IEICE TRANS. ELECTRON., VOL.E95–C, NO.11 NOVEMBER 2012

PAPER

Balanced Switching Schemes for Gradient-Error Compensation in Current-Steering DACs

Xueqing LI[†], **Qi WEI**[†], *Nonmembers*, **Fei QIAO**[†], *Member*, *and* **Huazhong YANG**[†a)], *Nonmember*

SUMMARY This paper introduces balanced switching schemes to compensate linear and quadratic gradient errors, in the unary current source array of a current-steering digital-to-analog converter (DAC). A novel algorithm is proposed to avoid the accumulation of gradient errors, yielding much less integral nonlinearities (INLs) than conventional switching schemes. Switching scheme examples with different number of current cells are also exhibited in this paper, including symmetric arrays and non-symmetric arrays in round and square outlines. (a) For symmetric arrays where each cell is divided into two parallel concentric ones, the simulated INL of the proposed round/square switching scheme is less than 25%/40% of conventional switching schemes, respectively. Such improvement is achieved by the cancelation of linear errors and the reduction of accumulated quadratic errors to near the absolute lower bound, using the proposed balanced algorithm. (b) For non-symmetric arrays, i.e. arrays where cells are not divided into parallel ones, linear errors cannot be canceled, and the accumulated INL varies with different quadratic error distribution centers. In this case, the proposed algorithm strictly controls the accumulation of quadratic gradient errors, and different from the algorithm in symmetric arrays, linear errors are also strictly controlled in two orthogonal directions simultaneously. Therefore, the INLs of the proposed non-symmetric switching schemes are less than 64% of conventional switching schemes.

key words: *digital-to-analog converter, gradient errors, nonlinearity, switching scheme, the integral nonlinearity*

1. Introduction

Segmentation is widely used in wide-band digital-to-analog converters (DACs) with a high dynamic range [1]–[9]. In these DACs, the most-significant bits (MSBs) are thermometer-coded to steer a unary equally-weighted current array. It has been reported that the DAC's performance strongly relies on the matching of the MSB current sources in the unary current array [2], [3], [10], [11].

However, random matching errors and gradient errors cause nonlinearities [2], [10]–[12]. To describe the amount of nonlinearities, the integral nonlinearity (INL) is usually used to indicate the maximum deviation of the actual analog output from the ideal output. It is revealed that larger current source area and larger gate-source overdrive voltage reduce random matching errors [13]. The function of overdrive voltage optimization is limited due to the supply and the output voltage swing. In order to keep the influence of random errors negligible, large transistors are often adopted. Consequently, the current source array becomes so large that the gradient errors must be well compensated to satisfy the matching requirements [4]–[7].

Manuscript received May 25, 2012.

[†]The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing, China.

a) E-mail: yanghz@tsinghua.edu.cn

DOI: 10.1587/transele.E95.C.1790

Some other methods to compensate the gradient errors are calibration [3], trimming [4], and dynamic element matching (DEM) [10]. These methods improve current matching and decrease the size of the current array. However, these methods also increase the complexity and power of the DAC, because they need on-chip analog-to-digital converters [3], post-adjustments [4], or DEM encoders [10], etc. One effective and conventional way to compensate the errors is the optimization of the switching scheme, i.e. switching sequence, of current sources.

A switching scheme for the current cells in a unary array is a number sequence that determines which current sources to switch on: when the digital input value of the MSB segment is k, current cells with switching sequence numbers $1, 2, \ldots, k$ are turned on to form a total current of the same value in analog. Figure 1(a) shows an 8-bit switching scheme in which each current source is split into two parallel concentric ones. Note that some cells in the array drive more current than average (positive errors), whereas some drive less (negative errors), so a proper switching scheme might avoid the excessive accumulation of positive errors and that of negative errors. There have been some typical types of switching schemes: row-column switching schemes [5], [6], Q^2 random walk switching scheme [7], Q^N rotated walk switching scheme [14], the gradient-error and edge-effect tolerant (GET) switching scheme [15], and others as proposed in [11], [16]–[18]. As indicated in [5], [7], an intrinsic accuracy of 12-bit or even 14-bit has been successfully realized using proper switching schemes.

The shortcomings of the existing high-accuracy switching schemes are the usage of comparatively large transistors, and the need to split each current source into smaller ones located concentrically. The interconnection lines of these split current sources bring additional parasitic capacitance at the output current route and degrade the dynamic performance, especially at high frequencies [1], [8]. Therefore, it is of significance to improve the switching schemes so that no split or less split is necessary. From another point of view, given certain INL requirement, an improved switching scheme allows more headroom for random matching errors. Thus, the requirement on the size and overdrive voltage of current source transistors is relaxed, which makes it easier to design a DAC with smaller area and/or lower supply voltage. Aimed at these purposes, this paper proposes a novel algorithm, namely balanced algorithm, to generate improved switching schemes.

