Switching Sequence Optimization for Gradient Error Compensation in Thermometer-Decoded DAC Arrays

Yonghua Cong, *Student Member, IEEE* and Randall L. Geiger, *Fellow, IEEE*

***Abstract*—This paper discusses switching schemes for gradient error compensation in unary (thermometer-decoded) arrays of digital-to-analog converters (DAC's). The absolute lower bound of integral nonlinearity (INL) by optimizing switching sequences is established and optimal switching sequences that meet the lower bound of INL are presented for linear error compensation in one-dimensional arrays. A rapidly converging algorithm is developed to obtain INL bounded switching sequences for any given type of gradient error compensation. Simulation results show that the new switching sequences substantially reduce the nonlinearity of DAC's due to gradient errors.**

***Index Terms*—Digital-to-analog converter, gradient error, nonlinearity, switching sequence, thermometer decoding.**

I. Introduction

SEGMENTED architectures are widely used in high-conversion-rate and high-accuracy digital-to-analog converters (DAC's) [1]–[6]. Generally, the least significant bits (LSB's) steer a binary weighted array, while the most significant bits (MSB's) are thermometer decoded and steer a unary array. The static performance of a segmented DAC is strongly dependent on the linearity of the unary array.

Linearity can be achieved by overcoming all possible random and systematic errors [1]. The random errors are determined by the inherent matching properties of the technology used. In a given process technology, for architectures that do not incorporate trimming or tuning, increasing the active area of each unit element in the arrays of DAC's is the most effective method for reducing random errors. When only random errors are considered, for each extra bit of DAC accuracy, the active area of the unary array increases by a factor of four. In high-accuracy DAC's, this results in large dimension arrays. The gradient errors in these arrays can become very significant and must be correctly compensated to keep the random error dominate.

Optimizing switching schemes can reduce the nonlinearity due to gradient errors. This potential has been seen in many current-steering DAC designs [1]–[6]. A switching scheme is actually a layout technique. In a current-steering DAC, the switching scheme determines the interconnection between the outputs of the thermometer decoder/latch and the control terminals of the switches in the current matrix. As an example, an 8-bit thermometer-decoded current-steering DAC is shown in Fig. 1. The unary array contains 256 current sources that are ideally identical. The switching scheme determines the order the current sources are switched on as the digital code increases from 1 to 255. The current sources are numbered $1, 2, \cdots, 255$ in the order they are switched on. The unused current source is a dummy current source. Instead of inserting a dummy current source there, that area is often used for the biasing circuits. In a segmented DAC, this "dummy" area of the unary array can be used for the binary weighted array. For the unary array in Fig. 1, there are totally 256! possible switching sequences.

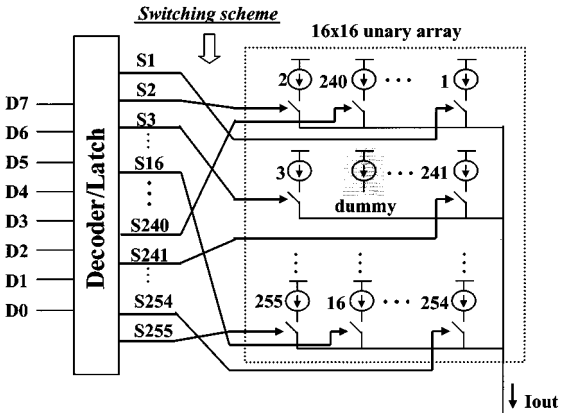


Fig. 1. Architecture of an 8-bit current-steering thermometer-decoded DAC

Several switching schemes and sequences have been heuristically derived in literature for gradient error compensation in unary arrays of DAC's [1]–[5]. However, no real analytical treatment has been attempted to verify whether these switching schemes and sequences are sufficient to compensate for the gradient errors, and the issue of whether better solutions exist has not been addressed. A general approach is necessary to find optimal or near optimal switching sequences under any given type of gradient condition.

An absolute lower bound of integral nonlinearity (INL) through optimizing the switching sequence is established in this paper. It will be shown that the conventional switching sequences result in linearity errors that are higher than this lower bound. *Optimal* switching sequences that meet the lower bound are presented for one-dimensional (1-D) arrays with linear gradient errors. A general approach is developed to find optimal or near optimal switching sequences for any given type of gradients.

Even though current-steering DAC's are used as examples in this paper, a similar analysis can be easily applied to other types of thermometer-decoded DAC's, such as capacitor array DAC's where charge rather than current is used. Even for resistor-string

Manuscript received November 1999; revised April 2000. This work was supported in part by Texas Instruments Incorporated and Rocketschips. This paper was recommended by Associate Editor B. Razavi.

Y. Cong and R. L. Geiger are with the Electrical and Computer Engineering Department, Iowa State University, Ames, IA 50010 USA.

Publisher Item Identifier S 1057-7130(00)05804-3.

DAC's, where the resistor strings are often laid out in several segments (e.g., a 10-bit resistor string containing 1023 resistors are laid out in 32 columns, each column, except one, containing 32 resistors), the order to interconnect the segments (or columns) can also be optimized using the approach described in this paper.

Before switching schemes are discussed, the linearity errors (INL and DNL) of thermometer-decoded DAC's are formulized in Section II. This formalization shows a strong dependence of INL on switching sequences. In Section III, typical gradient error distributions are illustrated and normalized. The conventional switching schemes are reviewed in Section IV. In Section V, an absolute lower bound of the INL for arbitrary switching sequences is derived and optimal sequences which meet this lower bound are given for 1-D linear error arrays. The idea is then expanded to any type of error arrays including two-dimensional (2-D) arrays and a rapidly converging algorithm—the INL bounded algorithm—is developed to find optimal or near optimal switching sequences for any given type of gradients. Simulation results are given in Section VI to compare different switching sequences.

II. Linearity Errors

In an n-bit thermometer-decoded DAC, the unary array generally consists of N ($N = 2^n$) elements that are ideally identical. However, mismatches between the elements always exist. For example, in a current-steering DAC, the actual current provided by current source j ($1 \le j \le N$), can be expressed as

$$I_j = \overline{I}(1 + \varepsilon_j) \tag{1}$$

where $\overline{I}$ is the average current provided by all the current sources in the array and ε_j is the relative deviation of I_j from $\overline{I}$. Hence

$$\overline{I} = \frac{\sum_{j=1}^{N} I_j}{N} \tag{2}$$

and the average value of ε_j ($1 \le j \le N$) is equal to zero.

For a real DAC, only $N-1$ elements are required while the extra element in the unary array is a dummy element. However, in some simplified DAC's as discussed later in this paper, the N elements in the unary array are all used and the digital input is in the range of $[0,\ N]$ instead of $[0,\ N-1]$. For notation convenience, the linearity errors of thermometer-decoded DAC's will be formulated in two cases: first for DAC's without dummy element, second, for DAC's with one dummy element.

