An analysis of interconnection network effects of dynamic element matching digital-to-analog converters

Hui Fu

University of Nevada, Las Vegas

Follow this and additional works at: https://digitalscholarship.unlv.edu/rtds
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.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0600

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
AN ANALYSIS OF INTERCONNECTION NETWORK EFFECTS
ON DYNAMIC ELEMENT MATCHING
DIGITAL-TO-ANALOG
CONVERTERS

by

Hui Fu
Bachelor of Science
Tsinghua University, P.R.China
1997

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree
Electrical and Computer Engineering
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
August 2000
Thesis Approval
The Graduate College
University of Nevada, Las Vegas

July 3, 2000

The Thesis prepared by

Hui Fu

Entitled

An Analysis of Interconnection Network Effects on Dynamic Element Matching Digital-To-Analog Converters

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

Examination Committee Chair

Dean of the Graduate College
ABSTRACT

An Analysis of Interconnection Network Effects on Dynamic Element Matching Digital-to-Analog Converters

by

Hui Fu

Dr. Peter Stubberud, Examination Committee Chair
Professor of Electrical and Computer Engineering
University of Nevada, Las Vegas

Digital-to-Analog Converters (DACs) convert digital signals into analog signals and thereby provide the link from digital systems to the analog world. Many DAC architectures use matched components to convert signals. However, practical DACs contain mismatched circuit components that cause nonlinear errors in the DAC's analog output signal. Dynamic Element Matching (DEM) is a method that can eliminate or reduce the effects of mismatch errors. In this thesis, DEM algorithms applied to a specific DEM DAC architecture that use several interconnection networks, including the Barrel Shifting network, the Generalized-Cube network, the Omega network and the Indirect Binary n-cube network, are mathematically analyzed. The hardware complexity of some networks is discussed and a new interconnection network is developed and is shown to be hardware efficient when applied to the specific DEM DAC architecture.
# TABLE OF CONTENTS

ABSTRACT ................................................................................................................................ iii

CHAPTER 1  INTRODUCTION ............................................................................................. 1

CHAPTER 2  OVERVIEW OF DYNAMIC ELEMENT MATCHING
DIGITAL-TO-ANALOG CONVERSION ..................................................................................... 4
  2.1 Overview of Digital to Analog Conversion ............................................................... 4
  2.2 Digital Dynamic Element Matching Technique .................................................... 7
    2.2.1 A DEM Flash DAC Architecture ...................................................................... 8
    2.2.2 An Analysis of DEM DAC Architecture ....................................................... 10
    2.2.3 Performance Criteria for DEM DACs .......................................................... 13

CHAPTER 3  MATHEMATICAL ANALYSES OF INTERCONNECTION NETWORK EFFECTS ON STOCHASTIC DYNAMIC ELEMENT MATCHING ................................................................. 17
  3.1 A Mathematical Analysis of stochastic controlled Barrel Shifting networks used in DEM DACs .................................................................................................................. 18
    3.1.1 Representation of Barrel Shifting network Outputs ........................................ 19
    3.1.2 Mathematical Analysis on $E\{T_{inl}[x(n),c(n)]|x(n)\}$ of Barrel Shifting Network .............................................................................................................. 21
    3.1.3 Mathematical Analysis on $E[GG^T|x(n)]$ of Barrel Shifting Network .......... 22
  3.2 A Mathematical Analysis of the Generalized-Cube Network used in DEM DACs ................................................................................................................................. 27
    3.2.1 Representation of the Generalized-Cube Network Outputs ............................ 29
    3.2.2 Mathematical Analysis on $E\{T_{inl}[x(n),c(n)]|x(n)\}$ of the Generalized-Cube Network .............................................................................................................. 30
    3.2.3 Mathematical Analysis on $E[GG^T|x(n)]$ of the Generalized-Cube Network .............................................................................................................. 32
  3.3 Mathematical Analyses of Shuffle/Exchange-Type Switching Networks used in DEM DACs ................................................................................................................................. 34
    3.3.1 The Omega Network in DEM DACs ............................................................... 34
    3.3.2 The Indirect Binary n-cube network and several other Shuffle/Exchange type Networks used in DEM DACs ................................................................. 37

CHAPTER 4  HARDWARE EFFICIENT DYNAMIC ELEMENT MATCHING ............ 39
  4.1 The Hardware Complexity of Some Interconnection Networks ....................... 39
    4.1.1 The Hardware complexity of the Generalized-Cube Network, ....... 39
the Omega Network, and the Indirect Binary n-cube Network.... 40
4.1.2 The Hardware complexity of the Benes Network .................. 40
4.2 The Compressed Butterfly-Based network.................................. 43
4.3 Hardware Efficiency of The compressed Butterfly-based Network .... 46
4.4 The Similarity of the Compressed Butterfly-based Network and a
Low-complexity Network proposed in [32]........................................... 49

CHAPTER 5 CONCLUSION................................................................................. 53

REFERENCES........................................................................................................... 55

VITA.......................................................................................................................... 58
CHAPTER 1

INTRODUCTION

The use of digital signal processing in applications such as modern audio systems, imaging processing systems and telecommunications systems has increased consistently over the last decades. Consequently, the field of data conversion systems, which provide the link between the analog world and digital signal processing systems, has also rapidly expanded. Data conversion systems perform two kinds of data conversion, Analog-to-Digital conversion and Digital-to-Analog conversion. Analog-to-Digital converters (ADCs) convert the analog signals into digital signals and Digital-to-Analog converters (DACs) convert digital signals into analog signals.

An ideal B-bit DAC transforms a B-bit digital signal, \( x(n) \), into an analog signal \( y(t) \), such that \( y(nT) = qx(n) \), where \( q \) is a constant, \( T \) is the DAC's sampling period and \( n \) is an integer that indexes the sequence \( x \). Therefore, an ideal DAC is a linear system. However, practical DACs contains mismatched circuit components that cause conversion errors in the analog output signal, \( y(t) \), such that \( y(nT) = qx(n) + e(n) \) where \( e(n) \) is a sequence representing the DAC's conversion errors. In general, the transformation mapping \( x(n) \) to \( e(n) \) is a nonlinear transformation, so practical D/A converters are generally nonlinear devices. The nonlinear errors give rise to harmonic
distortion that in many applications is the limiting factor in overall system performance. For example, the harmonic distortion caused by DAC nonlinearities limits the performance of direct digital frequency synthesizers \([1][2]\). The accuracy of delta-sigma A/D converters is also limited by a DAC's nonlinearity \([3]\).

Many DAC architectures rely on matched components to perform data conversion. However, perfectly matched components are impossible to fabricate \([2]\). Component mismatch errors can also be caused by temperature gradients across circuits and component noise \([4][5]\). Component mismatch errors add a nonlinear transformation to the DAC’s ideal linear transformation. As the component mismatching increases, the distortion in the output signal increases and the DAC’s output signal to noise ratio (SNR), signal to distortion ratio (SDR), signal to noise plus distortion ratio (SNDR), and spurious free dynamic range (SFDR) decrease.

Circuit operations and algorithmic techniques have been developed that reduce the effects of component mismatch errors and therefore improve DAC performance. For example, laser wafer trimming and fuse techniques are used to reduce mismatch errors \([6][7][8]\). Self-calibration techniques have been designed to detect mismatch errors and automatically compensate for these errors \([9][10][11]\).

Dynamic element matching (DEM) is another method that can eliminate or reduce the effects of mismatched circuits. DEM techniques dynamically rearrange the interconnections of mismatched components such that the time average of the equivalent components at each component position is nearly equal. Therefore, DEM can improve a DAC’s SNR, SDR, SNDR and SFDR. Both analog and digital DEM techniques have been developed. However, this thesis only discusses digital DEM techniques. Digital
DEM techniques are so named because they rearrange the interconnections of mismatched components using digital circuits and signals.

Many digital DEM techniques use switching networks called interconnection networks [12-16]. DEM DACs have used different interconnection networks to achieve excellent performance. To compare DEM algorithms, criteria have been developed to evaluate the performance of DEM DACs using different interconnection networks [15]. Also, certain efforts have produced low hardware-complexity DEM DACs [17][18].

In this thesis, DEM algorithms that use several classes of interconnection networks and are applied to a specific DEM DAC architecture are analyzed. Performances of these interconnection networks in DEM DACs can be compared using these results and the criteria developed in [15]. Also, a new interconnection network is developed and is shown to be hardware efficient when applied to the specific DEM DAC architecture.
CHAPTER 2

OVERVIEW OF DYNAMIC ELEMENT MATCHING

DIGITAL-TO-ANALOG CONVERSION

In this chapter, the overview of Digital-to-Analog conversion is presented. The basic analysis of digital Dynamic Element Matching technique is discussed and some of the performance criteria are described.

2.1 Overview of Digital to Analog Conversion

An ideal digital-to-analog (D/A) converter transforms a B-bit digital signal, \( x(n) \), into an analog signal, \( y(t) \), such that \( y(nT) = qx(n) \), where \( q \) is a constant that represents the DAC's code width, and \( T \) is the DAC's sampling period. Figure 2.1.1 shows an ideal linear 3-bit DAC's transfer characteristic.

Figure 2.1.2 shows a block diagram of a commonly used B-bit flash DAC architecture, which is also the architecture studied in this thesis. In Figure 2.1.2, the thermometer encoder converts the B-bit input signal, \( x(n) \), into a \( 2^B \)-bit thermometer coded signal, \( t(n) \), where \( t_k(n) \) represents the \( kth \) bit of \( t(n) \). The \( kth \) bit, \( t_k(n) \), controls the \( kth \) unit DAC. When \( t_k = 1 \), the \( kth \) unit DAC outputs the analog value, \( q \),

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 2.1.1 Ideal 3-bit DAC transfer characteristic

Figure 2.1.2 A B-bit Flash DAC
and when \( t_k = 0 \), the \( kth \) unit DAC outputs the analog value, \( 0 \). The outputs of the \( 2^b \) unit DACs are summed to generate the DAC's output, \( y(nT) \).

Practical DACs have circuit mismatch errors that cause the analog outputs of the unit DACs to differ from the identical value \( q \). For example, mismatched resistors in a voltage division DAC's resistor string cause the DAC's code widths to vary from the designed value \( q \). Similarly, mismatch current sources causes nonuniform code widths in current steering DACs.

