Constructive Placement and Routing for Common-Centroid Capacitor Arrays in Binary-Weighted and Split DACs

Nibedita Karmokar, Arvind K. Sharma, Jitesh Poojary,
Meghna Madhusudan, Ramesh Harjani, and Sachin S. Sapatnekar

***Abstract*—Process variations and the effect of interconnect parasitics can cause significant perturbations in the performance metrics of capacitive digital-to-analog converters (DACs). This paper develops fast constructive procedures for common-centroid placement and routing for binary-weighted and split capacitor array topologies of charge-sharing DACs. Our approach particularly targets FinFET technologies with high wire and via parasitics: in these technology nodes, we show that the switching speed of the capacitor array, as measured by the 3dB frequency, can be severely degraded by these parasitics, and develop techniques to place and route the capacitor array, for both binary-weighted and split DACs, to optimize the switching speed. A balance between 3dB frequency and DAC INL/DNL is shown by trading off via counts with dispersion. The approach delivers high-quality results with low runtimes.**

I. Introduction

Charge-scaling digital-to-analog converters (DACs) (Fig. 1(a)) are widely used in analog design, and their precision, accuracy and performance depend on building well-matched capacitor arrays with binary-weighted capacitor ratios [1], [2]. It is widely recognized that these ratios may be perturbed by systematic or random mismatch and designers have long used the common-centroid (CC) layout technique to cancel the impact of the linear systematic mismatch [3], [4]. This work considers the problem of CC layout for binary-weighted capacitor arrays for DACs, taking advantage of the problem structure to optimize DAC performance metrics. Several related works (e.g., [5], [6]) address CC layout in general, but do not leverage the specific properties of DACs. Among methods that specifically target DAC structures, many prior CC placement methods [6]–[8] ignore the impact of routing parasitics, which are critically important to DAC performance. Methods that do incorporate routing considerations [9]–[11] are based on computationally expensive stochastic search [1], [2], [9], [11]. In this paper, we develop a set of fast, constructive approaches to obtain routability-conscious performance-driven CC capacitor array layout for DACs, with a special emphasis on designing layouts for FinFET technology nodes.

One of the classical approaches for building a capacitive DAC uses a set of binary-weighted capacitors to provide the required capacitor ratios. For an N-bit DAC, the number of unit capacitors is 2^N: as a result, both the silicon area and total capacitance of a binary-weighted DAC increases exponentially with N, causing power dissipation to rise exponentially with the number of bits [12] and resulting in larger settling times due to the large charging time constant required to charge/discharge the capacitors. These problems can be addressed by using a split DAC [13], [14], in which an additional attenuation capacitor is used to separate an L-bit LSB capacitor array from an $(N-L)$-bit MSB array in the schematic. This is depicted in Fig. 1(b), where the LSB, D_1, corresponds to the leftmost switch, and the MSB, D_N, to the rightmost switch. Let C_T^{LSB} [C_T^{MSB}] is the total capacitance from the LSB [MSB] array. The value of the attenuation capacitor, C_A, is given by

$$C_A = (C_T^{LSB}/C_T^{MSB})C_u \tag{1}$$

This value is, in general, not an integer, and is typically not a multiple of the unit capacitor used to build the MSB and LSB capacitors. This may cause poor matching with other capacitors. To resolve this issue [15] proposed to use a unit bridge capacitor and removed the first capacitor from the LSB-side array. However, this solution suffers from a 1 LSB gain error. For metal-insulator-metal (MIM) capacitors, it has been shown that the implementation of non-unit sized capacitors can be handled by maintaining their perimeter to area ratio [16].

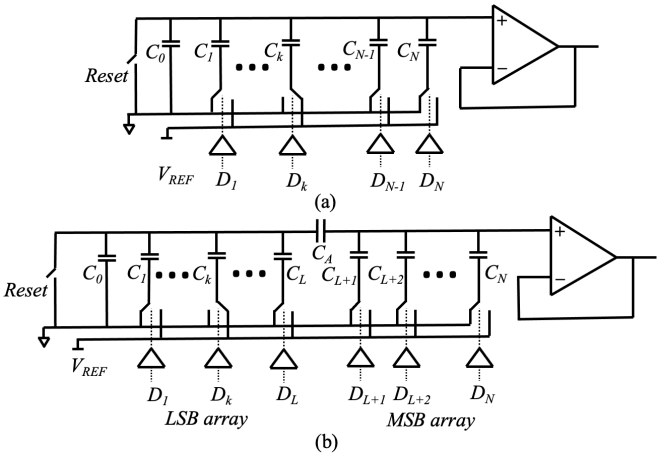


Fig. 1: Schematic circuit of a charge-scaling (a) binary, and (b) split DAC with the k^{th} bit set to 1 and all other bits set to 0.

This work focuses on the layout in FinFET nodes, which

This work is supported in part by the Semiconductor Research Corporation (SRC) and by the DARPA IDEA program (as part of the ALIGN project under SPAWAR Contract N660011824048).

bring forth several new challenges as compared to older bulk technologies. Although bulk technologies are generally better for analog designs than FinFET nodes, the need for integrating analog and digital circuitry together in the same chip forces designers to build analog designs in FinFET nodes. These advanced technologies result in several significant challenges for the design of passive capacitor arrays. First, the per-unit wire/via resistances become higher as technology nodes scale down, and are particularly acute in FinFET nodes. Second, lithography rules dictate that wiring in a single layer must be in the same direction, implying that a change in routing direction invokes the significant resistive penalty of going through a via to another layer. Prior techniques that address older bulk technology nodes do not face these restrictions and cannot easily be adapted to FinFET designs. For example, the routing detours and bends in [17], [18] incur high resistance penalties in FinFET nodes. Third, analog design in FinFET nodes favors Metal-oxide-metal (MOM) capacitors with high capacitance density and low-resistance connections (going through few vias) to the device layer, but prior efforts primarily target MIM capacitors in the upper layers of the interconnect stack. The high cost of going through multiple vias to reach the MIM capacitor layer can cause significant RC-induced degradation on the 3dB frequency at which the capacitor array can operate. Only a few efforts [10], [11] consider CC layout for MOM capacitors, and none address issues specific to FinFET nodes.

We summarize the chief results and novel aspects of this paper here. First, in contrast to prior iterative/stochastic layout generators for CC capacitor arrays, we build a fast, *constructive* approach for generating the array layout. Our constructive approaches present a new *spiral* CC layout that has very few bends, and thus low via counts, resulting in high 3dB frequencies. We also apply our approach to existing *chessboard* methods, and we devise a new family of *block chessboard* placements that can achieve a balance between the dispersion advantages of chessboard layouts and the low via counts of spiral layouts. Second, we address the layout of CC capacitor arrays for both binary-weighted DACs and split DACs; to the best of our knowledge, split DACs have not been addressed in prior work. For the split DAC, we propose a method to determine the size of the non-unit capacitor which is used as the attenuation capacitor. Third, unlike prior work that has focused mainly on bulk technologies, our method specifically targets FinFET technologies, addressing the deleterious impact of high via resistances and per-unit wire resistances. Our capacitor array layouts are specifically designed to have few vias, particularly for the most significant bits, which determine the 3dB frequency. Moreover, we reduce wire resistances on these connections through the use of parallel wires, which effectively result in the use of wider wires while obeying the FinFET technology requirements of allowing only quantized wire widths. Fourth, we explore the impact of using spiral, chessboard, and block chessboard methods on a variety of DACs and demonstrate the impact of parasitics on 3dB frequency: in particular, we show that chessboard layouts are very constraining in FinFET technologies. Our spiral placement approach provides large improvements in 3dB frequency over prior work, with some penalty in the INL/DNL, and our block chessboard approaches explore intermediate design points by trading off 3dB frequency with INL/DNL.

II. Background

A. Binary-weighted DACs

A digital input code is applied to the binary-weighted charge-scaling DAC shown in Fig. 1(a)‘, which is typically used as part of charge-redistribution ADC. This is used to sample a fixed reference voltage, and the common terminal of the capacitor array is connected to a unity-gain amplifier in Fig. 1(a) to produce the analog output voltage; however, this requires a rail-to-rail input and output buffer. If the voltages are not rail-to-rail, as may happen in a stand-alone DAC, it is possible to compensate for the difference by appropriately amplifying the signal by using a switched-capacitor amplifier with programmable gain [16]. Since our focus is on the binary-weighted capacitor array, for convenience we consider the case when the capacitors are connected to a unity-gain amplifier. The black capacitors correspond to the intentionally inserted capacitances, while those shown in blue, purple, orange, and grey correspond to parasitic capacitances.

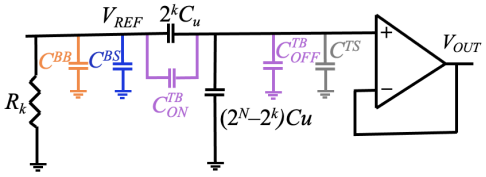


Fig. 2: Equivalent circuit of a charge-scaling DAC with the k^{th} bit set to 1 and all other bits set to 0.

For the ideal case, we assume that these parasitics are zero and consider only the capacitors shown in black. Let $D_k(i)$ be the k^{th} bit of code i; let C_T and $C_{ON}(i)$ [$C_{OFF}(i)$] be the total capacitance and sum of capacitors whose bottom plates are connected to V_{REF} [ground] and R_k is the parasitic resistance associated with the capacitor C_k. In a binary-weighted DAC, $C_k = n_k C_u$, where $n_k = 1$ for $k = 0$; $n_k = 2^{k-1}$ for $k \geq 1$. Noting that $C_0 = C_u$ is always grounded,

$$C_T(i) = C_u \cdot \left(1 + 1 + 2 + \cdots + 2^{N-1}\right) = 2^N C_u \tag{2}$$

$$C_{ON}(i) = \sum_{k=1}^{N} D_k(i) 2^{k-1} C_u \; ; \; C_{OFF}(i) = \sum_{k=1}^{N} \overline{D}_k(i) 2^{k-1} C_u$$

For perfect capacitor matching and an ideal opamp the output voltage can be represented as,

$$V_{OUT}^{ideal}(i) = V_{REF} \cdot \frac{C_{ON}(i)}{C_T} = V_{REF} \cdot \sum_{k=1}^{N} D_k(i) 2^{(k-N-1)} \tag{3}$$

Different routing parasitics are seen for capacitor C_i in the CC array [19] as shown in Fig. 2: (1) top-plate to ground parasitic capacitance, C_i^{TS}, (2) top-plate-to-bottom-plate parasitic capacitance, C_i^{TB}, (3) the bottom plate capacitance C_i^{BS} to ground, and (4) bottom [top] plate to bottom [top] plate C_{ij}^{BB} [C_{ij}^{TT}]. If the bottom plates [top plates] of two different capacitors are both at V_{DD} or both at ground, C_{ij}^{BB} [C_{ij}^{TT}] will be effectively zero, since it depends on the excitation.

However, for the binary-weighted DAC all the top plates of the capacitors have the same potential.

Depending on whether the bottom plate is switched to V_{DD} or ground, the top-plate-to-bottom-plate parasitic capacitance C_i^{TB} accumulates to C_{ON}^{TB} or C_{OFF}^{TB}, respectively. The parasitics associated with C^{TS} and C^{TB} can alter V_{OUT}^{ideal}, as shown analytically in Section III-A. Increasing the minimum unit capacitance value C_u can reduce these effects. However, it would increase the power consumption. Moreover, with the increase of C_u, the layout area also increases since in N-bit binary-weighted DAC most of the area is occupied by the 2^N unit capacitors. The parasitic capacitance associated with the bottom-plates, does not affect DAC linearity, but affects the load for V_{REF}, and impact the power and 3dB frequency. The impact of routing induced parasitics on 3dB frequency is explained in Section III-C.