A. DAC's without Dummy Element

Assume the actual analog output for a digital code k ($0 \le k \le N$) is denoted as $A(k)$, hence the offset of the DAC is $A(0)$. The linearity errors are generally in units of LSB's. The actual value of 1 LSB is defined as

$$A_{\mathrm{LSB}} = \frac{A(N) - A(0)}{N}. \tag{3}$$

After the offset is removed, the integral nonlinearity (INL) at each digital code is defined as the deviation of the analog output (in LSB's) at that code from the ideal transfer response which is a straight line from the output at 0 LSB to the output at N LSB. The differential nonlinearity (DNL) is the deviation of the analog step size from 1 LSB. Formally, the INL (in LSB's) at digital code k ($0 \le k \le N$) is given by

$$\mathrm{INL}(k) = \frac{A(k) - A(0)}{A_{\mathrm{LSB}}} - k = \frac{A(k) - A(0)}{A(N) - A(0)} * N - k. \tag{4}$$

It is easy to see that $\mathrm{INL}(0) = \mathrm{INL}(N) = 0$. Similarly, the DNL (in LSB's) at digital code k ($1 \le k \le *N$) is given by

$$\begin{aligned}\mathrm{DNL}(k) &= \frac{A(k) - A(k-1)}{A_{\mathrm{LSB}}} - 1 \\ &= \frac{A(k) - A(k-1)}{A(N) - A(0)} * N - 1.\end{aligned} \tag{5}$$

In a current-steering DAC without a dummy element, if all the N current sources are numbered $1, 2, \cdots, N$ in the order they are switched on, the actual output current for digital code k ($k = 1, 2, \cdots, N$) is thus given by

$$I(k) = \sum_{j=1}^{k} I_j + I(0) \tag{6}$$

where $I(0)$ is the offset current. It follows from (1)–(6) that INL and DNL (in LSB's) of the current-steering array can be expressed as

$$\mathrm{INL}(k) = \sum_{j=1}^{k} \varepsilon_j, \quad (0 \le k \le N) \tag{7}$$

$$\mathrm{DNL}(k) = \varepsilon_k, \quad (1 \le k \le N). \tag{8}$$

Therefore, both the INL and DNL are independent of the average current $\overline{I}$ and can be determined only by the relative errors ε_j ($1 \le j \le N$) of the current sources in the array. The INL and DNL of the overall DAC are defined as

$$\mathrm{INL}_{\mathrm{DAC}} = \max_{k=1}^{N}(|\mathrm{INL}(k)|) \tag{9}$$

$$\mathrm{DNL}_{\mathrm{DAC}} = \max_{k=1}^{N}(|\mathrm{DNL}(k)|). \tag{10}$$

From (8) and (10), it is apparent that thermometer-decoded DAC's can achieve very low $\mathrm{DNL}_{\mathrm{DAC}}$. For each element in the array, 50% variation is good enough to obtain a $\mathrm{DNL}_{\mathrm{DAC}}$ of 0.5 LSB.

However, it can be shown that with a poor switching sequence, the $\mathrm{INL}_{\mathrm{DAC}}$ can be very high when gradient errors are present. Our goal is to minimize $\mathrm{INL}_{\mathrm{DAC}}$ by optimizing the switching sequence.

B. DAC's with One Dummy Element

In this case, only $N-1$ elements in the unary array are required. The INL and DNL (in LSB's) at digital code k ($0 \le k \le N-1$) are given by

$$\mathrm{INL}(k) = \frac{A(k) - A(0)}{A(N-1) - A(0)} * (N-1) - k \tag{4'}$$

$$\mathrm{DNL}(k) = \frac{A(k) - A(k-1)}{A(N-1) - A(0)} * (N-1) - 1. \qquad (5')$$

In the current source array, $N-1$ current sources are used and numbered $1, 2, \cdots, N-1$ in the order they are switched on. If the dummy current source is numbered N, again the actual output current for digital code k is given by

$$I(k) = \sum_{j=1}^{k} I_j + I(0). \qquad (6')$$

Following from (1), (2) and (4')–(6'), $\mathrm{INL}(k)$ and $\mathrm{DNL}(k)$ for an array with one dummy element can be expressed as

$$\mathrm{INL}(k) = \frac{\sum_{j=1}^{k} \varepsilon_j + \frac{k}{N-1}\varepsilon_N}{1 - \frac{\varepsilon_N}{N-1}} \approx \sum_{j=1}^{k} \varepsilon_j \qquad (7')$$

$$\mathrm{DNL}(k) = \frac{\varepsilon_k + \frac{\varepsilon_N}{N-1}}{1 - \frac{\varepsilon_N}{N-1}} \approx \varepsilon_k \qquad (8')$$

and the INL and DNL of the overall DAC are

$$\mathrm{INL}_{\mathrm{DAC}} = \max_{k=1}^{N-1}(|\mathrm{INL}(k)|) \qquad (9')$$

$$\mathrm{DNL}_{\mathrm{DAC}} = \max_{k=1}^{N-1}(|\mathrm{DNL}(k)|). \qquad (10')$$

Comparing (7')–(10') with (7)–(10), the DNL and INL of DAC's in both cases are almost the same. Their minor difference depends on which current source is chosen as the dummy current source. This "dummy" effect is often so small that it can be ignored.

As we mentioned before, the DNL of a thermometer-decoded array is quite small while the INL may be quite high when gradient error accumulates. The switching sequence is optimized to minimize the error accumulation in the INL. Before addressing the switch sequence optimization, we will characterize the error distributions across the unary arrays of the DAC's.

III. Gradient Errors

If the errors or mismatch of the elements in a unary array are totally random and uncorrelated, the INL and the yield of the DAC can be estimated by Monte Carlo simulations [1], [2], [7]. With this estimation, the minimum requirement for the matching precision of the elements can be determined to meet given accuracy and yield specifications. Pelgrom [9] showed that random error can be reduced by increasing element area. For 1-bit accuracy improvement, the element area has to increase by a factor of four. When matching parameters for a given technology are available, the minimum active area required for the unit elements can be determined.

If only uncorrelated random errors are present, the INL and the yield of a DAC are independent of the switching sequence. However, as mentioned above, to reduce random errors so as to achieve high accuracy, the unary arrays of high-resolution DAC's generally occupy quite a large area, which results in large distances between elements. Large distances between elements introduce significant gradient errors [9]. To make the unary array compact, matrix configurations are often used and a square matrix is especially preferred. But even with compact layout, the distances between elements are still large.

A. Gradient Error Distributions

Gradient error distribution across a unary matrix can be approximated by a Taylor series expansion around the center of the unary array [1]. The gradient error of the element located at (x, y) can be expressed as

$$\varepsilon(x, y) = a_0 + a_{11}x + a_{12}y + a_{21}x^2 + a_{22}y^2 + a_{23}xy + \cdots \qquad (11)$$

It is generally assumed that the linear (the first order) and the quadratic (the second order) terms are adequate to model gradient effects [1], [2]. That is, the error distribution is typically linear or quadratic or the superposition of both as illustrated in Fig. 2. For example, in a current source matrix, the doping and the oxide thickness over the wafer or the voltage drop along the power supply lines have been reported to cause approximately linear gradient errors [1], [2], [4]. Temperature gradients and die stress may introduce approximately quadratic errors [8]. The overall systematic error distribution is given by superimposing these error components [4].

Assume in a $X \times Y$ matrix layout (X elements in the x direction and Y elements in the y direction), the location of each element is represented by its geometrical center and the spacing between two elements is Δx in the x direction and Δy in the y direction. Hence, the total layout area of the matrix is $(X * \Delta x) * (Y * \Delta y)$. If we use the center of the matrix as the origin as depicted in Fig. 2, then three typical gradient error distributions in this matrix can be formulized as follows.

1) Linear Error Distribution: The linear gradient error for an element located at (x, y) can be expressed as

$$\varepsilon_l(x, y) = g_l * \cos\theta * x + g_l * \sin\theta * y \qquad (12)$$

where θ is the angle of the gradient and g_l is the slope of the linear gradient. Apparently, the average error of the elements in the matrix is zero.

If the linear gradient is due to wafer gradient, and since the position of a die on the wafer is unknown, the gradient can occur in any direction, i.e., the gradient angle θ will vary randomly from 0°–360°.