In a DAC that contains mismatch errors, the DAC's output can be written as

\[
y(nT) = \sum_{k=0}^{x(n)-1} q_k = qx(n) + \epsilon(n),
\]

where \( q_k \) is the analog value that the \( kth \) unit DAC outputs when \( t_k = 1 \), \( q \) is the average of all \( q_k \)s, and \( \epsilon(n) \) is a sequence representing the DAC's output error.

To quantify a DAC's nonlinearity, performance metrics such as code width, INL, DNL, SNR, SDR and SNDR have been developed [19]. A DAC's code width, also called a quantization step size, is the change in analog output when the digital input changes from one code to an adjacent code. Figure 2.1.1 shows an ideal three-bit DAC that has identical code widths. Differential nonlinearity (DNL) is the difference between a DAC's actual quantization step size and its ideal quantization step size. Integral nonlinearity (INL) is the difference between the actual DAC's transfer characteristic and the DAC's ideal transfer characteristic, which is determined by the straight line, \( qx(n) \), where \( q \) is the DAC's average code width. Figure 2.1.3 shows a 3-bit DAC's transfer characteristic and shows its DNL and INL. Manufacturers usually report the maximum absolute value of the DAC's DNL and INL. Signal to noise ratio (SNR) is the ratio of the output signal
power to the total output noise. Signal to distortion ratio (SDR) is the ratio of the output signal power to the total output distortion. Signal to noise plus distortion ratio (SNDR) is the ratio of the output signal power to the total noise plus distortion power at the output.

2.2 Digital Dynamic Element Matching Technique

As mentioned early, Dynamic Element Matching (DEM) is a dynamic process that dynamically rearranges the positions of the mismatched components so that the time averages of the values of the mismatched components are equal or nearly equal. Digital
DEM algorithms often use interconnection networks to rearrange the connections of the mismatched components. In this thesis, several DEM algorithms that use interconnection networks are applied to the flash DAC architecture shown in Figure 2.2.1. To compare the performance of these DEM algorithms when they are applied to the architecture in Figure 2.2.1, performance criteria, such as the Mean INL conditioned on the input $x(n)$, the INL variance conditioned on the input $x(n)$, SDR and SNDR are used [16].

2.2.1 A DEM Flash DAC Architecture

In [16], P. Stubberud and J.W. Bruce analyzed the B bit flash DEM DAC architecture shown in Figure 2.2.1 and developed criteria for comparing DEM algorithms for this architecture. This DAC performs DEM by mapping a B-bit binary input signal, $x(n)$, where $x_0 \leq x(n) \leq x_0 + 2^B - 1$ to the $2^B$ unit DACs [12]. The natural binary...
converter transforms the input signal, $x(n)$, where $x_0 \leq x(n) \leq x_0 + 2^B - 1$, into the $B$-bit natural binary signal, $\chi(n)$, where $\chi(n) = x(n) - x_0$, which implies that $0 \leq \chi(n) \leq 2^B - 1$. The thermometer encoder converts the natural binary coded signal, $\chi(n)$, into a $2^B$-bit thermometer coded signal, $t(n)$. The interconnection network connects the $2^B$ bits of the modified thermometer coded signal, $t(n)$, to the $2^B$ unit DACs. Regardless of the interconnection network’s control signal, $c(n)$, the interconnection network’s output, $g(n)$, activates $\chi(n)$ unit DACs and deactivates the remaining $2^B - \chi(n)$ unit DACs. If each activated unit DAC generated an analog signal, $a$, and each deactivated unit DAC generates an analog signal, $d$, the DAC’s quantization step size, $q$, are the difference between $a$ and $d$, that is $q = a - d$ which is a constant. Also, the DAC’s output, $y(nT)$, which is the sum of all the unit DAC outputs, can be written as

$$y(nT) = a\chi(n) + d[2^B - \chi(n)] = (a - d)\chi(n) + d 2^B = q[x(n) - x_0] + d 2^B \quad (2.2.1)$$

In practice, mismatched components between each of the unit DACs prevent the analog outputs of the activated unit DACs from being identical. Similarly, the analog outputs of the deactivated unit DACs are not identical [12]. As a result, the DEM DAC’s output, $y(nT)$, is not only a function of $x(n)$, but also a function of the interconnection network, the control signal of the interconnection network, $c(n)$, and the activated and deactivated analog outputs generated by each unit DAC.

If the interconnection network’s control signal, $c(n)$, is deterministic, the mapping between the DAC’s input signal, $x(n)$, and the $2^B$ unit DACs is deterministic,
and the DAC is referred to as a deterministic DEM DAC. On the other hand, if \( c(n) \) is stochastic, the mapping will be stochastic and the DAC is referred to as a stochastic DEM DAC. In this thesis, only the performance of stochastic DEM DACs is analyzed.

2.2.2 An Analysis of DEM DAC Architecture

The analysis of the DEM DAC architecture shown in Figure 2.2.1 is derived in [16] and reviewed in this section. Let \( X_k(n) \) represent \( x(n) \)'s \( k \)th bit, where \( X_0(n) \) and \( X_{2^{b-1}}(n) \) are \( x(n) \)'s least significant bit (LSB) and most significant bit (MSB), respectively. Also, let vector, \( T(n) \), represent the thermometer coded signal, \( t(n) \), where

\[
T(n) = [t_0(n), t_1(n), t_2(n), ..., t_{2^{b-1}}(n)]^T,
\]

the superscript \( T \) denotes transpose and \( t_k(n) \) represents \( t(n) \)'s \( k \)th bit. The thermometer code discussed in Chapter 2 and Chapter 3 is defined such that

\[
t_0(n) = t_1(n) = ... = t_{2^{b-1}}(n) = 1.
\]

For example, for \( B = 3 \) and \( x(n) = 4 \),

\[
T(n) = [1, 1, 1, 0, 0, 0, 0]^T.
\]

To simply an interconnection network in Chapter 4, a modified thermometer coder is used such that \( t_0(n) = 0 \) and \( t_{2^{b-1}}(n) = ... = t_{2^{b-1}}(n) = X_{b-1}(n) \). For example, for \( B = 3 \),

\[
T(n) = [0, X_0(n), X_1(n), X_1(n), X_2(n), X_2(n), X_2(n), X_2(n)]^T.
\]

Regardless of the type of thermometer code, \( T(n) \) can be related to \( x(n) \) by

\[
T^T(n)T(n) = \sum_{i=0}^{2^{b-1}} t_i(n) = X(n) = x(n) - x_0
\]
To express the interconnection network's $2^n$-bit output signal, $g(n)$, as a function of the input signal, $x(n)$, let the vector, $G(n)$, represent $g(n)$, such that

$$G(n) = [g_0(n), g_1(n), g_2(n), ..., g_{2^n-1}(n)]^T,$$

where $g_k(n)$ represents $g(n)$'s $k$th bit. Therefore, $G(n)$ is a function of the interconnection network, its control signal, $c(n)$, and the thermometer coded signal, $t(n)$, which is a function of DAC's input $x(n)$. Regardless of the interconnection network and the control signal, $c(n)$,

$$G^T(n)G(n) = \sum_{i=0}^{2^n-1} g_i(n) = T^T(n)T(n) = \sum_{i=0}^{2^n-1} t_i = \chi(n) = x(n) - x_0$$  \hspace{1cm} (2.2.2)

Thus, the interconnection network's output $G(n)$, activates $\chi(n)$, or $x(n) - x_0$ unit DACs and deactivates $2^n - \chi(n)$ or $2^n - x(n) + x_0$ unit DACs.

To express the DAC's output, $y(nT)$, as a function of the interconnection network's output, $G(n)$, define the output, $y_k(nT)$, of the $k$th unit DAC as

$$y_k(nT) = \begin{cases} a_k & g_k(n) = 1 \\ d_k & g_k(n) = 0 \end{cases}$$

where $a_k$ and $d_k$ are the values of $k$th unit DAC's activated and deactivated outputs, respectively. If $\overline{a}$ and $\overline{d}$ are defined as the average values of the activated and deactivated unit DACs, respectively, that is

$$\overline{a} = \frac{1}{2^n} \sum_{k=0}^{2^n-1} a_k$$ \hspace{1cm} (2.2.3)

and
\[
\bar{d} = \frac{1}{2^{\beta}} \sum_{k=0}^{2^{\beta}-1} d_k
\]  

(2.2.4)

then \( \bar{q} \), the DAC's average code width can be written as \( \bar{q} = \bar{a} - \bar{d} \), and the output, \( y_k(nT) \), of the kth unit DAC can be written as

\[
y_k(nT) = \begin{cases} 
\bar{a} + h_k, & g_k(n) = 1 \\
\bar{d} + l_k, & g_k(n) = 0 
\end{cases}
\]

where \( h_k = a_k - \bar{a} \) and \( l_k = d_k - \bar{d} \).

By defining \( H = [h_1, h_2, h_3, \ldots, h_{2^\beta}(n)]^T \), \( L = [l_1, l_2, l_3, \ldots, l_{2^\beta}(n)]^T \), and \( 1 \) as a \( 2^\beta \times 1 \) vector of ones, the DAC's output, \( y(nT) \), can be written as

\[
y(nT) = \sum_{k=0}^{2^\beta-1} y_k(nT) \\
= \bar{a}G^T(n)G(n) + G^T(n)H + \bar{d}[1 - G(n)]^T [1 - G(n)] + [1 - G(n)]^T L \quad (2.2.5)
\]

Using (2.2.2),

\[
[1 - G(n)]^T [1 - G(n)] \\
= 1^T 1 - 2G^T(n)1(n) + G^T(n)G(n) \\
= 2^\beta - 2[x(n) - x_0] + [x(n) - x_0] \\
= 2^\beta - [x(n) - x_0] \quad (2.2.6)
\]

Substituting (2.2.2) and (2.2.6) into (2.2.5),

\[
y(nT) = \bar{a}[x(n) - x_0] + G^T(n)H + \bar{d}[2^\beta - x(n) + x_0] + [1 - G(n)]^T L \\
= (\bar{a} - \bar{d})[x(n) - x_0] + \bar{d}2^\beta + G^T(n)H + [1 - G(n)]^T L \\
= \bar{q}[x(n) - x_0] + \bar{d}2^\beta + G^T(n)(H - L) + 1^T L \quad (2.2.7)
\]