B. Split DACs

The split DAC structure, shown in Fig. 1(b), contains two sets of switched capacitances, corresponding to the LSB array and the MSB array. The top plates of the capacitors in each array are connected together at two different electrical nodes that are separated by an attenuation capacitor, C_A. The LSB and MSB arrays each consist of binary-weighted capacitors. In the LSB array, like the binary-weighted DAC, we have $C_k = n_k C_u$, where $n_k = 1$ for $k = 0$; $n_k = 2^{k-1}$ for $k \geq 1$, and with C_0 kept permanently grounded. In the MSB array, $C_k = n_k C_u$, but here, $n_k = 2^{k-1-L}$ – i.e., successive capacitors in both the LSB and MSB array are double the value of their predecessor, but the first capacitance in the MSB array is set to C_u. Hence, the largest capacitance is well below that in the binary-weighted DAC.

For a binary code corresponding to the integer input i, let $C_{ON}^{LSB}(i)$ and $C_{OFF}^{LSB}(i)$ be the sum of capacitances whose bottom plates are connected to V_{REF} and ground, respectively and R_{ON}^{LSB} [R_{ON}^{MSB}] is the parasitic resistance associated with the capacitor C_{ON}^{LSB}[C_{ON}^{MSB}], from the LSB capacitive array, and let $C_{ON}^{MSB}(i)$ and $C_{OFF}^{MSB}(i)$ denote the corresponding values for the MSB array. Quantitatively,

$$\begin{aligned} C_{ON}^{LSB}(i) &= \textstyle\sum_{k=1}^{L} D_k(i) 2^{k-1} C_u \\ C_{OFF}^{LSB}(i) &= \textstyle\sum_{k=1}^{L} \overline{D}_k(i) 2^{k-1} C_u + C_u \\ C_{ON}^{MSB}(i) &= \textstyle\sum_{k=L+1}^{N} D_k(i) 2^{N-k} C_u \\ C_{OFF}^{MSB}(i) &= \textstyle\sum_{k=L+1}^{N} \overline{D}_k(i) 2^{N-k} C_u \end{aligned}$$

The total LSB [MSB] array capacitance, C_T^{LSB} [C_T^{MSB}] is

$$C_T^{LSB} = C_{ON}^{LSB}(i) + C_{OFF}^{LSB}(i) \tag{4}$$

$$C_T^{MSB} = C_{ON}^{MSB}(i) + C_{OFF}^{MSB}(i) \tag{5}$$

For the scenario where only the capacitors from LSB array are connected to V_{REF} or ground corresponding to the code i, and all capacitors from the other array are connected to ground, as shown in Fig. 3(a), we denote the output voltage by $V_{OUT,ideal}^{LSB}$. For the similar case where the LSB capacitors are all grounded and the MSB array implements the corresponding digits of code i, shown in Fig. 3(b), we denote the response at the output as $V_{OUT,ideal}^{MSB}$. The figures show all capacitors

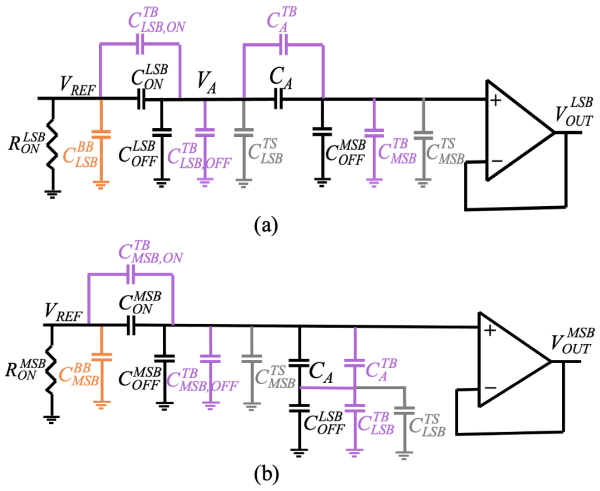


Fig. 3: Equivalent circuit of a split DAC with excitations applied to (a) the LSB array, and MSB excitations set to 0, (b) the MSB array, and the LSB excitations set to 0. The two cases are superposed to determine V_{OUT}^{LSB}.

in the system, including the parasitic capacitances shown in purple and grey.

The counterpart of Equation (3) is obtained by superposition, by adding the output voltages for the two scenarios above. From Fig. 3(a), denoting V_A as the voltage on the left terminal of C_A,

$$\begin{aligned} V_{OUT,ideal}^{LSB}(i) &= V_A \cdot \frac{C_A}{C_A + C_T^{MSB}} \\ &= V_{REF} \cdot \frac{C_{ON}^{LSB}(i)}{C_T^{LSB} + C_A^{MSB}} \cdot \frac{C_A}{C_A + C_T^{MSB}} \end{aligned} \tag{6}$$

where C_A^{MSB} is the equivalent capacitance of series connection between attenuation capacitor C_A and the MSB capacitors whose bottom plates are connected to the ground, i.e.,

$$C_A^{MSB} = \frac{C_A C_T^{MSB}}{C_A + C_T^{MSB}} \tag{7}$$

For the second component of superposition in Fig. 3(b),

$$V_{OUT,ideal}^{MSB}(i) = V_{REF} \cdot \frac{C_{ON}^{MSB}(i)}{C_T^{MSB} + C_A^{LSB}} \tag{8}$$

where C_A^{LSB} is the equivalent capacitance of series connection between attenuation capacitor C_A and the LSB capacitors whose bottom plates are connected to the ground, i.e.,

$$C_A^{LSB} = \frac{C_A C_T^{LSB}}{C_A + C_T^{LSB}} \tag{9}$$

For an applied code i, the expression for V_{OUT}^{ideal} is the superposition of the cases described in Eqs. (6) and (8), i.e.,

$$V_{OUT}^{ideal}(i) = V_{OUT,ideal}^{MSB}(i) + V_{OUT,ideal}^{LSB}(i) \tag{10}$$

Similar to the case of binary-weighted DACs, the parasitics C^{TS} and C^{TB} associated with LSB [MSB] array can alter $V_{OUT,ideal}^{LSB}$ [$V_{OUT,ideal}^{MSB}$], and therefore the V_{OUT}^{ideal} of the circuit. We will further elaborate on this in Section III-B.

Here too, increasing C_u can reduce these effects, at the cost of increased power. The impact of various types of parasitic capacitances on 3dB switching frequency is explained in Section III-C.

C. Modeling variations in a DAC capacitor array

To reduce systematic mismatch, the unit capacitors C_u, which are built as identical-sized capacitor cells (called *unit cells*) that are placed in a gridded *common-centroid matrix* [3], [4]. Unit capacitors may be built using a range of technologies. A MIM capacitor consists of two metal layers and a dielectric layer. MOM capacitor structures consist of a set of metal wires that can be connected in various configurations, e.g., alternate polarity MOMs (APMOMs), woven, parallel stacked, or vertical structures [10]. This work focuses on MOM structures in FinFET nodes, although many ideas can be applied to MIM structures.

Taking the origin to be at the center of an $r \times s$ CC array, we can obtain the location (x_k, y_k) of a unit capacitor in row r_k and column s_k by following [20] from its width and height, W and H, and the vertical and horizontal spacing, S_v and S_h between unit capacitors as shown below:

$$x_k = (s_k - (s+1)/2)\,(W + S_h)$$
$$y_k = ((r+1)/2 - r_k)\,(H + S_v)$$

1) Modeling systematic variation due to a linear gradient: Systematic variations are typically modeled as a gradient across the layout area [5]. At the center of the CC array, if t_0 denotes the spacing between MOM cap wires, and C_u is the unit capacitance, then the oxide thickness at (x_j, y_j) is $t_j = t_0 + \gamma(x_j \cos\theta + y_j \sin\theta)$. Here, γ and θ ($0 \le \theta \le 180°$) are the linear oxide gradient magnitude and angle at the origin, respectively, where γ and θ randomly differ from array to array (as they are subject to manufacturing variability), but are constant for a given array. A unit capacitor at (x_j, y_j) thus has value $C_u(t_0/t_j)$. Each capacitor value C_k is then shifted due to systematic variations to

$$C_k^* = \textstyle\sum_j C_u \times (t_0/t_j) \tag{11}$$

If an ideal capacitor ratio, $C_0 : \cdots : C_N$, shifts to $C_0^* : \cdots : C_N^*$ due to a process gradient, then the systematic ratio mismatch is:

$$M_{sys} = \max_{p,q \in \{0,\cdots,N\}, p \neq q} \left| \frac{(C_p^*/C_q^*) - (C_p/C_q)}{(C_p/C_q)} \right| \tag{12}$$

2) Modeling random variations: A unit capacitor has zero-mean random variations with variance $\sigma_u^2 = A_f^2/(WH)$ [21], where A_f is similar to a Pelgrom mismatch coefficient [22], W and H are the width and height of the unit capacitor. The correlation coefficient for two unit capacitors A at (x_1, y_1) and B at (x_2, y_2) in the $(r \times s)$ CC matrix is [6], [23]:

$$\rho_{AB} = (\rho_u)^{D(A,B)} \tag{13}$$

$$\text{where } D(A,B) = \left(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}\right)/L_c \tag{14}$$

Here $0 < \rho_u < 1$ and L_c are process-specific parameters.

If $C_p = pC_u$ [$C_q = qC_u$], with p [q] unit capacitors, correlation coefficient $\rho_{pq} = Cov(p,q)/(\sigma_p \sigma_q)$ where

$$\sigma_p^2 = \sigma_u^2\,(p + 2S_p);\ \sigma_q^2 = \sigma_u^2\,(q + 2S_q) \tag{15}$$

$$Cov(p,q) = \sigma_u^2 S_{pq} \tag{16}$$

$$S_p = \sum_{a=1}^{p-1} \sum_{b=a+1}^{p} \rho_{ab}\ ;\ S_q = \sum_{a=1}^{q-1} \sum_{b=a+1}^{q} \rho_{ab}\ ;\ S_{pq} = \sum_{a=1}^{p} \sum_{b=1}^{q} \rho_{ab}$$

III. Circuit-level metrics

In this section, we examine the impact of nonidealities due to parasitic capacitance mismatch on performance metrics for the binary-weighted and split DAC topologies.