2) Quadratic Error Distributions: It has been observed that the mismatch due to die stress is commonly a symmetrical function of the distance from the die center and that matching sensitivity to die stress is lowest at the die center [8]. As a design rule of thumb, the matching sensitive circuits are suggested to be placed symmetrically at the center of the die. It has also been reported that the stress in the y direction is nearly independent of the x coordinate [8]. In this paper, we assume that the matrix of the DAC is located at the center of the die and the quadratic gradients in both the x and y directions are independent and equal. Therefore, the quadratic gradient error for an element located at (x, y) can be expressed as

$$\varepsilon_q(x, y) = g_q * (x^2 + y^2) - a_0 \qquad (13)$$

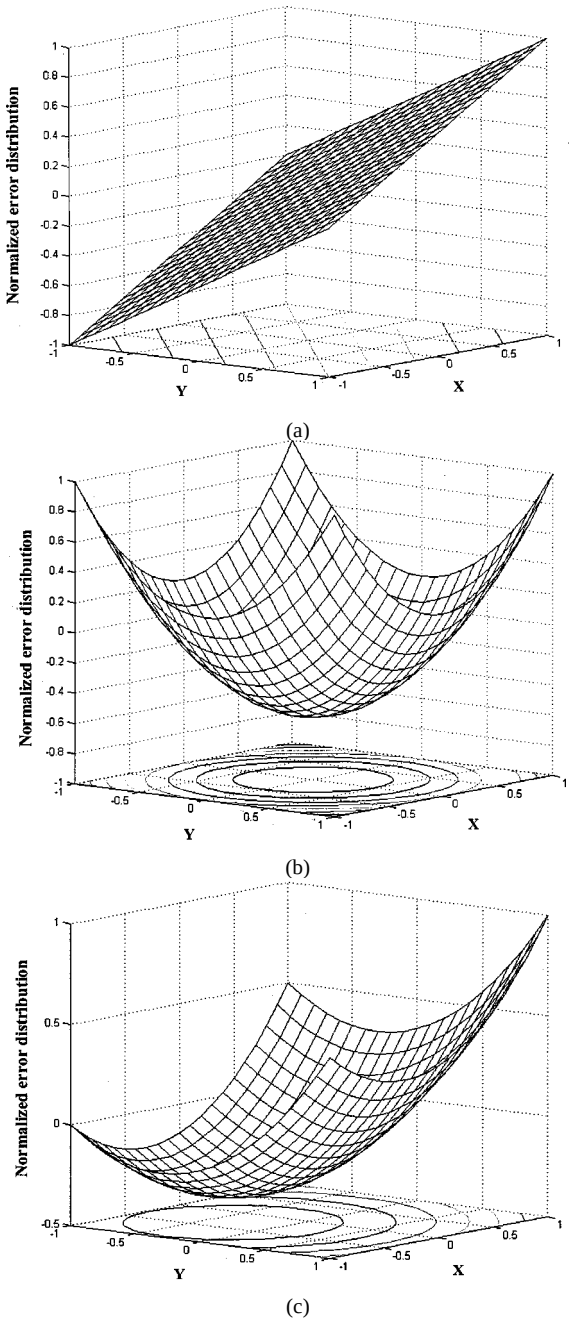


Fig. 2. Model of gradient error distribution. (a) Normalized linear error distribution. (b) Normalized quadratic error distribution. (c) Normalized joint error distribution.

where g_q and a_0 are technology parameters determined dominantly by the die bonding techniques. Notice that by definition, the average error of the elements in the matrix is zero.

3) Joint Error Distribution: In this case, the gradient error for an element located at (x, y) can be expressed as the superposition of a linear error component and a quadratic error component

$$\varepsilon(x, y) = \varepsilon_l(x, y) + \varepsilon_q(x, y). \tag{14}$$

Notice that (14) keeps the average error equal to zero.

B. Normalized Error Distributions

From (7), it can be seen that if the gradient errors in a unary array all scale by a common factor, then for any switching sequence, the INL of each digital code also scales by that same factor, i.e., the efficiency of switching sequences is independent of the scaling factor. If a sequence is good for a given error array, it is also good for any scaled version of this error array. Therefore, the comparison between different switching sequences can be made in normalized gradient error arrays.

In this paper, a square $M \times M$ matrix (for an 8-bit array, $M = 16$) will be normalized as an example. Since the layout of this matrix is square, the element spacing in the x direction is equal to that in the y direction, i.e., $\Delta x = \Delta y = \Delta$. The geometric position of this matrix layout can then be normalized so that all the elements are spatially distributed in the interval $[-1, 1]$ in both the x and y directions as shown in Fig. 2, i.e.,

$$x, y \in \left\{-1, -1 + \frac{2}{M-1}, -1 + 2 * \frac{2}{M-1}, \cdots, 1 - \frac{2}{M-1}, 1\right\}.$$

Hence, the geometric position of a real matrix can be obtained by multiplying the normalized position matrix by the scaling factor

$$S1 = \frac{(M-1) * \Delta}{2}. \tag{15}$$

In the normalized $M \times M$ matrix, the gradient errors are normalized so that the maximum error magnitude is equal to one. The denormalization scaling factors will be given under the three typical error distribution conditions.

1) Linear Error Distribution: Assume the linear error component is all due to wafer gradient. From die to die, the angle of the wafer gradient θ may vary randomly from 0° to 360°. The maximum possible magnitude of the linear errors occurs when $\theta = 45°$ and 135°, which is equal to $\sqrt{2}\, g_l$ [see (12)]. As shown in Fig. 2(a), we can normalize this magnitude as 1, i.e. in (12) $g_l = 1/\sqrt{2}$. The overall denormalization scaling factor in this case, including the position matrix scaling factor $S1$ in (15), is

$$S = \sqrt{2} * g_l * \frac{(M-1) * \Delta}{2}. \tag{16}$$

2) Quadratic Error Distribution: The maximum magnitude of the quadratic errors is equal to $2g_q - a_0$ [see (13)]. As shown in Fig. 2(b), if we normalize this magnitude as 1, then the denormalization scaling factor is

$$S = (2g_q - a_0) * \frac{(M-1) * \Delta}{2}. \tag{17}$$

3) Joint Error Distribution: The maximum magnitude of the overall gradient errors is equal to the summation of the maximum magnitudes of the linear and quadratic error components. If the linear error component is due to wafer gradient, the maximum possible magnitude of the overall gradient errors occurs when the linear gradient angle θ is 45° or 135°, and

is equal to $\sqrt{2}\,g_l + 2g_q - a_0$. As shown in Fig. 2(c), if we normalize this magnitude as one, then the denormalization scaling factor is

$$S = (\sqrt{2}\,g_l + 2g_q - a_0) * \frac{(M-1)*\Delta}{2}. \tag{18}$$

Optimization of switching sequences becomes rather complicated if both linear and quadratic errors are present. We have to consider not only the direction of the linear gradient, but also the ratio of the linear component to the quadratic component, which is defined as

$$w = \frac{\sqrt{2}\,g_l}{2g_q - a_0}. \tag{19}$$

This ratio can be estimated since the gradients of different types of errors have been experimentally quantified for the technology used.

The actual INL of the DAC can be obtained by multiplying the INL based on the normalized error array by the denormalization scaling factors given in (16)–(18).

IV. Conventional Switching Schemes

For a given type of gradient, the INL of DAC corresponding to a switching sequence can be calculated based on (7) and (9). To find the best sequence with the lowest INL, one may try to exhaust all possible sequences. However, the number of possibilities arises in a factorial fashion and thus becomes incredibly large for over 4-bit resolution. For a 4-bit unary array which consists of 16 unit elements, there are $16! = 2.1e13$ possible sequences, while a 5-bit array has $32! = 2.6e35$ possibilities.