Because \( d_k = \bar{d} + l_k \), (2.2.4) can be written as

\[
\bar{d} = \frac{1}{2^\beta} \sum_{k=0}^{2^\beta-1} (d + l_k) = \frac{1}{2^\beta} \sum_{k=0}^{2^\beta-1} \bar{d} + \frac{1}{2^\beta} \sum_{k=0}^{2^\beta-1} l_k = \bar{d} + \frac{1}{2^\beta} \sum_{k=0}^{2^\beta-1} l_k ,
\]
which implies that

$$\sum_{k=0}^{\infty} l_k = 1^T L = 0 \quad (2.2.8)$$

Similarly, it can be shown that

$$\sum_{k=0}^{\infty} h_k = 1^T H = 0 \quad (2.2.9)$$

Substituting (2.2.8) into (2.2.7),

$$y(nT) = q[x(n) - x_0] + d2^b + G^T(n)(H - L) \quad (2.2.10)$$

In (2.2.10), the term, $q[x(n) - x_0] + d2^b$, is similar to the ideal DAC’s output in (2.2.1) and represents the DAC’s linear output. The other term, $G^T(n)(H - L)$, represents the DAC’s integral nonlinearity (INL). Because $G(n)$ is a function of the interconnection network’s control signal, $c(n)$, and the DAC’s input $x(n)$, (2.2.10) can also be written as

$$y(nT) = q[x(n) - x_0] + d2^b + G^T[x(n), c(n)](H - L)$$

$$= q[x(n) - x_0] + d2^b + T_{\text{INL}}[x(n), c(n)] \quad (2.2.11)$$

where $T_{\text{INL}}$ represents the transformation of the DAC’s INL, that is,

$$T_{\text{INL}}[x(n), c(n)] = G^T[x(n), c(n)](H - L) \quad (2.2.12)$$

To develop SDR and SNDR equations for the DEM DAC’s output, the DC power in the DAC’s digital input, $x(n)$, and the DC power of DAC’s undistorted output, $q[x(n) - x_0] + d2^b$, must be equal. Thus, $x_0 = d2^b / q$, and $q[x(n) - x_0] + d2^b = qx(n)$. Therefore,

$$y(nT) = qx(n) + G^T[x(n), c(n)](H - L) = qx(n) + T_{\text{INL}}[x(n), c(n)] \quad (2.2.13)$$
2.2.3 Performance Criteria for DEM DACs

Using the analysis presented in the previous section, the DEM DAC's mean integral nonlinearity (INL), the variance of the DEM DAC's INL, the DEM DAC's output signal to distortion ratio (SDR), and the DEM DAC's output signal to noise plus distortion ratio (SNDR) can be calculated [16]. These performance criteria are used in Chapter 3 as criteria to compare the performance of DEM DACs using specific interconnection networks and control signals.

To calculate the mean of a stochastic DEM DAC's INL, the expectation operator conditioned on the input signal, \( x(n) \), is applied to the DAC's output in (2.2.13), that is,

\[
E[y(nT) \mid x(n)] = qx(n) + E\{T_{\text{INL}}[x(n), c(n)]\} = qx(n) + E\{G^T[x(n), c(n)] \mid x(n)\}(H - L). \tag{2.2.14}
\]

Using (2.2.14), the variance of the DEM DAC's INL conditioned on the input signal, \( x(n) \), can be written as

\[
\sigma^2_{y|x} = E\{[y - E(y)]^2 \mid x\}
= E\{([G^T \mid x - E(G^T \mid x)](H - L))^2\}
= (H - L)^T [E(GG^T \mid x) - E(G \mid x)E(G^T \mid x)](H - L) \tag{2.2.15}
\]

Two other criteria used to measure a DAC's performance are signal to distortion ratio (SDR) and signal to noise plus distortion ratio (SNDR).

For stochastic DEM DACs, the DAC's average signal plus distortion power, \( P_y \), can be written as

\[
P_y = E[y^2(nT)] = E\{E[y^2(nT) \mid x(n)]\} \tag{2.2.16}
\]

Substituting (2.2.13) into (2.2.16),
\[ P_y = E\left\{ \frac{-2}{q} x^2(n) + 2q x(n) T_{\text{INL}} [x(n), c(n)] + T_{\text{INL}}^2 [x(n), c(n) | x(n)] \right\} \]
\[ = E\left\{ \frac{-2}{q} x^2(n) + 2q E(x(n) T_{\text{INL}} | x(n)) + E[T_{\text{INL}}^2 | x(n)] \right\} \]
\[ = \frac{-q}{P_{st}} E[x^2(n)] + 2q E\{x(n) E[T_{\text{INL}} | x(n)]\} + E[T_{\text{INL}}^2] \]  \hspace{1cm} (2.2.17)

where \( P_{st} \) is the average power of the output of a linear DAC with code widths \( \bar{q} \) and \( P_{sd} \) is the average power of the output’s conversion errors or distortion. Therefore, the DAC’s SDR is

\[ SDR = \frac{P_{st}}{P_{sd}} = \frac{-q}{2q E(x(n) E[T_{\text{INL}} | x(n)]} + E[T_{\text{INL}}^2] \]  \hspace{1cm} (2.2.18)

If the DAC’s input, \( x(n) \), can be written as

\[ x(n) = s(n) + w(n) \]  \hspace{1cm} (2.2.19)

where \( s(n) \) is a signal component, and \( w(n) \) is an independent zero mean white noise component, then the DAC’s SNDR can be calculated by substituting (2.2.19) into (2.2.17),

\[ P_y = \frac{-q}{P_{ys}} E[x^2(n)] + \frac{-q}{P_{yn+d}} E[w^2(n)] + 2q E\{x(n) E[T_{\text{INL}} | x(n)]\} + E[T_{\text{INL}}^2] \]  \hspace{1cm} (2.2.20)

where \( P_{ys} \) is the average power of the output of a linear DAC with code widths \( \bar{q} \) and \( P_{yn+d} \) is the average power of the output’s noise plus distortion. Therefore, the DAC’s SNDR is

\[ \text{SNDR} = \frac{P_{ys}}{P_{yn+d}} = \frac{-q}{q^2 E[w^2(n)] + 2q E\{x(n) E[T_{\text{INL}} | x(n)]\} + E[T_{\text{INL}}^2]} \]  \hspace{1cm} (2.2.21)
For stochastic DEM DACs, interconnection networks are often chosen and controlled such that the DAC's mean transformation represents a linear DAC, that is

\[ E[y(nT) | x(n)] = qx(n) \]  

(2.2.22)

which implies that the stochastic DEM DAC's expected INL for a particular input, \( x(n) \), is zero, that is

\[ E[T_{\text{INL}}[x(n), c(n)] | x(n)] = E[G^T[x(n), c(n)] | x(n)](H - L) = 0 \]  

(2.2.23)

For these types of stochastic DEM DACs, the performance criteria described by (2.2.15), (2.2.18) and (2.2.21) can be simplified. Substituting (2.2.23) into (2.2.15),

\[ \sigma_{y_{x}}^2 = (H - L)^T \{ E(GG^T | x) - E(G | x)E(G^T | x) \}(H - L) \]

\[ = (H - L)^T \{ E(GG^T | x) \}(H - L) \]  

(2.2.24)

Therefore,

\[ \sigma_{y_{x}}^2 = (H - L)^T \{ E(GG^T | x) \}(H - L) = E[T_{\text{INL}}^2 | x]. \]  

(2.2.25)

Substituting (2.2.23) into (2.2.18),

\[ \text{SDR} = \frac{P_{y_{x}}}{P_{y_{d}}} = \frac{-\frac{1}{q}E[x^2(n)]}{E[T_{\text{INL}}^2]} \]  

(2.2.26)

Substituting (2.2.23) into (2.2.21),

\[ \text{SNDR} = \frac{P_{y_{x}}}{P_{y_{n+d}}} = \frac{-\frac{1}{q}E[s^2(n)]}{\frac{1}{q}E[w^2(n)] + E[T_{\text{INL}}^2]} \]  

(2.2.27)

where \( E[T_{\text{INL}}^2] = E\{E[T_{\text{INL}}^2 | x] \} = E[\sigma_{y_{x}}^2] \).
MATHEMATICAL ANALYSES OF INTERCONNECTION NETWORK EFFECTS ON STOCHASTIC DYNAMIC ELEMENT MATCHING

DEM algorithms are often implemented using interconnection networks [6-9] [19]. To compare the performance of DEM algorithms using the criteria discussed in Chapter 2 and developed in [16], the interconnection networks must be analyzed. As discussed in Chapter 2, interconnection networks in stochastic DEM DACs are typically chosen and controlled to satisfy (2.2.23), that is

$$E\{T_{\text{INL}}[x(n),c(n)] \mid x(n)\} = E\{G^T[x(n),c(n)] \mid x(n)\}(H - L) = 0.$$ 

Therefore, $E\{T_{\text{INL}}[x(n),c(n)] \mid x(n)\}$ for each interconnection network needs to be calculated.

Assuming that the DEM algorithm is controlled such that (2.2.23) is satisfied, the DEM algorithm can be evaluated using (2.2.25) which is

$$\sigma^2_{\text{INL}} = (H - L)^T [E(GG^T \mid x)](H - L) = E[T_{\text{INL}}^2],$$

(2.2.26) which is

$$SDR = \frac{P_{\text{INL}}}{P_{\text{dy}}} = \frac{\sigma^2_{\text{INL}}}{E[T_{\text{INL}}^2]}.$$
and (2.2.27) which is

\[
\text{SNDR} = \frac{P_{y_1}}{P_{y_{add}}} = \frac{-qE[s^2(n)]}{q^2E[w^2(n)] + E[T_{I_{NL}}^2]}.
\]

Because each of these criteria is a function of \(E[T_{I_{NL}}^2]\), which is a function of

\[E(GG^T[x(n),c(n)]|x(n)), E(GG^T[x(n),c(n)]|x(n))\] for each interconnection network needs to be calculated to use these criteria.

In this chapter, \(E(T_{I_{NL}}[x(n),c(n)]|x(n))\) and \(E(GG^T[x(n),c(n)]|x(n))\) are calculated for the Barrel Shifting network, the Generalized-Cube network and the Indirect Binary n-cube network. Using these formulas, DEM algorithms using these interconnection networks can be compared. In this thesis, the control signal \(c(n)\) is assumed to be a B-bit stochastic signal with a uniform pdf, that is

\[P[c(n) = m] = \begin{cases} 1/2^B, & 0 \leq m \leq 2^B - 1 \\ 0, & \text{otherwise} \end{cases}
\]

where \(P(.)\) is the probability function.

