text{otherwise.} \end{cases} \tag{83}$$

Proof: From Theorem A1, each one in $y[n]$ either immediately precedes or succeeds

a negative one. Thus, if there exist m values of $i \in \{1, \ldots, h-1\}$ such that $y[n+i] = 1$, then there exist m values of i such that $y[n+i] = -1$. Therefore, $2m$ must be less than h, which implies that $P_h(h, m) = 0$ if $m > (h-1)/2$.

Let $P(h, m, B)$ be the probability that:

a) $y[n] = 1$ and $y[n+h] = -1$;

b) There exist m values of $i \in \{1, \ldots, h-1\}$ such that $y[n+i] = 1$;

c) The given segment of $y[n]$ is a Type B segment.

As shown in the proof of Lemma A4, these conditions imply that $R_n[l] = l - N_{-1}[l]$, where $N_{-1}[l]$ is the number of negative ones in the segment $y[n], \ldots, y[n+l-1]$. Since $y[n+h] = -1$, it follows that $R_n[h] = R_n[h+1] = h - m$. Additionally, because this segment of $y[n]$ is Type B, it follows that $m \leq \lfloor \frac{h-1}{2} \rfloor$, which implies

$$R_n[h] = R_n[h+1] \geq \left\lfloor \frac{h+2}{2} \right\rfloor. \tag{84}$$

Since either h or $h+1$ is odd, it follows from the above inequality and Lemma A4 that $R_n[h] = R_n[h+1] = h - m$ if and only if the three events listed above hold. Thus, $P(h, m, B)$ can be written as

$$P(h, m, B) = P(R_n[h] = R_n[h+1] = h - m). \tag{85}$$

Recall that $R_n[h]$ is given by (60) as a function of the two independent and uniformly distributed (across the interval $(0, 1)$) random variables γ_1 and γ_2. Because $x \leq \lfloor x \rfloor < x + 1$, the event $R_n[h] = R_n[h+1] = h - m$ occurs if and only if

$$\frac{h-m}{h+2} + \frac{h+1}{h+2}\gamma_2 \leq \gamma_1 < \frac{h-m+1}{h+2} + \frac{h+1}{h+2}\gamma_2. \tag{86}$$

and

$$\frac{h-m}{h+1} + \frac{h}{h+1}\gamma_2 \leq \gamma_1 < \frac{h-m+1}{h+1} + \frac{h}{h+1}\gamma_2. \tag{87}$$

With $0 < \gamma_2 < 1$ and $h - m \geq 1$, the upper bound in (86) is more restrictive than that in (87), and the lower bound in (87) is more restrictive than that in (86). Therefore, (86) and (87) are satisfied if and only if

$$\frac{h-m}{h+1} + \frac{h}{h+1}\gamma_2 \leq \gamma_1 < \frac{h-m+1}{h+2} + \frac{h+1}{h+2}\gamma_2. \tag{88}$$

Thus, the probability that $R_n[h] = R_n[h+1] = h - m$ is equivalent to the probability that γ_1 and γ_2 satisfy (88). Since γ_1 and γ_2 are independent with uniform distributions, this probability is

$$P(h, m, B) = \int_0^1 \int_{\min\left\{\frac{h-m}{h-1} + \frac{h}{h+1}y, 1\right\}}^{\min\left\{\frac{h-m+1}{h+2} + \frac{h+1}{h+2}y, 1\right\}} dx\, dy. \tag{89}$$

Solving this integral gives

$$P(h, m, B) = \begin{cases} \frac{(m+1)^2}{h(h+1)(h+2)}, & \text{for } m \leq \frac{h-1}{2}; \\ 0, & \text{otherwise.} \end{cases} \tag{90}$$

Let $P(h, m, A)$ be the probability that events a) and b) occur along with the following:

c.′ The given segment of $y[n]$ is a Type A segment.

By Theorem A1, every one in a Type A segment of $y[n]$ is followed by a negative one. Thus, it follows that Events a), b), c′) occur if and only if Event c′) occurs along with the following events:

a′) $y[n+1] = -1$ and $y[n+h-1] = 1$;

b′) There exist $m - 1$ values of $i \in \{2, \ldots, h-2\}$ such that $y[n+i] = -1$.