A. Row–Column Switching Schemes

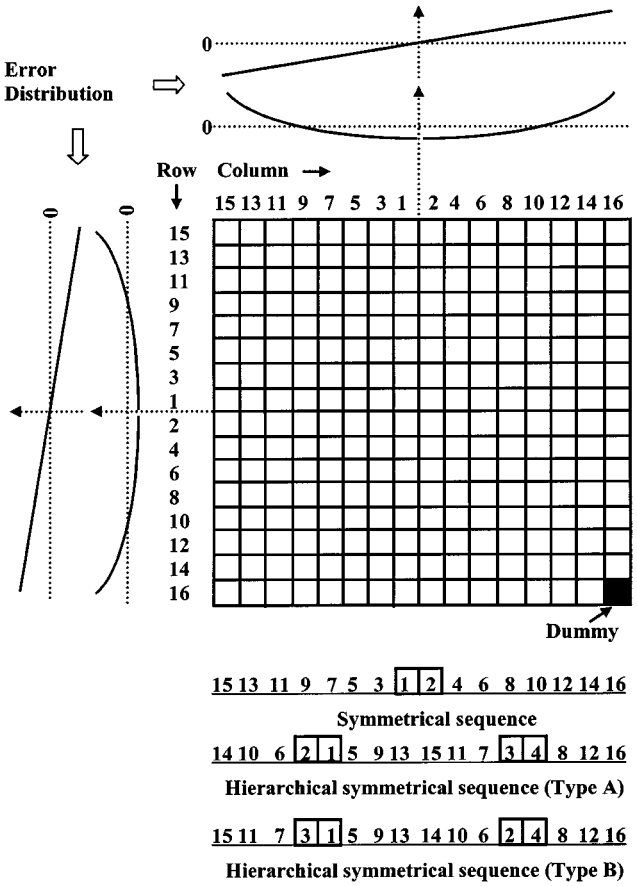
The well-known row–column switching scheme is commonly used in a heuristic attempt to optimize the switching sequence [2]–[6]. In this scheme, the spatial gradient errors are averaged in two directions, as shown in Fig. 3(a) and the sequences for row and column selection are optimized independently. The switching optimization problem is thus reduced to a 1-D space.

One switching sequence for the overall 8-bit matrix (16×16) in Fig. 3(a), where the "symmetrical sequence" [5] is used for row and column selection, is as follows:

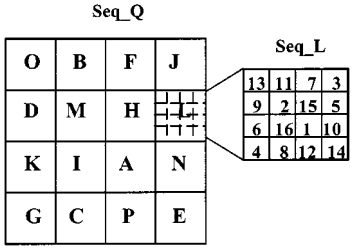
1. current source at (row 1, column 1)
2. current source at (row 1, column 2)
...
16. current source at (row 1, column 16)
17. current source at (row 2, column 1)
18. current source at (row 2, column 2)
...
254. current source at (row 16, column 14)
255. current source at (row 16, column 15).

The dummy current source is at (row 16, column 16). In this symmetrical sequence, linear errors are cancelled by every two current sources located symmetrically about the center, but quadratic errors accumulate.

To compensate for both linear and symmetrical (an approximation of quadratic) errors, hierarchical symmetrical sequences were proposed [4]. The hierarchical symmetrical sequences, as well as the symmetrical sequence for the 1×16 array, are all given in the lower part of Fig. 3(a). The hierarchical symmetrical sequence of Type A compensates for symmetrical errors at the first level and compensates for linear errors at the second level. Correspondingly, Type B sequence compensates for linear errors at the first level and compensates for symmetrical errors at the second level.



(a)



(b)

Fig. 3. Conventional switching schemes. (a) Row–column scheme. (b) Hierarchical scheme.

Even with good switching sequences, any of the row–column switching schemes are inherently insufficient for 2-D gradient error compensation. This can be seen by calculating the INL of the 8-bit DAC in Fig. 3(a) with (7). The gradient error of each unary array element can be divided into two parts: the column-related error and the row-related error. As the digital input increases from 1–16, the 16 elements in the first selected

TABLE I
SWITCHING SEQUENCES FOR A 1 × 8 LINEAR ERROR ARRAY AND THE INL CORRESPONDING TO THESE SEQUENCES

	Location →							
Original array :	4.65	4.75	4.85	4.95	5.05	5.15	5.25	5.35
Error array (%)	-7	-5	-3	-1	+1	+3	+5	+7

Sequences	Errors (%) of the elements: 1 2 3 4 5 6 7 8	INL (%) of digital code: 1 2 3 4 5 6 7 8	INL_{DAC} (%)
Sequential seq. 1 2 3 4 5 6 7 8	-7 -5 -3 -1 +1 +3 +5 +7	-7 -12 -15 **-16** -15 -12 -7 0	16
Symmetrical seq. 7 5 3 **1 2** 4 6 8	-1 +1 -3 +3 -5 +5 -7 +7	+1 0 +3 0 +5 0 **+7** 0	7
A new seq. **2 6 4** 8 5 **1** 7 **3**	+3 -7 +7 -3 +1 -5 +5 -1	+3 **-4** +3 0 +1 **-4** +1 0	4

row are turned on successively, and the column-related error is soon compensated due to the optimized column switching sequence. However, the error related to the first selected row accumulates and cannot be compensated until the second row is sequenced. The accumulation of row related errors also happen in other rows, which may lead to a large INL.

The advantage of the row–column scheme is its simplicity for design and layout. It is very straightforward to use a row–column decoder and put the local decoder and latch into each element cell. Since, in each cell, the area occupied by the local decoder and latch is often comparable to the active area itself, a possible way to reduce the matrix area and correspondingly the distances between elements, is to move all the digital parts to the outside of the matrix [2]. However, this method dramatically increases the number of matrix inputs, resulting in complex routing and ultimately a tradeoff between routing area and decoding area in the matrix.

B. Hierarchical Switching Scheme

An alternative to the row–column scheme is a two-step hierarchical switching scheme undertaken in the "Q^2Random Walk" current-steering DAC [1]. As shown in Fig. 3(b), the 8-bit (16×16) current matrix is divided into 16 regions (4×4) and each region has 16 elements (4×4). The switching of regions in [1] (designated as Seq_Q) is in the order of $A, B, \cdots, P$ which is intended to compensates for quadratic errors and the switching within each region (designated as Seq_L) is in the order of $1, 2, \cdots, 15, 16$, which is intended to compensates for the remaining linear errors. The overall switching sequence is:

1. current source 1 in region A
2. current source 1 in region B
...
16. current source 1 in region P
17. current source 2 in region A
18. current source 2 in region B
...
254. current source 16 in region N
255. current source 16 in region O.

The current source 16 in region P is the dummy source and provides biasing for the circuit. This hierarchical switching scheme allows optimization in 2-D space (even though the optimization is still constrained) with the penalty of complex interconnections.

For DAC design, a tradeoff has to be made between accuracy and complexity. To further optimize the switching sequences without dramatically increasing the complexity of DAC's, we will consider three options.

1) If the row–column switching scheme is used due to its simplicity, optimal sequences are wanted for gradient error compensation in 1-D arrays.
2) If the hierarchical scheme is used, optimal sequences are wanted for 2-D gradient error compensation. Reference [1] only gave the switching sequences for linear and quadratic error compensation in a 4×4 array. The derivation of the two sequences was not presented, and a general approach to find switching sequences for different size arrays was not described.
3) If we try to find optimal switching sequences through unconstrained optimization over 2-D arrays, it is necessary to derive a good algorithm that can find optimal or near optimal sequences for a given type of gradient very quickly. As a matter of fact, with such an algorithm, the problems in 1) and 2) are also solved.

