New Sequence Switching and Layout Technique for High-Speed High-Accuracy Current-Steering DACs

Tao Zeng, and Degang Chen, Senior Member, IEEE
Department of Electrical and Computer Engineering, Iowa State University
Ames, Iowa 50011 USA

Abstract—In this paper, new sequence switching and layout techniques are presented for the design of high-speed high-accuracy current-steering DACs. Our new sequence switching technique—after-fabrication programmable switching—rearranges the switching sequence of current sources after chip fabrication, which will guarantee to generate an optimal switching sequence to achieve high static accuracy. With this implementation, the area will be reduced dramatically, which will result the reduced parasitic effect and improved dynamic performance. Moreover, the gradient effect will become much less significant. To reduce the gradient errors even more, a new layout technique is introduced.

I. Introduction

In modern communication systems, there are a great number of demands for high-speed high-accuracy DACs. Of several different architectures, current steering DACs are more favored for these applications. With different selection methods—binary-weighted, thermometer-coded, and segment-coded, current steering DACs are implemented by an array of matched current sources. In order to achieve high matching accuracy, special techniques are needed.

The first technique is intrinsic-accuracy method, which ensures the matching of current source within a desired yield by using a large area. However, this becomes problematic or unrealistic for an intrinsic accuracy of more than 12 bit, because the gate area of the current source grows by a factor of 4 for each extra bit [1]. Furthermore, large area increases parasitic effect, as well as nonlinear gradient errors due to process variations, temperature gradients, mechanical stress, and voltage drops along the power lines [2]. These errors are extremely hard to compensate, and degrade the SFDR significantly at high frequencies.

The second technique is calibration or self-calibration, which adjusts or corrects the current value of each current source according to the result of comparison. This method has the advantage of smaller area, and it becomes more and more attractive for 14 bit or higher resolution DACs [3]-[4]. However, the drawback of this method is the additional analog circuitry, which makes very difficult for low-voltage technologies.

Recently, a new technique has been reported in the literature, which is switching-sequence post-adjustment calibration (SSPA) [5]. SSPA achieves the high static accuracy by dynamically adjusting the switching sequence of the current sources after chip implementation. It measures the current value of each current source, and the best switching sequence is calculated based on these current values. SSPA only requires a current comparator in the analog domain. However, SSPA is not the optimized switching strategy.

In this paper, an optimal switching strategy—after-fabrication programmable switching (AFPS), which is based on the principle of 1-D switching strategy in [6], is presented. AFPS guarantees to generate an optimal switching sequence to achieve the linearity where ½DNL ≤ INL ≤ DNL. Meanwhile, AFPS is only controlled by digital logic circuitry. This is more favored in the small-feature-size and low-voltage technologies. By this implementation, the area requirement can be dramatically reduced compared to that using intrinsic-accuracy method with the same yield requirement. This large area reduction gives us significantly reduced parasitic effect resulting in an improvement in settling and dynamic performance [3]. Furthermore, the gradient effect becomes much less significant. To reduce the gradient errors even more, a new layout technique is introduced.

This paper is organized as follows. In Section II, the principle of after-fabrication programmable switching is presented, while the new layout technique is proposed in section III. Conclusion is drawn in section IV.

II. After-Fabrication Programmable Switching

The principle of AFPS is very simple. At first, the current value of each current source is measured manually by the digital controller after fabrication. Then, the corresponding errors are calculated by subtracting the average from each current value. The optimal switching sequence is obtained based on these errors, and will be programmed into the same digital controller. With this optimal switching sequence, AFPS achieves very good static accuracy, since it guarantees ½DNL≤ INL ≤ DNL. This gives us a great deal of freedom to shrink the area of current-source array. Moreover, additional analog circuits are not required in AFPS, which makes the design suitable for low-voltage technologies.

A 6-bit thermometer-coded current-source array, which consists of 63 current sources, is used as an example to describe AFPS method. In general, random effect, gradient effect and finite output impedance effect are three main contributors to DAC's nonlinearities. At here, only random errors are considered, since the gradient errors and errors due to the finite output impedance can be significantly reduced by special layout patterns and cascoded structure. Random errors of current sources are cost by device mismatches [7]-[8], and can be characterized as a normal distribution with mean of zero and standard deviation of following:

978-1-4244-4495-3/09/$25.00 ©2009 IEEE

$$\sigma_I = \sqrt{\frac{A_\beta^2 + 4A_{VT}^2/(V_{GS} - V_T)^2}{2WL}}\,\bar{I} \qquad (1)$$

where A_β and A_{VT} are the process mismatch parameters, and $\bar{I}$ is the nominal unity current.

