Performance-Aware Common-Centroid Placement and Routing of Transistor Arrays in Analog Circuits

Arvind K. Sharma[1], Meghna Madhusudan[1], Steven M. Burns[2]
Soner Yaldiz[2], Parijat Mukherjee[2], Ramesh Harjani[1], Sachin S. Sapatnekar[1]
[1]University of Minnesota, Minneapolis, MN [2]Intel Corporation, Hillsboro, OR

***Abstract*—The common-centroid (CC) layout style is widely used to minimize the impact of variations among matched devices in analog blocks such as current mirror banks and differential pairs. This paper presents a constructive, performance-aware CC placement and routing algorithm for transistor arrays. Specifically, the proposed approach maximizes diffusion sharing, incorporates length of diffusion (LOD) based stress-induced performance variations, and mitigates resistive parasitics and electromigration (EM) hotspots, all of which are critical in modern technology nodes. The proposed algorithms are validated using cell- and circuit-level test cases in a commercial 12nm FinFET process. As compared to existing works, the cells generated using the proposed approach are shown to provide better performance in the presence of systematic variations, LOD, layout parasitics, and EM-induced degradation.**

I. Introduction

A. Motivation

Technology scaling has been an essential tool for improving integrated circuit performance. However, successive technology generations have resulted in complex variation patterns that cause unpredictability in circuit performance parameters. The use of layout structures that ensure device matching is critical to the design of high-performance analog circuits.

Variations in a design are typically attributed to process, voltage, and temperature [1]. Voltage variations can be controlled using well-designed supply networks, and temperature variations typically have a wide footprint. Process-induced on-chip variations can be categorized as either systematic variations, which can be modeled predictably, or random variations, which can only be represented statistically. Some sources of systematic variations are photo mask errors [2], and process parameter gradients over a die (e.g., across-chip length variation) [3]. Random dopant fluctuations (RDF) [4] and line edge roughness (LER) [5] are examples of sources of random variations and modeled as uncorrelated distributions.

Random variations can be reduced by increasing the device area [6]. To reduce the impact of systematic variations, common-centroid (CC) layout patterns are widely used to ensure matching in array structures such as transistors and capacitors. In a CC layout devices to be matched are divided into small devices called unit cells; these unit cells are placed such that the centroid of all devices in an array is identical [7]. This pattern is symmetric around both the X- and Y-axis: all devices are distributed uniformly, and aspect ratio of the layout is near to a square [7]. For example, Fig. 1 shows a CC layout pattern of a differential pair. The devices A and B are each divided into sixteen unit cells and placed such that the centroid coincides at **C**. A CC layout minimizes the impact of systematic variations but it is difficult and time consuming to generate an optimal CC placement manually. Moreover, a CC placement must also be routing-friendly: for example, resistive parasitics at the terminals of device A and B in a differential pair (Fig. 1) impact the transistor transconductance, and should be small and matched. This can be achieved by combining CC placement with CC routing.

This work is supported in part by the DARPA IDEA program, as part of the ALIGN project, under SPAWAR Contract N660011824048.

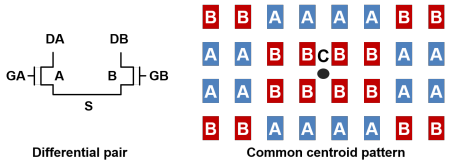


Figure 1: A common-centroid layout pattern of a differential pair.

Analog designs are more easily built in older technologies, but due to application-driven requirement of integrating analog on-chip functionality with digital processing, analog circuits are increasingly being designed at lower technology nodes. In such technologies, layout dependent effects (LDEs) and interconnect parasitics are critical and must be considered during layout generation. LDEs affect the device threshold voltage and mobility, and can cause differential mismatch between devices unless specifically countered during placement and routing. High resistive parasitics in advanced technologies can also shift circuit performance. In lower metal layers, where wire resistances for minimum-sized wires are high, it is important to identify sensitive wires and reduce their resistance by using wider connections, typically implemented using parallel wires in FinFET nodes. The use of wider connections is also effective in mitigating electromigration (EM) by reducing the current density. Via resistances are also significant in nanometer-scale technologies, and the requirement of unidirectional routing in lower metal layers requires CC layouts to use routes with a small number of vias.

Diffusion sharing between adjacent transistors is important in all technologies, because it can help reduce cell area and result in lower source/drain diffusion parasitics. Critically for nanometer-scale technologies, interconnect parasitics are also reduced: compact cells have lower routing lengths, and hence lower wire parasitics, and the use of diffusion sharing reduces the number of vias. Diffusion sharing is particularly tricky for CC layout because it must be performed uniformly in the array, such that each device is matched to have the same number of diffusion breaks to avoid mismatch.

B. Prior work

CC layouts to minimize systematic variations have been extensively studied [8]–[16]. In [8]–[11], CC placement and routing algorithms for capacitor arrays have been proposed. However, these algorithms are not applicable to transistor arrays, where considerations such as diffusion sharing and LDEs must be taken into account.

The work in [13], [14] present constructive algorithms to generate CC patterns for transistor arrays. Thermal effects are also considered for placement generation in [14]. However, neither of them addresses the routing problem, or the issue of diffusion sharing between transistors, or LDEs. In [15], a diffusion-sharing-aware CC placement and routing algorithm is presented. To maximize diffusion sharing,

the circuit is represented by a graph, with nodes represented by vertices and source-drain connections by edges, and an iterative approach that considers all possible Euler paths is used for placement: this enumeration can be expensive. However, LDEs and parasitic mismatches are not taken into account. In ALIGN a generator based approach is used to generate layout patterns for analog cells such as current mirrors and differential pairs [17]. The CC placement and routing patterns for these cells are specified by the developer/user.