Other methods for gradient error compensation include providing separate biasing for each quadrant of the current matrix [3], and splitting each current source into several units located symmetrically in the matrix [1], [2]. These methods effectively suppress the spatial gradient errors and the linearity of the DAC's are thus determined by the residual errors, which are further compensated with either the row–column switching scheme [2], [3] or the "random walk" scheme [1]. This implies that combining these methods with the switching schemes and sequences described in this paper may provide more effective gradient error compensation. However, this related topic is beyond the scope of this paper.

V. INL BOUNDED SWITCHING SEQUENCES

Since in the row–column switching schemes, the switching sequences either for row selections or for column selections are all optimized in 1-D space, we will first consider the switching optimization of 1-D arrays. As an example, a 1×8 unary array with linear gradient errors is given in Table I. Rows 1 and 2 show the actual values of the elements in the array and the relative error of each element, respectively. Three switching sequences are considered. For example, with the symmetrical sequence, as the digital input increases from 1 to 8, the element with error -1% is switched on first and numbered 1, the element with error $+1\%$ is switched on next and numbered 2, and so forth. Based on (7) and (9), the INL of the DAC with different switching sequences can be easily calculated as shown

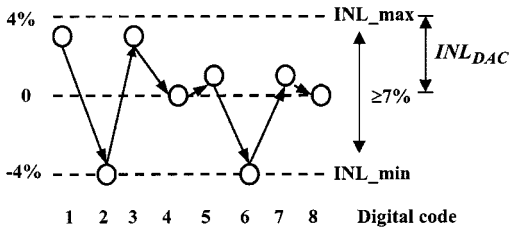


Fig. 4. Illustration of INL calculation for the new switching sequence in Table I

in the last two columns of Table I. The sequential sequence results in an $\mathrm{INL_{DAC}}$ of 16% due to sever error accumulation. In the symmetrical sequence, the linearity error caused by a certain element is canceled when the element located symmetrically is switched on. It results in an $\mathrm{INL_{DAC}}$ of 7%, which is equal to the maximum error magnitude in the error array. The new sequence is able to further reduce $\mathrm{INL_{DAC}}$ by about a factor of two. In what follows, it will become apparent that the new sequence in this example is an optimal sequence. An optimal switching sequence means for a given gradient, no other switching sequence can achieve an $\mathrm{INL_{DAC}}$ less than that achieved by this optimal sequence. Note that the definition of optimality says nothing about uniqueness. For a given type of gradient, there are often several or even many distinct optimal sequences.

To find optimal sequences, we will first determine a lower bound for $\mathrm{INL_{DAC}}$. For a unary array containing N elements without dummy element, define the maximum and minimum INL of a certain sequence by the expressions

$$\mathrm{INL\_max} = \max_{k=1}^{N} \mathrm{INL}(k) \tag{20}$$

$$\mathrm{INL\_min} = \min_{k=1}^{N} \mathrm{INL}(k). \tag{21}$$

Then, based on (9), $\mathrm{INL_{DAC}}$ can be given by

$$\mathrm{INL_{DAC}} = \max(\mathrm{INL\_max}, -\mathrm{INL\_min}). \tag{22}$$

As the digital input k increases one by one, the value of $\mathrm{INL}(k)$ moves between $\mathrm{INL\_max}$ and $\mathrm{INL\_min}$, as illustrated in Fig. 4. Each step size is determined by the error of the element currently switched on. The maximum step is equal to the maximum magnitude of the errors (donated as $E\ \max$) in the error array. Therefore, the spacing between $\mathrm{INL\_max}$ and $\mathrm{INL\_min}$ can be no less than $E\ \max$. This results in the inequality

$$\mathrm{INL\_max} - \mathrm{INL\_min} \geq E\ \max. \tag{23}$$

It can be observed from (22) that $\mathrm{INL_{DAC}}$ is minimized if $\mathrm{INL\_max}$ and $\mathrm{INL\_min}$ are symmetrical about zero, as depicted in Fig. 4. In this case

$$\mathrm{INL_{DAC}} = \mathrm{INL\_max} = -\mathrm{INL\_min}. \tag{24}$$

Returning to (23), a lower bound of $\mathrm{INL_{DAC}}$ is obtained

$$\mathrm{INL_{DAC}} \geq E\ \max/2. \tag{25}$$

This key inequality establishes an absolute lower bound on the INL of a DAC. It is not dependent upon the type of gradient present and applies to arrays of any dimension.

The formal proof of (25) is given as follows.

It is well known that if x and y are nonnegative real numbers, then

$$\max(x, y) \geq \frac{x+y}{2}$$

and $\max(x, y) = (x+y)/2$ if and only if $x = y$.

Observe that $\mathrm{INL\_max}$ and $-\mathrm{INL\_min}$ are nonnegative real numbers. It thus follows that

$$\mathrm{INL_{DAC}} \geq \frac{\mathrm{INL\_max} - \mathrm{INL\_min}}{2}. \tag{26}$$

With (23), (25) can be obtained and $\mathrm{INL_{DAC}}$ is equal to the lower bound $E\ \max/2$ if and only if

$$\mathrm{INL\_max} = -\mathrm{INL\_min} = E\ \max/2. \tag{27}$$

For a given error distribution, $E\ \max/2$ is an absolute lower bound of $\mathrm{INL_{DAC}}$. Since this is an absolute lower bound, no switching sequence can results in an $\mathrm{INL_{DAC}}$ lower than this lower bound. In the above example, $E\ \max = 7\%$, thus $E\ \max/2 = 3.5\%$. Since the resolution of the error array is 1%, the minimum achievable $\mathrm{INL_{DAC}}$ is 4%. The new sequence in Table I meets this lower bound, so it is optimal.

We are now in a position to make the following claim. This new switching sequence given in Table I is optimal for any 1×8 linear error array, independent of both the sign and magnitude of the gradient, because any linear gradient differs from that given in the example only by a constant scaling factor. As mentioned in Section III, the optimality of the sequence will not be impacted by this scaling factor.

The new optimal sequence given in Table I is not unique. There are several other optimal sequences, two of which are obtained if the elements in the array (from the left to the right) are numbered 3 5 1 8 7 6 4 2 and 4 6 2 8 1 7 3 5, respectively.

We can find optimal sequences by building a tree structure as shown in Fig. 5. Start with an element, the amplitude of whose relative error is equal to or less than the lower bound of $\mathrm{INL_{DAC}}$. In the above example, the lower bound of $\mathrm{INL_{DAC}}$ is 4%, so we can start with the elements that have errors of 3%, 1%, −1%, or −3%. They are surrounded with circles in Fig. 5. If we start with 3%, the INL for digital code "1" is also 3% shown beside the arrow. The next element is chosen so that the INL for digital code "2" is within [−4%, 4%]. The possible elements are those whose errors are within [−4%−3%, 4%−3%]=[−7%, 1%]. As shown in the second row of Fig. 5, −7%, −5%, −3%, −1%, and 1% can satisfy this requirement. Likewise, the third element is chosen so that the INL for digital code "3" is within [−4%, 4%]. The same process is repeated (if possible) until all eight elements are selected without repetition and thus the INL for all digital codes (1–8) are no larger than the lower bound. This yields an optimal sequence. Otherwise, if the selection is stuck somewhere in the middle, that is, none of the remaining elements can make the INL meet the lower bound, then the searching fails in this path and we have to go back to the upper level and try another path. Any path successfully going through all eight levels represents an optimal sequence. For example, in Fig. 5, the high lighted path: 3%, −7%, 7%, −5%, 5%, −3%,

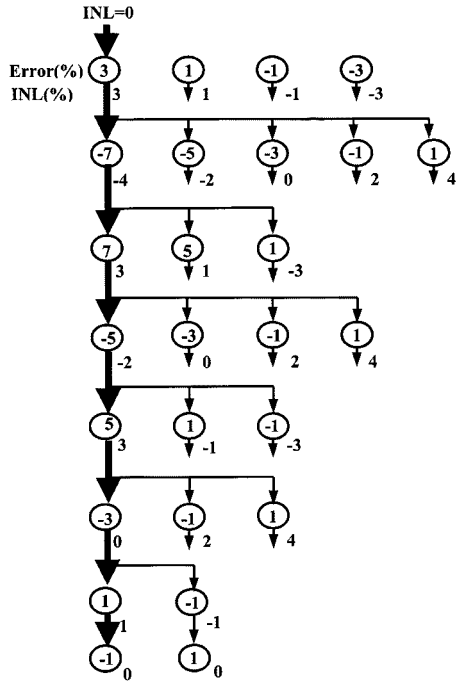


Fig. 5. Tree structure for searching optimal switching sequences

1%, −1%, which corresponds to the sequence 2 4 6 8 7 1 5 3, is another optimal sequence for a 1×8 linear error array.