The *differential nonlinearity (DNL)* is the difference between the ideal and nonideal step for input code $1 \le i \le 2^N - 1$. The DNL at input code i is given by:

$$DNL(i) = \left(V_{OUT}(i) - V_{OUT}(i-1) - V_{LSB}\right)/V_{LSB} \tag{17}$$

where $V_{LSB} = V_{REF}/2^N$. The *integral nonlinearity (INL)* is the deviation between the ideal output and the actual output in the presence of mismatch. By definition, $INL(0) = 0$, and for $1 \le i \le 2^N - 1$,

$$INL(i) = \left(V_{OUT}(i) - V_{OUT}^{ideal}(i)\right)/V_{LSB} \tag{18}$$

A. Errors in linearity metrics for a binary-weighted DAC

For a binary-weighted DAC, under nonidealities the output voltage can be represented as [8]:

$$V_{OUT}(i) = V_{REF} \cdot \frac{C_{ON}(i) + \Delta C_{ON}(i)}{C_T + \Delta C_T} \tag{19}$$

where $\Delta C_{ON}(i)$ [ΔC_T] is the shift in $C_{ON}(i)$ [C_T]. The shifts in $C_{ON}(i)$ and C_T in (2) due to nonidealities are:

$$\Delta C_{ON}(i) = \textstyle\sum_{k=1}^{N} D_k(i) \Delta C_k + C_{ON}^{TB} \tag{20}$$

$$\Delta C_T = \textstyle\sum_{k=0}^{N} \Delta C_k + C_{ON}^{TB} + C_{OFF}^{TB} + C^{TS} \tag{21}$$

where C_{ON}^{TB}, C_{OFF}^{TB}, and C^{TS} represent the parasitics illustrated in Fig. 2; ΔC_k is the sum of the statistical variations, ΔC_k^{sta}, and systematic variations. The systematic variation of the k^{th} capacitor ΔC_k^{sys} is given by:

$$\Delta C_k^{sys} = (C_k^* - n_k C_u) \tag{22}$$

We followed the statistical variation model from the preliminary conference version of this paper [19] using a 3σ model. The variances of $\Delta C_{ON}(i)$ and ΔC_T are:

$$\sigma^2_{\Delta C_{ON}(i)} = \textstyle\sum_{j=1}^{N} \sum_{k=1}^{N} D_j(i) D_k(i) Cov(j,k) \tag{23}$$

$$\sigma^2_{\Delta C_T} = \textstyle\sum_{j=0}^{N} \sum_{k=0}^{N} Cov(j,k) \tag{24}$$

where $Cov(i,j)$ is given by (16). Therefore,

$$\Delta C_{ON}(i) = \textstyle\sum_{k=1}^{N} D_k(i) \Delta C_k + \left(3\sigma_{\Delta C_{ON}(i)} + C_{ON}^{TB}\right)$$

$$\Delta C_T = \textstyle\sum_{k=0}^{N} \Delta C_k + \left(3\sigma_{\Delta C_T} + C_{ON}^{TB} + C_{OFF}^{TB} + C^{TS}\right)$$

B. Errors in linearity metrics for a split DAC

To model the split DAC under nonidealities, as before, we superpose the effect of the LSB and MSB arrays on the output voltage. For an applied code i, depending on whether the k^{th} bit, D_k, is 0 or 1, capacitors in each array may be connected to V_{REF} or ground, respectively; those in the other are grounded for the purposes of superposition calculations.

For a split DAC under nonideal conditions, we update $C^X_{ON}(i)$, C^X_T and C^X_A, where $X \in \{MSB, LSB\}$ in (6) and (8). We will denote the shifts in $C^X_{ON}(i)$, C^X_T, and C^X_A as $\Delta C^X_{ON}(i)$, ΔC^X_T, and $C^X_{A,nonideal}$, respectively. We calculate the output voltage V^{LSB}_{out} under nonidealities for the excitation to the LSB array (Fig. 3(a)) by first calculating voltage V_A as

$$V_A(i) = V_{REF} \cdot \frac{C^{LSB}_{ON}(i) + \Delta C^{LSB}_{ON}(i)}{C^{LSB}_T + \Delta C^{LSB}_T + C^{MSB}_{A,nonideal}} \tag{25}$$

The output voltage as in Eq. (6) under nonidealities is given by:

$$V^{LSB}_{OUT}(i) = V_A(i) \cdot \frac{C_A + C^{TB}_A}{C_A + C^{TB}_A + C^{MSB}_T + \Delta C^{MSB}_T} \tag{26}$$

For the MSB array (Fig. 3(b)), Eq. (8) is modified to:

$$V^{MSB}_{OUT}(i) = V_{REF} \cdot \frac{C^{MSB}_{ON}(i) + \Delta C^{MSB}_{ON}(i)}{C^{MSB}_T + \Delta C^{MSB}_T + C^{LSB}_{A,nonideal}} \tag{27}$$

The nonideal counterparts of Eq. (7) and Eq. (9) are represented as $C^X_{A,nonideal}$, where $X \in \{LSB, MSB\}$. These are based on the nonideal version of each of the corresponding capacitors, and are given by the equivalent series capacitance of $(C_A + C^{TB}_A)$ with $(C^X_T + \Delta C^X_T)$.

The ideal output voltage, $V_{OUT}(i)$ in Eq. (10) can be modified for the nonideal case superposing Eqs. (26) and (27). To evaluate these two equations, we require the shifted values of the LSB and MSB capacitors in Figs. 3(a) and (b), respectively, which are given by:

$$\begin{aligned}
\Delta C^{LSB}_{ON}(i) &= \textstyle\sum_{k=1}^{L} D_k(i)\Delta C_k + C^{TB}_{LSB,ON} \\
\Delta C^{MSB}_{ON}(i) &= \textstyle\sum_{k=L+1}^{N} D_k(i)\Delta C_k + C^{TB}_{MSB,ON} \\
\Delta C^{LSB}_T &= \textstyle\sum_{k=0}^{L} \Delta C_k + C^{TB}_{LSB,ON} + C^{TB}_{LSB,OFF} + C^{LSB}_{TS} \\
\Delta C^{MSB}_T &= \textstyle\sum_{k=L+1}^{N} \Delta C_k + C^{TB}_{MSB,ON} + C^{TB}_{MSB,OFF} \\
&\quad + C^{MSB}_{TS}
\end{aligned}$$

As before, ΔC_k represents the shift due to process variations and $D_0(i) = 0$ for all i as C_0 is connected to ground.

The extent of statistical variations is given by the 3σ points of the statistical summations of C_k terms for the variances of $\Delta C^{LSB}_{ON}(i)$ [$\Delta C^{MSB}_{ON}(i)$] and ΔC^{LSB}_T [ΔC^{MSB}_T] for LSB [MSB] bits are shown below:

$$\sigma^2_{\Delta C^{LSB}_{ON}(i)} = \textstyle\sum_{j=1}^{L}\sum_{k=1}^{L} D_j(i)D_k(i)Cov(j,k) \tag{28}$$

$$\sigma^2_{\Delta C^{MSB}_{ON}(i)} = \textstyle\sum_{j=L+1}^{N}\sum_{k=L+1}^{N} D_j(i)D_k(i)Cov(j,k) \tag{29}$$

$$\sigma^2_{\Delta C^{LSB}_T} = \textstyle\sum_{j=0}^{L}\sum_{k=0}^{L} Cov(j,k) \tag{30}$$

$$\sigma^2_{\Delta C^{MSB}_T} = \textstyle\sum_{j=L+1}^{N}\sum_{k=L+1}^{N} Cov(j,k) \tag{31}$$

Using the 3σ model, the shifts in $C^{LSB}_{ON}(i)$ [$C^{MSB}_{ON}(i)$] and C^{LSB}_T [C^{MSB}_T] can be written as:

$$\begin{aligned}
\Delta C^{LSB}_{ON}(i) &= \textstyle\sum_{k=1}^{L} D_k(i)\Delta C_k + 3\sigma_{\Delta C^{LSB}_{ON}(i)} + C^{TB}_{LSB,ON} \\
\Delta C^{MSB}_{ON}(i) &= \textstyle\sum_{k=L+1}^{N} D_k(i)\Delta C_k + 3\sigma_{\Delta C^{MSB}_{ON}(i)} \\
&\quad + C^{TB}_{MSB,ON} \\
\Delta C^{LSB}_T &= \textstyle\sum_{k=0}^{L} \Delta C_k + C^{TB}_{LSB,ON} + C^{TB}_{LSB,OFF} \\
&\quad + C^{LSB}_{TS} + 3\sigma_{\Delta C^{LSB}_T} \\
\Delta C^{MSB}_T &= \textstyle\sum_{k=L+1}^{N} \Delta C_k + C^{TB}_{MSB,ON} + C^{TB}_{MSB,OFF} \\
&\quad + C^{MSB}_{TS} + 3\sigma_{\Delta C^{MSB}_T}
\end{aligned}$$

C. Capacitor 3dB frequency

The switching response of a capacitor array is encapsulated in the 3dB frequency metric, which defines the maximum speed at which switches can close the bottom plates of the unit capacitors of the DAC. For high-resolution DACs, the routing-induced parasitics may substantially affect the 3dB frequency. In FinFET technologies in particular, high wire and via parasitics can result in high RC time constants, which degrade the 3dB frequency.

In our preliminary version of this paper [19], we had shown that this effect is particularly acute when CC layout techniques attempt to increase capacitor dispersion, which requires the connecting wires to use a large number of vias, resulting in high parasitic resistance. We had also shown a model based on [24] to obtain the critical frequency for the binary-weighted DAC. Since each cycle goes through a charge-discharge phase, the 3dB frequency, f_{3dB}, for a full cycle of a capacitor is represented as:

$$f_{3dB} = 1/(2(N+2)ln(2)\tau) \tag{32}$$

For each capacitor C_i, τ is measured by computing the Elmore delay of the RC mesh to the unit capacitors of C_i. Over all bits, the charging network with the maximum Elmore delay limits the frequency and is used as τ in (32). For a binary-weighted DAC, the Elmore delay for a mesh can be computed using standard techniques [25].

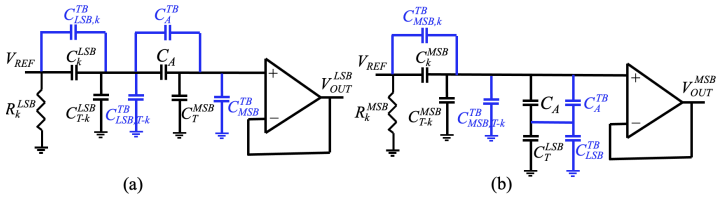


Fig. 4: Equivalent circuit of a split DAC with the k^{th} bit from (a) LSB side, (b) MSB side set to 1 and all other bits set to 0.

To determine the RC product associated with the switching of a specific bit k for a split DAC, we consider the case when bit k is set to 1 and all other bits are 0. The equivalent circuits for these two cases, when the k^{th} capacitor (C^X_k) is connected to V_{REF} and other $N-k$ capacitors (C^X_{T-k}) are connected to the ground in the LSB and MSB arrays, where

$X \in \{MSB, LSB\}$, are displayed in Figs. 4(a) and (b).[1] The equivalent capacitance of Fig. 4(a), excluding the k^{th} capacitor from the LSB side array which is connected to V_{REF} with all other capacitors grounded, can be written as

$$C_{A,eq}^{LSB} = C_{T-k}^{LSB} + C_{LSB,T-k}^{TB} + \frac{(C_A + C_A^{TB})(C_T^{MSB} + C_{MSB}^{TB})}{C_A + C_A^{TB} + C_T^{MSB} + C_{MSB}^{TB}} \tag{33}$$

The total equivalent capacitance for excitation to the LSB side array (Fig. 4(a)) can be represented as

$$C_{eq}^{LSB} = \frac{(C_k^{LSB} + C_{LSB,k}^{TB})C_{A,eq}^{LSB}}{C_k^{LSB} + C_{LSB,k}^{TB} + C_{A,eq}^{LSB}} \tag{34}$$

Similarly, the equivalent capacitance for Fig. 4(b) at the right side of the k^{th} bit from the MSB array which is set to 1, with all other capacitors connected to the ground is given by

$$C_{A,eq}^{MSB} = C_{T-k}^{MSB} + C_{MSB,T-k}^{TB} + \frac{(C_A + C_A^{TB})(C_T^{LSB} + C_{LSB}^{TB})}{C_A + C_A^{TB} + C_T^{LSB} + C_{LSB}^{TB}} \tag{35}$$

The total equivalent capacitance for Fig. 4(b) is shown below

$$C_{eq}^{MSB} = \frac{(C_k^{MSB} + C_{MSB,k}^{TB})C_{A,eq}^{MSB}}{C_k^{MSB} + C_{MSB,k}^{TB} + C_{A,eq}^{MSB}} \tag{36}$$

To calculate the 3dB frequency, the resistance for the RC product corresponds to the parasitic resistance R_k^{LSB} [R_k^{MSB}] which is associated with the capacitor from the LSB [MSB] array whose bottom plate is connected to V_{REF} and capacitance corresponds to the equivalent capacitance from the equivalent circuits (Fig. 4). The frequency calculation is performed by using (32) for each capacitor from the capacitor array, and the minimum of these frequencies is considered as the 3dB frequency of the circuit.