In [12], the notion of dispersion, the degree to which the unit cells of a transistor are distributed throughout a layout, is used to compare layouts and methods for generating maximally dispersive layouts are presented. However, the proposed techniques can only be applied to arrays with two transistors. A CC placement for FinFETs considering the impact of gate misalignment is studied in [16]. The placement algorithm is diffusion-sharing-aware and maximizes the dispersion of the unit cells to minimize random mismatch, and a parasitic aware routing algorithm is used. However, parasitic mismatch due to routing, dominant LDE modes such as length of diffusion (LOD), and EM are not accounted for. Moreover, the algorithms are developed for current mirror structures, therefore, limiting their use.

C. Our Contributions

Our contributions are summarized as follows:

- We present a generalized constructive CC placement algorithm for transistor arrays that maximizes diffusion sharing between devices and is routing-friendly.
- We incorporate LDE-awareness into the CC placement algorithm, which is very critical at lower technology nodes.
- We develop a parasitic-mismatch-aware routing algorithm that also incorporates EM considerations.
- We demonstrate experimentally that in comparison with existing approaches, the transistor arrays placed and routed using our approach perform better in the presence of systematic variations, LDEs, layout parasitics, and EM-induced degradation.

The rest of the paper is organized as follows: Section II reviews on-chip variations and LDEs at lower technology nodes. Section III presents constructive CC placement and routing algorithms. Section IV demonstrate validation of the proposed algorithms on different test cases and Section V concludes the paper.

II. Preliminaries

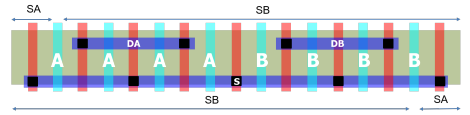
A. On-chip variations

On-chip variations can be classified as systematic or spatial variations; random variations are included in the latter class. Systematic variations are generally modeled as a gradient across the chip, while spatial variations are further categorized as having either a short or long correlation distance [6]. Spatial variations with correlation distance much smaller than the transistor dimension, e.g., random dopant fluctuations (RDF) [4], are commonly known as random variations [6]. In this paper, we validate our proposed algorithms against all these variations. We simulate spatial variations using random fields [18] using a methodology similar to [19].

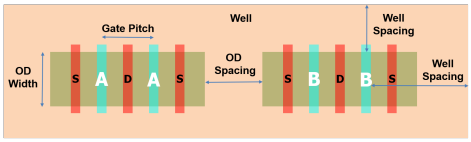
B. Layout dependent effects

At advanced technology nodes, LDEs [20]–[22] induce shifts in transistor performance parameters stemming from relative position in the layout. The most common LDEs (Fig. 2) are discussed next.

Well proximity effect (WPE) At nanoscale CMOS nodes, to minimize the latchup effect high-energy ions are used to create a deep retrograde well profile [22]. However, the high-energy ions scatter at the edge of photo resist and change the doping profile that modifies



(a) SA and SB parameters for LOD effect.



(b) OD width and spacing, gate pitch, and well proximity effects.

Figure 2: Layout dependent effects.

V_{th} of a device based on its distance from the well edge. This effect is commonly knows as WPE [22]. The well spacing is shown for device B in Fig. 2 (b). The WPE induced mismatch can be minimized by keeping well edges far from devices or by maintaining equal well spacing for the devices to be matched.

Process-induced stress has been intentionally used at nanoscale nodes to improve a transistor performance. However, the improvement depends on a device layout and its proximity, therefore, result in LDEs. The main LDEs due to the stress are as follows:

Length of diffusion (LOD) One of the most significant LDEs is caused by LOD effect [23], whereby the stress on a transistor, and hence its V_{th}, varies with the length of the diffusion region. The impact of LOD [23] is described by two parameters, SA and SB, the distances from poly-gate to the diffusion/active edge on either side of the device. For a device of gate length L_g, and n unit cells [24]:

$$\Delta V_{th} \propto \frac{1}{\text{LOD}} = \sum_{i=1}^{n} \left(\frac{1}{\text{SA}_i + 0.5L_g} + \frac{1}{\text{SB}_i + 0.5L_g} \right) \quad (1)$$

Fig 2(a) shows SA and SB parameters for a unit cell of device A and B. Devices to be matched must have same values of SA and SB, in order to match their threshold voltage shift, ΔV_{th}.

Oxide definition (OD) spacing and width Spacing between the OD regions (active areas), shown in Fig. 2(b), changes stress induced in a transistor; therefore, V_{th} varies as a function of OD spacing [20]. The effect is also known as oxide spacing effect (OSE). Moreover, stress induced in a transistor varies with the OD width (active area width). These effects can be avoided by maintaining same OD width and spacing for devices to be matched. For analog cells a unit cell based approach is used in which devices to be matched are divided into unit cells, therefore, same OD width is maintained for different devices. Further, the same OD spacing is used across unit cells. Moreover, the unit cells are placed such that devices to be matched have same number of diffusion breaks (i.e., OD breaks).

Gate pitch Stress induced in a transistor is also a function of gate pitch or poly pitch [20]. Gate pitch is shown in Fig. 2(b) for device A. As gate pitch increases the volume of the stressor material around the poly increases, this results in increased induced stress in the transistor channel, consequently, V_{th} varies. In analog cells, the effect is minimized by using a same poly pitch for devices to be matched.

In this work, we use a unit cell approach that is designed to cancel out all LDEs except LOD and WPE. Specifically, the gate/poly pitches are uniform for the analog blocks we place in CC; by construction, the unit cell approach ensures that the OD width is uniform; the y-direction OD spacing (OSE) is uniform for each transistor due to the use of a row-based unit cell placement approach, and the x-direction spacing is uniform due to diffusion sharing. Therefore, we focus on optimizing LOD and WPE mismatch through the use of dummies and using placement techniques.