By the Symmetry Property in Lemma 1, the probability that Events a′), b′), and c′) occur is equivalent to the probability that:

a″) $y[n+1] = 1$ and $y[n+h-1] = -1$;

b″) There exist $m - 1$ values of $i \in \{2, \ldots, h-2\}$ such that $y[n+i] = 1$;

c″) The segment $y[n+1], \ldots, y[n+h-1]$ is a Type B segment.

This implies that $P(h, m, A) = P(h-2, m-1, B)$ for $m < (h-1)/2$ and $h > 2$: in other words,

$$P(h, m, A) = \begin{cases} \frac{m^2}{h(h-1)(h-2)}, & \text{for } h > 2,\ m < \frac{h-1}{2}: \\ 0, & \text{otherwise.} \end{cases} \tag{91}$$

Since, by virtue of Theorem A1, there are only Type A or B segments of $y[n]$, then $P_h(h, m) = P(h, m, A) + P(h, m, B)$. This, (90), and (91) imply (83).

■

Theorem A3: Given l is a positive integer,

$$P_h(l) = \begin{cases} \frac{1}{4l}, & \text{for } l \text{ odd:} \\ \frac{l}{4(l-1)(l+1)}, & \text{for } l \text{ even.} \end{cases} \tag{92}$$

Proof: For l even, $P_h(l) = P_h(h, m)$ with $h = l-1$, and $m = (l-2)/2$. In other words,

$$P_h(l) = \frac{l}{4(l-1)(l+1)}. \tag{93}$$

when $l > 0$ is even.

For l odd, $P_h(l)$ can be written as

$$P_h(l) = P\left(R_n[m] = \left\lfloor \tfrac{m+1}{2} \right\rfloor, \text{ for } m = 1, \ldots, l\right). \tag{94}$$

Given $R_n[l] = (l+1)/2$ and l is odd, it follows from Lemma A4 that $y[n] = 1$, and the segment $y[n], \ldots, y[n+l-1]$ is a Type B segment. However, this does not imply that $y[n+i] = (-1)^i$ for $i = 0, \ldots, l-1$.

Suppose $l > 1$. It is shown next that the two events $R_n[l] = (l+1)/2$ and $R_n[l-1] = (l-1)/2$ occur if and only if $y[n+i] = (-1)^i$ for $i = 0, \ldots, i-1$. Necessity follows by computing $R_n[l]$ and $R_n[l-1]$. Now, sufficiency is proven by supposing that $R_n[l] = (l+1)/2$ and $R_n[l-1] = (l-1)/2$ both hold. As

shown in the proof of Lemma A4, because the given segment of $y[n]$ is Type B. $R_n[i] = i - N_{-1}[i]$, where $N_{-1}[i]$ is the number of negative ones in the segment $y[n].....y[n+i-1]$. Suppose $y[n+i] = 0$ for some $i = 1.....l-2$. This implies that $N_{-1}[l-1] \leq (l-3)/2$, and $R_n[l-1] \geq (l+1)/2$. This contradicts the fact that $R_n[l-1] = (l-1)/2$. Therefore, $y[n+i] \neq 0$ for $i = 0.....l-2$, and since $y[n+l-2] = -1$, it follows that $y[n+l-1] = 1$ because, as previously shown, this segment of $y[n]$ is a Type B segment. So, the expressions $R_n[l-1] = (l-1)/2$ and $R_n[l] = (l+1)/2$ imply that $y[n+i] = (-1)^i$ for $i = 0.....l-1$.