Section 2 introduces the gradient errors in a unary ar-

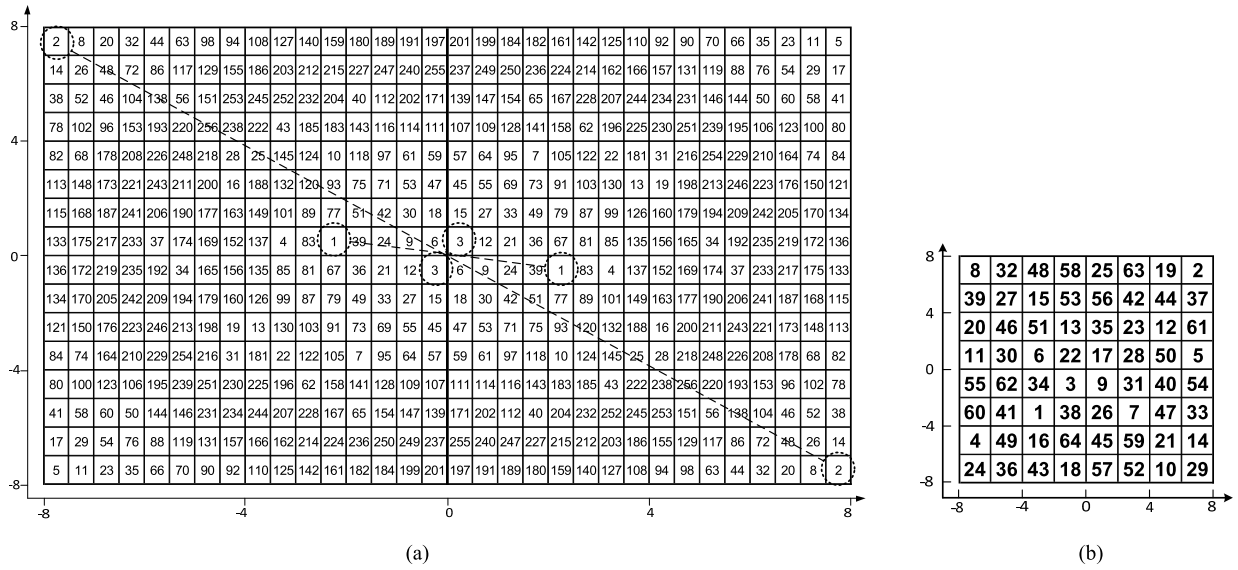

Fig. 1 Switching scheme examples generated by the proposed technique: (a) An 8-bit symmetric scheme; (b) A 6-bit non-symmetric scheme.

ray, and discusses the proposed algorithm in detail. Nonlinearity simulation results under different gradient error distributions will be given in Sect. 3. Finally, Sect. 4 gives the conclusion.

2. Proposed Balanced Switching Algorithm

2.1 Gradient Errors, *INL(k)*, and the Absolute INL Lower Bound

Oxide thickness, dopant variation, heat dissipation, mechanical stress, and voltage drop on the ground and supply lines are major sources of gradient errors [12], [16]. Given a horizontal coordinate x and a vertical coordinate y, the gradient errors could be described by the Taylor series as

$$\varepsilon(x,y) = a_0 + a_{11}x + a_{12}y + a_{21}x^2 + a_{22}y^2 + a_{23}xy + \ldots, \quad (1)$$

where the constant component a_0 contributes to no nonlinearities, a_{11} and a_{12} contribute to linear errors, a_{21}, a_{22} and a_{23} contribute to quadratic errors [7]. It has been reported that a_{21} has the same polarity as a_{22}, and that the errors along x and y directions are uncorrelated, i.e. $a_{23} \approx 0$ [11], [12], [18]. It is generally assumed that linear and quadratic errors are adequate to model nonlinearities, so higher-order errors are not considered in this paper [11], [14]–[18]. Figure 2 depicted typical zero-averaged models of linear and centered quadratic error distributions.

The *INL* of a DAC is universally defined as the maximum deviation of the DAC's actual analog output from the ideal output at all possible input code k, $k = 1, 2, \ldots, 2^{NOB} - 1$, where *NOB* is the number of bits of the DAC. In this paper, *INL(k)* defines the deviation of the actual analog output from the ideal output at input code k.

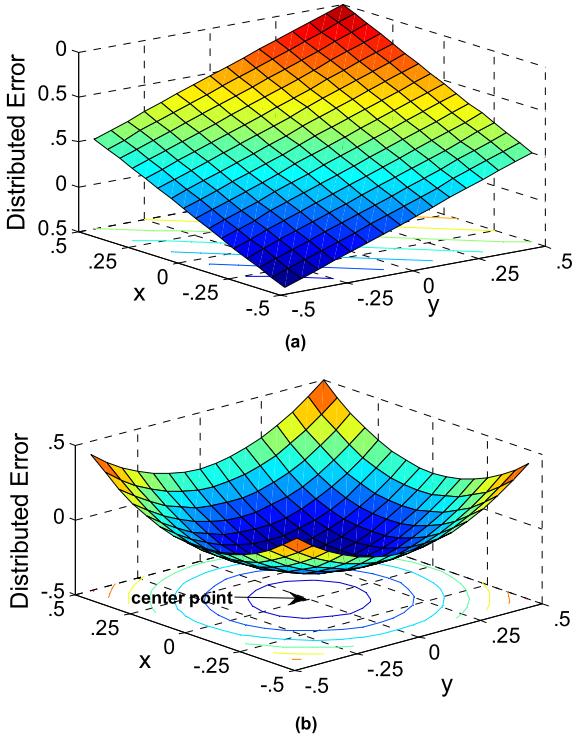

Fig. 2 Typical zero-averaged models of error distributions: (a) linear error; (b) centered quadratic error.

The absolute INL lower bound for a unary array is defined as the minimum absolute INL of the array that could be achieved. Assuming a zero-averaged error distribution $\varepsilon(x,y)$ has a maximum value $\varepsilon_{\max}$ (positive) and a minimum value $\varepsilon_{\min}$ (negative), the absolute INL lower bound

$$INL_{abs\_lower\_bound} \geq \max\{\varepsilon_{\max}/2, -\varepsilon_{\min}/2\}. \quad (2)$$

This conclusion has been proved in [11]. The equal

sign may not be achieved sometimes. For example, if the error sequence is {−1, −3, 4}, then the absolute INL lower bound is 3, rather than max{4/2, −(−3)/2} = 2.

2.2 Symmetric Unary Arrays and Non-Symmetric Unary Arrays

In this paper, a symmetric unary array is a unary array where each current source is split into two or more parallel equally-weighted cells and located concentrically. The current of the parallel cells of the same current source is gathered together using interconnections in the chip layout. Symmetric unary arrays have been applied in many designs as in [2], [6], [7], [15]. Different from symmetric unary arrays, a non-symmetric unary does not split current sources into smaller ones and locate them concentrically. The readers should note that the words "symmetric" and "non-symmetric" here do not refer to the geometrical outline of the array, but the fact whether or not a current source is split and located concentrically.

The introduction section of this paper has already defined the switching schemes. Corresponding to symmetric and non-symmetric unary arrays, there are symmetric and non-symmetric switching schemes. Figure 1(a) is an example of 8-bit symmetric switching scheme and Fig. 1(b) is an example of 6-bit non-symmetric switching scheme. There is essential difference between symmetric and non-symmetric switching schemes. In symmetric unary arrays, linear errors are fully canceled, and the change of the quadratic error distribution center has no effect on the INL performance [16]. For one thing, it is because the positive and negative linear errors of each pair of concentric cells have the same amount and sum to zero when forming a current source. For another, it is because the change of the quadratic error distribution center is identically to the change of linear error distributions, which are canceled in symmetric arrays. Therefore, the proposed symmetric switching schemes focus on minimizing only the accumulation of quadratic errors, while non-symmetric switching schemes minimize the accumulation of linear and quadratic errors simultaneously.