3.1 A Mathematical Analysis of stochastic controlled Barrel Shifting networks used in DEM DACs

The Barrel Shifting network is so called because it performs circular shifts. A B-bit Barrel Shifting DEM DAC uses a \(2^B\)-port wrap-around shifter as the interconnection network in Figure 2.2.1. Figure 3.1.1 shows a thermometer coder and a B-bit Barrel Shifting network. In Figure 3.1.1, \(x(n)\) is the B-bit input signal to the thermometer.
coder, where $\chi(n) = x(n) - x_0$ and $0 \leq \chi(n) \leq 2^b - 1$, and $t(n)$ is the $2^b$-bit output of the thermometer coder and is the $2^b$-bit input to the Barrel Shifting network. The Barrel Shifting network's output signal, $g(n)$, is also a $2^b$-bit signal. Because $G(n)$ is a circular shift of the Barrel Shifting network's input signal, $T(n)$, the Barrel Shifting network's output can be written as,

$$g_i = t_{(i-c(n)) \mod 2^b}$$

for $0 \leq i \leq 2^b - 1$ (3.1.2)

### 3.1.1 Representation of Barrel Shifting network Outputs
To calculate $E\{T_{N^L}[x(n), c(n)] \mid x(n)\}$ and $E\{GG^T[x(n), c(n)] \mid x(n)\}$ for the Barrel Shifting network, its output, $G(n)$, needs to be represented as a function of the DAC's input signal, $x(n)$, and the control signal, $c(n)$. In this section, $G(n)$ is expressed as a function of $\chi(n)$, where $\chi(n) = x(n) - x_0$, and $c(n)$.

(3.1.2) can be written as

$$g_i(n) = \sum_{k=0}^{2^b-1} \delta(k - c(n))$$

for $0 \leq i \leq 2^b - 1$.

where $\delta(k)$ is the unit impulse sequence, and $\tilde{t}_k(n)$ and $\tilde{\delta}(k)$ represent the $2^b$ point periodic extensions of $t_k(n)$ and $\delta(k)$ in the k-dimension, respectively. Letting $m = i - k$, (3.1.3) can be written as

$$g_i(n) = \sum_{m=0}^{2^b-1} \tilde{t}_m(n) \tilde{\delta}(i - c(n) - m)$$

(3.1.4)

Because $t_m(n) = \tilde{t}_m(n)$ for $0 \leq m \leq 2^b - 1$, (3.1.4) can be written as

$$g_i(n) = \sum_{m=0}^{2^b-1} t_m(n) \delta(i - c(n) - m)$$

(3.1.5)

Because $T(n)$ is a $2^b$-bit thermometer coded signal that equals the Natural Binary Encoder's B-bit output signal, $\chi(n)$, where $0 \leq \chi(n) \leq 2^b - 1$,

$$t_k(n) = \begin{cases} 1, k \leq \chi(n) - 1 \\ 0, k > \chi(n) - 1 \end{cases}$$

for $0 \leq k \leq 2^b - 1$.

(3.1.6) can also be written as

$$t_k(n) = u(\chi(n) - 1 - k)u(k)$$

(3.1.7)
where \( u(k) \) is unit step sequence. Figure 3.1.2 shows \( t_k(n) \) as a function of \( x(n) \).

\[ \begin{array}{c}
0 & 1 & 2 & \cdots & \cdots & \cdots & \cdots & k \\
\chi(n) & -1 & \cdots & \cdots & \cdots & \cdots & \cdots & k
\end{array} \]

Figure 3.1.2 \( t_k(n) = u(\chi(n) - 1 - k)u(k) \)

Substituting (3.1.7) into (3.1.5),

\[
g_1(n) = \sum_{m=0}^{2^n-1} u(\chi(n) - 1 - m)u(m)\delta(i - c(n) - m) \\
= \sum_{m=0}^{\chi(n) - 1} \delta(i - c(n) - m)
\]

which expresses \( g_1(n) \) as a function of \( x(n) \) and \( c(n) \).

3.1.2 Mathematical Analysis on \( E\{T_{\text{inv}}[x(n), c(n)]|x(n)\} \) of Barrel Shifting network

Because

\[
T_{\text{inv}}[x(n), c(n)] = G^T[x(n), c(n)](H - L),
\]

where \( H \) and \( L \) are not a function of \( x(n) \),

\[
E\{T_{\text{inv}}[x(n), c(n)]|x(n)\} = E\{G^T[x(n), c(n)](H - L)|x(n)\} \\
= E\{G^T[x(n), c(n)]|x(n)\}(H - L).
\]

To calculate \( E\{G^T[x(n), c(n)]|x(n)\} \), the expectation operator conditioned on \( x(n) \) is applied to (3.1.8). This yields
\[ E(\text{g}_i(n) \mid x(n)) = E\{ \sum_{m=0}^{\chi(n)-1} \delta(i - c(n) - m) \mid x(n)\} \]

\[ = \sum_{m=0}^{\chi(n)-1} E\{ \delta(i - c(n) - m) \mid x(n)\} \]

\[ = \sum_{m=0}^{\chi(n)-1} \frac{1}{2^b} \]

\[ = \frac{\chi(n)}{2^b} \]  

Therefore,

\[ E(\mathbf{G}^T [x(n), c(n)] \mid x(n)) = \left[ \frac{\chi(n)}{2^b}, \frac{\chi(n)}{2^b}, \frac{\chi(n)}{2^b}, \ldots, \frac{\chi(n)}{2^b} \right] \]

\[ = \frac{\chi(n)}{2^b} \mathbf{1}^T. \]  

(3.1.11)

Substituting (3.1.11) into (3.1.9),

\[ E(T_{\text{INL}} [x(n), c(n)] \mid x(n)) = E(\mathbf{G}^T [x(n), c(n)] \mid x(n))(\mathbf{H} - \mathbf{L}) \]

\[ = \frac{\chi(n)}{2^b} \mathbf{1}^T (\mathbf{H} - \mathbf{L}) \]

\[ = \frac{\chi(n)}{2^b} \mathbf{1}^T \mathbf{H} - \frac{\chi(n)}{2^b} \mathbf{1}^T \mathbf{L} \]  

(3.1.12)

Substituting (2.2.8) and (2.2.9) into (3.1.12),

\[ E(T_{\text{INL}} [x(n), c(n)] \mid x(n)) = \frac{\chi(n)}{2^b} \mathbf{0} - \frac{\chi(n)}{2^b} \mathbf{0} = 0 \]  

(3.1.13)

Therefore, (2.2.23) is satisfied for DEM DACs using the Barrel Shifting network when \( c(n) \) is a stochastic signal with a uniform pdf.

3.1.3 Mathematical Analysis on \( E[\mathbf{GG}^T \mid x(n)] \) of Barrel Shifting network
Because $G(n) = [g_0, g_1, g_2 \cdots g_{2^b-1}]^T$, $GG^T$ can be written as

$$GG^T = [g_0, g_1, g_2 \cdots g_{2^b-1}]^T [g_0, g_1, g_2 \cdots g_{2^b-1}] = [g_i g_j]_{2^b \times 2^b},$$

(3.1.14)

where $[g_i g_j]_{2^b \times 2^b}$ represents the $2^b \times 2^b$ matrix where $g_i g_j$ is the element in the $i+1$st row and the $j+1$st column for $0 \leq i \leq 2^b-1$ and $0 \leq j \leq 2^b-1$. Therefore,

$$E[GG^T | x(n)] = E\left\{ [g_i g_j]_{2^b \times 2^b} | x(n) \right\} = [E(g_i g_j | x(n))]_{2^b \times 2^b}$$

(3.1.15)

Using (3.1.3),

$$g_i(n) g_j(n) = \sum_{m=0}^{2^b-1} \tilde{t}_{i-m}(n) \delta(m - c(n)) \sum_{k=0}^{2^b-1} \tilde{t}_{j-k}(n) \delta(k - c(n))$$

$$= \sum_{m=0}^{2^b-1} \tilde{t}_{i-m}(n) \tilde{t}_{j-m}(n) \delta(m - c(n))$$

(3.1.16)

Letting $l = i - m$, (3.1.16) can be written as

$$g_i(n) g_j(n) = \sum_{l=i}^{2^b-1} \tilde{t}_i(n) \tilde{t}_{j-i+l}(n) \delta(i - l - c(n))$$

$$= \sum_{l=0}^{2^b-1} \tilde{t}_i(n) \tilde{t}_{j-i+l}(n) \delta(i - l - c(n))$$

(3.1.17)

Applying the expectation operator to (3.1.17) conditioned on $x(n)$,

$$E[g_i(n) g_j(n) | x(n)] = E\left\{ \sum_{l=i}^{2^b-1} \tilde{t}_i(n) \tilde{t}_{j-i+l}(n) \delta(i - l - c(n)) | x(n) \right\}$$

$$= \sum_{l=0}^{2^b-1} \tilde{t}_i(n) \tilde{t}_{j-i+l}(n) E\{\delta(i - l - c(n)) | x(n)\}$$

$$= \frac{1}{2^b} \sum_{l=0}^{2^b-1} \tilde{t}_i(n) \tilde{t}_{j-i+l}(n)$$

(3.1.18)
Because $E[GG^T|x(n)]$ is a symmetric matrix, only the case where $i \leq j$ needs to be discussed. If $i \leq j$ and $k = j - i$, (3.1.18) can be written as

$$E[g_i(n)g_{i+k}(n)\mid x(n)] = \frac{1}{2^b} \sum_{l=0}^{2^b-1} t_i(n)\tilde{t}_{i+k}(n)$$

$$= \frac{1}{2^b} \sum_{l=0}^{2^b-1} t_i(n)\tilde{t}_{i+k}(n)$$

(3.1.19)

The summation in (3.1.19) can be calculated by determining the nonzero samples of $\tilde{t}_{k-l}(n)$, for $0 \leq l \leq \chi(n)-1$. $\tilde{t}_{k-l}(n)$'s starting point and ending point of non-zero samples depend on the comparison of $\chi(n)$ and $k$. If $k < \chi(n)$, $\tilde{t}_{k-l}(n)$ is shifted so that $\tilde{t}_{k-l}(n) = 1$ for $0 \leq l \leq \chi(n)-1-k$ or $2^b - k \leq l \leq 2^b - 1$. Therefore, if $k < \chi(n)$ and $\chi(n)-1 < 2^b - k$,

$$E[g_i(n)g_{i+k}(n)\mid x(n)] = \frac{1}{2^b} \sum_{l=0}^{\chi(n)-1-k} t_i(n)\tilde{t}_{i+k}(n)$$

$$= \frac{1}{2^b} \sum_{l=0}^{\chi(n)-1-k} 1$$

$$= \frac{\chi(n) - k}{2^b}.$$

(3.1.20)

Figure 3.1.3 (a) illustrates this case.