The same idea can be applied to any error arrays including two-dimensional arrays. The general form of two algorithms are described as follows.

A. Sort and Group (SG) Algorithm

1) Sort the whole error array in either ascending or descending order. Assuming a unary array consists of $2L$ elements, after sorting, we get a new 1-D error array labeled as

$$\varepsilon_{-L}, \varepsilon_{-(L-1)}, \cdots \varepsilon_{-2}, \varepsilon_{-1}, \varepsilon_1, \varepsilon_2, \cdots, \varepsilon_{L-1}, \varepsilon_L.$$

Through this step, any 2-D error matrix is reduced to a 1-D array. According to (25), the absolute lower bound of $\mathrm{INL}_{\mathrm{DAC}}$ is $\max(|\varepsilon_L|, |\varepsilon_{-L}|)/2$.

2) A simple method to optimize switching sequence is to group the above sorted error array in the same way as in the 1-D linear gradient error array, hence the name "SG" algorithm. For example, using the grouping method of the new sequence in Table I results in an error sequence

$$\underbrace{\varepsilon_{L/2}, \varepsilon_{-L}, \varepsilon_L, \varepsilon_{-L/2}}\underbrace{\varepsilon_{L/2-1}, \varepsilon_{-(L-1)}, \varepsilon_{L-1}, \varepsilon_{-(L/2-1)}},$$
$$\cdots\underbrace{\varepsilon_1, \varepsilon_{-(L/2+1)}, \varepsilon_{L/2+1}, \varepsilon_{-1}}.$$

The corresponding switching sequence is an SG sequence. Likewise, using the grouping method of the sequence obtained in Fig. 5 results in another SG sequence.

The optimal sequences for 1-D linear error arrays of size 8, 16, and 32 are often needed. Some SG sequences for 1×8 arrays have already been given in the above example. The following are two SG sequences, one for 1×16 arrays and one for 1×32 arrays, which are optimal for linear gradient of any magnitude and sign:

1×16 array: 2 6 10 14, 4 8 12 16, 13 9 5 1, 15 11 7 3

1×32 array: 2 6 10 14 18 22 26 30, 4 8 12 16 20 24 28 12,
29 25 21 17 13 9 5 1, 31 27 23 19 15 11 7 3

Although SG sequences are optimal for 1-D linear error arrays, their efficiency for other types of gradient distribution has not been theoretically investigated. Simulation results show that they may not be the optima if quadratic gradients present. Even for 2-D linear error arrays, the optimality of SG sequences can not be guaranteed, because after sorting, the 2-D linear error matrix turns into a 1-D error array which is not simply linear. In these cases, some of the SG sequences may perform better than the others. However, for typical gradient distributions as given in Section III, simulation results show that the SG sequences can usually achieve better linearity than conventional switching sequences.

B. INL Bounded Algorithm

A more general approach that may allow further reducing $\mathrm{INL}_{\mathrm{DAC}}$ is to build a tree, as shown in Fig. 5. Notice that the lower bound given in (25) may be an overly optimistic estimation. It is possible that a switching sequence meeting this absolute lower bound does not exist. A practical approach using the tree of Fig. 5 is to relax the bound of $\mathrm{INL}_{\mathrm{DAC}}$. For example, a value between the absolute lower bound given by (25) and the $\mathrm{INL}_{\mathrm{DAC}}$ achieved by the SG sequences could be established. This relaxed bound enhances the possibility of convergence. The sequence obtained by this algorithm is hence termed as an "INL bounded" sequence. These INL bounded sequences are often optimal or near optimal and can sufficiently compensate for any given type of gradient errors, which will be demonstrated in the next section.

The INL bounded algorithm is a simple algorithm, which has not been optimized to minimize computing time although computation minimization strategies could be explored. We have, however, succeeded in obtaining near optimal switching sequences with small computation times for many examples. Even for a unary array with 8-bit resolution, the simple INL bounded algorithm converges pretty fast, and the obtained sequences, as shown in the Section VI, achieve sufficiently low $\mathrm{INL}_{\mathrm{DAC}}$.

VI. Simulation Results

To demonstrate the application of the new switching optimization algorithm and compare the INL bounded sequences with the conventional sequences, the 16×16 matrix of an 8-bit thermometer-decoded DAC is used as an example. The error distributions across the matrix are normalized as in Section III. Thus, the $\mathrm{INL}_{\mathrm{DAC}}$ obtained in the following simulations are normalized. They are only used for comparison, the actual $\mathrm{INL}_{\mathrm{DAC}}$ would need to be denormalized as described in Setion III to reflect the actual gradient effects present in the process. The simulations will be done for both

row–column and hierarchical switching schemes and the INL bounded sequences will be given under three typical error distribution conditions.

A. Using Row–Column Switching Scheme

The sequences that compensate for the gradient errors in a 1×16 array serve as the row and column selection sequences. The symmetrical sequence and the hierarchical symmetrical sequence for a 1×16 array have already been given in Fig. 3(a). The INL bounded sequence described in the previous section will be derived under three conditions.

1) Normalized Linear Gradient Error: We have already obtained optimal switching sequences (SG sequences) for 1-D linear error array in Section V. In one SG sequence for a 1×16 array, the elements (from the left to the right) are numbered

$$\frac{2\ 6\ 10\ 14,\ 4\ 8\ 12\ 16,\ 13\ 9\ 5\ 1,\ 15\ 11\ 7\ 3}{\text{An SG Sequence (also an INL bounded sequence)}}.$$

If the linear gradient is due to wafer gradient, assuming the gradient can have any direction across the array with equal probability, then the INL of the DAC versus the angle of the gradient and the yield can be obtained as shown in Fig. 6. Here, only the symmetrical sequence is compared with the SG sequence, because when only linear gradients are present, the hierarchical sequence of Type B has the same performance as the symmetrical sequence while the Type A hierarchical sequence performs poorly compared to the Type B sequence.

As we expect, the SG sequence results in an INL_{DAC} nearly 1/2 less than that obtained by the symmetrical sequence and the yield for a given INL_{DAC} can be substantially enhanced.

2) Normalized Quadratic Gradient Error: We obtained the following INL bounded sequence to compensate for 1-D quadratic errors

$$\frac{8\ 6\ 4\ 2\ 5\ 1\ 12\ 7\ 9\ 15\ 10\ 13\ 3\ 16\ 14\ 11}{\text{INL bounded sequence}}.$$

As shown in Table II, the INL bounded sequence results in an INL_{DAC} that is only 1/3 of that attained by the hierarchical symmetrical sequence of Type A. The Type B sequence and the symmetrical sequence are not well suited for managing quadratic errors and thus are not included in the comparison.