C. Electromigration and parasitics

At nanometer-scale technologies EM has become a major reliability concern, especially for analog and mixed-signal circuits where substantial DC biasing currents can flow for extended periods. Prolonged current flow through metal wires can result in the physical migration of metal atoms due to the electron wind. Over a period, this aging phenomenon can increase the wire resistance or lead to open-circuit failures on lines with high current atomic flux. In older technologies, EM was primarily an issue in upper metal layers, but as wire thicknesses have scaled down, these issues are most critical in lower metal layers where the current densities are high.

Parasitics are critical in analog layouts and can degrade the circuit performance considerably and also cause circuit failure. Nodes can be sensitive to resistive or capacitive parasitics or both. Typically, in nanometer-scale technology nodes, the resistivity of the lower metal layers is very high [25]. Moreover, uni-directional routing for lower metal layers results in increased parasitics due to an increased number of vias. Hence, resistance parasitics tend to dominate at the analog cell level. In current mirrors, these routing parasitics can cause mismatch in the source voltage of matched devices and may result in a current ratio shift. In differential pair circuits, these parasitics affect the transconductance of the circuit (G_m) which can further degrade the performance of the analog circuit such as the gain and bandwidth of an operational trans-conductance amplifier (OTA).

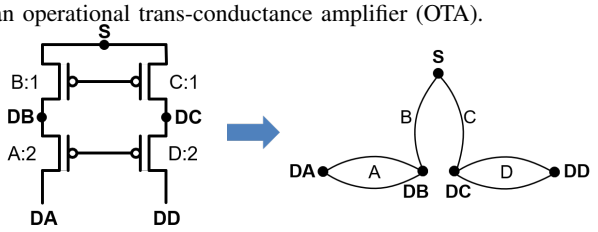


Figure 3: (a) A PMOS cascoded load and (b) its corresponding graph.

III. Common-centroid Placement and Routing

A. Graph representation of analog cells

We develop a CC layout algorithm for the transistor-based building-block cells (without passives) that are commonly used in analog circuits (e.g., current mirrors, differential pairs, cascoded differential pair, cascoded load, etc.), which we will refer to as "analog cells."

We represent the transistor netlist of an analog cell as a graph, $G(V, E)$. The set of vertices V represents nodes in the schematic/netlist, and the set of edges E corresponds to source-drain connections of the transistors, where the number of edges for a device is equal to the number of unit cells for the device. Fig. 3(a) shows the schematic of a PMOS cascoded load and its corresponding graph. The cascoded load has four devices, where devices A and D have two unit cells each, while B and C have one unit cell each. The corresponding graph is shown in Fig. 3(b).

B. Common-centroid placement

In Algorithm 1, we present a procedure for placing the devices in an analog cell in CC pattern. In addition to canceling out systematic process variations in the devices, which is ensured by CC placement, the algorithm optimizes the area and source/drain parasitics of the layout by maximizing diffusion sharing and incorporating LDEs. The inputs to the algorithm are the analog cell netlist, listing the number of unit cells for each device, and the unit cell aspect ratio (K).

The algorithm can be explained using a current mirror bank. Fig. 4(a) shows a schematic of the example circuit consists of five devices, A, B, C, D, and E, whose multiplicity matrix $M = [2, 2, 4, 8, 8]$

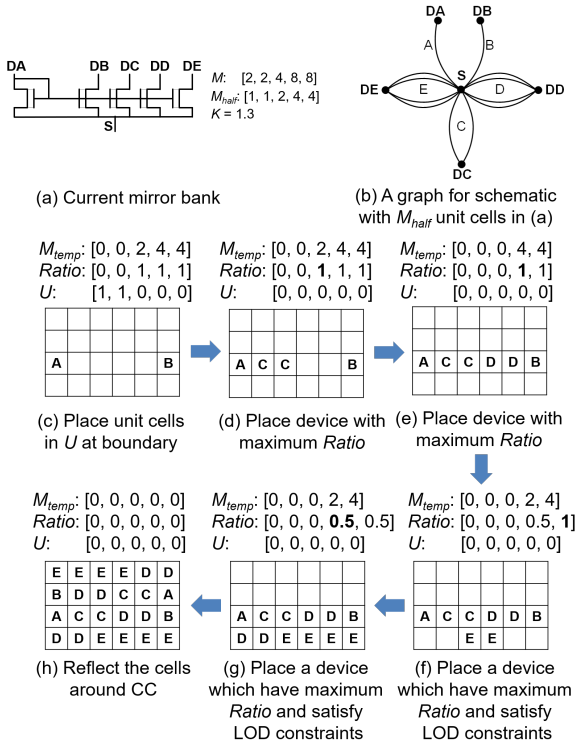


Figure 4: The application of the proposed common-centroid algorithm on a current mirror bank in (a). (b) is the graph for the schematic with M_{half} and (c)–(f) shows intermediate steps of the algorithm.

represents, in the same order, the number of unit cells of these five devices. The graph for the circuit is shown in Fig. 4(b). The algorithm proceeds through the following steps:

Step 1: Preprocessing First, the list of devices with an odd number of unit cells in M is stored in a list U (line 4). These odd unit cells will be divided into half-cells (i.e., cells with the same height as unit cells, but with half the active width compared to the unit cells). This transformation ensures that the number of unit cells is even for all devices, thus enabling CC layout. However, since these half-cells cannot share diffusion with other "full-cells" and must be placed at the edges of the layout matrix, X, we add them to a list, U, of cells that must be at the edge of X.

Next, the remaining unit cells are divided in two halves and stored in the list M_{half} (line 5). In the succeeding steps, we will first place the unit cells in M_{half} in the matrix X in the lower half of the array; when the matrix has an odd number of rows, the left half of the middle row is also populated. Later, in Step 5, we will reflect this placement to the other half of the matrix through the CC point.