If $k < \chi(n)$ and $\chi(n)-1 \geq 2^b - k$,

$$E[g_i(n)g_{i+k}(n)\mid x(n)] = \frac{1}{2^b} \sum_{l=0}^{\chi(n)-1} t_i(n)\tilde{t}_{i+k}(n)$$

$$= \frac{1}{2^b} \left( \sum_{l=0}^{\chi(n)-1-k} 1 + \sum_{l=2^b-k}^{\chi(n)-1} 1 \right)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 3.1.3 $t_l(n)$ and $\tilde{t}_{l+k}(n)$ when $0 \leq l \leq 2^b + 1$

(a) $k < \chi(n)$ and $\chi(n) - 1 < 2^b - k$  
(b) $k < \chi(n)$ and $\chi(n) - 1 \geq 2^b - k$  
(c) $k \geq \chi(n)$ and $\chi(n) - 1 \geq 2^b - k$  
(d) $k \geq \chi(n)$ and $\chi(n) - 1 < 2^b - k$

\[
= \frac{2\chi(n) - 2^b}{2^b}. \quad (3.1.21)
\]

Figure 3.1.3(b) illustrates this case.

If $k \geq \chi(n)$, $\tilde{t}_{l+k}(n)$ is shifted so that $\tilde{t}_{l+k}(n) = 1$ for $2^b - k \leq l \leq 2^b + \chi(n) - 1 - k$. Therefore, if $k \geq \chi(n)$ and $\chi(n) - 1 \geq 2^b - k$,

\[
E[g_l(n)g_{l+k}(n) \mid x(n)] = \frac{1}{2^b} \sum_{l=0}^{\chi(n) - 1} t_l(n)\tilde{t}_{l+k}(n)
\]
\[
\left( \sum_{i=0}^{\chi(n)-1} \frac{X(n) + k - 2^b}{2^b} \right)
\]

Figure 3.1.3(c) illustrates this case.

If \( k \geq \chi(n) \) and \( \chi(n) - 1 \geq 2^b - k \),

\[
E[g_i(n)g_{i+k}(n) | x(n)] = \frac{1}{2^b} \sum_{i=0}^{\chi(n)-1} t_i(n) t_{i+k}(n)
\]

\[= 0. \quad (3.1.23)\]

Figure 3.1.3(d) illustrates this case.

Therefore,

\[
E(g_i g_{i+k} | x(n)) = \begin{cases} 
\frac{\chi(n) - k}{2^b} & \text{if } k < \chi(n) \text{ and } \chi(n) - 1 < 2^b - k \\
\frac{2\chi(n) - 2^b}{2^b} & \text{if } k < \chi(n) \text{ and } \chi(n) - 1 \geq 2^b - k \\
\frac{\chi(n) + k - 2^b}{2^b} & \text{if } k \geq \chi(n) \text{ and } \chi(n) - 1 \geq 2^b - k \\
0 & \text{otherwise}
\end{cases}
\]

\[= \begin{cases} 
\frac{\chi(n) - k}{2^b} & \text{if } k < \chi(n) \leq 2^b - k \\
\frac{2\chi(n) - 2^b}{2^b} & \text{if } \chi(n) > k \text{ and } \chi(n) > 2^b - k \\
\frac{\chi(n) + k - 2^b}{2^b} & \text{if } 2^b - k < \chi(n) \leq k \\
0 & \text{otherwise}
\end{cases} \quad (3.1.24)\]

Substituting (3.1.24) into (3.1.15), \( E(\mathbf{G}^\mathbf{G}^T | x) \) matrix can be calculated to evaluate the performance of Barrel Shifting DEM DACs.
3.2 A Mathematical Analysis of the Generalized-Cube Network used in DEM DACs

A B-bit Generalized-Cube network (GCN) [20-23] is a multistage cube-type network that has \(2^B\) inputs, \(2^B\) outputs and \(B\) stages where each stage consists of a set of \(2^B\) lines (links) connected to \(2^{B-1}\) binary switches. Each binary switch is a two-input, two-output device, which has two states, straight and swap, as shown in Figure 3.2.1. Figure 3.2.2 shows an example of the Generalized-Cube network where \(B = 3\). The inputs are labeled 0 to \(2^B - 1\) from top to bottom, and the labels on the upper and lower inputs of each switch are also used as labels for its upper and lower outputs, respectively. The switches are labeled 'A' to 'L'.

![Figure 3.2.1 Two states of the switch: straight when \(c = 0\) and swap when \(c = 1\)](image)

The characteristic of the Generalized-Cube Network is that the binary representations of the upper and lower labels of a switch in stage \(i\) differ only in the \(i\)th bit position. For example, in Figure 3.2.2, switch A is in stage 3 and its inputs, 0 = (000)\(_2\) and 4 = (100)\(_2\), differ only in the 3\(^\text{rd}\) bit. As a result of this characteristic, the control signal to connect an input \(i = (i_B i_{B-1} \ldots i_2 i_1)\(_2\) to an output \(j = (j_B j_{B-1} \ldots j_2 j_1)\(_2\) can
be determined by performing an XOR operation on each bit of the input and each corresponding bit of the output, that is,

\[ c_k = i_k \oplus j_k. \] (3.2.1)

To illustrate how the Generalized-Cube network generates different permutations, consider the three-stage generalized-cube network in Figure 3.2.3 where a connection from source \( 2 = (010)_2 \) to destination \( 4 = (100)_2 \) is highlighted. Because \( (010)_2 \oplus (100)_2 = (110)_2 \), stage 1 is set to straight and stages 3 and 2 are set to swap.
Another characteristic of the Generalized-Cube Network is that it is a full access network, which implies that a connection exists such that any input signal \( i \), for \( 0 \leq i \leq 2^a - 1 \), can be connected to any output signal \( j \), for \( 0 \leq j \leq 2^a - 1 \).

### 3.2.1 Representation of the Generalized-Cube Network Outputs

When the Generalized-cube network is used in a DEM DAC, the input to the Generalized-Cube network is the \( 2^a \times 1 \) vector, \( T(n) \), where \( T(n) = [t_0, t_1, t_2, \ldots, t_{2^a - 1}]^T \).
and the output of the Generalized-Cube Network is the \(2^B \times 1\) vector, \(G(n)\), where 
\[
G(n) = [g_0, g_1, g_2 \cdots g_{2^B-1}]^T.
\]

Because the Generalized-cube network is a full access network, the output bit, \(g_i\),
where \(0 \leq i \leq 2^B - 1\), can be expressed as a function of all input bits and control bits. To
determine this function, binary notation for the input and output labels is used. The input
bit \(i_k\) where \(0 \leq i \leq 2^B - 1\) is represented by \(i_{[i_k, \cdots, i_1]}\), where \(i_k\) is the \(k\)th bit of \(i\), and
the output bit \(g_j\) where \(0 \leq j \leq 2^B - 1\) is represented by \(g_{[j_B, j_{B-1}, \cdots, j_1]}\), where \(j_k\) is the \(k\)th
bit of \(j\). Similarly, the control signal \(c(n)\) is represented by \((c_B c_{B-1} \cdots c_1)\), where \(c_k\) is
the \(k\)th bit of \(c(n)\). In a GCN network, \(g_{[j_B, j_{B-1}, \cdots, j_1]} = t_{[i_k, \cdots, i_1]}\) when \(c_k = i_k \oplus j_k\) for
\[1 \leq k \leq B\]. Therefore,
\[
g_j = g_{[j_B, j_{B-1}, \cdots, j_1]}^0 = t_{(c_B \oplus j_B, c_{B-1} \oplus j_{B-1}, \cdots, c_1 \oplus j_1)}.
\] (3.2.2)

For example, if \(B = 3\), control signal 110 maps \(t_{110}\) to \(g_{000}\) and control signal 101 maps
\(t_{011}\) to \(g_{110}\). (3.2.2) can also be written as
\[
g_{[j_B, j_{B-1}, \cdots, j_1]}^0 = \sum_{i_B=0}^{1} \sum_{i_{B-1}=0}^{1} \cdots \sum_{i_1=0}^{1} t_{[i_B, i_{B-1}, \cdots, i_1]} \delta(c_B - (i_B \oplus j_B), c_{B-1} - (i_{B-1} \oplus j_{B-1}), \cdots, c_1 - (i_1 \oplus j_1))
\] (3.2.3)

where \(\delta(k_B, k_{B-1}, \cdots, k_1)\) is the \(B\)-dimensional unit impulse sequence.

3.2.2 Mathematical Analysis of \(E\{T_{INL}[x(n), c(n)] \mid x(n)\}\) for the Generalized-Cube
Network
To calculate $E\{T_{bl}(x(n), c(n)) | x(n)\}$ for a DEM DAC that uses a Generalized-Cube Network, the expectation operator conditioned on the input signal, $x(n)$, is applied to (3.2.3). This yields

$$
E [g_{(i_1, i_2, \cdots, i_{\beta})} | x(n)]
= E \left[ \sum_{i_g=0}^{\beta} \sum_{i_{i_g}=0}^{1} \cdots \sum_{i_{l_{i_{i_g}}}=0}^{1} t_{(i_{i_g}, i_{i_{i_g}}), (i_{l_{i_{i_g}}}, i_{l_{i_{i_g}}})} \delta[c_{\beta} - (i_\beta \oplus j_\beta), c_{\beta-1} - (i_{\beta-1} \oplus j_{\beta-1}), \cdots, c_1 - (i_1 \oplus j_1)] | x(n) \right]
= \sum_{i_g=0}^{\beta} \sum_{i_{i_g}=0}^{1} \cdots \sum_{i_{l_{i_{i_g}}}=0}^{1} t_{(i_{i_g}, i_{i_{i_g}}), (i_{l_{i_{i_g}}}, i_{l_{i_{i_g}}})} E [\delta[c_{\beta} - (i_\beta \oplus j_\beta), c_{\beta-1} - (i_{\beta-1} \oplus j_{\beta-1}), \cdots, c_1 - (i_1 \oplus j_1)] | x(n)]
$$

(3.2.4)

Because the control signal is assumed to be a stochastic signal with a uniform pdf,

$$
P(c_i = 1) = P(c_i = 0) = \frac{1}{2} \quad \text{which implies that for any fixed} \ (k_{\beta}, k_{\beta-1}, \ldots, k_1),$$

$$
E [\delta(c_{\beta} - k_{\beta}, c_{\beta-1} - k_{\beta-1}, \ldots, c_1 - k_1)] = \frac{1}{2^\beta}.
$$