D. Non-unit capacitor aspect ratio

In the split DAC, the top plates from the LSB and MSB array are connected via an attenuation capacitor. In a conventional split DAC, the attenuation capacitor is a non-unit capacitor (i.e., not an integer multiple of the unit capacitor C_u used to build the binary-weighted (sub)array(s)). The overall dimensions are usually set to between one and two times the unit-sized capacitor (C_u), and the capacitor has a rectangular bounding box [16]. To obtain the correct non-unit capacitive ratio, we will show that the relative error due to process variation for the MOM capacitors is similar to the MIM capacitors, where the relative error is equal to the ideal perimeter-to-area ratio.

Let N be the number of coupled segments as shown in Fig. 5. The capacitance can be represented as

$$C = N\epsilon\left(\frac{lt}{d}\right) \tag{37}$$

[1]Note that this differs from the scenario in Fig. 3, where multiple bits in the MSB/LSB array may be switched on; here, to calculate the 3dB frequency for bit k, only bit k is activated and all others are set to 0.

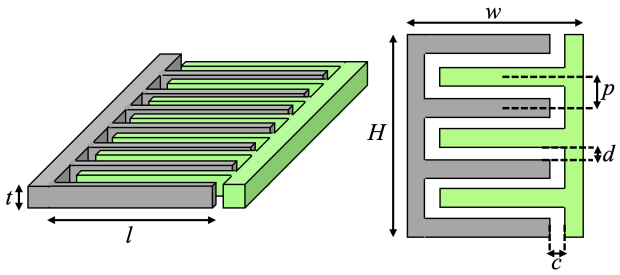


Fig. 5: Cross-section of a MOM capacitor in one metal layer.

where ϵ, l, t and d are the dielectric constant, coupling length, thickness, and distance between the fingers, respectively. Here,

$$l = w - c\ ;\ H = Np \tag{38}$$

where c is a constant, corresponding to the spacing between the top and bottom plates along the direction of l as shown in Fig. 5, p is the pitch between two adjacent wires, and w and H are the width and height of the MOM capacitor. Since $N = H/p$,

$$C = \left(\frac{\epsilon t}{dp}\right) lH \tag{39}$$

Under process variations, the total capacitance is represented as

$$C + \Delta C = \left(\frac{\epsilon t}{dp}\right)(l + \Delta l)(H + \Delta H) \tag{40}$$

$$\text{i.e., } \Delta C = \left(\frac{\epsilon t}{dp}\right)(l\Delta H + H\Delta l) \tag{41}$$

Neglecting $\Delta l \cdot \Delta H$, the relative error in the capacitance is

$$\frac{\Delta C}{C} = \frac{l\Delta H + H\Delta l}{lH} = \frac{\Delta H}{H} + \frac{\Delta l}{l} \tag{42}$$

Using notation consistent with the analysis in [16] for MIM capacitors, we assume that under process variations, the coupling length, width, and height of the MOM capacitor change by 2δ, i.e.,

$$l \to l + 2\delta\ ;\ w \to w + 2\delta\ ;\ H \to H + 2\delta \tag{43}$$

The relative error in the capacitor is therefore given by

$$\frac{\Delta C}{C} = \frac{2\delta}{H} + \frac{2\delta}{l} \tag{44}$$

$$\frac{\Delta C}{C} = 2\delta\left(\frac{H + l}{Hl}\right) = 2\delta\left(\frac{H + w - c}{H(w - c)}\right) \tag{45}$$

Thus the relative capacitor error for a MOM capacitor due to process variations has a similar form as that for the MIM capacitor [16], where the relative error is proportional to the the ideal perimeter to ideal area ratio: the difference here is in the use of $w - c$ instead of w in the MIM case. for a given capacitance value C, we set the relative error to be the same as that for a unit capacitor on the left-hand side of (45). Together with (39), this yields two equations in two variables that are solved to obtain H and l.

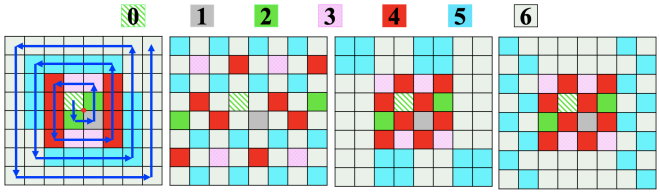
IV. Common-Centroid Placement and Routing

Our constructive routing-friendly common-centroid placement flow optimizes mismatch, interconnect wirelength, parasitic RCs, and 3dB frequency for both the binary-weighted and split DACs; next, a routing step optimizes DAC performance. We first present the placement and routing algorithms for a binary-weighted DAC in Sections IV-A and IV-B, respectively. Next, we shown how these algorithms are modified to generate the layout for a split DAC in Section IV-C. In our work [19] we presented the formulation for obtaining array size for N-bit binary-weighted DAC.

A. CC placement for a binary-weighted DAC

1) Placement tradeoffs between wire resistance and dispersion: Good matching under random variations is ensured through *dispersion*, which reflects the spread of the unit capacitances of C_0 through C_N in the CC array. An additional major consideration is to build routing-friendly placements that optimize interconnect parasitic effects. Previous efforts have not addressed the specific needs of FinFET technologies, with high wire resistance and higher via resistances. As FinFET technologies used reserved-direction routing, especially in the lower metal layers that are used for MOM capacitors, every bend in a wire incurs a via resistance cost due to a layer change.

Reducing via count is critical for reducing interconnect resistance and improving 3dB frequency. An extension of high-dispersion chessboard placement [8] matches capacitive routing parasitics but neglects resistance: results show high via counts. We consider a range of new constructive placement solutions – **spiral** placement and **block chessboard (BC)** methods – to trade off interconnect parasitics with dispersion for the binary-weighted DAC.



(a) Spiral (b) Chessboard (c) BC (coarser) (d) BC (finer)

Fig. 6: An example illustrating our CC placement algorithm.

Spiral placement for optimized interconnect parasitics This placement algorithm reduces the number of bends in the connections and is illustrated for a 6-bit DAC in Fig. 6(a). Since the number of unit capacitors in C_0 and C_1 is 1, an odd number, it is not possible to achieve a common-centroid placement. Instead, we place these as close to the common-centroid as possible to limit the impact of process variations. Here, we place C_0 and C_1 diagonally opposite each other near the center. Next, we place all the capacitors of C_2, then C_3, and so on, in a spiral sequence from the center.

Whenever we place a unit capacitor at a location along a spiral, we also place another unit capacitor at its reflection to maintain the CC property. Considering the CC point as the origin (the red dot in the figure), if we place a unit capacitor in a square (d_1, d_2), it will be accompanied by another unit capacitor at location $(-d_1, -d_2)$. For example, when the spiral places a unit capacitor of C_2 at $(-1, -1)$ in the figure, we place another unit capacitor of C_2 at $(+1, +1)$. We place the unit capacitors of C_3 at the first empty location along the spiral, first at $(1, 2)$ and its reflection at $(-1, -2)$, and so on.

Beyond C_2, this technique naturally aligns numerous unit cells of a capacitor to lie in the same row or column, the method reduces the number of vias (corresponding to wire "bends") required to connect them. This approach maintains adequate dispersion while using a number of bends (corresponding to vias) for the routing connections. Although our proposed placement algorithm is simple, it is different from previous methods: the nearest similar methods are [5], with a mix of rectangles and circles for placement, and [26] with interleaved rows, but it does not achieve good dispersion.

Chessboard placement for optimized dispersion [8] At the other extreme, [8] optimizes dispersion by interspersing unit capacitors in a chessboard pattern, as illustrated for a 6-bit DAC in Fig. 6(b). For a 6-bit DAC, the 32 unit capacitors of C_6 are first placed in an 8×8 array on the black squares of a "chessboard"; then the 16 unit capacitors of C_5 are placed; and so on. However, the routing resistance costs here are large.

Block chessboard (BC) approaches A block chessboard approach attempts to find the best of both worlds, by achieving the dispersion of the chessboard approach and the lower routing costs of the spiral approach. Examples of this approach for a 6-bit DAC are shown in Figs. 6(c) and 6(d). The inner core of this structure is a conventional chessboard layout for the capacitors with a smaller number of unit cells (here, C_0 through C_4): this provides good dispersion, and while it has a high number of bends/vias, its wire RCs are typically smaller than those of the larger capacitors C_5 and C_6, and do not constrain the 3dB frequency, which is determined by the worst-case time constant. The outer corridor here has a width of 2 cells. Since $n_6 : n_5 = 2 : 1$, we first lay out half the cells of C_6 in clusters and then perform chessboard placement, alternating the remaining cells of C_6 with C_5. Two layouts are shown for different granularities in the outer corridor.

Other block chessboard structures may be built with the inner full-chessboard core of $C_0 - C_k$, and an outer block structure for $C_{k+1} - C_N$. MSB capacitors do not greatly affect the DAC accuracy since their variation is averaged over more unit capacitors than LSB capacitors.[2] We design the block chessboard scheme so that MSB capacitors use fewer vias than in the chessboard scheme, resulting in an improved (higher) 3dB frequency. Our block chessboard placement method is outlined in Algorithm 1. The approach consists of three steps: performing the inner complete chessboard (Step 1), creating the outer block chessboard pattern, and completing the block chessboard (Step 2).

For the block chessboard placement, we focus on reducing

[2]In case of uncorrelated random variations, MSB capacitors will show a lower σ/μ variation, and for a sum of n random variables, $\mu = n\mu_u$, $\sigma = \sqrt{n}\sigma_u$, where μ_u and σ_u are the variances of C_u. As a result, in general, we observe that the relative variations in MSB capacitance values is less than for LSB capacitors. Note that the deviation in the capacitor ratio (i.e., the ratio of C_k/C_0) depends on C_0, which shows significant variations.

the number of vias in the highest-order bits, since they have the largest RCs, and to provide good dispersion for the lower-order bits, which have small capacitors that are susceptible to variation effects. For this reason, we perform an inner complete chessboard for $C_0, \cdots, C_k$ and block chessboard layout for $C_{k+1}, \cdots, C_N$. In practice, it is reasonable to assume that k and $N-k$ is even to obtain a placement without dummies.

We first calculate the number of rows, (r_c), and columns, (s_c), for the inner complete chessboard (line 5). Next, we lay out the inner complete chessboard for $C_0 - C_k$ in a manner similar to [8].

In step 2 (line 7), we lay out $(N-k)$ capacitors in the outer corridor of the inner complete chessboard in a block chessboard pattern using the selected block size, b_s. For finer [coarser] granularity, a smaller [larger] block size can be chosen. To create a block chessboard layout, starting from $i = k+1$, we place the blocks in chessboard fashion using the selected block size, b_s, for C_i. First, we determine the number of blocks for the i^{th} capacitor (i.e., we define the range of the outer two loops), which can be placed at the upper half of the placement across the block columns and rows. Here, a block column [row] consists of the number of columns [rows] of size b_s.

The unit capacitors of C_i are placed in lines 17–21. We then reflect the solution simultaneously at its diagonal symmetric location with respect to the center to maintain CC symmetry. Next, we calculate the initial column location for other blocks in the row and place them successively. After finishing the placement of blocks in one block row, we reset the starting column location across each block row (lines 25–29) and increment the row by a block size (line 30) to perform the next set of block chessboard placement. Once all unit capacitors of C_i have been placed, we place the unit capacitors of the next largest capacitor (C_{i+1}) in the rest of the locations of the outer corridor (line 33). To define the next outer corridor for $i = k+3$ if $N-k > 2$, here at line 34 we reset the row and column $(r_c,\ s_c)$ of inner chessboard placement. We increment i and repeat the process until $i = N-1$.