Therefore, (94) can be simplified to

$$P_b(l) = P\left(R_n[l] = \tfrac{l+1}{2}, R_n[l-1] = \tfrac{l-1}{2}\right). \tag{95}$$

Given $R_n[0] = 0$, (95) holds for all odd l, including $l = 1$. Recall that $R_n[l]$ is given by (60) as a function of the two independent and uniformly distributed (across the interval $(0,1)$) random variables γ_1 and γ_2. Because $x \leq \lfloor x \rfloor < x+1$, the event $R_n[l] = R_n[l-1] + 1 = (l+1)/2$ occurs if and only if

$$\frac{l-1}{2l} + \frac{l-1}{l}\gamma_2 \leq \gamma_1 < \frac{l+1}{2l} + \frac{l-1}{l}\gamma_2. \tag{96}$$

and

$$\frac{1}{2} + \frac{l}{l+1}\gamma_2 \leq \gamma_1 < \frac{l+3}{l+1} + \frac{l}{l+1}\gamma_2. \tag{97}$$

Because γ_1 and γ_2 are restricted to the interval $(0,1)$, the upper inequality of (96) is more restrictive than that of (97), while the lower inequality of (97) is more restrictive than that of (96). Therefore, (96) and (97) are satisfied if and only if

$$\frac{1}{2} + \frac{l}{l+1}\gamma_2 \leq \gamma_1 < \frac{l+1}{2l} + \frac{l-1}{l}\gamma_2. \tag{98}$$

Therefore, $P_b(l)$ is equal to the probability that γ_1 and γ_2 satisfy (98). Since γ_1 and γ_2 are independent random variables that are uniformly distributed across

the interval $(0,1)$. this probability can be evaluated as follows:

$$P_b(l) = \int_0^1 \int_{\min\left\{\frac{l-1}{2l}+\frac{l-1}{l}y,1\right\}}^{\min\left\{\frac{1}{2}+\frac{l}{l-1}y,1\right\}} dx\,dy. \tag{99}$$

Evaluating (99) gives

$$P_b(l) = \frac{1}{4l}. \tag{100}$$

for odd $l > 0$. Therefore, (93) and (100) imply (92).

■

Theorem A4: The probability function $P_k(h,m)$ distributes through the DAC as follows:

$$P_{k-1}(h,m) = \sum_{i=m}^{\left\lfloor\frac{h-1}{2}\right\rfloor} \left(\frac{1}{2}\right)^{i+3} \left(\binom{i}{m} + 2\binom{i-1}{m-1}\right) P_k(h,i). \tag{101}$$

where $k > 1$ and $\binom{a}{b}$ is the combination function with $\binom{-1}{-1}$ *defined* to be 1.

Proof: By assumption, $P_k(h,m)$ is independent of the layer depth; therefore, $P_{k-1}(h,m)$ is derived from $P_k(h,m)$ by considering the top output of $S_{k,r}$, which is given by (2). As shown in the proof of Lemma 1 (Claim 3), either $x_{k,r}[n] = z_{k,r}[n]o_{k,r}[n]$ or $x_{k,r}[n] = -z_{k,r}[n]o_{k,r}[n]$. Let $\hat{P}_{k-1}(h,m,i,1)$ be the conditional probability that

a) $x_{k-1,2r-1}[n] = 1$ and $x_{k-1,2r-1}[n+h] = -1$;

b) There exist m values of $j \in \{1,\ldots,h-1\}$ such that $x_{k-1,2r-1}[n+j] = 1$;

given that

1. $x_{k,r}[n] = 1$ and $x_{k,r}[n+h] = -1$;
2. There exist i values of $j \in \{1,\ldots,h-1\}$ such that $x_{k,r}[n+j] = 1$;
3. $x_{k,r}[n] = z_{k,r}[n]o_{k,r}[n]$ for all n.

Under this condition, the segment $x_{k,r}[n],\ldots,x_{k,r}[n+h]$ gives rise to $i+1$ symbols in the switching sequence during these same sample times.