(3.2.5)

Using (3.2.5), (3.2.4) can be written as

$$
E (g_{(i_1, i_2, \cdots, i_{\beta})} | x(n)) = \frac{1}{2^\beta} \sum_{i_g=0}^{\beta} \sum_{i_{i_g}=0}^{1} \cdots \sum_{i_{l_{i_{i_g}}}=0}^{1} t_{(i_{i_g}, i_{i_{i_g}}), (i_{l_{i_{i_g}}}, i_{l_{i_{i_g}}})} (n)
= \frac{1}{2^\beta} \sum_{i=0}^{2^\beta-1} t_i (n)
$$

(3.2.6)

Substituting (2.2.2) into (3.2.6),

$$
E (g_j | x(n)) = E (g_{(i_1, i_2, \cdots, i_{\beta})} | x(n)) = \frac{1}{2^\beta} \sum_{i=0}^{2^\beta-1} t_i (n) = \frac{1}{2^\beta} \chi (n)
$$

(3.2.7)

Therefore,

$$
E [G^T | x(n)] = E [g_0, g_1, g_2, \cdots, g_{2^\beta-1} | x(n)] = \frac{\chi (n)}{2^\beta} 1^T
$$

(3.2.8)

Substituting (3.2.8) into (2.2.23),
\[ E(T_{\text{INL}}[x(n), c(n)] | x(n)) = E(G^T [x(n), c(n)] | x(n)) (H - L) \]

\[ = \frac{X(n)}{2^\alpha} 1^T (H - L) \]

\[ = \frac{X(n)}{2^\alpha} 1^T H - \frac{X(n)}{2^\alpha} 1^T L \quad (3.2.9) \]

Substituting (2.2.8) and (2.2.9) into (3.2.9),

\[ E(T_{\text{INL}}[x(n), c(n)] | x(n)) = \frac{X(n)}{2^\alpha} 0 - \frac{X(n)}{2^\alpha} 0 = 0 \quad (3.2.10) \]

which implies that (2.2.23) is satisfied.

### 3.2.3 Mathematical Analysis of \( E[GG^T | x(n)] \) of the Generalized-Cube Network

To calculate \( E[GG^T | x(n)] \), \( g_j g_k \) needs to be expressed as a function of the input signal and the control signal. Using (3.2.2), \( g_j \) can be written as

\[ g_j = g_{(a_j b_{a+1} \ldots b_{a+1})_2} \]

\[ = t_{(c_r \oplus a_j, c_r \oplus b_{a+1} \ldots c_r \oplus b_{a+1})_2} \cdot \]

Similarly,

\[ g_k = g_{(a_k a_{a+1} \ldots a_{a+1})_2} \]

\[ = t_{(c_r \oplus a_k, c_r \oplus a_{a+1} \ldots c_r \oplus a_{a+1})_2} \cdot \]

Therefore,

\[ g_j g_k \]

\[ = g_{(a_j b_{a+1} \ldots b_{a+1})_2} g_{(a_k a_{a+1} \ldots a_{a+1})_2} \]

\[ = t_{(c_r \oplus a_j, c_r \oplus b_{a+1} \ldots c_r \oplus b_{a+1})_2} t_{(c_r \oplus a_k, c_r \oplus a_{a+1} \ldots c_r \oplus a_{a+1})_2} \cdot \quad (3.2.11) \]

Using (3.2.2) and (3.2.3),
\[ t_{(c_B \oplus j_B, \ldots, c_j \oplus j_j)_2} \]

\[ = \sum_{l_g=0}^{l_g} \sum_{l_{g-1}=0}^{l_{g-1}} \cdots \sum_{l_2=0}^{l_2} t\left( i_g \oplus j_g, i_{g-1} \oplus j_{g-1}, \ldots, i_2 \oplus j_2, i_1 \oplus j_1 \right)_2 \delta \left[ c_B - (i_B \oplus j_B), c_{B-1} - (i_{B-1} \oplus j_{B-1}), \ldots, c_1 - (i_1 \oplus j_1) \right]. \] (3.2.12)

and

\[ t_{(c_B \oplus j_B, \ldots, c_j \oplus j_j)_2} \]

\[ = \sum_{l_g=0}^{l_g} \sum_{l_{g-1}=0}^{l_{g-1}} \cdots \sum_{l_2=0}^{l_2} \sum_{l_1=0}^{l_1} t\left( i_g \oplus j_g, i_{g-1} \oplus j_{g-1}, \ldots, i_2 \oplus j_2, i_1 \oplus j_1 \right)_2 \delta \left[ c_B - (i_B \oplus k_B), c_{B-1} - (i_{B-1} \oplus k_{B-1}), \ldots, c_1 - (i_1 \oplus k_1) \right]. \] (3.2.13)

Letting \( l_i = i_m \oplus j_m \oplus k_m \), which implies \( i_m = l_m \oplus j_m \oplus k_m \), for \( 1 \leq m \leq B \),

\[ t_{(c_B \oplus j_B, \ldots, c_j \oplus j_j)_2} \]

\[ = \sum_{l_g=0}^{l_g} \sum_{l_{g-1}=0}^{l_{g-1}} \cdots \sum_{l_2=0}^{l_2} \sum_{l_1=0}^{l_1} t\left( i_g \oplus j_g, i_{g-1} \oplus j_{g-1}, \ldots, i_2 \oplus j_2, i_1 \oplus j_1 \right)_2 \delta \left[ c_B - (l_B \oplus j_B), c_{B-1} - (l_{B-1} \oplus j_{B-1}), \ldots, c_1 - (l_1 \oplus j_1) \right]. \] (3.2.14)

Substituting (3.2.12) and (3.2.13) into (3.2.11),

\[ g_j g_k = \sum_{l_g=0}^{l_g} \sum_{l_{g-1}=0}^{l_{g-1}} \cdots \sum_{l_2=0}^{l_2} \sum_{l_1=0}^{l_1} t\left( i_g \oplus j_g, i_{g-1} \oplus j_{g-1}, \ldots, i_2 \oplus j_2, i_1 \oplus j_1 \right)_2 \delta \left[ c_B - (i_B \oplus j_B), c_{B-1} - (i_{B-1} \oplus j_{B-1}), \ldots, c_1 - (i_1 \oplus j_1) \right] \]

\[ \sum_{l_g=0}^{l_g} \sum_{l_{g-1}=0}^{l_{g-1}} \cdots \sum_{l_2=0}^{l_2} \sum_{l_1=0}^{l_1} t\left( i_g \oplus j_g, i_{g-1} \oplus j_{g-1}, \ldots, i_2 \oplus j_2, i_1 \oplus j_1 \right)_2 \delta \left[ c_B - (l_B \oplus j_B), c_{B-1} - (l_{B-1} \oplus j_{B-1}), \ldots, c_1 - (l_1 \oplus j_1) \right] \]

\[ = \sum_{l_g=0}^{l_g} \sum_{l_{g-1}=0}^{l_{g-1}} \cdots \sum_{l_2=0}^{l_2} \sum_{l_1=0}^{l_1} t\left( i_g \oplus j_g, i_{g-1} \oplus j_{g-1}, \ldots, i_2 \oplus j_2, i_1 \oplus j_1 \right)_2 \delta \left[ c_B - (l_g \oplus j_g), c_{B-1} - (l_{B-1} \oplus j_{B-1}), \ldots, c_1 - (l_1 \oplus j_1) \right] \]

Therefore,

\[ E[g_j g_k | x(n)] = E[g_{(j_B, j_{B-1}, \ldots, j_2, j_1)} g_{(k_B, k_{B-1}, \ldots, k_2, k_1)} | x(n)] \]
\[ \begin{aligned} &= E \left\{ \sum_{i_B=0}^{1} \sum_{i_B=0}^{1} \ldots \sum_{i_B=0}^{1} t_{(i_B \oplus j_B) \oplus k_B \oplus j_B, i_B \oplus j_B, i_B \oplus j_B, i_B \oplus j_B} \right\} \\
\delta[c_B - (i_B \oplus j_B), c_{B-1} - (i_{B-1} \oplus j_{B-1}), \ldots, c_1 - (i_1 \oplus j_1) \mid x(n)] \\
&= \sum_{i_B=0}^{1} \sum_{i_B=0}^{1} \ldots \sum_{i_B=0}^{1} t_{(i_B \oplus j_B) \oplus k_B \oplus j_B, i_B \oplus j_B, i_B \oplus j_B, i_B \oplus j_B} \right\} \\
E \{ \delta[c_B - (i_B \oplus j_B), c_{B-1} - (i_{B-1} \oplus j_{B-1}), \ldots, c_1 - (i_1 \oplus j_1)] \mid x(n) \} \\
&= \sum_{i_B=0}^{1} \sum_{i_B=0}^{1} \ldots \sum_{i_B=0}^{1} t_{(i_B \oplus j_B) \oplus k_B \oplus j_B, i_B \oplus j_B, i_B \oplus j_B, i_B \oplus j_B} \right\} \\
\delta[c_B - (i_B \oplus j_B), c_{B-1} - (i_{B-1} \oplus j_{B-1}), \ldots, c_1 - (i_1 \oplus j_1)] \mid x(n) \} \tag{3.2.15} \end{aligned} \]

Substituting (3.2.5) into (3.2.15),

\[ E[g_j g_k \mid x(n)] = \frac{1}{2^B} \sum_{i_B=0}^{1} \sum_{i_B=0}^{1} \ldots \sum_{i_B=0}^{1} t_{(i_B \oplus j_B) \oplus k_B \oplus j_B, i_B \oplus j_B, i_B \oplus j_B, i_B \oplus j_B} \right\} \\
\delta[c_B - (i_B \oplus j_B), c_{B-1} - (i_{B-1} \oplus j_{B-1}), \ldots, c_1 - (i_1 \oplus j_1)] \mid x(n) \} \tag{3.2.16} \]

where \( j = (j_B j_{B-1} \ldots j_2 j_1)_2 \) and \( k = (k_B k_{B-1} \ldots k_2 k_1)_2 \).

3.3 Mathematical Analyses of Shuffle/Exchange -Type Switching Networks used in DEM DACs

Shuffle/Exchange class networks are interconnection networks that are constructed of repeated copies of shuffle connections followed by a column of binary switches. Shuffle/Exchange class networks include the Omega Network and the Indirect Binary n-cube Network.