Algorithm 1 Block chessboard placement

1: **Input**: $C = [C_0, \cdots, C_N]$; r, row; s, column; k, capacitor number for inner complete chessboard; $b_s[i]$, a list of block sizes for capacitor C_i.
2: **Output**: Common-centroid placement
3: *//Step 1: Perform inner complete chessboard*
4: *//Calculate row and column for inner complete chessboard for $C_0 - C_k$ capacitors*
5: $r_c \leftarrow \left\lceil \sqrt{\sum_{i=0}^{k} C_i} \right\rceil,\ s_c \leftarrow \left\lceil \left(\sum_{i=0}^{k} C_i\right) / r_c \right\rceil$
6: Fill out the inner black and white squares of a chessboard with row and column size of $(r_c,\ s_c)$ using [8].
7: *//Step 2: Perform outer block chessboard placement*
8: *//Place $(C_{k+1} - C_N)$ capacitors at the outer corridor of inner chessboard*
9: **for** $i = k+1$ to N by 2 **do**
10: *//Calculate row and column for BC placement for $C_i - C_{i+1}$ capacitors*
11: $r_i \leftarrow \left\lceil \sqrt{\sum_{j=0}^{i+1} C_j} \right\rceil,\ s_i \leftarrow \left\lceil \left(\sum_{j=0}^{i+1} C_j\right) / r_i \right\rceil$
12: $(x_s,\ y_s) \leftarrow (s_i/2 - s_c/2,\ r_i/2 - r_c/2)$ *//Initial (row, column) position*
13: *//Create block chessboard pattern using capacitor C_i*
14: **for** $p = 1$ to $\left\lceil \frac{r_i/2}{b_s[i]} \right\rceil$ **do** *//Over number of blocks in a block column*
15: **for** $q = 1$ to $\left\lceil \frac{(s_i - s_c)}{b_s[i]} \right\rceil$ **do** *//Over number of blocks in a block row*
16: *//Create a block of block size b_s for capacitor C_i*
17: **for** $x = x_s$ to $x_s + b_s[i]$ **do** *//Over columns*
18: **for** $y = y_s$ to $y_s + b_s[i]$ **do** *//Over rows*
19: Place a unit of C_i at (x, y) and diagonal symmetric of (x, y) location.
20: **end for**
21: **end for**
22: Calculate starting column location for the next block across a block row.
23: **end for**
24: *//Update initialization of starting column location for the next set of blocks*
25: **if** $p\%2 == 0$ **then**
26: $x_s \leftarrow 0$
27: **else**
28: $x_s \leftarrow b_s[i]$
29: **end if**
30: $y_s \leftarrow y_s + b_s[i]$ *//Update starting row location for the next set of blocks*
31: **end for**
32: Place the units of C_{i+1} at the leftover location, outside of $(r_c,\ s_c)$
33: within $(r_i,\ s_i)$.
34: $r_c \leftarrow r_i,\ s_c \leftarrow s_i$ *//Reset the outer corridor for the next set of capacitors*
35: **end for**

B. Routing for a binary-weighted DAC

There are several routing induced parasitics as discussed in [19], we minimize C_i^{TB} as in [9] with nonoverlapped routing that separates the wires that route the top-plate and bottom-plate. A detailed description on the bottom plate routing is presented in [19].

1) Connected unit capacitor group formation: To connect all bottom plates of unit capacitors of each C_i, we first create a connected capacitor group of neighboring unit capacitors of each C_i. We represent unit capacitors by nodes in graph G, with edges between nodes for these neighboring unit capacitors.

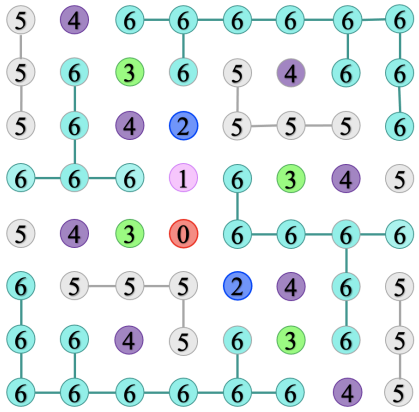


Fig. 7: Graph representation of connected capacitor groups.

We apply a breadth first search (BFS) algorithm on graph G to find its connected components (*connected capacitor groups*). The bottom plates of neighboring unit capacitors in the BFS tree are connected using *branch wires*: each connection is immediately mirrored to the unit capacitor at the diagonally symmetric location in the common-centroid placement, maintaining symmetric routing. The graph representation of the connected unit capacitor groups for a 6-bit CC array for the placement presented in [9] is shown in Fig. 7.

2) Bottom-plate routing: Bottom-plate routing requires separate routes to connect the unit capacitor groups of each C_i. For a DAC, the bottom-plate terminals in the capacitive array are connected to switches and drivers that are clustered together outside the array, since these are noisy digital structures and the terminals must go to the bottom of the array.

We use three types of wires for routing (Fig. 8): *branch wires* are used to connect unit capacitors within capacitor

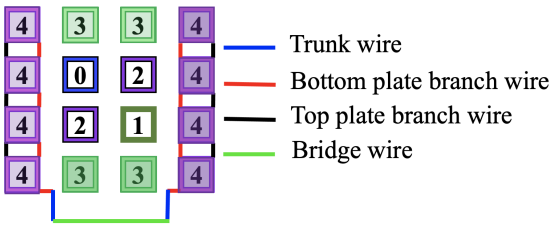


Fig. 8: Routing topology for C_4.

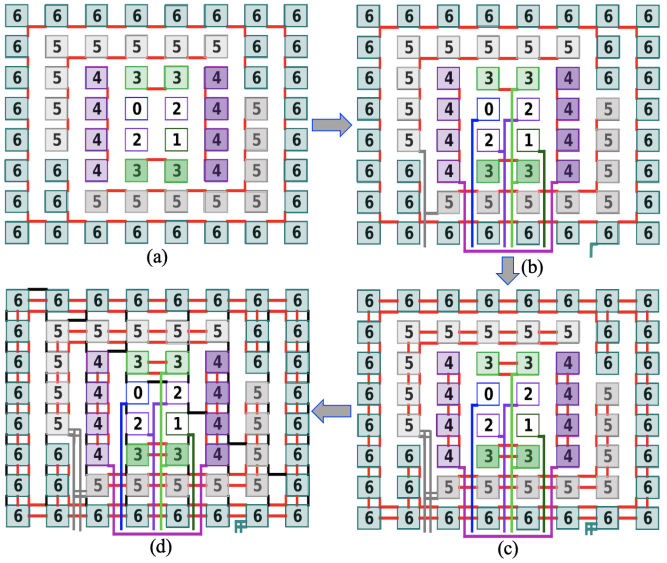


Fig. 9: Routing for a CC placement of a 6-bit DAC, (a) Connected capacitor groups formation [bottom-plate connections are shown in red], (b) Bottom plate routing, (c) Parallel wire routing, (d) Top-plate connections are shown in black.

groups, or unit capacitor groups to trunk wires, *trunk wires* connect disjoint connected capacitor groups along the vertical tracks, and *bridge wires* connect trunk wires at the periphery of the array (the definition of all the wires are represented in Fig. 8). The detailed routing method is outlined in an Algorithm in our previous work [19], and consists of three steps: channel selection, track assignment, and routing.

In FinFET nodes under discrete wire widths we use multiple parallel wires for critical bits to reduce resistance and improve 3dB frequency, we use multiple parallel wires for critical bits. Using multiple parallel wires allow multiple parallel vias with every change in wire direction. Fig. 9(c) shows bottom-plate connections using two parallel wires.

3) Top-plate routing: The objective of top-plate routing is to minimize the top-plate to substrate parasitic capacitance (C^{TS}). We create a graph G such that each vertex $v \in G$ is a unit capacitor for any C_i (since all C_i top plates must be connected). Each unit capacitor is connected to its north, south, east, and west neighbor (if they exist), with an edge weight corresponding to the horizontal or vertical spacing, as applicable. In our case, since the vertical space between unit capacitors is less than the horizontal spacing for channels, the minimum spanning tree (MST) can be built by simply connecting all unit capacitors in each column using branch wires, and then connecting the unit capacitors in adjacent columns using a branch wire. The use of this MST minimizes, shown in Fig. 9(d), the parasitic capacitance, C_i^{TS}.

C. Layout generation for a split DAC

There are some differences between the common-centroid placement and routing algorithm for the split DAC and the binary-weighted DAC, since in the split DAC, the ratio of the capacitors and the circuit connection diagram is different from the binary-weighted DAC. This section outlines those differences.

1) Array size calculation: As in the binary-weighted DAC, to minimize the impact of systematic variations, the aspect ratio of the rectangular CC array in the split DAC is made as close to a square as possible. We calculate the array size, $r \times s$, by using an approach similar to presented in [19] for the LSB and MSB arrays. For an N-bit split DAC, the capacitor ratios are $[n_0 : n_1 : n_2 : \cdots : n_L : n_{L+1} : n_{L+2} : \cdots : n_N] = [1 : 1 : 2 : \cdots : 2^{L-1} : 1 : 2 : \cdots : 2^{N-L-1}]$. The required number of dummy unit capacitors to complete the array is given by $D_C = (r \times s) - \left(\sum_{i=0}^{N} n_i + 2\right)$. Here, the addition of the number 2 considers the case associated with the non-unit attenuation capacitor, which is implemented using two slots in the capacitor array.

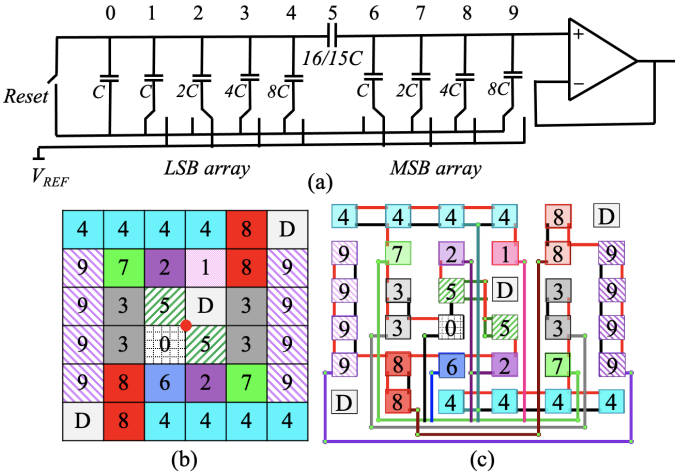


Fig. 10: CC placement and routing for a split DAC.

2) Placement: A modified spiral method can be applied for the placement of the split DAC and is illustrated for 8-bit DAC using $L = 4$ in Fig. 10(b), with a non-unit attenuation capacitor. The capacitors are represented by the numbers shown at the top of the schematic in Fig. 10(a). To obtain a common-centroid placement among the unit capacitors of both LSB and MSB capacitive arrays, first, we place two slots of the non-unit attenuation capacitor C_5, near the center of the array while maintaining the CC property, since it is at the center of the split DAC, connecting LSB and MSB arrays (Fig. 10(a)). Note that the actual size of each of the two capacitors for C_5 are smaller than C_u. Therefore, the relative perturbations in their values due to the routing capacitance is liable to be the largest, and can be minimized if they are placed as close to the center as possible.