To place this half, a graph $G(V, E)$ is created for the unit cells in M_{half} (line 7): note that the number of edges here is different from Fig. 3, which shows the graph for M. A graph for the current mirror bank testcase is shown in Fig. 4(b). Next, the unit cells with odd multiplicity in M_{half} are detected: these cells must lie at the end point of an Eulerian path and can only be placed without a diffusion break at the boundary of the CC placement matrix, X. All such cells are added to the list U (lines 8–14). This arises when an element of M_{half} is odd (i.e., it must be at an end point of an Eulerian path) and its source or drain has no other connections other than to the devices in M_{half}. For the current mirror bank testcase, the lists, M_{half} and U, are shown in Fig. 4(c).

Step 2: Aspect ratio calculation (lines 16–22) In this step, the number of rows and columns ($r \times q$) of the matrix X are calculated

Algorithm 1 Common-centroid placement of analog cells

1: **Input**: Analog cell netlist; Unit cell height to width ratio, K; Device size vector $M = [M_1, M_2, \cdots, M_N]$, where, M_i is the number of unit cells for device i
2: **Output**: Common-centroid placement X in an array of size $r \times q$.
3: *// Step 1: Preprocessing*
4: $U \leftarrow$ list of M_i with odd unit cells
5: $M_{half} \leftarrow [\lfloor M_1/2 \rfloor, \lfloor M_2/2 \rfloor, \cdots, \lfloor M_N/2 \rfloor]$
6: *// Create layout for half the matrix; Reflect other half around CC*
7: Create netlist graph $G(V, E)$, use M_{half} for edge multiplicities
8: **for** $i = 1$ to N **do** *// Over all devices*
9: **if** $M_{half}[i]$ is *odd* $\wedge$ [(deg($M_{i,source}$)==$M_{half}[i]$) $\vee$
10: (deg($M_{i,drain}$)==$M_{half}[i]$)] **then**
11: $U.add(M_i)$ *// Add odd unit cell of M_i in U*
12: $M_{half}[i] = M_{half}[i] - 1$
13: **end if**
14: **end for**
15: *// Step 2: Aspect ratio $r \times q$ calculation*
16: $r = Round\left(\sqrt{\sum M}/K\right)$ *// Row calculation*
17: **if** $len(U)$ is *odd* and r is *even* **then**
18: $r = r + 1$
19: **end if**
20: $q = \lceil \sum M/r \rceil$ *// Column calculation*
21: $q = 2\lfloor \frac{q+1}{2} \rfloor$ *// Make column even*
22: Calculate common-centroid point (C_X, C_Y) = ($\frac{r}{2}$, $\frac{q}{2}$)
23: *// Step 3: Placement of devices in List U*
24: **if** $len(U)$ is *odd* **then**
25: $X[C_X][C_Y] = U[1]$
26: $U.delete(1)$
27: **end if**
28: $t = len(U)$ *// length of list U*
29: $n \leftarrow 1$ *// Counter*
30: **for** $i = 0$ to $(\frac{t}{2} - 1)$ **do** *// Place cells in U at boundary*
31: $X[n * \frac{r}{2} - i][n] = U[1]$ and $X[n * \frac{r}{2} - i][q + 1 - n] = U[2]$
32: $U.delete(1, 2)$
33: **if** $(n * \frac{r}{2} - i - 1) == 0$ **then**
34: $n = n + 1$
35: **end if**
36: **end for**
37: *// Step 4: Placement of devices in List M_{half}*
38: Sort M_{half} in *ascending* order; set $M_{temp} = M_{half}$
39: **for** $i = \frac{r}{2}$ to 1 **do** *// Over half rows*
40: $Z = 1$ *// If $Z = 1$, a cell is placed at the left of CC*
41: $Z_l = C_Y$ and $Z_r = C_Y + 1$ *// Left and right counters*
42: **while** *Row i* is not filled **do**
43: $Ratio = [k/l$ for (k, l) in $(M_{temp}, M_{half})]$
44: **if** $Z == 1$ **then** *// Place cell at the left of the CC*
45: Select M_X which can share the diffusion region
46: with device at $X[i][Z_l + 1]$ and have maximum *Ratio*
47: **if** $M_X == X[j$ for j in $(\frac{r}{2}, \frac{r}{2} - 1, \cdots, i - 1)][Z_l]$ **then**
48: *// M_X already placed in the column*
49: Go to line 46 and select another M_X
50: *// To minimize LOD mismatch*
51: **end if**
52: $X[i][Z_l] = M_X$
53: $Z_l = Z_l - 1$ and $M_X = M_X - 1$
54: **else** *// Z=0, place cell at the right of the CC*
55: Select M_X which can share the diffusion region
56: with device at $X[i][Z_r - 1]$ and have maximum *Ratio*
57: **if** $M_X == X[j$ for j in $(\frac{r}{2}, \frac{r}{2} - 1, \cdots, i - 1)][Z_r]$ **then**
58: *// M_X already placed in the column*
59: Go to line 56 and select another M_X
60: *// To minimize LOD mismatch*
61: **end if**
62: $X[i][Z_r] = M_X$
63: $Z_r = Z_r + 1$ and $M_X = M_X - 1$
64: **end if**
65: **if** (r is *even* $\vee$ i ! $= \frac{r}{2}$) **then**
66: $Z = \overline{Z}$
67: **end if**
68: **end while**
69: **end for**
70: *// Step 5: Postprocessing*
71: Reflect the remaining half devices around *CC* in X
72: Calculate ΔV_{th}^{max} using (1) between two devices
73: Calculate # of dummies using (1) to ensure $\Delta V_{th}^{max} < \varepsilon \cdot V_{th}$

so that a near-square aspect ratio is obtained. The number of rows is calculated using line 16 and adjusted according to the unit cells in list U (lines 17–19): we will elaborate on this in Step 3. Finally, the number of columns and CC point (C_X, C_Y) are calculated (lines 20–22). For the current mirror bank testcase, the array size is 4×6, and CC point is at (3, 2).

Step 3: Placement of unit cells in U (lines 24–36) In this step, the unit cells in U are placed at the boundary in X. If the total number of unit cells is odd in U (i.e., length of U is odd), then one of the unit cells is placed at the center of the odd row without a diffusion break (lines 24–27); in Step 2, we had ensured that when the length of U is odd, the total number of rows is odd (lines 17–19).