3.3.1 The Omega Network in DEM DACs
The Omega network was first introduced by Lawrie [24]. Similar to generalized-cube network topology, a B-bit Omega network has B stages, where each stage consists of a set of shuffles followed by exchange switches. However, the Omega network uses perfect shuffles.

A perfect shuffle operator maps the source with label \( x = (x_B x_{B-1} \ldots x_2 x_1) \) to the destination with label \( y \) such that

\[
y = \text{shuffle}(x) = \text{shuffle}((x_B x_{B-1} \ldots x_2 x_1)_2) = (x_{B-1} x_{B-2} \ldots x_1 x_B)_2,
\]

which is equivalent to assigning the label \( y \) to be the left circular shift of label \( x \). The inverse perfect shuffle, \( \text{shuffle}^{-1} \), can be defined as

\[
\text{shuffle}^{-1}(x) = \text{shuffle}^{-1}((x_B x_{B-1} \ldots x_2 x_1)_2) = (x_1 x_B x_{B-1} \ldots x_2 x_1)_2
\]

The exchange operations between perfect shuffles are defined as

\[
\text{exchange}((x_B x_{B-1} \ldots x_2 x_1)_2) = (x_B x_{B-1} \ldots x_2 x_1)_2
\]

and occur when the control signal is set to 1.

Figure 3.3.1 shows an 8-input / 8-output Omega Network with 3 stages. In Figure 3.3.1, the network’s inputs and outputs and the switch inputs and outputs for each stage are labeled from 0 to \( 2^8 - 1 \) based on their vertical ordering. The switches in Figure 3.3.1 are labeled from A to L.

In [20-22], Siegel and Smith described the connections in the generalized-cube network with the cube interconnection function:

\[
cube_i((p_B p_{B-1} \ldots p_2 p_1)_2) = (p_B p_{B-1} \ldots p_{i-1} \overline{p_i} p_{i-1} \ldots p_2 p_1)_2,
\]

that is, the \( \text{cube}_i \) connects link \( p \) to link \( \text{cube}_i(p) \), where \( \text{cube}_i(p) \) is the link whose label differs from \( p \) in just the \( i \)th bit.
Siegel and Smith discussed the relationship between the generalized-cube network and the omega network. Using the network topology description functions of (3.3.1), (3.3.3) and (3.3.4), they proved that the permutations generated by the Omega network are the same as that generated by the Generalized-Cube network when the switches are controlled on a stage-by-stage basis [22]. Therefore, all the mathematical

Figure 3.3.1 The Omega Network Topology for $B = 3$
results in Section 3.2 can also be applied to the Omega network. As a result, (3.2.10) and (3.2.16) can be used to mathematically analyze the performance of Omega Network used in DEM DACs.

3.3.2 The Indirect Binary n-cube network used in DEM DACs

![Diagram of the Indirect Binary n-cube Network Topology for B = 3](image)

Figure 3.3.2 The Indirect Binary n-cube Network Topology for $B = 3$

A B-bit Indirect Binary n-cube network consists of $B2^{B-1}$ switches, B control signals and B stages which are labeled consecutively 1 to B from the input stage to output [24]. Figure 3.3.2 shows an Indirect binary n-cube network for $B = 3$. Similar to the
Generalized-Cube network, the structure of the Indirect Binary n-cube network is defined such that the two inputs to a switch in stage \( i \) differ only in the \( i \)th bit. Thus, stage \( i \) implements \( \text{cube}_i \).

In [20-21], Siegel and Smith discuss the relationship between the Generalized-Cube network and the Indirect Binary n-cube network. They showed that the permutations generated by the Indirect Binary n-cube network is the same as that generated by the Generalized-Cube network when switches are controlled on a stage-by-stage basis, except that the stages are in reversed order. Therefore, \( E(T_{\text{INL}}[x(n), c(n)] | x(n)) \) and \( E[GG^T | x(n)] \) of the Indirect Binary n-cube network can be calculated using the mathematical results developed in Section 3.2 for the Generalized-Cube network. To illustrate this, the effect of the reversed control bit order is considered.

In Section 3.2, (3.2.5) which is

\[
E[\delta(c_b - k_b, c_{b-1} - k_{b-1}, ..., c_1 - k_1)] = \frac{1}{2^b},
\]

is applied to (3.2.3) and (3.2.15) respectively to calculate \( E(g_i | x) \) and \( E[GG^T | x(n)] \).

Because the control bits are independent and uniformly distributed, the value of \( E[\delta(c_b - k_b, c_{b-1} - k_{b-1}, ..., c_1 - k_1)] \) does not change as the order of the control bits reverses. Therefore, for DEM DACs using the Indirect Binary n-cube network, (3.2.5) can also be applied to derive (3.2.10). Similarly, (3.2.5) can also be applied in the process to calculate \( E(GG^T | x) \) and derive (3.2.16) for DEM DACs using the Indirect Binary n-cube network. Therefore, (3.2.16) can be also used to analyze the performance of the Indirect Binary n-cube network DEM DACs.
CHAPTER 4

HARDWARE EFFICIENCY DYNAMIC ELEMENT MATCHING

Interconnection networks have been used in DEM DACs to dynamically rearrange the interconnection of the mismatched components and reduce the DAC's distortion. However, many previously reported interconnection networks used in DEM DACs require an excessive amount of digital hardware. Therefore, efforts have been made to reduce hardware complexity while preserving performance.

In this Chapter, the hardware complexity of several DEM DAC interconnection networks, including the Benes network, the Generalized-cube network, the Omega network, and the Indirect Binary n-cube network, is compared in this chapter. A DEM interconnection network is developed that uses a small amount of hardware while having the same performance as the Indirect Binary n-cube network. It is also shown that a low-complexity network proposed in [25] is similar to this network.

4.1 The Hardware Complexity of Some Interconnection Networks

The hardware complexity of an interconnection network can be evaluated with the number of the binary switches and the number of control bits that the network requires.
Using these criteria, the hardware complexity of several interconnection networks, including the Generalized-Cube network, the Omega Network, the Indirect Binary n-cube network and the Benes network, is compared in this section.

4.1.1 The Hardware complexity of the Generalized-Cube Network, the Omega Network, and the Indirect Binary n-cube network.

The Generalized-Cube network, the Omega network, and the Indirect Binary n-cube network are B-bit interconnection networks that have B stages and $B2^{B-1}$ binary switches. If the networks are operated in *Stage Mode*, that is, all the binary switches in each stage are controlled by a single control signal, each of these networks require B control signals and the outputs of these networks can be rearranged into $2^B$ permutations [21]. For example, the 3-stage Generalized-Cube network shown in Figure 3.2.2 can generate 8 permutations when operated in Stage Mode. If the networks are operated in *Switch Mode*, that is, each switch is controlled by an independent control signal, each of these networks requires $B2^{B-1}$ control signals and the outputs of these networks can be rearranged into $2^{2^{B-1}}$ permutations [23]. For example, the 3-stage Generalized-Cube network shown in Figure 3.2.2 can generate 4096 permutations when operated in Switch Mode.

4.1.2 The Hardware complexity of the Benes Network

The Benes network is capable of generating all possible permutations of the input signals. Although a DEM DAC that uses a Benes network has never been implemented, an analysis of such a DEM DAC is provided in [7]. A B-bit Benes network consists of 2B
stages and $B2^B$ binary switches. The Benes network can be seen as an Indirect Binary n-cube network followed by a Generalized-Cube network. As a result, a $B$-bit Benes network has twice the number of binary switches as a $B$-bit Generalized-Cube network or a $B$-bit Indirect Binary n-cube network. Figure 4.1 shows a 3-bit Benes network.

![Diagram of a 3-bit Benes network](image)

Figure 4.1 The Benes network with 8 inputs/outputs, 6 stages, and 24 binary switches. This network is fully rearrangeable when operated in Switch Mode, meaning that all $8!$ Permutations of the input signals can be realized at the output.

When operated in Switch Mode, the Benes network can generate all the possible permutations which is significantly more than that the Generalized-Cube network, the Omega network and the Indirect Binary n-cube network can generate. For example, when
operated in Switch Mode, the 8 input/output Benes network in Figure 4.1 can generate $8!$, or 40320, permutations which is nearly ten times more than the $2^{3\cdot 2^8}$, or 4096, permutations that an 8 inputs/outputs Generalized-Cube can generate. However, when operated in Stage Mode, a Benes network generates the same number of permutations as a Generalized-Cube network, an Omega network and an Indirect Binary n-cube network. To illustrate, when operated in Stage Mode, the permutation generated by control signal $c_{B}, c_{B-1}, \ldots, c_{1}, c_{1}, \ldots, c_{B}$ in a B-bit Benes network is the same permutation that is generated by control signal $c_{B} \oplus c_{B}, c_{B-1} \oplus c_{B-1}, \ldots, c_{1} \oplus c_{1}$ in a B-bit Indirect Binary n-cube network.

If an interconnection network used in a DEM DAC is operated in Switch Mode, the number of independent control signals required to operate each of the binary switches increases exponentially as the DAC’s resolution increases. Therefore, interconnection networks operated in Switch Mode can be impractical for a DAC with more than 6 bits. For example, an 8-bit Benes network operated in Switch Mode requires 2048 random control signals. Therefore, in the following sections, only interconnection networks that are operated in Stage Mode are discussed.

The number of binary switches required by the Benes network and the Shuffle/Exchange class networks increases exponentially as the DAC’s resolution increases. For example, a 3-bit flash DAC that uses the Generalized-Cube network requires 12 binary switches and an 8-bit flash DAC that uses the Generalized-Cube network requires 1024 binary switches. Although current fabrication technologies can meet the requirements, other networks that require fewer switches can perform DEM as effectively.
4.2 The Compressed Butterfly-Based network

As shown in Figure 2.2.1, a DEM DAC's input is a B-bit input signal that is thermometer coded into a $2^B$-bit signal which is the input to the $2^B$-input / $2^B$-output interconnection network. The thermometer coder can be a modified thermometer coder defined such that $t_0(n) = 0$ and $t_{2^i-1}(n) = ... = t_{2^{i-1}-1}(n) = X_{k-1}(n)$ for example, for $B=3$,

$$T(n) = [0, X_0(n), X_1(n), X_1(n), X_2(n), X_2(n), X_2(n), X_2(n)]^T.$$

Figure 4.2 shows a 3-stage Indirect Binary n-cube network whose input bits are the output bits of this type of 3-bit modified thermometer coder.

