Optimal Switching Sequences for One-dimensional Linear Gradient Error Compensation in unary DAC Arrays

Yonghua Cong and Randall L. Geiger
Iowa State University
Durham 351, Ames, IA50011

***Abstract*—Gradient error can be compensated by optimizing switching sequences of DAC arrays. This paper establishes an absolute lower bound of integral nonlinearity (INL) which may be achieved by optimizing switching sequences. Optimal switching sequences that meet this lower bound are presented for one-dimensional linear gradient error compensation in unary (thermometer decoded) DAC arrays. The sequences can be used in row-column decoded current or capacitor unary arrays as well as R-string DACs, where the switching sequence is optimized in one dimension.**

I. INTRODUCTION

The accuracy of DACs depends on the matching property of the elements in the DAC arrays. Gradient error always exists. In high resolution DACs, it becomes very significant due to the large dimension of the arrays and must be correctly compensated.

Optimizing switching schemes can reduce the nonlinearity due to gradient errors. This potential has been seen in many current-steering DAC designs [1]-[7]. A switching scheme is actually a layout technique. In a current steering or charge redistributed DAC, the switching scheme determines the interconnection between the outputs of the thermometer decoder/latch and the control terminals of the switches in the unary array. In another words, it determines the order the unit elements are switched on as the digital code increases. For a R-string DAC, the R-string is often laid out in several segments, the order to interconnect the segments can also be optimized using the approach described in this paper.

DAC arrays are generally either binary weighted or thermometer decoded. Thermometer based DACs have many advantages over their binary counterparts, such as guaranteed monotonicity, lower DNL and less glitch. In a high accuracy DAC, the MSBs usually drive a thermometer-decoded unary array that determines the linearity of the overall DAC. This paper focuses on the switching optimization in thermometer-decoded arrays.

Row-column decoding is most widely used in DAC design due to its simplicity for design and layout [1]-[5]. A DAC using row-column decoding generally consists of a row and a column decoder as well as a unary array. Each cell in the array contains a local decoder, a latch and a unit element. In this scheme, the spatial gradient errors are averaged in both the x and y directions as shown in Fig.1 and the sequences for the row and column selections are optimized independently. One switching sequence for the overall 8-bit matrix (16x16) in Fig.1, where the "Symmetrical Sequence" [1] is used for the 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). The switching optimization problem is thus reduced to a one-dimensional space. The goal of this paper is to derive the optimal switching sequence for one-dimensional array with linear gradient dominant. In a current source matrix, the doping and the oxide thickness over the wafer or the voltage drop along the power supply lines can all cause approximately linear gradient errors [1][2].

II. LINEARITY ERRORS

Assume a unary array consists of N elements. For example, in Fig.1, the 16 rows form a 1x16 array, hence N=16, and each row is represented by a current source with current equal

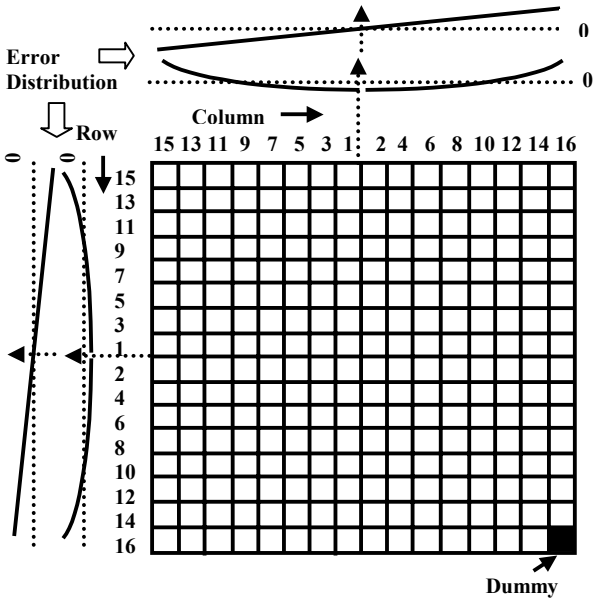

Fig.1 A unary array using row-column decoding

to the average current of the 16 sources in the row. Ideally, the elements in the array are identical. However, mismatches between the elements always exist. In a current array, as an example, the actual current provided by current source j ($1 \le j \le N$), can be expressed as