The remaining unit cells in U are placed at the boundary of X (lines 28–36). For this, first, we initialize a counter n (line 29), which selects a column from the leftmost and rightmost ends of X for unit cell placement. Once the leftmost and rightmost columns are filled, the counter is increased and the next columns are selected (lines 33–35). For example, in the current mirror testcase there are two unit cells in U one each from device A and B. These are placed at the left and right boundary location as shown in Fig. 4(c). In this case, one column at each edge suffices; the role of the counter is to populate a second or third column, if necessary.

Step 4: Placement of unit cells in M_{half} The unit cells in M_{half} are sorted in ascending order and stored in M_{temp} (line 38), which represents the set of cells that are yet to be placed. Thereafter, the unit cells are placed over half of the rows, starting at line 39. These unit cells in each row are placed alternately at the left/right of the CC point. The starting location for the cells to be placed in a row is set by two variables Z_l and Z_r (line 41). Initially, Z_l and Z_r are set to C_Y and $C_Y + 1$, respectively (line 41). After placement of a unit cell at the left (right) of the CC point, Z_l (Z_r) is decreased (increased) by one and the location is updated. In other words, Z_l and Z_r move to the left/right of the CC location after a cell placement at the left/right of the CC. The cells from M_{temp} are then successively placed in a row until it gets filled (line 42).

The order in which the unit cells are populated into rows is based on the parameter, $Ratio$, that is computed for each device (line 43): this is the ratio of unplaced unit cells for that device in M_{temp} to the total number of unit cells M_{half}. The principle is that we choose a device for placement if, relatively speaking, a smaller fraction of its unit cells have been placed so far. This helps ensure better dispersion of the devices. Using this principle, the algorithm now selects a device from M_{temp} (if possible, that can share the diffusion region) and has maximum $Ratio$ (lines 46 and 56).

In each row, the method alternately places cells to the left and to the right of the CC point. The Boolean counter Z is used to enforce this by verifying whether it is 0 or 1. The exception to this alternation is when the total number of rows is odd and the CC placement occurs in the middle row: in this row, the cells are placed at the left of the CC only. As we will explain later, this left half-row will be reflected to right half-row in Step 5 about the CC point. Thus, the Boolean counter Z is inverted each time a unit cell is placed in a row, except when the total number of rows is odd and cells are placed in the middle row (lines 65–67). Moreover, to minimize LOD mismatch if the device has already been placed in the column (in a different row), other devices are prioritized over this one (lines 48–51 and 58–61)

For example, in the current mirror bank testcase, first, device C is selected: at this point no device can share the diffusion region, and the C is a device with the highest $Ratio$ value. Its placement in X is shown in Fig. 4(d). Thereafter, $Ratio$ is updated, and device D, which now has the largest value in $Ratio$, is placed as shown in the figure. At this point, the row is filled and we move to the next

row. The procedure is repeated until all cells are placed, as shown in Fig. 4(f)–(g).

Step 5: Postprocessing The algorithm, as explained so far, places half of the devices (in M_{half}) in the lower part array. The remaining half of the devices are reflected around the CC point in X. The reflection is carried out about a horizontal line through the CC point. If the number of rows is odd, an additional step is required for the row in the middle: its left half is mirrored on to the right half to create CC symmetry. This is illustrated in Fig. 4(h) (line 71).

Finally, the maximum threshold voltage mismatch between two devices, Δ_V^{max}, due to the LOD effect is calculated using (1). The SA/SB values for each unit cell are first calculated from the placement, and thereafter (1) is used to calculate Δ_V^{max}. The mismatch can be minimized using dummies on the left/right of X (this will increase SA and SB as shown in Fig. 2, and consequently will reduce Δ_V^{max}). To minimize Δ_V^{max} within $\varepsilon \cdot V_{th}$ (ε is a user-defined tolerance) the values of SA/SB are calculated using (1), and the required number of dummy unit cells on the left/right of X are calculated to meet the SA/SB criteria. WPEs are also best addressed through the use of dummy cells that ensure a minimum distance to the well edge.

C. EM and IR drop aware routing

Once placement is complete, our next task is to route the CC array. Due to interconnect bottlenecks in nanometer-scale analog circuits, CC layouts must be carefully routed to avoid performance degradation. As stated in Section II-C, both EM and resistive parasitics are important considerations in routing. For wires that carry substantial DC currents for extended periods, EM considerations must be factored in to control the current density in wires, particularly in lower metal layers. Moreover, especially in advanced technologies, high wire resistance in lower metal layers and high via resistance implies that IR drops along these lines can be high and/or mismatched, thus significantly shifting circuit performance metrics. Both effects can be reduced by identifying sensitive wires and using multiple parallel connections between nodes, thus effectively reducing resistance as well as the current density for EM.

Algorithm 2 describes a CC routing approach that is parasitic mismatch aware and EM-aware. To meet constraints from the current density limit or reduce the IR drop, the algorithm effectively widens wire widths to satisfy these constraints. In FinFET technologies, due to coloring rules, wire widening implies that multiple parallel wires must be used. IR drops along wires shift the transistor bias points and affect offset and matching. Since the current through a transistor depends on $(V_{GS} - V_{th} - V_{IR})$, we use the random V_{th} mismatch, which is an uncontrollable variation for a netlist with a specified number of unit cells, as a reference. We constrain the allowable IR drop to be a fraction of the random mismatch in V_{th}. Specifically, the standard deviation, $\sigma(\Delta V_{GS})$, is calculated as follows [26]:

$$\sigma^2(\Delta V_{GS}) = \sigma^2(\Delta V_{th}) + \frac{1}{(g_m/I)^2}\left(\frac{\sigma(\Delta\beta)}{\beta}\right)^2 \quad (2)$$

where the transistor $\beta = \mu C_{ox}(W/L)$, where all terms have their usual meanings. The maximum allowable IR drop, V_{IR}^{Max}, is defined through a user-specified ε' as:

$$V_{IR}^{Max} = \varepsilon' \cdot \sigma(\Delta V_{GS}) \quad (3)$$

The inputs to the routing algorithm are the circuit netlist, the list of terminals, a CC placement of unit cells from Algorithm 1, the fraction ε', bias current and voltages, per unit length resistance and EM constraints for routing layers, and process constants (e.g., A_{VT}, β, etc.). We assume that schematic simulation provides the bias voltages and the current per unit cell. We describe the routing algorithm and illustrate it on the testcase in Fig. 4.