Since $t_{2^i-1}(n) = ... = t_{2^{i-1}-1}(n) = X_{k-1}(n)$, any binary switch that exchanges any two of \{t_{2^i-1}(n), t_{2^{i-1}-1}(n), ... , t_{2^0-1}(n)\} is not necessary because the two input bits to the switch are identical. For example, the binary switches in the first two stages of the Indirect Binary n-cube network in Figure 4.2 are labeled A through H. The two inputs to switch B are both $X_1$. The two inputs to switch C and the two inputs to switch D are $X_2$. As a result, both inputs to switch G and switch H are $X_2$. Therefore, switches B, C, D, G and H are unnecessary. Removing these unnecessary switches, the network in Figure 4.2 can be redrawn as shown in Figure 4.3. In this thesis, this network is named the Compressed Butterfly-based network. When operated in Stage Mode, the Compressed Butterfly-based network can generate the same permutations as the Indirect Binary n-cube network, the Generalized-Cube network, the Omega network and the Benes network and therefore provide the same DEM DAC performance as these networks.

A B-bit Compressed Butterfly-based network consists of B stages and has $2^{i-1}$ binary switches in stage $i$ where $1 \leq i \leq B$. As a result, a B-bit Compressed Butterfly-
Modified Thermometer coder

Figure 4.2 A 3-stage Indirect Binary n-cube network whose input bits are the output bits of a 3-bit modified thermometer coder

based network consists of a total of \(2^8 - 1\) switches. This is much less than those of the Generalized-Cube network, the Omega network, the Indirect Binary n-cube network and the Benes network. Table 4.1 shows the comparison of the switches required for these interconnection networks and the Compressed Butterfly-based network.
Figure 4.3 The Compressed Butterfly-based network with 3 inputs/8 outputs, 3 stages, and 7 binary switches. All the redundant switches in Figure 4.2 are removed.

<table>
<thead>
<tr>
<th>DAC Bits</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benes network</td>
<td>24</td>
<td>64</td>
<td>160</td>
<td>384</td>
<td>896</td>
<td>2048</td>
</tr>
<tr>
<td>Generalized-Cube network</td>
<td>12</td>
<td>32</td>
<td>80</td>
<td>192</td>
<td>448</td>
<td>1024</td>
</tr>
<tr>
<td>Omega network</td>
<td>12</td>
<td>32</td>
<td>80</td>
<td>192</td>
<td>448</td>
<td>1024</td>
</tr>
<tr>
<td>Indirect Binary n-cube network</td>
<td>12</td>
<td>32</td>
<td>80</td>
<td>192</td>
<td>448</td>
<td>1024</td>
</tr>
<tr>
<td>Compressed Butterfly-Based Network</td>
<td>7</td>
<td>15</td>
<td>31</td>
<td>63</td>
<td>127</td>
<td>255</td>
</tr>
</tbody>
</table>

Table 4.1 A comparison of hardware complexities between the DEM techniques using the Benes network, the Generalized-Cube Network, the Omega network, the Indirect Binary n-cube and the Compressed Butterfly-based Network. It lists the number of required binary switches.
4.3 Hardware Efficiency of The compressed Butterfly-based Network

In this section, it is shown that the Compressed Butterfly-based network has the lowest hardware complexity in the sense of binary switches when compared with the hardware complexity of the Benes network and the Shuffle/Exchange networks discussed earlier. To show this, the interconnection networks are illustrated with network graphs.

In a network graph, each signal is represented by a node and therefore each binary switch is represented by 4 nodes, 2 cross edges and 2 straight edges. Figure 4.4(a) illustrates the network graph of a binary switch. Using network graphs, a 3-bit Generalized-Cube network, a 3-bit Indirect Binary n-cube network and a 3-bit Benes network can be represented by the graphs in Figure 4.5 (a), (b), and (c) respectively. Figure 4.6 shows the network graph for a 3-bit Compressed Butterfly-based network.

To ensure the time average of all the equivalent components at each component position is equal, interconnection networks used in DEM DAC need to be a full-access network. This implies that each bit, including the least significant bit, of the B-bit input must be able to be mapped to every output.
Figure 4.5 Network graphs: (a) A 3-bit Generalized-Cube network
(b) A 3-bit Indirect Binary n-cube network (c) A 3-bit Benes network

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 4.6 Network graph of a 3-bit Compressed Butterfly-based network

Figure 4.7 Binary tree from $\chi_0$ to 8 outputs. Each node is a binary switch
By tracking the paths of the least significant bit being mapped to each output in the network graph, it can be seen that the paths can be treated as a binary tree from the root to $2^B$ leaves. In the binary tree, each node is a binary switch which provides the ability to split the signal into two signals. Figure 4.7 shows the binary tree from $x_0$ to 8 outputs in Figure 4.6. To split one signal to $2^B$ signals, B stages and $2^{i-1}$ switches at $i$th stage are required. Therefore, at least $\sum_{i=1}^{B} 2^{i-1} = 2^B - 1$ binary switches are required. Therefore, if $L$ represents the least number of binary switches required for any $2^B$-input/output interconnection network to implement a $B$-bit DEM DAC,

$$L \geq 2^B - 1.$$  \hspace{1cm} (4.1)

Since the Compressed Butterfly-based network can implement DEM technique with $2^B - 1$ binary switches when a modified thermometer coder is used and $2^B - 1$ is the lower bound of $L$,

$$L = 2^B - 1.$$  \hspace{1cm} (4.2)

Therefore, the Compressed Butterfly-based network can implement DEM technique to DACs with the least number of binary switches. That is, when a modified thermometer is used in the DEM DAC architecture illustrated in Figure 2.2.1, the Compressed Butterfly-based network has the lowest hardware complexity compared to those previously discussed interconnection networks.

4.4 The Similarity of the Compressed Butterfly-based Network and a Low-complexity Network proposed in [25]
In [25], H. T. Jensen and J. F. Jensen proposed a low-complexity network for use in DEM DACs. In this thesis, this network is called J-J network. Generally, at the \( i \)th stage of a J-J network, the \( i \)th LSB of \( x(n) \) can be exchanged with the outputs of stage \( i-1 \). For example, the LSB can be exchanged with logic “0” at stage 1. At stage 2, the second LSB can be exchanged with the outputs of stage 1. At stage 3, the third LSB can be exchanged with the outputs of stage two. An \( i+1 \)-bit J-J network can be built using an \( i \)-bit J-J network by attaching \( 2^i \) binary switches to an \( i \)-bit J-J network to exchange the MSB input signal with the outputs of the \( i \)-bit J-J network. Figure 4.8 shows a 3-bit J-J network built based on a 2-bit J-J network.

Similarly, an \( i+1 \)-bit Compressed Butterfly-based network can also be built using an \( i \)-bit Compressed Butterfly-based network by attaching \( 2^i \) binary switches to exchange the MSB input signal with the outputs of the \( i \)-bit Compressed Butterfly-based network. Figure 4.9 shows a 3-bit Compressed Butterfly-based network based on a 2-bit Compressed Butterfly-based network. Although the J-J network and the Compressed Butterfly-based network are similar, have the same hardware complexity and generate the same number of permutations, these two networks generate different permutations.
Figure 4.8 A 3-bit J-J network built based on a 2-bit J-J network
Figure 4.9 A 3-bit Compressed Butterfly-based network built based on a 2-bit Compressed Butterfly-based network.
CHAPTER 5

CONCLUSION

In this thesis, interconnection networks have been analyzed for the specific DEM DAC architecture shown in Figure 2.2.1. This DEM DAC architecture consists of a thermometer coder, an interconnection network, B unit DACs and an analog summation unit. In this thesis, the criteria developed in [15] are used to compare the performance of DEM DACs that use different interconnection networks. According to these criteria, the DEM DAC's mean integral nonlinearity (INL), $E(T_{\text{INL}}[x(n), c(n)] | x) = E(G^T [x(n), c(n)] | x)(H - L)$, and the DEM DAC's conditional variance, $E[T_{\text{INL}}^2 | x] = (H - L)^T E(GG^T [x(n), c(n)] | x)(H - L)$, are the two critical features needed to evaluate the performance of DEM DACs.

In Chapter 3, the above two critical features are determined for the Barrel Shifting network and the Generalized-cube network. After calculating the matrices $E(G^T [x(n), c(n)] | x)$ and $E(GG^T [x(n), c(n)] | x)$, it is possible to calculate the mean and the variance of the integral nonlinearity. In practice, the circuit production process determines the matrix $H - L$. By putting in the real $H - L$, $E[T_{\text{INL}}^2 | x]$ can be calculated and SNR, SDR, SNDR and SFDR of DACs can be calculated and therefore evaluate the...
performance of certain interconnection networks.

It is also shown in Chapter 3 that the mathematical analysis of the Generalized-cube interconnection network can also be used to the Omega network. This is because these two interconnection networks generate same permutations with the same control signals. Also it is shown that the Binary n-cube network generates the same permutation as the Generalized-cube network if the control signals are in reverse order. Because the derivation process for $E(G^T[x(n),c(n)]|x)$ and $E(GG^T[x(n),c(n)]|x)$ of the Generalized-cube network does not depend on the order of the stages, the mathematical analysis for the Generalized-Cube network can be applied to the Indirect Binary n-cube network. Therefore, for all these Shuffle/Exchange Type networks, including Omega network and Indirect Binary n-cube network, their performance in DEM DACs can be evaluated with the same mathematical result derived for the Generalized-Cube network.

The final topic is the hardware efficiency of the interconnection networks used in DEM DACs. A Compressed Butterfly-based network has been proposed and it is shown that this network is hardware efficient when used with a modified thermometer coder in the DEM architecture indicated in Chapter 2. It is also shown that mathematically an interconnection network developed by Jensen [25] is similar to the Compressed Butterfly-based network and these two networks generate same number of permutations.
REFERENCES


VITA

Graduate College
University of Nevada, Las Vegas

Hui Fu

Local Address:
1676 Crystal Downs Avenue
Las Vegas, Nevada, 89123

Home Address:
Wuhan Research Institute of Telecommunications and Posts
Wuhan, 430074, Hubei Province, P.R.China

Degrees:
Bachelor of Science, Electronic Engineering, 1997
Tsinghua University, P.R.China

Thesis Title: An Analysis of Interconnection Network Effects on Dynamic Element Matching Digital-to-Analog Converters

Thesis Examination Committee:
Chairperson, Associate Professor, Dr. Peter Stubberud, Ph.D.
Committee Member, Professor, Dr. Ashok Iyer, Ph.D.
Committee Member, Assistant Professor, Dr. Henry Selveraj, Ph.D.
Committee Member, Assistant Professor, Dr. Zhiyong Wang, Ph.D.