3) Normalized Joint Gradient Error [assuming $w = 1$ in (19)]: In this case, if the linear gradient is due to wafer gradient so that the angle of the linear gradient (θ) is random, the error distribution in both the row and column direction changes with θ. The switching sequence that is optimal for one angle may not be optimal for other angles. Assuming θ varies from 0° to 360° with equal probability, our goal is to find a sequence that results in a low INL_{DAC} with high yield. In other words, the sequence can achieve low INL_{DAC} for all possible error arrays due to the random linear gradient angle. In the INL bounded algorithm, ideally the bound of INL_{DAC} should be applied to all possible error arrays. In practice, we can apply the bound of INL_{DAC} in the error arrays with some typical values of θ, for example, 0°, 45°, 90°, 135°, and 180°.

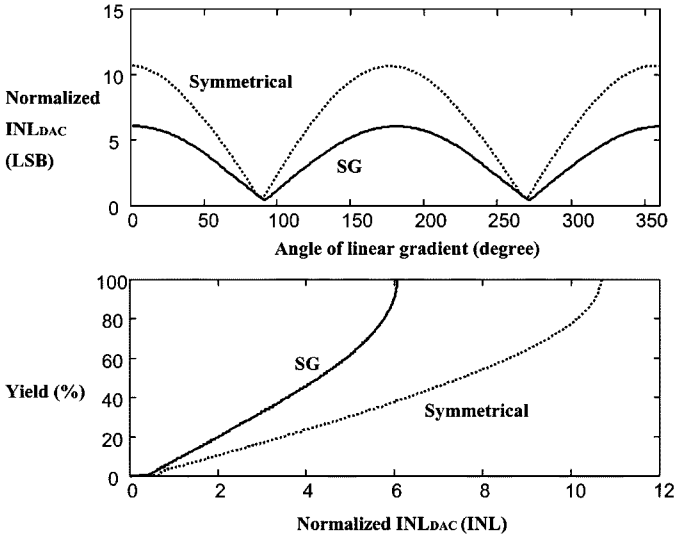


Fig. 6. INL and yield of the DAC (with linear gradient) using row–column switching scheme

TABLE II
INL OF THE DAC (WITH QUADRATIC GRADIENT) USING ROW–COLUMN SWITCHING SCHEME

Sequences	Normalized INL_{DAC}
Hierarchical Symmetrical (Type A)	11.37
INL bounded	4.25

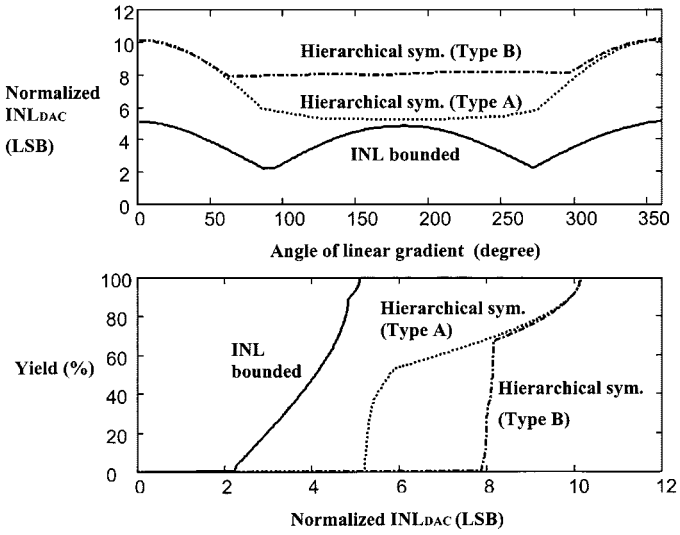


Fig. 7. INL and yield of the DAC (with joint gradient) using row–column switching scheme

With this approach, we obtained the following INL bounded sequence for a 1×8 array

$$\frac{13\ 4\ 8\ 1\ 16\ 7\ 9\ 12\ 5\ 14\ 2\ 10\ 3\ 15\ 11\ 6}{\text{INL bounded sequence}}.$$

As shown in Fig. 7, if 95% yield is required, the INL of the DAC when using the new sequence is only half of that when hierarchical symmetrical sequences are used.

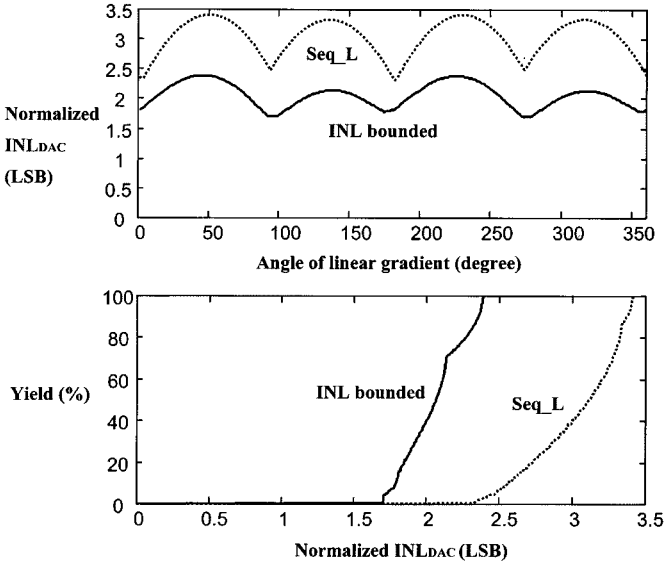


Fig. 8. INL and yield of the DAC (with linear gradient) using hierarchical switching scheme

B. Using Hierarchical Switching Scheme

As in the "random walk" DAC introduced in Section IV, the 16×16 array of the 8-bit DAC is divided into 16 regions (4×4) and each region contains 16 (4×4) elements. The switching sequences optimized for a 4×4 matrix control the region switching and the switching within each region. Similarly, the INL bounded sequences are given under three conditions.

1) Normalized Linear Gradient Error: If the linear gradient is due to wafer gradient, a low $\mathrm{INL_{DAC}}$ and high yield sequence as shown below can be obtained by applying an $\mathrm{INL_{DAC}}$ bound for the 4×4 error arrays with several typical linear gradient angles

13	15	7	5
11	9	1	3
2	8	16	10
4	6	14	12

.

As shown in Fig. 8, if we use Seq_L of the "random walk" scheme [see Fig. 3(b)], which is claimed to have the potential to compensate for linear gradient errors, for both the region selection and the switching within each region, the INL of the DAC varies from 2.3–3.4 when the angle of the gradient changes from 0° to 360°. If the INL bounded sequence given above is used, the INL of the DAC varies between 1.7–2.3.

2) Normalized Quadratic Gradient Error: Seq_Q of "random walk" scheme [see Fig. 3(b)] happens to be an INL bounded sequence. Therefore, we still use Seq_Q for the region selection. We can assume the residual gradient within each region is approximately linear. If we use the INL bounded sequence obtained in B–1 of this section to control the switching within each region, the INL of the DAC is 1.29. Instead, if Seq_L is used, the INL of the DAC is 1.63. These results are summarized in Table III.