As introduced in Sect. 1, doubling the number of the current cells may lead to poor dynamic performance. Therefore, all symmetric unary arrays in this paper divide each current source into only two half-size concentric cells.

2.3 Generation of Symmetric Switching Schemes for Quadratic Error Compensations

For a symmetric unary array composed of several concentric blocks, switching sequence generation of one block is identical to that of the whole array. As illustrated in Fig. 3(a), the following steps are taken to generate the sequence number for each current cell in a block:

Step I: Definitions and initializations of the variables and constants as depicted in Table 1. In addition, the quadratic errors are initialized as a zero-averaged distribution centered in the middle of the block.

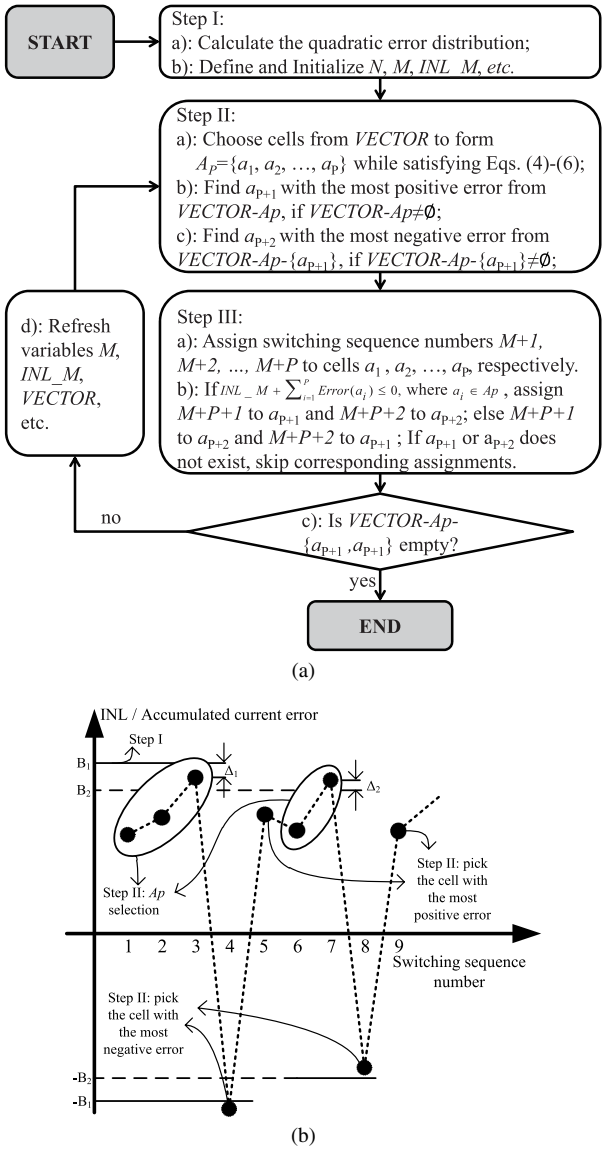

Fig. 3 Proposed algorithm to generate symmetric switching schemes: (a) The flow chart; (b) An example.

Table 1 Definitions and initializations of the proposed generation of symmetric switching schemes.

Constants, Variables	Definitions and descriptions	Initial Values
N	The number of the current cells in one block of the symmetric array	Constant value
M	The number of all the current cells that have been assigned a switching sequence number	0
INL_M	The current error sum of all the *M* current cells that have been assigned a switching sequence number	0
VECTOR	A vector that contains all the *N-M* cells that have NOT been assigned a switching sequence number.	All the cells
a_i	Representing a cell in *VECTOR*, *i*=1, 2, …, *N-M*	/
$Error(a_i)$	The current error of cell a_i in *VECTOR*	/
B	The absolute lower bound of *VECTOR*	/
Δ	The allowed margin between the absolute INL lower bound and the estimated INL yielded by this proposed algorithm	≈0 *
Ap	A sequence number segment generated in each loop	Ø

*: Δ should be slightly larger than zero at the first loop; If the generations in step II and step III are successful, Δ can be lowered and the algorithm may continue.

Step II: Generation of new sequence number segment. Firstly, randomly choose as few cells as possible from *VECTOR* to form a sequence *Ap*:

$$Ap \triangleq \{a_1, a_2, \ldots, a_P\}, \tag{3}$$

while satisfying

$$\left| INL\_M + \sum_{i=1}^{m} Error(a_i) \right| \le B + \Delta, \text{ where } m = 1, 2, \cdots, P, \tag{4}$$

$$\left| INL\_M + \left(\sum_{i=1}^{P} Error(a_i) \right) + B \right| \le \Delta, \text{ when } INL\_M \le 0, \tag{5}$$

$$\left| INL\_M + \left(\sum_{i=1}^{P} Error(a_i) \right) - B \right| \le \Delta, \text{ when } INL\_M > 0. \tag{6}$$

The purpose of Eq. (4) is to make sure that the accumulated current errors do not exceed the absolute INL lower bound of *VECTOR*, when cells $a_1, a_2, \ldots, a_P$ are assigned in step III. The purpose of Eqs. (5) and (6) is to keep the accumulated current errors close to B, which guarantees the accumulated current errors under control after the $(M + P + 1)$th and $(M + P + 2)$th cells are assigned in step III. Usually it needs only one or two cells under typical error distributions to form *Ap* in Eq. (3) while satisfying Eqs. (4)–(6), at the assumption of a reasonable Δ.

Next, if *VECTOR-Ap* $\neq \emptyset$, find a_{P+1} with the most *positive* current error from *VECTOR-Ap*. If *VECTOR-Ap*-$\{a_{P+1}\}$ $\neq \emptyset$, find a_{P+2} with the most *negative* current error from *VECTOR-Ap*-$\{a_{P+1}\}$.

Step III: Refreshment of the sequence numbers and variables. If $INL\_M + \sum_{i=1}^{P} Error(a_i) \le 0$, where $a_i \in Ap$, as in the case of Eq. (5), assign switching sequence numbers $M + 1, M + 2, \ldots, M + P, M + P + 1, M + P + 2$ to the current cells $a_1, a_2, \ldots, a_P, a_{P+1}, a_{P+2}$, respectively. Else, as in the case of Eq. (6), assign sequence numbers $M + 1$, $M + 2, \ldots, M + P$, $M + P + 1$, $M + P + 2$ to the current cells a_1 , $a_2, \ldots, a_P$, a_{P+2}, a_{P+1}, respectively. Note that if one or two cells of a_{P+1} or a_{P+2} does not exist, skip the corresponding assignment of sequence numbers $M + P + 1$ or $M + P + 2$. Next, refresh variables M, B, INL_M, *VECTOR*, etc., according to their definitions. If one or more current cells has not been assigned a switching sequence number, go back to step II and loop again. Else, the generation flow is finished.