At here, let us assume that we are designing a 14-bit DAC, and the 6-bit array represents the most significant bits (MSBs). In this case, we can use intrinsic-accuracy method to design the 8-bit low segments to get 9 or 10 bit accuracy. This is very easy to achieve with a reasonable area in today's technologies. However, it is unrealistic to apply the same technique for the MSB array. Therefore, we need to adopt other techniques to reduce the area requirement for the MSB array, and meanwhile, obtain the same matching accuracy. Fig. 1 shows the corresponding INL and DNL of unsorted current sources for the 6-bit array due to the random errors. From the plot, DNL is 0.5155 LSB, and INL is 1.5849 LSB, where LSB refers to the 14-bit level. The MSB INL is quite large, and therefore, this is an unacceptable design for the 14-bit DAC. In order to improve the static performance, AFPS is implemented by generating an optimal switching sequence. The details are discussed as follows.

At first, the current sources are sorted by their corresponding errors ε_i's in the ascending order, and all the errors are partitioned into negative, zero and positive groups. Then, a search is performed among the current sources, and a current source, whose error is close to -½DNL, becomes the start point of the switching scheme. In the following steps, a current source, with the smallest negative error (in magnitude) available to make INL (k) approach or just pass -½DNL, will be selected; after this happens, a current source, which has the largest positive error available to make INL (k) approach or just pass +½DNL, will be selected; after this happens, a current source, with the smallest negative error (in magnitude) with the same condition as the first step, will be selected, and so on and so forth. The selection continues as proposed until all the current sources have been selected. To make this strategy much clearer, the pseudo-code is described as follows.

Pseudo-code of AFPS:

```
Sort all error $\varepsilon_i$'s into three groups ($\varepsilon_n$, $\varepsilon_0$, and $\varepsilon_p$), where i = 1:$2^n$-1 (i=1:63 at here);
Step 1:  DNL (1) = $\varepsilon_{n\,(-½DNL)}$; INL (1) = DNL (1);
         remove $\varepsilon_{n\,(-½DNL)}$;
Step i+1: $\varepsilon_n$ =max ($\varepsilon_n$); $\varepsilon_p$ =max ($\varepsilon_p$);
         Case 1: INL (i) < 0
                 (1) if INL(i) + $\varepsilon_n$ ≥ -½DNL
                     choose $\varepsilon_n$ ; DNL(i+1) = $\varepsilon_n$;
                     INL (i+1) = INL (i) + $\varepsilon_n$;
                     remove $\varepsilon_n$;
                 (2) else if .|INL(i)+$\varepsilon_n$ | > |INL(i) +$\varepsilon_p$ |
                     choose $\varepsilon_p$ ; DNL(i+1) = $\varepsilon_p$;
                     INL (i+1) = INL (i) + $\varepsilon_p$;
                     remove $\varepsilon_p$;
                 (3) else
                     perform (1)
         Case 2: INL (i) > 0
                 It is symmetric to Case 1 with
                 proper modifications;
         Case 3: INL (i) =0
                 perform a search to find the available
                 $\varepsilon_{n(-½DNL)}$ in $\varepsilon_n$;
                 DNL (i+1) = $\varepsilon_{n(-½DNL)}$;
                 INL (i+1) = INL (i) + $\varepsilon_{n(-½DNL)}$;
Final step: pick all the current sources in $\varepsilon_0$ group;
```

The INL value in AFPS is bounded by DNL, namely, ½DNL ≤ INL ≤ DNL. The proof of this statement can be found in [6]. After performing AFPS, a better INL result has been shown in Fig. 2. From the plot, INL is 0.2593 LSB, which is improved by a factor of 6.1 compared to the previous value. To better understand how much the linearity can be improved by AFPS, 10000 simulations have been performed for the 6-bit array, and Fig. 3 shows the statistical distribution of DNL, and INL before and after AFPS. From this, the average INL improvement factor by using AFPS is found to be about 4.6. Also, it is noted that in the AFPS algorithm |INL (i)| is being pushed very close to ½DNL. Therefore, INL after AFPS is very close to ½DNL, which is the absolute optimal INL obtainable by any switching strategies.