Recall from (55) that the dither sequence chooses the type of a given symbol and determines whether or not $x_{k-1,2r-1}[n]$ is $x_{k,r}[n]$ or 0 at sample times within this symbol. In the given segment of $x_{k,r}[n]$, there are $i+1$ symbols and thus $i+1$ symbol type choices, all of which are made independently with a probability of 1/2. To satisfy the event characterized in $\hat{P}_{k-1}(h, m, i, 1)$, the first and last dither choices in this segment must ensure Event a) holds, while the remaining $i-1$ dither choices must ensure that Event b) holds. There are $\binom{i-1}{m-1}$ unique combinations of dither choices that ensure this, which implies that

$$\hat{P}_{k-1}(h, m, i, 1) = \left(\frac{1}{2}\right)^{i+1} \binom{i-1}{m-1}. \tag{102}$$

Let $\hat{P}_{k-1}(h, m, i, -1)$ be the same probability function as $\hat{P}_{k-1}(h, m, i, 1)$, except Condition 3 is changed to the following:

3′. $x_{k,r}[n] = -z_{k,r}[n]o_{k,r}[n]$ for all n.

In this case, symbols always start in $s_{k,r}[n]$ at sample times when $x_{k,r}[n]$ is -1. Furthermore, the segment $x_{k,r}[n], \ldots, x_{k,r}[n+h]$ includes $i+2$ dither choices (the two choices which determine $x_{k-1,2r-1}[n]$ and $x_{k-1,2r-1}[n+h]$, and the i choices between these two samples). To satisfy the event characterized in $\hat{P}_{k-1}(h, m, i, -1)$, the first and last dither choices for the described segment of $x_{k-1,2r-1}[n]$ must ensure that Events a) holds while the remaining i dither choices must ensure that Event b) holds. There are $\binom{i}{m}$ unique combinations of dither choices that ensure this, which implies that

$$\hat{P}_{k-1}(h, m, i, -1) = \left(\frac{1}{2}\right)^{i+2} \binom{i}{m}. \tag{103}$$

Because the initial state of $s_{k,r}[n]$ is taken to be an independent, uniform random variable, the probability that $x_{k,r}[n] = z_{k,r}[n]o_{k,r}[n]$ and the probability that $x_{k,r}[n] = -z_{k,r}[n]o_{k,r}[n]$ are both equal to 1/2. Therefore, by averaging the condi-

tional probabilities that are derived above, it follows that

$$P_{k-1}(h,m) = \frac{1}{2}\sum_{i=m}^{\left\lfloor \frac{h-1}{2} \right\rfloor} \left(\hat{P}_{k-1}(h,m,i,-1) + \hat{P}_{k-1}(h,m,i,1) \right) P_k(h,i). \qquad (104)$$

Substituting (102) and (103) into (104) gives (101).

■

Theorem A5: Given $k > 1$, the probability function $P_k(l)$ distributes through the DAC as follows:

$$P_{k-1}(l) = \frac{1}{2}\left(\left(\frac{1}{2}\right)^{\left\lfloor \frac{l+1}{2} \right\rfloor} + \left(\frac{1}{2}\right)^{\left\lceil \frac{l+1}{2} \right\rceil} \right) P_k(l). \qquad (105)$$

Proof: It follows from assumption that the value of $P_{k-1}(l)$ can be determined by analyzing the top output of $S_{k,r}$. As shown in the proof of Lemma 1 (Claim 3), either $x_{k,r}[n] = z_{k,r}[n]o_{k,r}[n]$ or $x_{k,r}[n] = -z_{k,r}[n]o_{k,r}[n]$. The probability of either event is 1/2 by assumption. Let $\hat{P}_{k-1}(l,1)$ be the conditional probability that:

a) $x_{k-1,2r-1}[n+i] = (-1)^i$ for $i = 0,\ldots,l-1$:

given that

1. $x_{k,r}[n+i] = (-1)^i$ for $i = 0,\ldots,l-1$:
2. $x_{k,r}[n] = z_{k,r}[n]o_{k,r}[n]$.

Given Condition 2, symbols in $s_{k,r}[n]$ begin at sample times only when $x_{k,r}[n]$ is 1. Therefore, given Conditions 1 and 2, Event a) occurs if and only if $d_{k,r}[n+2m] = 1$ for $m = 0,\ldots,\lfloor (l+1)/2 \rfloor - 1$. Since the dither sequence consists of i.i.d. uniform random variables, the probability of this event is

$$\hat{P}_{k-1}(l,1) = \left(\frac{1}{2}\right)^{\left\lfloor \frac{l+1}{2} \right\rfloor}. \qquad (106)$$

Let $\hat{P}_{k-1}(l,-1)$ be the same conditional probability as $\hat{P}_{k-1}(l,1)$ except Condition 2 is altered to be

2.′ $x_{k,r}[n] = -z_{k,r}[n]o_{k,r}[n]$.