Algorithm 2 Routing within analog cells

1: Input: A placement of transistors from Algorithm 1, netlist, terminal list (T), tolerable mismatch as a percentage of random mismatch (ε'), bias current and voltages, per unit length resistnace and EM constraints for routing layers, and process constants (e.g., A_{VT}, β, etc.)
2: Output: Routed layout of the analog cell
3: *// Step 1: Calculating the number of routing tracks*
4: Calculate vertical tracks (N_V) using (4)
5: Calculate horizontal tracks in a row (N_H) using (4)
6: *// Step 2: IR drop constraints*
7: Calculate V_{IR}^{Max} for source/gate terminals using (2)-(3)
8: Calculate V_{IR}^{Max} for other terminals as ε' percentage of bias volatge
9: *// Step 3: Vertical track assignment*
10: Sort terminal list T in descending order of # devices connected to the terminal
11: Represent unit cells as current sources and horizontal/vertical connections as resistances (an example is shown in Fig. 6)
12: $\delta = 0$ *// δ is the distance of the vertical track from the center*
13: **for** $k \in T$ **do**
14: Select vertical routing track at $Loc = \frac{N_V}{2} + \delta$
15: Set the *pin* location to be at the center of the vertical track
16: Determine the maximum IR drop from *pin*
17: Determine the IR drop and wire current in the
18: vertical track (V_V) and horizontal track (V_H) by solving
19: the resistive network when one track is used for routing
20: Calculate required # vertical tracks n_V using (6), (7)
21: Compute # of vertical tracks, n_V^{EM}, due to EM constraints
22: based on current in the wire (n_V^{EM})
23: $n_V = \max(n_V, n_V^{EM})$
24: $\delta = \delta + n_V/2$
25: $V\_list.add(k, n_V)$
26: **end for**
27: *// Step 4: Horizontal track assignment*
28: **for** $k \in T$ **do**
29: **for** $row_i \in \{1, \cdots row\}$ **do**
30: Determine the maximum $V_{H,i}$ in row_i
31: Find out V_V for row_i
32: Calculate required # horizontal tracks $n_{H,i}$ using (6)
33: Compute $ of vertical tracks, $n_{H,i}^{EM}$, due to EM constraints
34: based on current in the horizontal wire
35: $n_{H,i} = Max(n_{H,i}, n_{H,i}^{EM})$
36: $H\_list.add(k, row_i, n_{H,i})$
37: **end for**
38: **end for**

The core of the algorithm is highlighted in Fig. 5 for the terminal S of the testcase. Our overall scheme is to connect all cells of the same device in a row using horiztontal wires, and to connect cells across rows using vertical wires. First, we calculate the total number of vertical tracks (N_V) and horizontal tracks (N_H) in a row using Step 1. Thereafter, the maximum allowable IR drop for the terminals is calculated in Step 2. Next, an initial track assignment is carried out where single wires are used to connect all the unit cells to a terminal. This is shown for the terminal S of the testcase in Fig. 5(a). The IR drop constraints are satisfied by optimizing the number of parallel vertical/horizontal wires in Step 3 and 4. For example, vertical wires are assigned in Fig. 5(b)–(d) and horizontal wires are assigned in Fig. 5(e)–(f). Each routing metal layer is restricted to be unidirectional, which is common at advanced FinFET technology nodes due to lithography-driven considerations. We begin routing with (Metal2, Metal3) in the (Horizontal, Vertical) direction; further, higher metal layers are used if required. Next, we discuss each step.

Step 1: Calculating the number of tracks (lines 4–5) Given the placement, we first calculate N_V and N_H. Fig. 6(a) illustrates an array of size $W \times H$. If p_H and p_V are the pitches of horizontal and vertical wires, respectively, and R_H is the row height, which corresponds to the unit cell height, then

$$N_V = \frac{W}{p_V}\ ,\ N_H = \frac{R_H}{p_H} \quad (4)$$

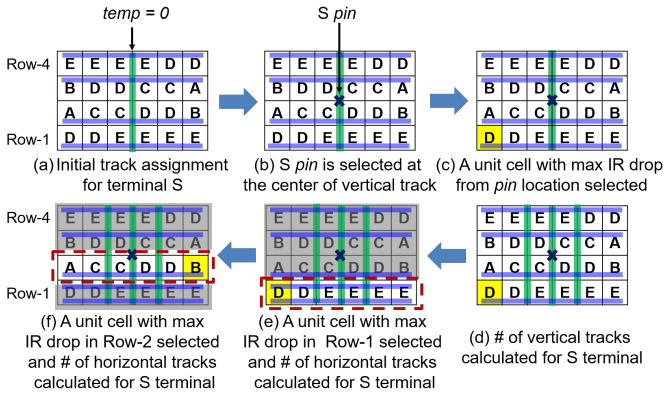


Figure 5: An illustration of vertical and horizontal routing for the S terminal in the testcase of Fig. 4.