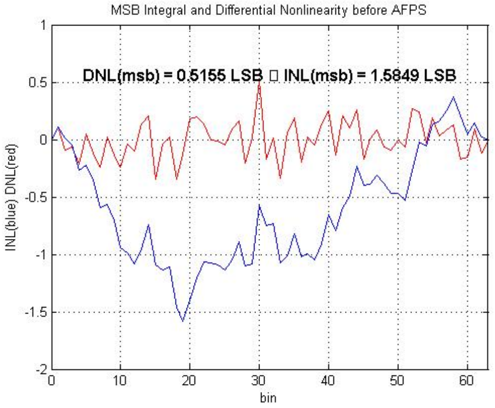


Fig. 1 DNL & INL before performing AFPS for 6-bit array

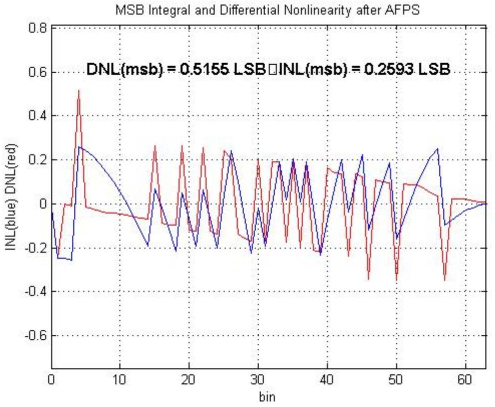


Fig. 2 DNL & INL after performing AFPS for 6-bit array

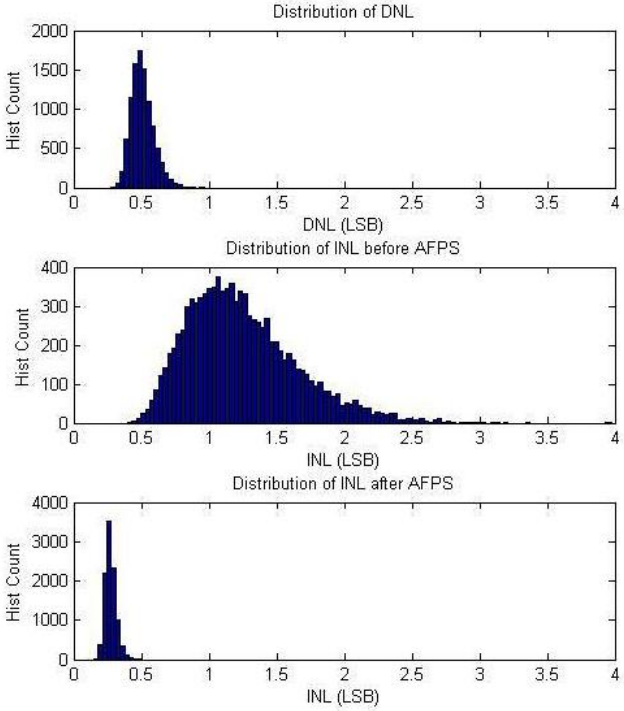


Fig. 3 Statistical distribution of DNL, INL before and after AFPS

Not only the static performance is improved, but also the chip area is reduced dramatically by using AFPS method. For the condition that INL < ½LSB, DNL needs to be less than ½LSB since AFPS guarantees INL ≤ DNL. To achieve the same INL level without AFPS or any other calibration methods, DNL needs to be less than $1/(2\sqrt{N})$ LSB for the thermometer-coded array, since

$$INL \ \leq \ \sqrt{N} \times DNL \tag{2}$$

where N is the total number of current sources in the array. Therefore, for a 6-bit array DNL needs to be less than 1/16LSB in order to achieve INL < ½LSB. Since only random errors are considered and the DNL requirement after using AFPS is 8 times less, according to (1), the area for the 6-bit array with using AFPS is 64 times less than that without using it! This not only gives us a smaller die area, but also it reduces the parasitic effects and results in an improvement in settling and dynamic performance. Furthermore, the gradient effects become less significant.