Figure 3(b) shows an example of generating a symmetric scheme. Current cells {1, 2, 3} and {6, 7} are two *Ap*'s generated in step II, {4, 5} are two cells with the most negative and positive current errors assigned in step III in the case of Eq. (6). {8, 9} is similar to {4, 5}.

Usually, the formation of *Ap* in step III is not unique, so the proposed algorithm is able to generate a series of switching schemes with similar INL performances. The INL simulation results will be presented in Sect. 3, along with the comparisons with other typical switching schemes.

2.4 Generation of Non-symmetric Switching Schemes for the Compensation of Linear and Quadratic Errors

It is more complicated to generate non-symmetric switching schemes than symmetric switching schemes, because both linear and quadratic errors must be handled at the same time in non-symmetric switching schemes. When generating a switching scheme for a non-symmetric unary array, the most difficult problem to deal with is that the INL performance deteriorates as the quadratic error distribution center moves. If the distribution center moves from one side of the array to another, the error distribution changes greatly.

The proposed algorithm solves this problem by compensating the quadratic errors as in symmetric arrays, while keeping the accumulation of linear errors strictly constrained. In detail, the algorithm controls linear errors by minimizing two orthogonal linear errors. This comes from the fact that any quadratic error distribution can be a linear combination of two orthogonal linear errors and one centered quadratic error distribution:

$$\begin{aligned}
&\varepsilon_{\text{general_quadratic}}(x, y) \\
&\quad = a_0 + a_{11}x + a_{12}y + a_{21}x^2 + a_{22}y^2 \\
&\quad = a_{21}(x - x_0)^2 + a_{22}(y - y_0)^2 + A(x - x_0) + B(y - y_0) + C \\
&\quad \overset{\substack{x'=x-x_0 \\ y'=y-y_0}}{\longleftrightarrow} \varepsilon_{\text{centered_quadratic}}(x', y') + Ax' + By' + C',
\end{aligned} \tag{7}$$

where

$$\begin{aligned}
&A = a_{11} + 2a_{21}x_0, B = a_{12} + 2a_{22}y_0, \\
&C = a_0 - a_{21}x_0^2 - a_{22}y_0^2, C' = C + Ax_0 + By_0, \\
&\varepsilon_{\text{centered_quadratic}}(x', y') = a_{21}(x - x_0)^2 + a_{22}(y - y_0)^2
\end{aligned} \tag{8}$$

and (x_0, y_0) represents the coordinate of the array center. Therefore, an intuitional solution is to deal with the quadratic and linear errors separately. In fact, it is effective because these errors are independent of each other. Furthermore,

$$\begin{aligned}
&INL_{\text{general_quadratic}} = \max\left\{ abs\left(\sum_{i=1}^{k} \varepsilon_{\text{general_quadratic}}(x_i, y_i) \right) \right\} \\
&\le \max\left\{ abs\left(\sum_{i=1}^{k} \varepsilon_{\text{centered_quadratic}}(x_i, y_i) \right) \right\} \\
&+ \max\left\{ abs\left(\sum_{i=1}^{k} (Ax_i) \right) \right\} + \max\left\{ abs\left(\sum_{i=1}^{k} (By_i) \right) \right\} \\
&= INL_{\text{centered_quadratic}} + INL_{x_\text{linear}} + INL_{y_\text{linear}}.
\end{aligned} \tag{9}$$

If $INL_{\text{centered\_quadratic}}$, $INL_{\text{x\_linear}}$, and $INL_{\text{y\_linear}}$ in Eq. (9) are strictly constrained, so will the $INL_{\text{general\_quadratic}}$. As a result, in the way of minimizing the accumulated quadratic errors, the proposed algorithm is similar to the algorithm in generating symmetric switching schemes, but different in the need of constraining $INL_{\text{x\_linear}}$, and $INL_{\text{y\_linear}}$. On the other hand, the proposed algorithm compensates linear errors by avoiding large $\text{abs}(\Sigma(Ax_i))$ and $\text{abs}(\Sigma(By_i))$ in Eq. (9). Further, in the process of sequence generation, small $\text{abs}(\Sigma(Ax_i))$ and $\text{abs}(\Sigma(By_i))$ at the same

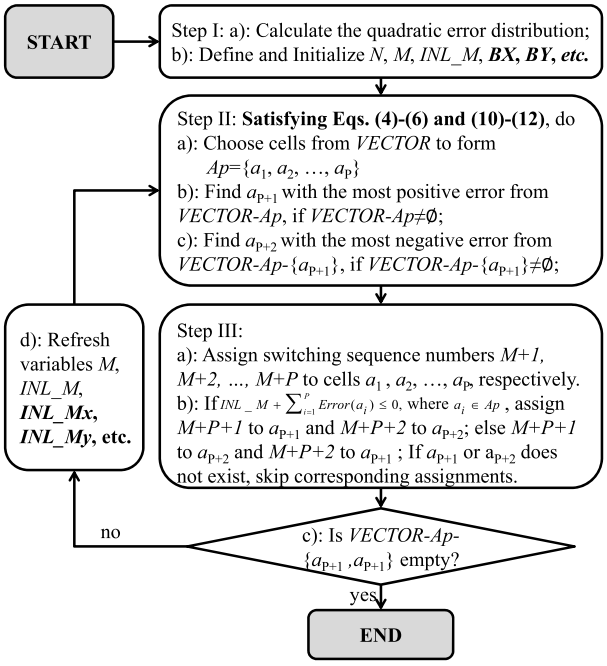

Fig. 4 Flow chart of the proposed algorithm to generate non-symmetric switching schemes.

time are also avoided, in case both of them are inevitably large because the linear error coefficients A and B in Eq. (8) may have similar distributions.

By the balanced control on both quadratic and linear errors, the final yielded INL is strictly controlled no matter where the quadratic error distribution center moves or how the linear error distribution coefficients change. This is the reason why the name of the proposed algorithm is balanced algorithm. Finally, it should be emphasized that the two orthogonal linear errors in Eq. (7) can also be others, for example, $x + y$ and $x - y$.