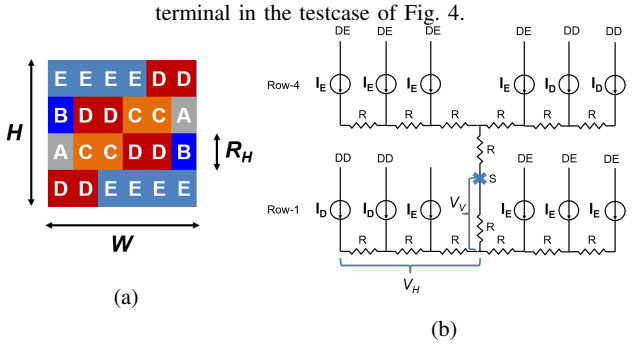


Figure 6: (a) A CC placement for the example in Fig. 4 and (b) its equivalent representation in terms of unit current sources and routing resistive parasitics of Row-1 and 4 for S terminal.

Step 2: IR drop constraints (lines 7–8) In this step, we calculate maximum allowable IR drop for the terminals/nodes. At a unit cell source/gate terminal, the maximum allowable voltage drop is computed using Equations (2) and (3), while at any other node, it is calculated as a small fraction of the bias voltages: we use the same multiplier ε' as in (3). These bias voltages are provided as inputs to this procedure and can be obtained from a SPICE simulation of the schematic netlist of the analog cell.

Step 3: Vertical track assignment (lines 10–26) We first sort the terminal list in descending order of the number of devices connected to the terminal, with ties broken arbitrarily. For example, the sorted list of terminals for the testcase in Fig. 4 is [S, DA, DB, DC, DD, DE]. Next, to calculate the maximum IR drop at a terminal/node, we represent the unit cells in the CC placement as current sources, and horizontal/vertical routing wire as a resistive network. The current per unit cell is an input to the procedure (e.g., in a current mirror it is input current per unit cell of reference device), and the parasitic resistances are estimated using the placement, the unit cell size, and the per unit length for the routing layers in the given technology.

For Fig. 4, an equivalent network in terms of unit current sources and resistances is shown in Fig. 6(b) (for simplicity, only Row 1 and Row 4 are shown for the S terminal only).

We begin routing vertical wires from the center outwards, and maintain a variable, δ, that corresponds to the distance of an available vertical track from the center; this is initialized to zero. In each step, we assign a vertical track as close to the center of the array as possible, in vertical track $\frac{N_V}{2} + \delta$ of the placement. and defines the location of a vertical track from the center. The track assignment alternates to the left and right of the CC point. In the example of Fig. 5, a vertical track is assigned for terminal S at the center (i.e., $\delta = 0$) as shown in Fig. 5(a). We also show horizontal tracks in each row for routing terminal S in the figure.

To minimize the maximum IR drop for a unit cell and maintain symmetry in routing, a *pin* for the terminal being routed is created at the center of the vertical track (line 15). This is shown in Fig. 5(b) for the terminal S. Next, we find the maximum IR drop from the *pin* location to a unit cell (line 19). For example, for *pin* S in the testcase, due to the structure of the connection, the devices at the edges of Rows 1 and 4 (i.e., the rightmost unit cell of E in Row 1 and D in Row 4, and the leftmost unit cell of device D in Row 1 and E in Row 4) will have the largest IR drop (we highlight the unit cell of device D in Fig. 5(c)). The IR drop from the pin to the unit cell can be divided into two parts: the drop, V_V, across the vertical track and the drop, V_H, across the horizontal track when a single wire/track is used for routing. We require that the total voltage drop,

$$V_V + V_H \leq V_{IR}^{Max} \tag{5}$$

If the criterion is not satisfied using a single wire, multiple wires must be used. For our optimization, we use the variables n_V and n_H to denote the number of vertical and horizontal wires, respectively. Temporarily dropping the requirement that n_V and n_H must be integers, this implies that to just satisfy the constraint, we require:

$$\frac{V_V}{n_V} + \frac{V_H}{n_H} = V_{IR}^{Max} \tag{6}$$

We add a second constraint that spreads the load of IR drop reduction evenly: if N_H is higher than N_V, a lower drop will be budgeted for N_H than for N_V (i.e., more wires, n_H, will be used). Quantifying this, we set the relative drop in the horizontal and vertical segments to be in inverse proportion to the number of available horizontal and vertical wires, i.e.,

$$\left(\frac{V_V}{n_V}\right) \Big/ \left(\frac{V_H}{n_H}\right) = \frac{N_H}{N_V} \text{ i.e., } \left(\frac{V_V}{n_V}\right) \cdot N_V = \left(\frac{V_H}{n_H}\right) \cdot N_H \tag{7}$$

We solve Eqs. (6) and (7) to find the optimal value of n_V (line 20). In most cases, this may be a fraction, and we round it to the nearest integer. Note that rounding will mean that the proportionality in (7) will not be perfect, but that is a guideline rather than a hard constraint. If n_V is rounded down, we will make up for the larger IR drop on vertical wires when we choose n_H in the next step. For the testcase as shown in Fig. 5(d), we obtain $n_V = 3$ for routing terminal S.

Further, the maximum current is calculated in the vertical tracks, and based on EM current density limit, the number of parallel vertical tracks to avoid EM (n_V^{EM}) are calculated. Finally, the larger of the two values, n_V^{EM} and n_V, is used for vertical routing (line 23). Now that this terminal is routed, a pin is added to the center of each vertical track. The variable δ is modified to represent that n_V vertical tracks are already assigned around the center track location, i.e., $\delta = \delta + n_V/2$, and the same procedure is repeated for each terminal. Some special handling is required for the case where n_V is odd/even, but this is a minor programming detail that is omitted from the psuedocode to keep it readable.

Step 4: Horizontal track assignment (lines 28–38) In this step, horizontal tracks are assigned for terminals in each row. Note that the value of n_H obtained from solving (6) and (7) is valid for the row with the maximum IR drop, but not for other rows. In this step, we compute n_H for all rows.