Next, we consider C_0, C_1, and C_6, each of which has one unit capacitor, i.e., an odd number. The total number of capacitors having an odd number of unit capacitors is three, also an odd number, and this may cause asymmetry in the CC array. To achieve a CC placement, we place capacitor C_0 and D (dummy) diagonally opposite to each other near the center of the CC placement to reduce the impact of process variations. We place C_1 at the first empty location along the spiral and C_6 at its diagonally symmetric location near the CC point. We also evaluated an alternative approach that places C_0, C_1, C_2, and a dummy in the inner four squares, and the two unit capacitors of C_5 in symmetric locations in the next ring: we did not find a significant difference in performance metrics between this approach and the one outlined above.

After completing the placement of the odd-numbered capacitors, we place all unit capacitors of C_2 from the LSB array, and then the unit capacitors of C_7 from the MSB array into the CC matrix, by following the spiral order from the center. The process is continued alternately (e.g., by placing all unit capacitors of C_3 next, then C_8, and so on). As in the binary-weighted DAC, whenever we place a unit capacitor at a location along a spiral, we also place another unit capacitor at its reflection to maintain the CC property.

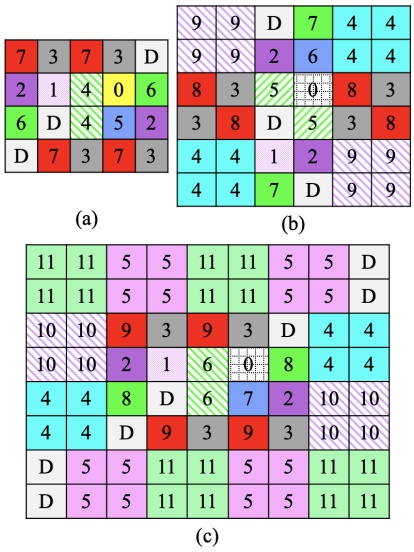


Fig. 11: Block chessboard placement for (a) 6-bit, (b) 8-bit, and (c) 10-bit split DAC.

Chessboard and block chessboard placement method can also be applied on split DAC. As in binary-weighted DAC we can create a complete chessboard pattern using the smaller capacitors and block chessboard pattern using larger capacitors where same block size can be applied for the larger capacitors from the LSB and MSB arrays for the same sized capacitors (e.g., block size of C_4 and C_9 can be same). During placement we place the blocks of the unit capacitors of same capacitor at the diagonal symmetric location with respect to the center as shown in Fig. 11. Block chessboard pattern can help to increase dispersion among the unit capacitors of same type of capacitor and can improve INL/DNL. However, due to increased dispersion routing induced parasitics will increase.

3) Routing: The following routing method is followed to route the capacitors of the split DAC in the CC array:
(1) Bottom plate routing: For routing the bottom plates of a split DAC for a modified spiral method, we followed a similar routing algorithm as for the binary-weighted DAC by constructing the connected capacitor groups for different capacitors as presented in Section IV-B1; next, we connect the bottom plates of those capacitor groups (Fig. 10(c)) by following the routing Algorithm outlined in [19]. The routing for a CC placement of an 8-bit split DAC shown in Fig. 10(c), where different shades of the same color is used to represent connected capacitor groups as used for the binary-weighted DAC (Fig. 9), and top-plate [bottom-plate] connections are shown in red [black]. The parallel wire routing method can also be applied on split DAC to reduce resistance and improve the 3dB frequency as presented in [19].
(2) Attenuation capacitor routing: Unlike binary-weighted DAC in split DAC there is a non-unit attenuation capacitor and to connect two slots of the attenuation capacitor we first check for their locations in the CC array. If the slots are adjacent to each other, they can be connected during connected capacitor group formation, but if the slots of the attenuation capacitor are not adjacent an extra vertical track must be assigned to connect them.
(3) Top plate routing: The top plate connection of the capacitors in split DAC is different from the binary-weighted DAC, since the top plates of the capacitors in each side array are connected together at two different electrical nodes that are separated by an attenuation capacitor. The top plate connection between two adjacent unit capacitors of a capacitor for the LSB and MSB capacitive arrays in the horizontal and vertical directions is performed during connected capacitor group formation using breadth-first search (BFS) as presented in Section IV-B1. Later, we again use BFS on the connected capacitor groups of both LSB, and MSB arrays to connect the adjacent capacitor groups as shown in Fig. 10(c) of different capacitors.

The top plates of the connected capacitor groups which are not adjacent to any other capacitor groups in both the LSB and MSB side arrays can be connected using a minimum spanning tree (MST). We create a graph G such that each vertex $v \in G$ is a connected capacitor group for any C_i from LSB [MSB] side array. We build an MST over this graph to find an optimal connection between these capacitor groups and complete the top plate routing by connecting the top plates of different capacitors (C_i) from each side array. This type of routing helps to minimize the top plate to substrate parasitics, C^{TS}_{LSB}, and C^{TS}_{MSB} (Fig. 3(a) and (b)). Later we connect the attenuation capacitor between the top plates of LSB and MSB side arrays. To minimize the routing induced parasitics, we connect the attenuation capacitor with the top plates of the adjacent unit capacitors from each side array.

V. Results and Discussion

Our approach is implemented in Python and evaluated on a commercial 12nm technology for N-bit binary-weighted DAC arrays with capacitors ratios of $1:1:2:4:\cdots:2^{N-1}$, with

N ranging from 6 to 10. We evaluate four techniques: the placement in [1]; the chessboard placement [8]; spiral placement ("S") and the block chessboard ("BC") [Section IV-A]. Several BC structures are considered, as shown in Fig. 12 and the **best BC result** is reported. All even/odd bit DACs use the same BC structure with a full chessboard for the inner core.

Our routing approach is applied to the S and BC methods. Since [8] only proposes a placement without routing, we use our router on their placement (Section IV-B).

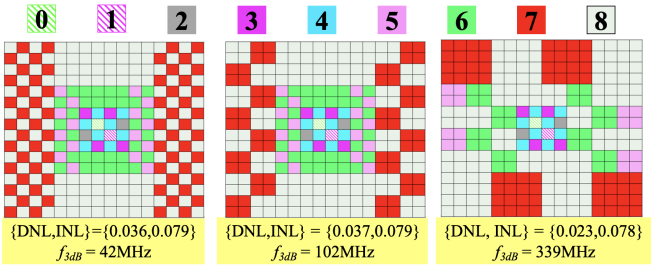


Fig. 12: 8-bit block chessboard layouts at various granularities.

For systematic variations, the wire spacing t_0 is based on a wire pitch of 64nm. Parasitic extraction is performed by considering the foundry-provided per-unit wire resistance r and per-unit wire capacitance c for each metal layer, the resistance and capacitance of a wire segment of length l are, respectively, $(r \cdot l)$ and $(c \cdot l)$. For adjacent wires, if the per-unit coupling capacitance is $c_c(s)$ for a spacing of s, the coupling capacitance between two segments with overlap length $l_{overlap}$ is $(c_c(s) \cdot l_{overlap})$. The per-unit models for resistance, capacitance to ground, and coupling-capacitance are taken from a commercial 12nm process. The systematic variation parameters were set to $\gamma = 10$ppm, $\rho_u = 0.9$, $L_c = 1$mm [1], [9], and we use $A_f^2 = 0.85\% \times 1$fF [21]. A unit capacitance value of 5fF is used for all cases for binary-weighted DACs. The MOM capacitors are built in three metal layers, with the bottom-plate and top-plate terminals available in metal1 and metal2, respectively.

Table I shows various parameters associated with the RC parasitics for routing the binary-weighted DAC. The capacitances, ΣC^{TS} (total top-plate-to-substrate), ΣC^{wire} (total wiring capacitance), and ΣC^{BB} (total bottom-plate to bottom-plate), represent the parasitics shown in Fig. 2; C^{TB} is negligible due to nonoverlapped routing (Section IV-B). The next set of metrics – ΣN_V, the total number of vias and ΣL the total wire length – are correlated with the total resistance. Since f_{3dB} only depends on resistances on the critical bit with the largest RC delay, the last column shows the total via resistance, R_V, and the total wire+via resistance, R_{total} for the critical bit.

Here, S has low resistive parasitics; BC has moderate parasitics, much lower than [1], [8]. The impact of parasitic resistance can be observed in Fig. 13 for different bit DACs for different placement methods. Both S and BC use our proposed parallel routing method: when parallel routing is used on the MSB, the second-most MSB, then the third-most MSB, etc., may become critical, and parallel routing is used there too. If parallel wires need channel resources, the spacing between columns is increased appropriately. For any number of bits for S, the only vias are at the input connection. Unit capacitors use nearest-neighbor connections using the same metal layer with no vias.

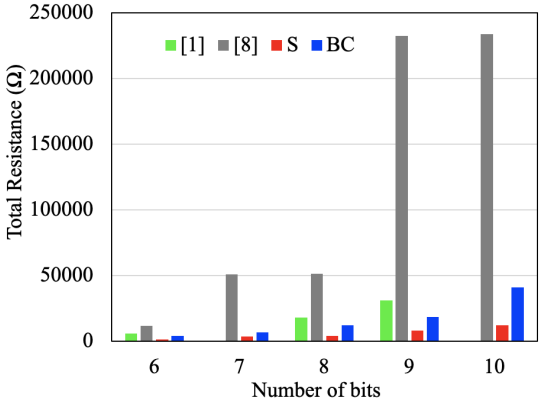


Fig. 13: Parasitic resistance for different bit DACs.

The C^{TS} values for S and BC are better than those of [1]; we apply the same solution to our routing for [8] (although their subsequent work [18] leads to higher top-plate wire lengths, i.e., higher C^{TS}). For other metrics (C^{wire}, C^{BB}, N_V, L and R), the spiral approach provides the best solution and the chessboard method [8] the worst, and the block chessboard method provides an intermediate solution.

Table II shows circuit-level metrics: Area of the routed CC binary-weighted array; $|DNL|$ / $|INL|$ the maximum absolute $DNL(i)$ (Eq. 17) / $INL(i)$ (Eq. 18); f_{3dB}, the 3dB frequency (Eq. (32)). Like [1], [8], we evaluate DNL/INL under capacitor nonidealities, assuming an ideal opamp. Area is lowest for the spiral method due to low routing overhead and comparable with other methods (except 7-bit and 9-bit solutions for [8], which double the unit capacitors). The table shows the INL/DNL vs. f_{3dB} tradeoff (particularly for >8 bits): *S has the best f_{3dB} but the worst INL/DNL; [8] is the opposite; BC is a good compromise.* All INL/DNL values are below 0.5LSB and are acceptable.

Figs. 14(a) and (b) show a Python-generated view of the placement and routing for an 8-bit DAC in a commercial 12nm process, using the approach in [8] and our spiral placement

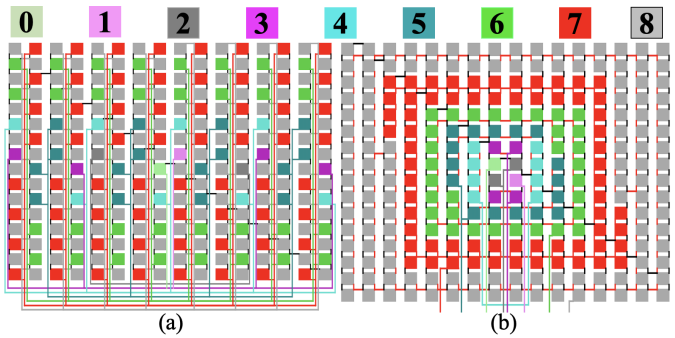


Fig. 14: CC layout in (a) [8] (b) spiral approach. High wirelength for [8] is inevitable: cells are spread for high dispersion.

TABLE I: CC array: Electrical metrics for binary-weighted DACs of various resolutions (C_u = 5fF).