Since the switching sequence is not determined before chip fabrication, each current source in the array will have 2^n possible routing alternatives, where n is the resolution of the current-source array. This means that the total complexity for applying AFPS is $(2^n)^{2^n}$. Therefore, there is a trade-off existing between the digital circuit area and linearity for using AFPS technique. However, applying AFPS to a 6 bit current-source array is quite easy even though the complexity is 64^{64}. One of the approaches is to use 64×64 1-bit ROM array with 6-bit row and column decoders. The memory array produces a complexity of $2^{64\times64}$, which is much larger than 64^{64}. Therefore, this approach will allow us to perform AFPS method without any problems.

III. New Layout Technique

Gradient errors are determined by process variations, temperature gradients, mechanical stress, and voltage drops along the power lines, and can be expressed as following for a current source at the location (x, y):

$$\varepsilon(x,y) = \varepsilon_1(x,y) + \varepsilon_2(x,y) + \cdots \tag{3}$$

The first term is the linear gradient error, where

$$\varepsilon_1(x,y) = g_1(a_{11}x\cos\theta + a_{12}y\sin\theta) \tag{4}$$

and g_1 and θ are the strength and angle of the linear gradient. The second term is the quadratic gradient error, where

$$\varepsilon_2(x,y) = g_2(a_{21}x^2 + a_{22}y^2 + a_{23}xy) \tag{5}$$

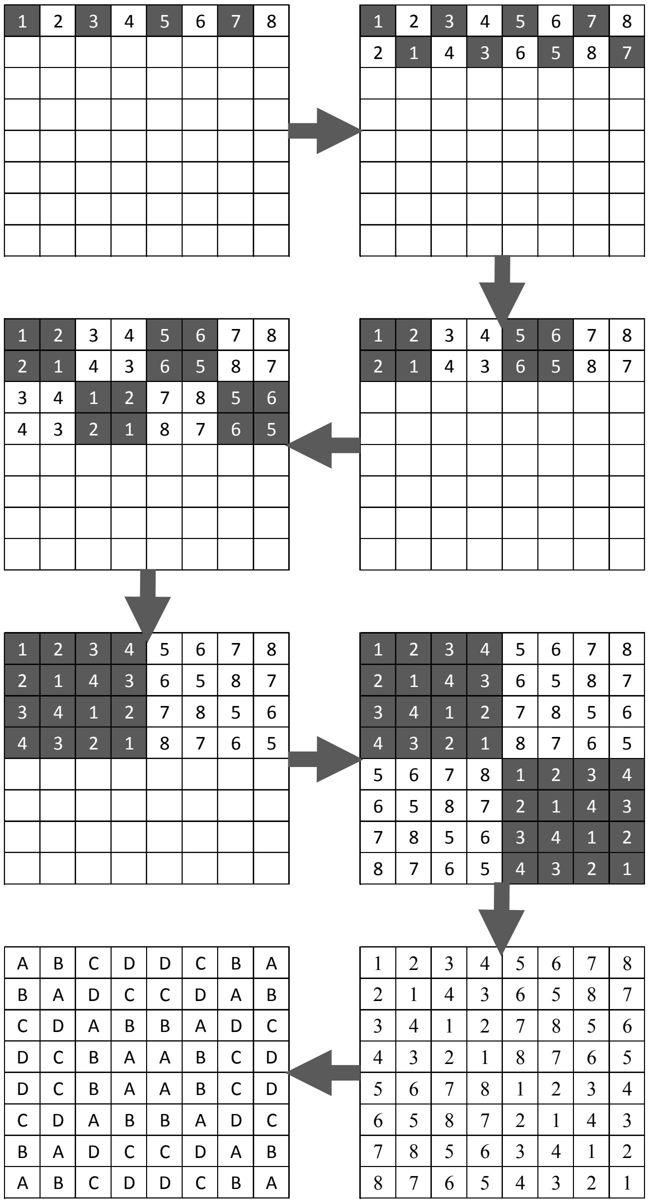
and g_2 is the strength of the quadratic gradient. Other high order gradients may also exist.

As discussed in Section II, after applying AFPS technique, gradient effect becomes less significant due to the area reduction. However, in order to reduce the gradients even more, special layout patterns have to be used.

Some well-known layout methods, including row-column switching scheme [1], Q^2 random walk scheme [2] and INL bounded switching scheme [9], are limited by factors such as insufficiency for reductions in two-dimensional gradient errors, very complex routing and high computational search complexity in worst case. In the following, a new layout technique is presented. With this layout technique, linear and quadratic gradients are cancelled, as well as some higher odd order gradients. Furthermore, it produces a certain pattern, which makes the routing much easier, and it does not require any search algorithms.