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

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

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

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

Assume the actual current output for a digital code k ($0 \le k \le N$) is denoted as I(k) and the offset is I(0). The linearity errors are generally in units of LSBs. The actual value of 1 LSB is defined as

$$I_{LSB} = \frac{I(N)-I(0)}{N} \tag{3}$$

The integral nonlinearity (INL) (in LSBs) at digital code k ($0 \le k \le N$) can be given by

$$INL(k) = \frac{I(k)-I(0)}{I_{LSB}} - k = \frac{I(k)-I(0)}{I(N)-I(0)} * N - k \tag{4}$$

It is easy to see that INL(0) = INL(N) = 0. The differential nonlinearity (DNL) (in LSBs) at digital code k ($1 \le k \le N$) is given by

$$DNL(k) = \frac{I(k)-I(k-1)}{I_{LSB}} - 1 = \frac{I(k)-I(k-1)}{I(N)-I(0)} * N - 1 \tag{5}$$

If all the N current sources are numbered 1,2,…,N in the order they are switched on, the actual output current for digital code k (k=1,2, …, N) is thus given by

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

follows from (1)-(6) that INL and DNL (in LSBs) can be expressed as

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

$$DNL(k) = \varepsilon_k \qquad (1 \le k \le N) \tag{8}$$

It can be seen the INL and DNL are independent of the average current $\bar{I}$ 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 array are defined as

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

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

From (8) and (10), it is apparent that thermometer decoded DACs can achieve very low DNL_{DAC}. For each element in the array, 50% variation is good enough to obtain a DNL_{DAC} of 0.5LSB. However, it can be shown that with a poor switching sequence, the INL_{DAC} can be very high due to gradient error accumulation. Our goal is to minimize INL_{DAC} by optimizing the switching sequence.

III. OPTIMAL SWITCHING SEQUENCES

As an example, a 1x8 unary array with linear gradient errors is given in Table.1. Row 1 and Row 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 in the last two columns of Table.1. The sequential sequence results in an 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 INL_{DAC} of 7%, which is equal to the maximum error magnitude in the error array. The new sequence is able to further reduce INL_{DAC} by about a factor of 2. In what follows, it will become apparent that the new sequence in this example is an optimal sequence. An optimal switching

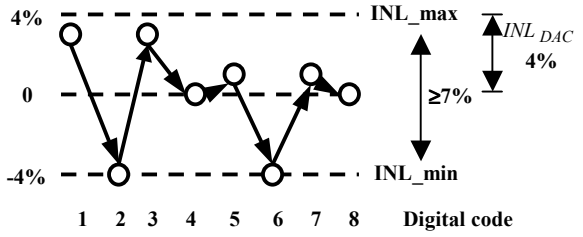

Fig.2 Illustration of INL calculation

Table.1 Switching sequences for a 1x8 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

sequence means for a given gradient, no other switching sequence can achieve an 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 INL_{DAC}. For a unary array containing N elements, define the maximum and minimum INL of a certain sequence as

$$INL\_\max = \max_{k=1}^{N} INL(k) \tag{11}$$

$$INL\_\min = \min_{k=1}^{N} INL(k) \tag{12}$$

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

$$INL_{DAC} = \max(INL\_\max, -INL\_\min) \tag{13}$$

As the digital input k increases one by one, the value of INL(k) moves between INL_max and INL_min as illustrated in Fig.2. 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 Emax) in the error array. Therefore, the spacing between INL_max and INL_min can be no less than Emax. This results in the inequality:

$$\text{INL\_max} - \text{INL\_min} \geq E\max \tag{14}$$

It can be observed from (13) that INL_{DAC} is minimized if INL_max and INL_min are symmetrical about 0 as depicted in Fig.2. In this case,

$$INL_{DAC} = INL\_\max = -INL\_\min \tag{15}$$

Returning to (14), a lower bound of INL_{DAC} is obtained:

$$INL_{DAC} \geq E\max/2 \tag{16}$$

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 (16) are given as follows:

It is well known that if x and y are non-negative real numbers, then

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

and $\max(x, y) = \frac{x+y}{2}$ if and only if x=y.