#	$\sum C^{TS}$ (fF)				$\sum C^{wire}$ (fF)				$\sum C^{BB}$ (fF)				($\sum N_V$, $\sum L$) (μm)				(R_V, R_{total}) (KΩ) for critical bit			
bits	[1]	[8]	S	BC	[1]	[8]	S	BC	[1]	[8]	S	BC	[1]	[8]	S	BC	[1]	[8]	S	BC
6	0.10	0.10	0.10	0.10	3.8	8.3	2.8	7.1	5.3	9.3	2.1	5.6	(34, 321)	(81, 686)	(118, 243)	(154, 583)	(0.2, 1.2)	(1.1, 3.6)	(0.002, 0.05)	(0.01, 1.1)
7	–	0.38	0.20	0.22	–	36.9	4.5	7.0.0	–	70.3	3.7	9.1	–	(295, 2993)	(76, 385)	(110, 591)	–	(4.1, 14.0)	(0.002, 0.09)	(0.002, 1.7)
8	0.40	0.38	0.38	0.38	12.3	36.6	5.8	18.5.0	17.6	73.3	6.1	15.8	(60, 1042)	(295, 3012)	(60, 514)	(234, 1535)	(0.3, 4.1)	(4.1, 14.0)	(0.002, 0.16)	(0.02, 0.67)
9	0.72	1.54	0.76	0.78	15.3	162.8	10.1	19.7	61.0	633.0	9.6	22.8	(143, 1319)	(1126, 13299)	(78, 907)	(190, 1682)	(1.2, 6.9)	(15.8, 56.3)	(0.002, 0.28)	(0.002, 3.3)
10	–	1.54	1.50	1.50	–	163.9	17.6	49.5	–	634.7	16.7	66.6	–	(1126, 13308)	(92, 1596)	(473, 4165)	–	(15.8, 56.3)	(0.002, 0.58)	(0.02, 1.60)

Notes: (1) [8] doubles the number of unit capacitors for odd bits $\Rightarrow$ {7-bit, 8-bit}, {9-bit, 10-bit} results are similar. (2) 7-bit, 9-bit DACs not reported in [1].

TABLE II: CC array: Performance metrics for binary-weighted DACs of various resolutions (C_u = 5fF).

#	Area (μm^2)				{$\lvert DNL\rvert, \lvert INL\rvert$} (LSB)				f_{3dB} (MHz)			
bits	[1]	[8]	S	BC	[1]	[8]	S	BC	[1]	[8]	S	BC
6	629	634	622	634	{0.00,0.02}	{0.00,0.02}	{0.01,0.02}	{0.01,0.02}	947	313	21179	3883
7	–	2541	1293	1421	–	{0.01,0.03}	{0.02,0.04}	{0.02,0.03}	–	18	5505	1197
8	2641	2541	2509	2541	{0.00,0.08}	{0.01,0.06}	{0.05,0.08}	{0.02,0.07}	56	16	1411	339
9	5354	10627	5085	5224	{0.01,0.14}	{0.02,0.14}	{0.12,0.16}	{0.11,0.15}	15	0.9	361	123
10	–	10627	10076	10157	–	{0.05,0.30}	{0.22,0.31}	{0.11,0.30}	–	0.8	81	29

TABLE III: CC array: Performance metrics ({$\lvert DNL\rvert, \lvert INL\rvert$} (LSB)) for binary-weighted DACs of various resolutions, varying the linear process gradient coefficient, γ (C_u = 5fF, $\rho_0 = 0.9$).

γ	[8]			S		
	6-bit	8-bit	10-bit	6-bit	8-bit	10-bit
1	{0.00,0.02}	{0.01,0.08}	{0.05,0.31}	{0.01,0.02}	{0.06,0.08}	{0.27,0.31}
10	{0.00,0.02}	{0.01,0.08}	{0.05,0.31}	{0.01,0.02}	{0.05,0.08}	{0.22,0.31}
100	{0.01,0.02}	{0.02,0.09}	{0.06,0.34}	{0.01,0.02}	{0.27,0.17}	{4.93,2.56}

TABLE IV: CC array: Performance metrics ({$\lvert DNL\rvert, \lvert INL\rvert$} (LSB)) for binary-weighted DACs of various resolutions, varying the correlation coefficient ρ_u (C_u = 5fF, γ = 10ppm).

ρ_u	[8]			S		
	6-bit	8-bit	10-bit	6-bit	8-bit	10-bit
0.10	{0.00,0.02}	{0.00,0.08}	{0.00,0.31}	{0.00,0.02}	{0.00,0.08}	{0.05,0.30}
0.50	{0.00,0.02}	{0.01,0.08}	{0.01,0.31}	{0.02,0.02}	{0.02,0.08}	{0.03,0.30}
0.70	{0.00,0.02}	{0.01,0.08}	{0.02,0.31}	{0.02,0.02}	{0.05,0.08}	{0.07,0.30}
0.75	{0.00,0.02}	{0.01,0.08}	{0.03,0.31}	{0.02,0.02}	{0.06,0.08}	{0.09,0.30}
0.80	{0.00,0.02}	{0.01,0.08}	{0.03,0.31}	{0.01,0.02}	{0.06,0.08}	{0.13,0.30}
0.85	{0.00,0.02}	{0.01,0.08}	{0.04,0.31}	{0.01,0.02}	{0.06,0.08}	{0.18,0.30}
0.90	{0.00,0.02}	{0.01,0.08}	{0.05,0.31}	{0.01,0.02}	{0.05,0.08}	{0.22,0.30}
0.95	{0.00,0.02}	{0.01,0.08}	{0.04,0.31}	{0.01,0.02}	{0.03,0.08}	{0.19,0.30}
0.99	{0.00,0.02}	{0.00,0.08}	{0.01,0.31}	{0.00,0.02}	{0.01,0.08}	{0.02,0.29}
0.999	{0.00,0.02}	{0.00,0.08}	{0.00,0.31}	{0.00,0.02}	{0.00,0.08}	{0.04,0.29}

method, respectively. While the former requires five vertical tracks in the vertical channels, the spiral approach, even with parallel routes, requires two routing tracks, resulting in lower C^{BB}, as documented by the total C^{BB} number in Table I. The total routing wirelength is significantly higher for the placement of [8], leading to higher C^{wire} parasitics, as shown in the same table. Both factors degrade the 3dB frequency of [8]. This effect is worse as the number of bits in the DAC increases.

In FinFET nodes, N_V adversely affects performance due to high via resistance. The spiral method uses the fewest vias of all methods, and the chessboard method [8] uses the most. Fig. 15(a) shows the impact of using parallel routes, which reduce interconnect resistance, on the 3dB frequency for spiral placement. We show the frequency improvement factor, where the ratio of the 3dB frequency using k wires vs. using one wire (i.e., a value of ≥ 1 represents an improvement). The increase in parasitic capacitance due to parallel wires is minimal and is dominated by the capacitance in the array, but the wire resistance reduction is significant. As k increases, we see diminishing returns. Similar trends are seen for the block chessboard scheme. When two parallel wires are used, the frequency improvement factor exceeds 2: for this resistance-dominated case, the connection from the trunk wire to a branch wire creates a 2×2 mesh with four vias. The gain lies between $2\times$ (wire-dominated case) and $4\times$ (via-dominated case). With more wires, the wire capacitance becomes noticeable, leading to lower improvement and we do not get the improvement in terms of frequency by a factor of k. Fig. 15(b) shows the impact of parallel wires for all methods, normalized to the 3dB frequency for S. Both BC and [1] improve, but have much lower baseline frequencies than S; chessboard [8] is bottlenecked by high via counts.

Table V shows the results for the split DAC with two parallel wires for routing for C_u = 5fF for 6-bit to 12-bit and 10fF for 14-bit for both spiral and block chessboard placement. Here, we use similar systematic variation parameters as for the binary-weighted DAC. Various parameters associated with the RC parasitics for routing for the split DAC are shown in the table. The capacitances, ΣC^{TS}, ΣC^{wire}, and ΣC^{BB}, represent the total parasitics for the circuit shown in Fig. 3; noting that the parasitics from the LSB and MSB side arrays are added; C^{TB} from each side array is negligible due to nonoverlapped routing used for connected capacitor group formation. The total number of vias ΣN_V, and the total wire length, ΣL, of the capacitor array are listed next. We

TABLE V: CC array: Electrical and performance metrics for split DACs.

# bits	$\sum C^{TS}$ (fF)		$\sum C^{wire}$ (fF)		$\sum C^{BB}$ (fF)		($\sum N_V$, $\sum L$) (μm)		(R_V, R_{total}) (KΩ) for critical bit		Area (μm^2)		$\{\|DNL\|, \|INL\|\}$ (LSB)		f_{3dB} (MHz)	
bits	S	BC	S	BC	S	BC	S	BC	S	BC	S	BC	S	BC	S	BC
6	0.03	0.03	2.37	2.97	0.58	0.72	(45, 194)	(70, 243)	(0.02, 0.34)	(0.05, 0.67)	194	193	(0.01, 0.01)	(0.03, 0.00)	47569	33572
8	0.06	0.06	3.87	5.29	1.68	1.60	(65, 319)	(85, 433)	(0.03, 0.38)	(0.03, 0.73)	362	353	(0.06, 0.04)	(0.04, 0.03)	19977	15256
10	0.12	0.11	6.47	9.34	6.62	7.85	(68, 534)	(105, 766)	(0.02, 0.71)	(0.05, 0.78)	724	719	(0.14, 0.10)	(0.12, 0.08)	8153	3615
12	0.20	0.21	7.98	12.4	7.16	12.23	(65, 666)	(100, 1024)	(0.02, 0.72)	(0.03, 0.84)	1302	1339	(0.17, 0.16)	(0.13, 0.17)	2136	1504
14	0.43	0.41	11.4	23.4	24.48	26.55	(72, 970)	(126, 1932)	(0.03, 1.61)	(0.05, 1.64)	2809	2719	(0.49, 0.31)	(0.51, 0.20)	484	169

TABLE VI: Runtimes for the proposed CC layout algorithms for binary-weighted DACs

#bits	6-bit	7-bit	8-bit	9-bit	10-bit
Spiral	0.02s	0.04s	0.12s	0.35s	1.11s
BC	0.03s	0.05s	0.19s	0.38s	2.25s

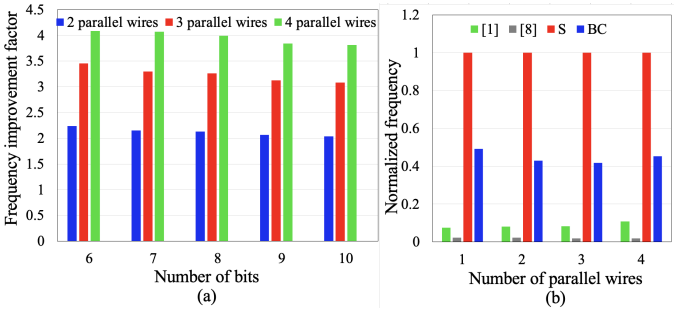
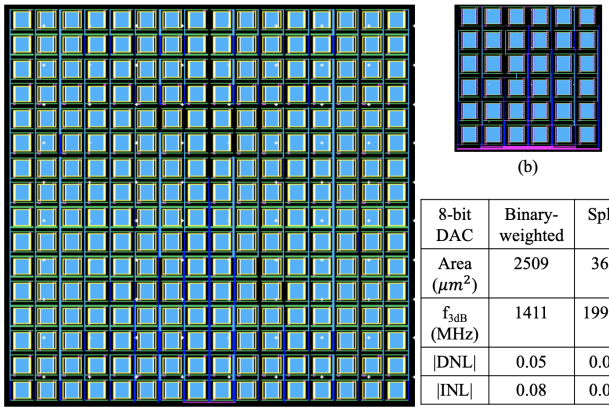


Fig. 15: 8-bit CC array with parallel wires: 3dB frequency improvements for (a) the spiral method. (b) all methods.