To better describe the layout technique, an 8×8 current-source array is considered. In this array, we are going to generate 4 current sources, where their linear and quadratic gradients are cancelled. In order to do that, a switching scheme has to be performed. Fig. 4 shows the seven steps of the switching procedure.

First of all, a sequence from 1 to 8 is filled in the first row of the current-source array. Second, odd numbers and even numbers are swapped, and the new sequence is placed in the second row. Third, a block consisting of 4 current sources are grouped, and there are four blocks in total. Fourth, block 1 and block 2 are swapped, and the same happens to the last two blocks. The new sequences are put in the third and fourth row. Fifth, first 4 blocks are re-grouped to a big block, and similarly the other 4 blocks are grouped to another big block. Sixth, the two big blocks are swapped again, and the new sequences are filled up with the rest of the array. Last, the current sources named 1 are combined with the ones named 8 to be current source A; the ones named 2 are grouped with the ones named 7 to be current source B; the ones named 3



A	B	C	D	D	C	B	A
B	A	D	C	C	D	A	B
C	D	A	B	B	A	D	C
D	C	B	A	A	B	C	D
D	C	B	A	A	B	C	D
C	D	A	B	B	A	D	C
B	A	D	C	C	D	A	B
A	B	C	D	D	C	B	A

1	2	3	4	5	6	7	8
2	1	4	3	6	5	8	7
3	4	1	2	7	8	5	6
4	3	2	1	8	7	6	5
5	6	7	8	1	2	3	4
6	5	8	7	2	1	4	3
7	8	5	6	3	4	1	2
8	7	6	5	4	3	2	1

Fig. 4 Layout switching sequence procedure

and 6 are combined to be current source C; and the remaining ones are formed current source D. In this way, four current sources with multiplier of 16 are generated, and their linear and quadratic gradients are cancelled, as well as some higher odd order gradients. Furthermore, there is a certain pattern in this layout sequence, which eases the layout routing, and more importantly makes the nonlinearities distribute equally for each current source.

Overall, this new layout technique is very effective for gradient error compensation, and it has less interconnect parasitic effect than the most of other layout techniques due to the easy layout routing, which improves the dynamic performance even more along with AFPS implementation.

IV. Conclusion

In this paper, two new techniques—after-fabrication programmable switching and the layout switching sequence—are introduced for the design of high-speed high-accuracy current steering DACs. AFPS method has broken the tradition of assigning switching sequence before chip implementation. Instead, it rearranges the sequence of current sources after fabrication and obtains the optimum linearity by generating an optimal switching sequence. The new layout technique has very easy routing and is very effective to compensate the gradient errors. By applying these two new techniques, it definitely reduces the chip area for the same or better static accuracy, and indeed improves the DAC's speed due to the reduced parasitic effect. The two methods will be implemented in a design of high resolution current-steering DAC, and the details will be provided shortly.

Reference

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

[2] 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.

[3] A. Rugeja and B. Song, "A self-trimming 14-b 100-MS/s CMOS DAC," *IEEE J. Solid-State Circuits*, vol. 35, pp. 1841–1852, Dec. 2000.

[4] Y. Cong and R. Geiger, "A 1.5-V 14-bit 100-MS/s self-calibrated DAC," *IEEE J. Solid-State Circuits*, vol. 38, pp. 2051 -2060, Dec. 2003.

[5] T. Chen and G. Gielen, "A 14-bit 200-MHz current-steering DAC with switching-sequence post-adjustment calibration," *IEEE J. Solid-state Circuits*, vol. 42, No. 11, pp. 2386-2394, Nov. 2007.

[6] Z. Yu, D. Chen, and R. Geiger, "1-D and 2-D switching strategies achieving near optimal INL for thermometer-coded current steering DACs," in *Proc. IEEE ISCAS*, pp. 909–912, May 2003.

[7] K. Lakshmikumar, R. Hadaway, and M. Copeland, "Characterization and modeling of mismatch in MOS transistors for precision analog design," IEEE J. Solid-State Circuits, vol. SC-21, pp. 1057–1066, Dec. 1986.

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

[9] Y. Cong and R. Geiger, "Switching sequence optimization for gradient error compensation in thermometer-decoded DAC arrays," IEEE Trans. Circuits Syst. II, vol. 47, pp. 585–595, Jul. 2000.