Observe that INL_max and –INL_min are non-negative real numbers. It thus follows that

$$INL_{DAC} \geq \frac{INL\_\max - INL\_\min}{2} \tag{17}$$

With (14), (16) can be obtained and INL_{DAC} is equal to the lower bound Emax/2 if and only if

$$\text{INL\_max} = -\text{INL\_min} = E\max/2 \tag{18}$$

For a given error distribution, Emax/2 is an absolute lower bound of INL_{DAC}. Since this is an absolute lower bound, no switching sequence can results in an INL_{DAC} lower than this lower bound. In the above example, Emax=7%, thus Emax/2=3.5%. Since the resolution of the error array is 1%, the minimum achievable INL_{DAC} is 4%. The new sequence in Table.1 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.1 is optimal for any 1x8 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. From (7), it can be seen that the INL for any sequence scales by the magnitude of that constant.

The new optimal sequence given in Table 1 is not unique. There are several other optimal sequences, two of which are obtained if the elements in the array (left to 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.3. Start with an element, the amplitude of whose relative error is equal to or less than the lower bound of INL_{DAC}. In the above example, the lower bound of 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

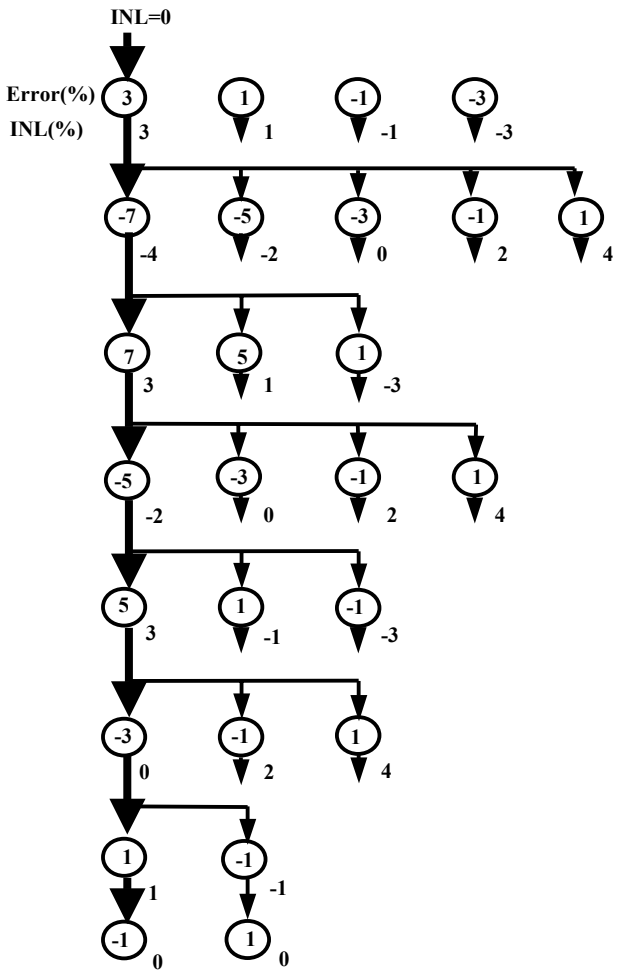

Fig.3 Tree structure searching for optimal switching sequence

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 8 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 though all 8 levels represents an optimal sequence. For example, in Fig.5, the high lighted path: 3%, –7%, 7%, -5%, 5%, -3%, 1%, –1% which corresponds to the sequence 2 4 6 8 7 1 5 3 is another optimal sequence for a 1x8 linear error array.

Using the same approach, the optimal sequences for one-dimensional linear error arrays of any size can be obtained. The following two sequences are for 1x16 and 1x32 arrays respectively. They are optimal for linear gradient of any magnitude and sign.

1x16 array: 2 6 10 14, 4 8 12 16, 13 9 5 1, 15 11 7 3
1x32 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

IV. CONCLUSIONS

An absolute lower bound of the integral nonlinearity error (INL) due to gradient effects is developed for switching sequence optimization in thermometer decoded DAC arrays. This lower bound is 1/2 of 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 one-dimension arrays were introduced. Compared to the symmetrical sequence, the new switching sequences can reduce the linearity errors due to gradient mismatch by up to 50%.

ACKNOWLEDGEMENT

This work was supported, in part, by TI Inc., RocketChips Inc. and R. J. Carver Trust.

REFERENCES

[1] 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.

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

[4]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.

[5] 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.

[6] 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. 1708-11718, Dec. 1999.

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