The flow chart of generating a non-symmetric switching scheme is depicted in Fig. 4, which is similar to Fig. 3(a). To make the description concise, only the differences in each step are listed below, and the coordinate of the array center is set to be (0, 0).

Step I: Except for those in Table 1, additional definitions and initializations are summarized in Table 2.

Step II: Along with conditions in Eqs. (4)–(6), the following conditions also need to be satisfied when choosing the current cells $a_1, a_2, \ldots, a_P, a_{P+1}$ and a_{P+2}:

$$\left| INL\_Mx + \left(\sum_{i=1}^{m} \mathrm{X}(a_i) \right) \right| \le BX + \Delta_x, \quad m = 1, 2, \cdots, P+2, \quad (10)$$

$$\left| INL\_My + \left(\sum_{i=1}^{m} \mathrm{Y}(a_i) \right) \right| \le BY + \Delta_y, \quad m = 1, 2, \cdots, P+2, \quad (11)$$

$$||\mathrm{X}(a_i)| + |Y(a_i)| - BXY| \le \Delta_{xy}, \quad m = 1, 2, \cdots, P+2. \quad (12)$$

Equations (10) and (11) keep $\sum_{i=1}^{k}(Ax_i)$ and $\sum_{i=1}^{k}(By_i)$ in Eq. (9) tightly constrained, and Eq. (12) is used to avoid large error accumulations under combinational linear error distributions: $\varepsilon(x, y) = Ax + By$ where $AB \neq 0$.

Step III: two additional variables *INL_Mx* and *INL_My* need to be refreshed.

Figure 1(b) demonstrates a 6-bit non-symmetric square array. Figure 5 shows three corresponding INL curves of it under centered quadratic error distribution $\varepsilon(x, y) = (x^2 + y^2)/100$, linear error distribution $\varepsilon(x, y) = 0.1x$, and $\varepsilon(x, y) = 0.1y$. While minimizing the accumulated quadratic errors, the accumulated linear errors in x direction and y direction are never too small or large at the same time. By doing this, the accumulated errors are strictly controlled even if the coefficients of linear errors or the center of the quadratic error changes, as proved by the simulation results in Sect. 3.

Table 2 Additional definitions and initializations of the proposed algorithm for generating non-symmetric switching schemes.

Constants& Variables	Definitions	Initial Values
$X(a_i)$	x-coordinate of current cell a_i in *VECTOR*	/
$Y(a_i)$	y-coordinate of current cell a_i in *VECTOR*	/
BX	Half of the maximum x-coordinate of the current cells in the array	Constant value
BY	Half of the maximum y-coordinate of the current cells in the array	Constant value
BXY	$BX+BY$	Constant value
INL_Mx	The sum of x-coordinates of all the cells that have been assigned a sequence number	0
INL_My	The sum of y-coordinates of all the cells that have been assigned a sequence number	0
Δ_x	The allowed margin between INL_Mx and BX	/
Δ_y	The allowed margin between INL_My and BY	/
Δ_{xy}	The allowed margin between $\|X(a_i)\|+\|Y(a_i)\|$ and BXY	/

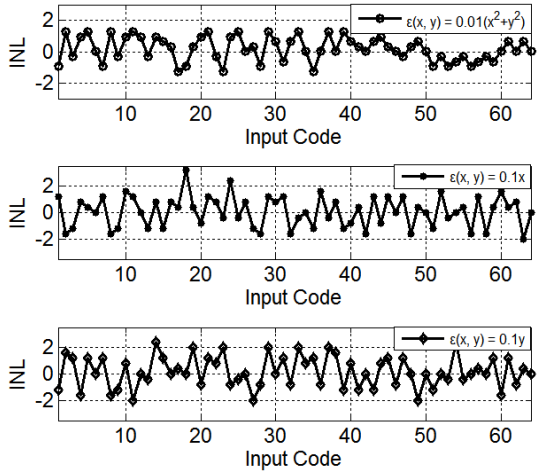

Fig. 5 Three INL curves of the non-symmetric switching scheme in Fig. 1(b) under the centered quadratic error $\varepsilon(x, y) = (x^2 + y^2)/100$, linear error along x: $\varepsilon(x, y) = x/10$, and linear error along y: $\varepsilon(x, y) = y/10$. Each current source occupies an area of 2×2 as explained in Sect. 3.

2.5 Proposed Switching Schemes with a Round Shape for Unary Arrays

Conventional unary arrays have a square outline. Such design fits the shape of a regular chip. However, under linear and quadratic gradient error distributions, the errors in the

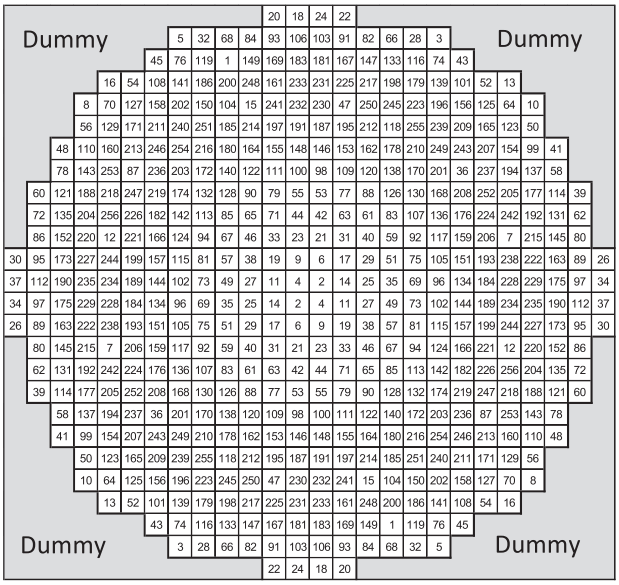

Fig. 6 An 8-bit round symmetric switching scheme generated by the proposed algorithm.