TABLE III
INL OF THE DAC (WITH QUADRATIC GRADIENT) USING HIERARCHIAL SWITCHING SCHEME

Sequences	Normalized INL_{DAC}
Random walk	1.63
INL bounded	1.29

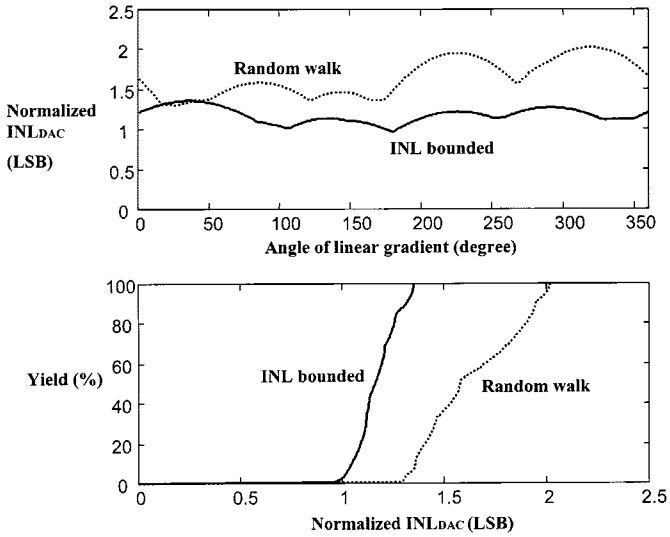


Fig. 9. INL and yield of the DAC (with joint gradient) using hierarchical switching scheme

3) Normalized Joint Gradient Error [assuming $w = 1$ in (19)]: Assume the linear gradient has random directions, the INL bounded sequence for region selection is shown as follows:

14	1	9	11
3	16	5	6
7	13	10	2
4	12	8	15

.

We still assume the gradient within each region is approximately linear, so the INL bounded sequence obtained in VI-B-1 is used for the switching within each region. In Fig. 9, this switching scheme is compared with the "random walk" scheme. If 95% yield is required, The $\mathrm{INL_{DAC}}$ of the "random walk" scheme is 2.0, while the $\mathrm{INL_{DAC}}$ of the new scheme is only 1.3.

Table IV summarizes the performance of the above switching schemes under the three error distribution conditions. The hierarchical switching schemes show a big advantage over the row–column schemes in gradient error compensation. The INL bounded sequences can further reduce the linearity errors. When row–column switching schemes are used, compared with the conventional sequences, the INL bounded switching sequences

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT SWITCHING SCHEMES UNDER THREE ERROR DISTRIBUTION CONDITIONS

Error distribution	Worst-case INL			
	Row-column switching scheme		Hierarchical switching scheme	
	Conventional	INL bounded	Random walk	INL bounded
Linear	10.7	6.1	3.4	2.4
Quadratic	11.4	4.2	1.6	1.3
Joint (50% linear+50% quadratic)	10.1	5.1	2.0	1.4

can reduce the INL of the DAC by approximately 50%. When hierarchical schemes are used, the INL bounded switching sequences reduce the INL of the DAC by about 30% compared with the "random walk" sequences.

SG and INL bounded algorithms were introduced. With these algorithms, optimal or near optimal switching sequence can be obtained when some gradient information, such as the ratio of the linear component to the quadratic component of the gradient, is available. The "Q^2 Random Walk" switching scheme [1] was established based on the good systematic error profile information from a test chip. With this scheme, 14-bit intrinsic accuracy was achieved without trimming or tuning. It has been shown that the two-step hierarchical switching optimization in the "Q^2 random walk" DAC provides much more flexibility for switching optimization than the classical row–column scheme. Unfortunately, the method of determining the switching sequences Seq_Q and Seq_L were not well explained in [1] making it difficult to extend this hierarchical switching scheme to arrays of different size and different types of error profiles. The INL bounded algorithm introduced in this paper provides this flexibility. Furthermore, even for linear gradient compensation, the $\mathrm{INL}_{\mathrm{DAC}}$ of the proposed sequence in Section VI-B-1 is lower than that achieved with the Seq_L sequence.

VII. CONCLUSION

An absolute lower bound of the INL due to gradient effects is developed for switching sequence optimization in thermometer-decoded DAC arrays. This lower bound is half the maximum deviation of all the elements in the gradient error array. Optimal switching sequences that meet this lower bound for linear error compensation in 1-D arrays were introduced. A rapidly converging algorithm was developed to provide INL bounded switching sequences under any given type of gradient error condition. Simulation results show that hierarchical switching schemes outperform the row–column schemes in the presence of linear and/or quadratic gradients. Compared with what is attainable with the best published switching sequences, the INL bounded switching sequences can reduce the linearity errors due to gradient mismatch by up to 50%.

REFERENCES

[1] G. Van der Plas, J. Vandenbussche, W. Sansen, M. Steyaert, and G. Gielen, "A 14-bit intrinsic accuracy Q^2 random walk CMOS DAC," *IEEE J. Solid-State Circuits*, vol. 34, pp. 1708–1718, Dec. 1999.

[2] J. Bastos, A. Marques, M. Steyaert, and W. Sansen, "A 12-bit intrinsic accuracy high-speed CMOS DAC," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1959–1969, Dec. 1998.

[3] C. Lin and K. Bult, "A 10-b, 500-Msamples/s CMOS in 0.6 mm^2," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1948–1958, Dec. 1998.

[4] Y. Nakamura, T. Miki, A. Maeda, H. Kondoh, and N. Yazawa, "A 10-bit 70 MS/s CMOS D/A converter," *IEEE J. Solid-State Circuits*, vol. 26, pp. 637–642, Apr. 1991.

[5] T. Miki, Y. Nakamura, M. Nakaya, S. Asai, Y. Akasaka, and Y. Horiba, "An 80-MHz 8-bit CMOS D/A converter," *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 983–988, Dec. 1986.

[6] B. Henriques and J. Franca, "A high-speed programmable CMOS interface system combining D/A conversion and FIR filtering," *IEEE J. Solid-State Circuits*, vol. 29, pp. 972–977, Aug. 1994.

[7] J. Bastos, M. Steyaert, and W. Sansen, "A high yield 12-bit 250-MS/s CMOS D/A converter," in *Proc. IEEE 1996 Custom Integrated Circuits Conf.*, May 1996, pp. 431–434.

[8] J. Bastos, M. Steyaert, A. Pergoot, and W. Sansen, "Influence of die attachment on MOS transistor matching," *IEEE Trans. Semiconduct. Manufact.*, vol. 10, pp. 209–217, May 1997.

[9] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, pp. 1433–1439, Oct. 1989.

Yonghua Cong (S'00) was born in Shandong, China, in 1972. She received the B.S. degree in computer communications in 1993 and the M.S. degree in circuits and systems in 1996 from the University of Electronic Science and Technology of China, Chengdu, China. She is currently pursuing the Ph.D. degree at the Electrical and Computer Engineering Department, Iowa State University, Ames.

In 1996–1997, she was a Research Assistant at the Electrical Engineering Department of Tsinghua University, Beijing, China. Her current research interests are high-performance digital-to-analog converters and high-speed filters.

Randall L. Geiger (S'75–M'77–SM'82–F'90) received the B.S. degree in electrical engineering and the M.S. degree in mathematics from the University of Nebraska, Lincoln, in 1972 and 1973, respectively, and the Ph.D. degree in electrical engineering from Colorado State University, Fort Collins, in 1977.

He spent 13 years as a Faculty Member in the Electrical Engineering Department, Texas A&M University, College Station. Since 1991, he has been a Professor in the Electrical and Computer Engineering Department, Iowa State University, Ames. His current teaching and research activities are in the area of analog and mixed-signal VLSI design.

Dr. Geiger was an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS during 1983–1985 and for *IEEE Circuits and Devices Magazine* during 1986–1989. He was Vice President of Publications for the Circuits and Systems (CAS) Society (1989–1990), and a Member (1990–1997) and Chair (1989–1990) of the Transactions Committee of the IEEE Periodicals Council. He was a Member of the IEEE Publications Board during 1990–1992 and 1994–1996, and Chairman of the Special Sessions (1990) and Workshop Chairman (1993) for the IEEE International Symposium on CAS. He is a former President-Elect (1991) and President (1992) of the CAS Society, and recipient of the Myril B. Reed Prize Paper Award (1981) and a Fellow Award "for contributions to discrete and integrated analog circuit design."