8-bit DAC	Binary-weighted	Split
Area (μm^2)	2509	362
f_{3dB} (MHz)	1411	19977
|DNL|	0.05	0.06
|INL|	0.08	0.04

(c)

Fig. 16: CC layout of 8-bit (a) binary-weighted, and (b) split DAC, (c) comparison result table.

also list the total via resistance, R_V, and the total wire+via resistance, R_{total} for the critical bit. The capacitive routing-induced parasitics for the split DAC are comparable with the binary-weighted DAC. The resistive parasitics for the split DAC for the critical bit are higher than the binary-weighted DAC since in binary-weighted DAC all the unit capacitors for the critical bit form a connected capacitor group and it requires a very short routing wire to connect the connected capacitor group with the switches and drivers; therefore, we use two parallel wires for routing the capacitors of the split DAC, whereas we used four parallel wires for routing the binary-weighted DAC.

In Table V we also have listed Area; the maximum absolute $DNL(i)/INL(i)$; f_{3dB} for the split DAC. The INL/DNL for the split DAC as shown in the Table V is comparable with the INL/DNL of the binary-weighted DAC (Table II), at the same time it can provide significantly better area and frequency since the total amount of unit capacitors is lower for the split DAC. The main advantage of the split DAC is that it can provide good performance in terms of INL/DNL and 3dB frequency (f_{3dB}) while using a smaller silicon area than binary-weighted DAC. From the table it can be observed that for block chessboard method the wire parasitics and wire lengths are slightly higher than the spiral, but the INL/DNL is better for block chessboard method from the spiral method.

Fig. 16(a), and (b) show the generated layout, visualized using Virtuoso, based on the GDS generated by our proposed spiral placement approach for 8-bit binary-weighted and split DAC, respectively. As can be observed from the figures and the comparison table (Fig. 16(c)), the layout area for the split DAC is significantly lower than the binary-weighted DAC. Since the number of unit capacitors for the split DAC is much less than the binary-weighted DAC, 3dB frequency (f_{3dB}) is considerably higher for the split DAC. Finally, the absolute maximum INL/DNL values are comparable for both of the DACs as shown in the table attached in Fig. 16(c).

The CPU times for both the spiral method and for each block chessboard are similar and are reported in Table VI for binary-weighted DACs. Because the method is constructive, it is much faster than stochastic optimization, while providing excellent quality of result. For split DAC the number of unit capacitors is much smaller than the binary-weighted DAC with the same number of bits, and the CPU times are therefore much smaller than these already small runtimes for the binary DACs.

The choice of $\rho_0 = 0.9$ and $\gamma = 10$ppm is based on prior work [1], [20]. In [23], a variational model for resistors, rather than capacitors is used, and this model has been assumed to be reasonable for other passives such as capacitors; work in [27] develops an exponential model for transistor variations. To our knowledge, there is no reported measured value for correlations in capacitive structures: correlation is alluded to in [21], but only data for uncorrelated variations are reported. The value of $\rho_0 = 0.9$ and $\gamma = 10$ppm has cascaded down from older papers such as [6].

Therefore, we show the INL and DNL for a 6-bit, 8-bit, and 10-bit binary-weighted DAC, for a range of values of ρ_0 and γ in Tables IV and III, respectively, under spiral placement and chessboard placement. When ρ_0 is varied, we keep γ fixed at 10ppm, and when γ is varied, we keep ρ_0 fixed at 0.9. It can be seen that the precise dependency of the INL and DNL on ρ_0 and γ is not monotonic. This is because the INL and DNL expressions depend on V_{out}, which in turn depend on the relative values of the uncorrelated capacitance variations,

correlated capacitance variations, and the wire parasitics. At different entries of these tables, different components become more dominant, leading to this nonmonotonicity.

It should be noted that in Table III, as γ is increased, the variation in the oxide thickness t increases linearly. Since the capacitance $C = C_u(t_0/t)$ is nonlinearly related to t, the linear approximation to C becomes increasingly inaccurate as t deviates from its nominal value, t_0. Thus, for higher values of γ, these nonlinearities become more prominent, reducing the ability of common-centroid layouts to cancel variations. As a result, we see that the DNL and INL generally worsen for large values of γ, overcoming smaller nonmonotonicities at lower values of γ.

VI. Conclusion

A set of routing-conscious, constructive common-centroid placement methods, followed by a routing step, are proposed in this work. These methods attempt to optimize systematic and random mismatches in order to improve performance and accuracy for both binary-weighted and split DACs. To the best of our knowledge, this is the first work in the literature to propose models to evaluate tradeoffs between linearity metrics as well as the 3dB frequency for binary-weighted DACs, and the first to automate CC layout generation for split DACs. For the split DAC, a method to determine the size of the non-unit capacitor, which is used as the attenuation capacitor, is also presented. Two different types of placement methods, the spiral and block chessboard methods, have been presented and experimental results show that a balance between INL/DNL and 3dB frequency can be achieved by trading off wire parasitics with dispersion.

References

[1] M. P.-H. Lin, Y.-T. He, V. W.-H. Hsiao, R.-G. Chang, and S.-Y. Lee, "Common-Centroid Capacitor Layout Generation Considering Device Matching and Parasitic Minimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, pp. 991–1002, 2013.

[2] W.-H. Hsiao, Y.-T. He, M. P.-H. Lin, R.-G. Chang, and S.-Y. Lee, "Automatic Common-Centroid Layout Generation for Binary-Weighted Capacitors in Charge-Scaling DAC," in *Proceedings of the IEEE International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, 2012, pp. 173–176.

[3] A. Hastings, *The Art of Analog Layout*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[4] N. Karmokar, M. Madhusudan, A. K. Sharma, R. Harjani, M. P.-H. Lin, and S. S. Sapatnekar, "Common-Centroid Layout for Active and Passive Devices: A Review and the Road Ahead," in *Proceedings of the Asia-South Pacific Design Automation Conference*, 2022.

[5] D. Sayed and M. Dessouky, "Automatic Generation of Common-Centroid Capacitor Arrays with Arbitrary Capacitor Ratio," in *Proceedings of the Design, Automation & Test in Europe*, 2002, pp. 576–580.

[6] P.-W. Luo, J.-E. Chen, C.-L. Wey, L.-C. Cheng, J.-J. Chen, and W.-C. Wu, "Impact of Capacitance Correlation on Yield Enhancement of Mixed-Signal/Analog Integrated Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, pp. 2097–2101, 2008.

[7] C.-C. Huang, J.-E. Chen, and C.-L. Wey, "PACES: A Partition-Centering-Based Symmetry Placement for Binary-Weighted Unit Capacitor Arrays," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, pp. 134–145, 2016.

[8] F. Burcea, H. Habal, and H. E. Graeb, "A New Chessboard Placement and Sizing Method for Capacitors in a Charge-Scaling DAC by Worst-Case Analysis of Nonlinearity," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, pp. 1397–1410, 2015.

[9] M. P.-H. Lin, V. W.-H. Hsiao, C.-Y. Lin, and N.-C. Chen, "Parasitic-Aware Common-Centroid Binary-Weighted Capacitor Layout Generation Integrating Placement, Routing, and Unit Capacitor Sizing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, pp. 1274–1286, 2017.

[10] N.-C. Chen, P.-Y. Chou, H. Graeb, and M. P.-H. Lin, "High-Density MOM Capacitor Array with Novel Mortise-Tenon Structure for Low-Power SAR ADC," in *Proceedings of the Design, Automation & Test in Europe*, 2017, pp. 1757–1762.

[11] P.-Y. Chou, N.-C. Chen, M. P.-H. Lin, and H. Graeb, "Matched-Routing Common-Centroid 3-D MOM Capacitors for Low-Power Data Converters," *IEEE Transactions on VLSI Systems*, vol. 25, pp. 2234–2247, 2017.

[12] Y. Zhu, U.-F. Chio, H.-G. Wei, S.-W. Sin, U. Seng-Pan, and R. P. Martins, "Linearity Analysis on a Series-Split Capacitor Array for High-Speed SAR ADCs," in *2008 51st Midwest Symposium on Circuits and Systems*. IEEE, 2008, pp. 922–925.

[13] R. J. Baker, *CMOS Circuit Design, Layout and Simulation*, 3rd ed. Piscataway, NJ: IEEE Press, 2010.

[14] Y. Chen, X. Zhu, H. Tamura, M. Kibune, Y. Tomita, T. Hamada, M. Yoshioka, K. Ishikawa, T. Takayama, J. Ogawa *et al.*, "Split Capacitor DAC Mismatch Calibration in Successive Approximation ADC," *IEICE Transactions on Electronics*, vol. 93, no. 3, pp. 295–302, 2010.

[15] A. Agnes, E. Bonizzoni, P. Malcovati, and F. Maloberti, "A 9.4-ENOB 1V 3.8 μW 100kS/s SAR ADC with Time-Domain Comparator," in *Proceedings of the IEEE International Solid-State Circuits Conference*. IEEE, 2008, pp. 246–610.

[16] D. A. Johns and K. Martin, *Analog Integrated Circuit Design*. John Wiley & Sons, 2008.

[17] K.-H. Ho, H.-C. Ou, Y.-W. Chang, and H.-F. Tsao, "Coupling-Aware Length-Ratio-Matching Routing for Capacitor Arrays in Analog Integrated Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, pp. 161–172, 2014.

[18] Y. X. Ding, F. Burcea, H. Habal, and H. E. Graeb, "PASTEL: Parasitic Matching-Driven Placement and Routing of Capacitor Arrays With Generalized Ratios in Charge-Redistribution SAR-ADCs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, pp. 1372–1385, 2019.

[19] N. Karmokar, A. K. Sharma, J. Poojary, M. Madhusudan, R. Harjani, and S. S. Sapatnekar, "Constructive common-centroid placement and routing for binary-weighted capacitor arrays," in *Proceedings of the Design, Automation & Test in Europe*. IEEE, 2022, pp. 166–171.

[20] C.-W. Lin, J.-M. Lin, Y.-C. Chiu, C.-P. Huang, and S.-J. Chang, "Mismatch-Aware Common-Centroid Placement for Arbitrary-Ratio Capacitor Arrays Considering Dummy Capacitors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, pp. 1789–1802, 2012.

[21] V. Tripathi and B. Murmann, "Mismatch Characterization of Small Metal Fringe Capacitors," *IEEE Transactions on Circuits and Systems I*, vol. 61, pp. 2236–2242, 2014.

[22] M. J. Pelgrom and A. C. Duinmaijer, "Matching Properties of MOS Transistors," in *Proceedings of the European Solid-State Circuits Conference*, 1988, pp. 327–330.

[23] C. S. Conroy, W. Lane, and M. Moran, "Statistical design techniques for d/a converters," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 4, pp. 1118–1128, 1989.

[24] F. Maloberti, *Data Converters*. New York, NY: Springer, 2007.

[25] S. S. Sapatnekar, *Timing*. Boston, MA: Kluwer, 2004.

[26] G. Chen, B. Liu, S. Nakatake, and B. Yang, "Routability of Twisted Common-Centroid Capacitor Array Under Signal Coupling Constraints," in *Proceedings of the IEEE International Midwest Symposium on Circuits and Systems*, 2016.

[27] J. Xiong, V. Zolotov, and L. He, "Robust extraction of spatial correlation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 4, pp. 619–631, 2007.