This implies that a symbol in $s_{k,r}[n]$ always begins at sample times when $x_{k,r}[n]$ is -1. Given Conditions 1 and 2′. Event a) occurs if and only if a dither sample gave rise to $x_{k,r}[n] = 1$ and for $l > 1$. $d_{k,r}[n+2m+1] = -1$ for $m = 0, \ldots, \lceil (l+1)/2 \rceil - 2$. Therefore. the probability of this event is given by

$$\hat{P}_{k-1}(l,-1) = \left(\frac{1}{2}\right)^{\left\lceil \frac{l-1}{2} \right\rceil}. \tag{107}$$

Since Conditions 2 and 2′ occur with equal probability. $P_{k-1}(l)$ is obtained by averaging the conditional probabilities as follows

$$P_{k-1}(l) = \frac{1}{2}\left(\hat{P}_{k-1}(l,1) + \hat{P}_{k-1}(l,-1)\right) P_k(l). \tag{108}$$

Substituting (106) and (107) gives (105).

■

Theorem 1: See Section IV for the theorem statement.

Proof: First. consider the switching sequence variances and the head-length probabilities for switching sequences in layer $k > 1$. As shown in (28) and (32). these statistics depend on $P_k(l)$. By recursively applying (105). it follows that

$$P_k(l) = \left(\frac{1}{2}\left(\left(\frac{1}{2}\right)^{\left\lfloor \frac{l+1}{2} \right\rfloor} + \left(\frac{1}{2}\right)^{\left\lceil \frac{l+1}{2} \right\rceil}\right)\right)^{b-k} P_b(l). \tag{109}$$

It follows from (92) that $P_b(1) = 1/4$. and with $l = 1$. (109) gives

$$P_k(1) = \left(\frac{1}{2}\right)^{b-k+2}. \tag{110}$$

Substituting (110) into (28) gives (17). Moreover, the head-length probabilities for a switching sequence in layer $k > 1$, as given in (18), follow by substituting (110) into (32).

Now, consider the probability that a head of length $h > 1$ samples occurs in a layer-1 switching sequence. It follows from the Symmetry Property in Lemma 1 and (29) that

$$P(H_1[m] = h) = \frac{2P\left(x_{1,r}[n] = x_{1,r}[n+h] = 0, x_{1,r}[n+i] = (-1)^i,\ i = 1, \ldots, h-1\right)}{P(x_{1,r}[n] = 0)}. \tag{111}$$

Because the nonzero samples of $x_{1,r}[n]$ always alternate between 1 and -1, it follows from the Law of Total Probability that

$$\begin{aligned} P\left(x_{1,r}[n+i] = (-1)^i,\ i = 1, \ldots, h-1\right) = \\ P\left(x_{1,r}[n+i] = (-1)^i,\ i = 0, \ldots, h-1\right) \\ + P\left(x_{1,r}[n] = x_{1,r}[n+h] = 0, x_{1,r}[n+i] = (-1)^i,\ i = 1, \ldots, h-1\right) \\ + P\left(x_{1,r}[n+i] = (-1)^i,\ i = 1, \ldots, h\right). \end{aligned} \tag{112}$$

The Symmetry Property in Lemma 1 implies that all of the probability functions in (112) that do not include $x_{1,r}[n] = 0$ can be determined by the function $P_1(l)$ to give

$$\begin{aligned} P_1(h-1) - 2P_1(h) = \\ P\left(x_{1,r}[n] = x_{1,r}[n+h] = 0, x_{1,r}[n+i] = (-1)^i,\ i = 1, \ldots, h-1\right). \end{aligned} \tag{113}$$

Substituting (113) into (111) gives

$$P(H_1[m] = h) = \frac{2\left(P_1(h-1) - 2P_1(h)\right)}{P(x_{1,r}[n] = 0)}. \tag{114}$$

Since $\sigma_1^2 = P(x_{1,r}[n] = 0)$, the expression for σ_1^2 given in (17) can be substituted into (114) to give (18) for $k = 1$ and $h > 1$.