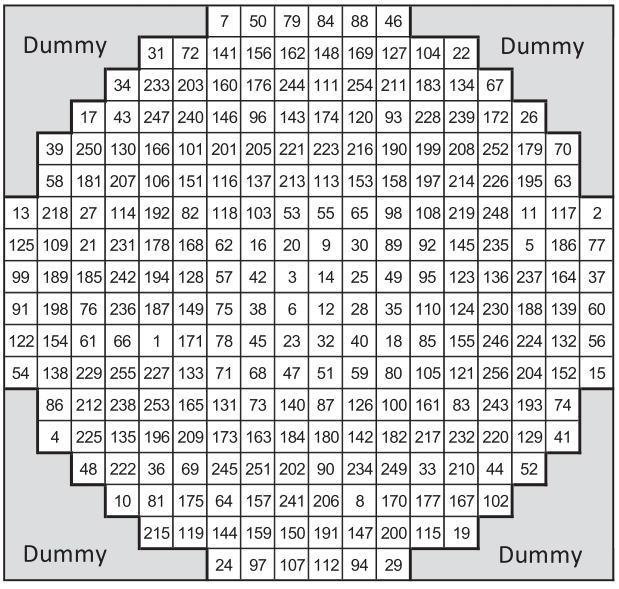

Fig. 7 An 8-bit round non-symmetric switching scheme generated by the proposed algorithm.

four corners of the square are the largest, and they directly determine the absolute INL lower bound. Therefore, an intuitional optimization is to reshape the square array to be round [17]. Two 8-bit round switching sequences generated by the proposed algorithm are shown in Fig. 6 and Fig. 7, representing one symmetric array and one non-symmetric array, respectively.

3. Simulations

3.1 Simulation Settings

The proposed switching schemes in Fig. 1(a) (8-bit, symmetric, square), Fig. 1(b) (6-bit, non-symmetric, square), Fig. 6 (8-bit, symmetric, round) and Fig. 7 (8-bit, non-symmetric, round) are simulated and compared with traditional switching schemes.

For the purpose of convenience, the origin (0, 0) is located at the array center of both symmetric and non-symmetric unary arrays in the simulations. This setting is different from the origin of a symmetric unary in the flow chart of the proposed algorithm. In that case, the origin is located in the center of one *block* of the unary array, rather than the center of the whole *array*.

For a current-steering DAC with a resolution of 12-bit or more, the number of bits in the MSB segment is usually 6 to 8 [1]–[3], [5], [7], [8]. Although the number of bits in the MSB segment varies in different designs, the area of the MSB segment is nearly the same to satisfy the matching requirement under random mismatch errors [5]–[7]. Therefore, the second setting is that all 6-bit and 8-bit unary arrays occupy the same area, and have the same square or round outline. As a result, one cell in the non-symmetric array in Fig. 7 has twice the area of one in the symmetric arrays in Fig. 1(a) and Fig. 6. In addition, one current cell in a 6-bit unary array occupies an area of 2×2, which is the same size as four current cells in an 8-bit non-symmetric unary array shown in Fig. 7.

In the simulations, linear errors are expressed as

$$\varepsilon(x, y) = a_{11}x + a_{12}y, \tag{13}$$

where the coefficients a_{11} and a_{12} varies from −0.1 to 0.1. Quadratic errors in the simulations are expressed as

$$\varepsilon(x, y) = a_{21}(x - x_0)^2 + a_{22}(y - y_0)^2, \tag{14}$$

where the coefficients $a_{21} = a_{22} = 0.01$, and the error distribution center (x_0, y_0) varies inside the whole array.

It is important to note that the yielded INL in the simulations do not have a unit of "least significant bit (LSB)" or "μA". This is because the magnitudes of the linear and quadratic errors are for the purpose of performance comparisons among different switching schemes, and are unlikely the exact gradient errors in fabricated chips.

3.2 Simulations for Symmetric Unary Arrays

Figure 8 shows the INL curve under quadratic errors in different 8-bit symmetric switching schemes. Figures 8(a)–(d) illustrate the INL curves of conventional switching schemes, i.e. the row-column switching scheme [4], Q^2 switching scheme [7], GET switching scheme [15], and SPBR switching scheme [16], respectively. The resulted INLs are 3.92, 1.39, 0.99, and 0.93, respectively. Figure 8(e) is the INL

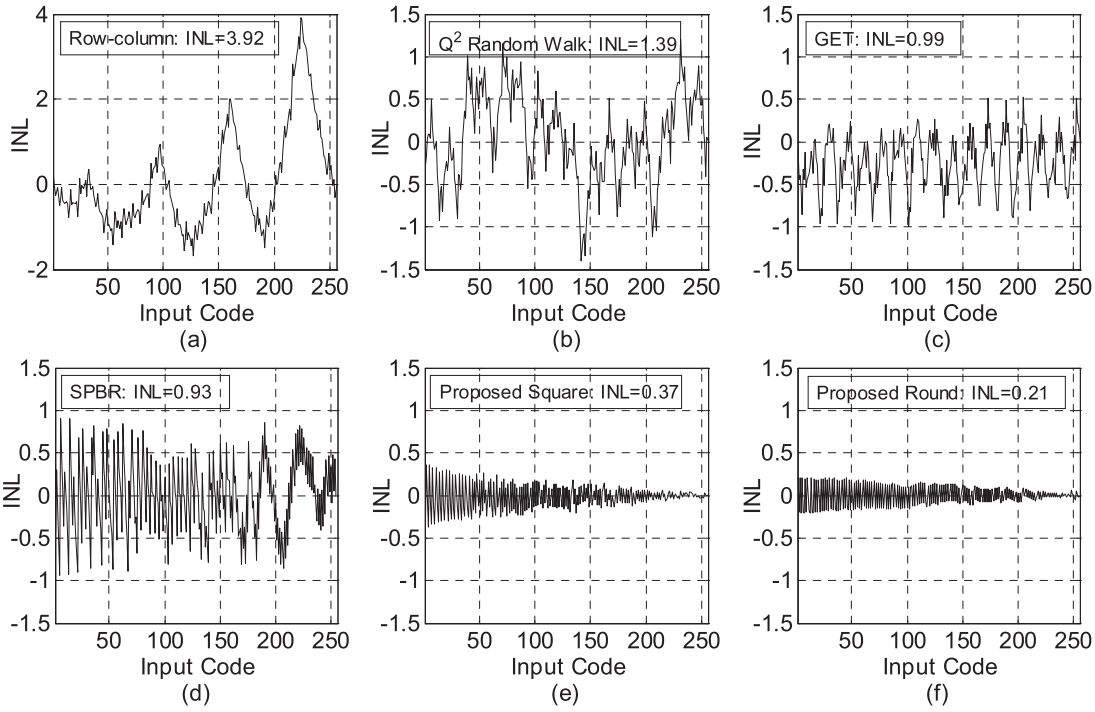

Fig. 8 Simulated INL curves under quadratic errors in symmetric arrays: (a) Row-column switching scheme, (b) Q^2 Random Walk switching scheme, (c) GET switching scheme, (d) SPBR Switching scheme, (e) proposed symmetric square switching scheme in Figs. 1(a), (f) proposed symmetric round switching scheme in Fig. 6.

curve of the proposed symmetric *square* switching sequence in Fig. 1(a), and the yielded INL is 0.375, exceeding the absolute INL lower bound 0.369 slightly. Compared with the referenced conventional square switching schemes, the INL of the symmetric square switching scheme is reduced to less than 40%.