Finally, consider the probability that a head of length 1 occurs in $s_{1,r}[n]$. The head-length probability is given by

$$P(H_1[m]=1)=\frac{P(x_{1,r}[n]=x_{1,r}[n+1]=0)}{P(x_{1,r}[n]=0)}. \tag{115}$$

The Law of Total Probability indicates

$$\begin{aligned}P(x_{1,r}[n]=0)=P(x_{1,r}[n]=x_{1,r}[n+1]=0)+P(x_{1,r}[n]=0,x_{1,r}[n+1]=-1)\\ +P(x_{1,r}[n]=0,x_{1,r}[n+1]=1).\end{aligned} \tag{116}$$

Additionally,

$$\begin{aligned}P(x_{1,r}[n+1]=1)=P(x_{1,r}[n]=0,x_{1,r}[n+1]=1)\\ +P(x_{1,r}[n]=-1,x_{1,r}[n+1]=1).\end{aligned} \tag{117}$$

Given the definition of $P_k(l)$ and the Symmetry Property of Lemma 1, (117) implies that

$$P(x_{1,r}[n]=0,x_{1,r}[n+1]=1)=P_1(1)-P_1(2). \tag{118}$$

and upon substituting this into (116), it follows that

$$P(x_{1,r}[n]=x_{1,r}[n+1]=0)=P(x_{1,r}[n]=0)-2(P_1(1)-P_1(2)). \tag{119}$$

Substituting (119) into (115) gives

$$P(H_1[m]=1)=1-\frac{2(P_1(1)-P_1(2))}{P(x_{1,r}[n]=0)}. \tag{120}$$

With $P(x_{1,r}[n]=0)=\sigma_1^2$ as described in the previous case, (17) and (120) imply (18) for $k=1$ and $h=1$.

With $P(l)\equiv P_1(l)$, substituting (92) into (109) gives (19). Moreover, (21) and (20) follow from Theorems A2 and A4, respectively.

■

CHAPTER ACKNOWLEDGMENT

The text of Chapter 4 is to be submitted, in part or in full, for publication as a Regular Paper in the *IEEE Transactions on Information Theory*. The dissertation author was the primary researcher. Ian Galton supervised the research which forms the basis of this paper.

REFERENCES

1. B. H. Leung, S. Sutarja, "Multi-bit sigma-delta A/D converter incorporating a novel class of dynamic element matching techniques," *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 39, no. 1, pp. 35-51, Jan. 1992.

2. M. J. Story, "Digital to analogue converter adapted to select input sources based on a preselected algorithm once per cycle of a sampling signal," U.S. Patent No. 5,138,317, Aug. 11, 1992.

3. R. T. Baird, T. S. Fiez, "Linearity enhancement of multi-bit $\Delta\Sigma$ A/D and D/A converters using data weighted averaging," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 12, pp. 753-762, Dec. 1995.

4. R. Schreier, B. Zhang, "Noise-shaped multi-bit D/A converter employing unit elements," *Electronics Letters*, vol. 31, no. 20, pp. 1712-1713, Sept. 28, 1995.

5. R. W. Adams, T. W. Kwan, "Data-directed scrambler for multi-bit noise shaping D/A converters," U.S. Patent No. 5,404,142, Apr. 4, 1995.

6. I. Galton, "Spectral shaping of circuit errors in digital-to-analog converters," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 10, pp. 808-817, Oct. 1997.

7. W. Chou, R.M. Gray, "Dithering and its effects on sigma-delta and multistage sigma-delta modulation," *IEEE Transactions on Information Theory*, vol.37, no.3, part 1, pp.500-13, May 1991.

8. I. Galton, "Granular quantization noise in a class of delta-sigma modulators," *IEEE Transactions on Information Theory*, vol.40, no.3, pp.848-59, May 1994.