Figure 8(f) shows the simulated INL curve under quadratic errors of the proposed symmetric *round* switching scheme shown in Fig. 6. The simulated INL curve varies from −0.208 to 0.209, exceeding the absolute INL lower bound 0.203 slightly. The INL of 0.209 is less than 25% of the referenced conventional switching schemes.

3.3 Simulations for Non-symmetric Unary Arrays

Different from symmetric arrays, non-symmetric arrays need to compensate both linear errors with varied coefficients and quadratic errors with varied distribution centers at the same time. In the simulations, the 8-bit round non-symmetric switching sequence shown in Fig. 7, and the 6-bit non-symmetric square switching sequence shown in Fig. 1(b) are simulated and compared with conventional switching sequences. The INL simulation results for GET switching sequence, Cretti's switching scheme, and proposed switching sequence are plotted in Fig. 9. The other conventional switching sequences result in larger INLs.

All simulations results are added into the INL simulation summary in Table 3. In the 8-bit non-symmetric case, it is clear that the proposed algorithm reduces the INL much more effectively than conventional algorithms. The average simulated INLs under linear and quadratic errors are 0.54 and 0.90, respectively. These results are less than 64% of the referenced best conventional 8-bit non-symmetric switching

Table 3 Summary of simulated INL under linear and quadratic errors.

	Switching Schemes	Average INL by linear errors*	Average INL by quadratic errors*
8-bit symmetric	Row-Column [4]	0	3.92
	Q^2 Random Walk [7]	0	1.39
	GET [15]	0	0.99
	SPBR [16]	0	0.93
	Proposed square scheme	**0**	**0.37**
	Proposed round scheme	**0**	**0.21**
	$INL_{abs\_lower\_bound,square}$	0	0.37
	$INL_{abs\_lower\_bound,round}$	0	0.21
8-bit Non-symmetric	Row-Column [4]	6.49	13.0
	Q^2 Random Walk [7]	2.7	4.08
	The INL-bounded [11]	1.9	3.31
	GET [15]	0.92	1.80
	Cretti's Patent [17]	0.84	1.40
	Proposed round scheme	**0.54**	**0.90**
	$INL_{abs\_lower\_bound,square}$	0.42	0.95
	$INL_{abs\_lower\_bound,round}$	0.37	0.71
6-bit Non-symmetric	Q^N rotated walk [14]	4.47	4.79
	Kuo's scheme [18]	5.21	15.0
	Proposed square scheme	**2.18**	**2.62**
	$INL_{abs\_lower\_bound,square}$	1.56	2.24

*: The average INL: For linear errors in Eq. (13), the coefficients a_{11} and a_{12} uniformly varies from -0.1 to 0.1; For quadratic errors in Eq. (14), the distribution center (x_0, y_0) uniformly varies inside the whole array.

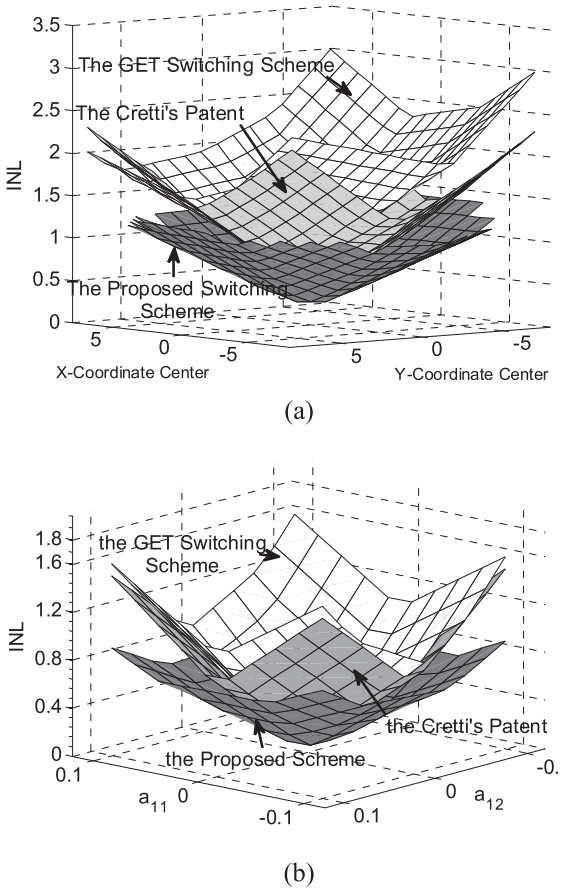

Fig. 9 INL simulation results of 8-bit non-symmetric arrays under: (a) quadratic errors with varied center; (b) linear errors with varied x coefficient a_{11} and y coefficient a_{12}.

schemes.

In the 6-bit case, the referenced best switching scheme gives a result of 4.47 and 4.79 under linear and quadratic errors, respectively, by the Q^N rotated-walk switching scheme. The INL of the proposed switching scheme under linear and quadratic errors is 2.18 and 2.62, respectively. The proposed 6-bit switching scheme improves the INL by a reduction of more than 45.3%.

4. Conclusion

This paper has presented the novel balanced algorithm for generating switching schemes to compensate the gradient errors in current-steering DACs. It is applied successfully in both symmetric and non-symmetric, round and square arrays. The generation flow chart of the proposed algorithm has been described in detail. Although the algorithm has not been tested with fabricated chips, the simulations results have proved that the proposed switching schemes yield a much smaller INL than conventional switching schemes. In the near future, fabricated DAC chips with different resolutions and array shapes will be statistically tested and compared with conventional schemes. Future work of applying the new switching sequences in current-steering DACs is of significance.

References

[1] W.H. Tseng, C.W. Fan, and J.T. Wu, "A 12-bit 1.25-GS/s DAC in 90 nm CMOS with 70 dB SFDR up to 500 MHz," IEEE J. Solid-State Circuits, vol.46, no.12, pp.2845–2856, Dec. 2011.

[2] K. Doris, J. Briaire, D. Leenaerts, M. Vertregt, and A.v. Roermund, "A 12b 500 MS/s DAC with >70 dB SFDR up to 120 MHz in 0.18 μm CMOS," IEEE International Solid State Circuits Conf. (ISSCC) Digest of Technical Papers, pp.116–117, San Francisco, CA, Feb. 2005.

[3] W. Schofield, D. Mercer, and L.S. Onge, "A 16b 400 MS/s DAC with < -80 dBc IMD to 300 MHz and < -160 dBm/Hz noise power spectral density," IEEE International Solid State Circuits Conf. (ISSCC) Digest of Technical Papers, pp.126–127, San Francisco, CA, Feb. 2003.

[4] J. Hyde, T. Humes, C. Diorio, M. Thomas, and M. Figueroa, "A 300-MS/s 14-bit digital-to-analog converter in logic CMOS," IEEE J. Solid-State Circuits, vol.38, no.5 pp.734–740, May 2003.

[5] J. Bastos, A.M. Marques, M.S.J. Steyaert, and W. Sansen, "A 12-Bit intrinsic accuracy high-speed CMOS DAC," IEEE J. Solid-State Circuits, vol.33, no.12, pp.1959–1969, Dec. 1998.

[6] A.V. d. Bosch, M.A.F. Borremans, M.S.J. Steyaert, and W. Sansen, "A 10-bit 1-GSample/s Nyquist current-steering CMOS D/A converter," IEEE J. Solid-State Circuits, vol.36, no.3, pp.315–324, March 2001.

[7] G.A.M. Van der Plas, J. Vandenbussche, W. Sansen, M.S.J. Steyaert, and G.E. Gielen, "A 14-bit intrinsic accuracy Q2 random walk CMOS DAC," IEEE J. Solid-State Circuits, vol.34, no.12, pp.1708–1718, Dec. 1999.

[8] C-H. Lin, F. van der Goes, J. Westra, J. Mulder, Y. Lin, E. Arslan, E. Ayranci, X. Liu, and K. Bult, "A 12b 2.9 GS/s DAC with IM3 $<$ -60 dBc Beyond 1 GHz in 65 nm CMOS," IEEE International Solid State Circuits Conf. (ISSCC) Digest of Technical Papers, pp.4–75, San Francisco, CA, Feb. 2009.

[9] D.H. Lee, T.H. Kuo, and K.L. Wen, "Low-cost 14-Bit current-steering DAC with a randomized thermometer-coding method," IEEE Trans. Circuits Syst. II, vol.56, no.2, pp.137–141, Feb. 2009.

[10] K.L. Chan, J.Y. Zhu, and I. Galton, "Dynamic element matching to prevent nonlinear distortion from pulse-shape mismatches in high-resolution DAC's," IEEE J. Solid-State Circuits, vol.43, no.9, pp.2067–2078, Sept. 2008.

[11] Y.H. Cong and R.L. Geiger, "Switching sequence optimization for gradient error compensation in thermometer-decoded DAC arrays," IEEE Trans. Circuits Syst. II, vol.47, no.7, pp.585–595, July 2000.

[12] J. Bastos, M. Steyaert, B. Graindourze, and W. Sansen, "Influence of die attachment on MOS transistor matching," IEEE Trans. Semicond. Manuf., vol.10, no.2, pp.209–217, May 1997.

[13] 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, no.5, pp.1433–1439, Oct. 1989.

[14] D.H. Lee, Y.H. Lin, and T.H. Kuo, "Nyquist-rate current-steering digital-to-analog converters with random multiple data-weighted averaging technique and Q^N rotated walk switching scheme," IEEE Trans. Circuits Syst. II, vol.53, no.11, pp.1264–1268, Nov. 2006.

[15] J. Deveugele, G. Plas, M. Steyaert, G. Gielen, and W. Sansen, "A gradient-error and edge-effect tolerant switching scheme for a high-accuracy DAC," IEEE Trans. Circuits Syst. I, vol.51, no.1, pp.191–195, Jan. 2004.

[16] M. Vadipour, "Gradient error cancellation and quadratic error reduction in unary and binary D/A converters," IEEE Trans. Circuits Syst. II, vol.50, no.12, pp.1002–1007, Dec. 2003.

[17] F. Cretti and F. Albertoni, "Method of generating a switching sequence for an unary array of conducting branches and a relative thermometrically decoded digital-to-analog converters," U.S. Patent 20050012650, 2005.

[18] K.C. Kuo and C.W. Wu, "A switching sequence for linear gradient error compensation in the DAC design," IEEE Trans. Circuits Syst. II, vol.58, no.8, pp.502–506, Aug. 2011.

Xueqing Li was born in Changting, Fujian Province, in 1985. He received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2007. He became a PHD candidate in microelectronics since 2007. His current interests are wide-band high-dynamic-range data converters.

Qi Wei was born in ningxia province in 1983. He received the Ph.D. degree from Tsinghua University, BeiJing, in 2010. He is currently a postdoctor in Department Of Electronic Engineering Tsinghua University. His research interests focus on analog IC design and high performance data convert including high performance operational amplifier, Pipeline ADC SAR ADC current steer DAC.

Fei Qiao is currently a faculty member of NICS (Nanoscale Integrated Circuits and Systems) Laboratory, Tsinghua University, Beijing, China. He received his B.S. degree from Lanzhou University in July, 2000, then furthered his study in Dept. of Electronic Engineering, Tsinghua University and obtained his Ph.D. degree in July, 2006.

Huazhong Yang (M'97-SM'00) was born in Ziyang, Sichuan Province, China, on Aug. 18, 1967. He received the B.S. degree in microelectronics, and the M.S. and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1989, 1993, and 1998, respectively. In 1993, he joined the Department of Electronic Engineering, Tsinghua University, Beij ing, where he has been a Full Professor since 1998. His research interests include chip design for communication and multimedia applications, synthesis of analog integrated circuits (IC), power estimation and synthesis of digital ICs, noise and delay estimation of deep submicron ICs, yield enhancement, optimization and modeling. He has been in charge of several projects, including projects sponsored by the 863 program, NSFC, the 9th five-year national program and several international cooperation projects. He has authored and coauthored over 100 technical papers and 6 books. Dr. Yang was recognized as '2000 National Palmary Young Researcher by NSFC.