9. N. He, F. Kuhlmann, A. Buzo, "Multiloop sigma-delta quantization," *IEEE Transactions on Information Theory*, vol.38, no.3, pp.1015-28, May 1992.

10. T. W. Kwan. R. W. Adams. R. Libert. "A stereo multibit sigma delta DAC with asynchronous master-clock interface." *IEEE Journal of Solid-State Circuits*. vol. 31. no. 12. pp. 1881-1887. Dec. 1996.

11. R. Adams. K. Nguyen. K. Sweetland. "A 113-dB SNR oversampling DAC with segmented noise-shaped scrambling." *IEEE J. Solid-State Circuits*. vol. 33. no. 12. pp. 1871-1878. Dec. 1998.

12. T. Brooks. D. Robertson. D. Kelly. A. Del Muro. S. Harston. "A cascaded sigma-delta pipeline A/D converter with 1.25 MHz signal bandwidth and 89 dB SNR." *IEEE J. Solid-State Circuits*. vol. 32. no. 12. pp. 1896-1906. Dec. 1997.

13. A. Yasuda. H. Tanimoto. T. Iida. "A third-order $\Delta\Sigma$ modulator using second-order noise-shaping dynamic element matching." *IEEE J. Solid-State Circuits*. vol. 33. no. 12. pp. 1879-1886. Dec. 1998.

14. I. Fujimori. L. Longo. A. Hairapetian. K. Seiyama. S. Kosic. J. Cao. S. Chan. "A 90dB SNR. 2.5 MHz output-rate ADC using cascaded multibit delta-sigma modulation at 8x oversampling ratio." *IEEE Journal of Solid-State Circuits*. vol. 35. no. 12. pp. 1820-1828. Dec. 2000.

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

16. E. Fogleman. J. Welz. I. Galton. "An audio ADC delta-sigma modulator with 100dB SINAD and 102dB DR using a second-order mismatch-shaping DAC." *IEEE Journal of Solid State Circuits*. vol. 36. no. 3. pp. 339-48. March 2001.

17. O.J.A.P. Nys. R.K. Henderson. "An analysis of dynamic element matching techniques in sigma-delta modulation." *Proceedings of the IEEE International Symposium on Circuits and Systems*. May 1996. pp.231-4.

18. J. Welz. I. Galton. E. Fogleman. "Simplified logic for first-order and second-order mismatch-shaping digital-to-analog converters." *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*. vol. 48. no. 11. Nov. 2001.

19. J. Welz. I. Galton. "The mismatch-noise PSD from a tree-structured DAC in a second-order delta-sigma modulator with a midscale input." *Proceedings of the IEEE International Conference on Acoustics. Speech. and Signal Processing*. vol. 4. pp. 2625-2628. May 7-11. 2001.

20. J. Grilo. I. Galton. K. Wang. R. Montemayor. "A 12-mW ADC delta-sigma modulator with 80 dB of dynamic range integrated in a single-chip Bluetooth

transceiver." *IEEE Journal of Solid-State Circuits*. vol. 37. no.3 . pp. 271-278. March 2002.

21. J. Welz. I. Galton. "The PSD of the DAC noise in the dithered first-order low-pass tree-structured DAC. " *IEEE Transactions on Information Theory*. in preparation.

22. M.J.M. Pelgrom. A.C.J. Duinmaijer: A.P.G. Welbers. "Matching properties of MOS transistors." *IEEE Journal of Solid-State Circuits*. vol.24. no.5. p.1433-9. Oct. 1989.

23. JU.-B. Shyu. G.C. Temes. K. Yao. "Random errors in MOS capacitors." *IEEE Journal of Solid-State Circuits*. vol.SC-17. no.6. p.1070-6. Dec. 1982.

24. J.-B. Shyu. G.C. Temes. F. Krummenacher. "Random error effects in matched MOS capacitors and current sources." *IEEE Journal of Solid-State Circuits*. vol.SC-19. no.6. p.948-56. Dec. 1984.

25. J. Welz. I. Galton. "Necessary and sufficient conditions for mismatch shaping in multi-bit DACs." under review in *IEEE Transactions on Circuits and Systems II—Analog and Digital Signal Processing*.