Row i has a total of $N_{H,i}$ horizontal tracks for routing (note that this number of tracks may be unequal after the first terminal is routed). For routing a terminal in the row, we find the maximum IR drop to any unit cell in the horizontal wire, i.e., $V_{H,i}$, for that

terminal (line 30). For example, in case of the terminal S, this drop corresponds to the leftmost unit cell of device D and the rightmost unit cell of E in Row 1 (the former is highlighted in Fig. 5(e)).

Next, the number of parallel horizontal wires in the row, i.e., n_H, to minimize V_H is calculated. We beging $V_{V,i}$ for this row i, which was computed earlier for the case where a single wire is used for the terminal (note that in general, this value will not be the same as the V_V value in Step 3, which was the worst case V_V over all rows): when we use n_V vertical wires (computed in Step 3), the voltage drop along the vertical wires becomes $V_{V,i}/n_V$. Based on this, and using $V_{V,i}$ and $V_{H,i}$, we use Eq. (6) to compute $n_{H,i}$, and round it to the next higher integer (line 32).

Finally, we use the maximum current in the horizontal wire to calculate the number of parallel wires, $n_{H,i}^{EM}$, in row i to make it EM-safe. As before, the larger of two values, $n_{H,i}$ and $n_{H,i}^{EM}$ is used for horizontal rooting in the row (line 35). This process is repeated for all rows. In the example, for routing terminal S, $n_V = 3$. We compute $n_{H,1} = 2$ for Row 1 and $n_{H,2} = 1$ for Row 2. Fig. 5(e)–(f) shows the routing for terminal S in these two rows.

IV. Results and Discussion

In analog circuits such as OTAs, comparators, and DACs, CC is required between transistor groups such as current mirrors and differential pairs. In this section we apply our CC placement and routing algorithms on a set of analog cells: current mirror banks and cascoded differential pairs. These algorithms are also valid for other analog cells that require CC layout – cross-coupled-pairs, differential and cascoded loads, etc. We compare our work with the algorithms presented in [15], [16] and highlight the advantages of our approach. We present a qualitative comparison for several circuit examples and also show the post-layout simulation results for a subset of these circuits.

A. Qualitative comparison

We begin by validating our CC placement and routing algorithm using testcases shown in Fig. 7–9. We compare our algorithms with the algorithms presented in [15], [16]. The results are compared based on five figures-of-merit (FOMs) that were discussed in Section I and II:

(1) Systematic variatoin tolerance: whether the placement is CC.
(2) Diffusion sharing: whether diffusion sharing is maximized.
(3) LDE: whether the placement considers the impact of LDE.
(4) Parasitic mismatch: whether parasitic mismatch is minimized.
(5) EM: whether EM constraints are addressed during routing.

In Fig. 7, four device current mirror bank (CMB) testcases are shown. A critical performance metric for a CMB is the current ratio, and it is degraded significantly by resistive parasitics: in our approach, to maintain the correct ratio, parasitics at the device terminals are reduced such that the IR drop is a small fraction ε' of the random V_{th} or the bias voltage (Section (III-C)). For a CMB, it is important for V_{th} mismatch due to LDEs to be low, and we incorporate this during placement by positioning the devices appropriately and adding dummies where necessary. Finally, diffusion sharing is also important for CMBs as it reduces area and output capacitance, which will be critical for high-speed designs.

Three different examples of **four-device CMBs** are used for the first comparison, shown in Fig. 7(b)–(d). The placements employing our proposed approach and algorithms in [15], [16] are also shown in the figure. We compare the five FOMs for these approaches. All approaches use *CC* placement and routing, and hence the results are tolerant to systematic variations. However, unlike our method, none of the other approaches considers *EM* during routing and they are

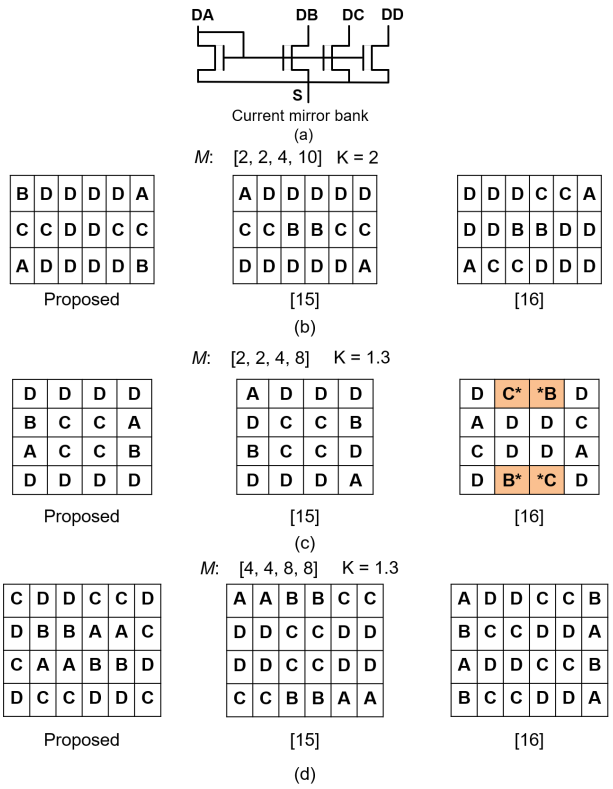


Figure 7: Four-device current mirror bank testcase: Comparing the proposed algorithm with the results of [15], [16]. Shaded cells show locations where diffusion breaks must be inserted.

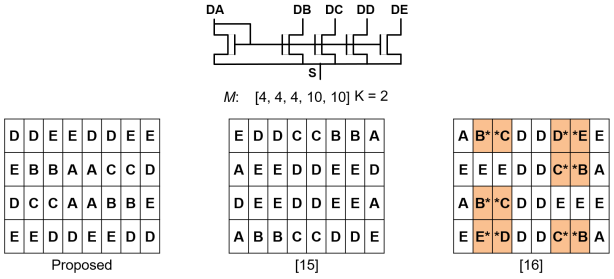


Figure 8: Five-device current mirror testcase: Comparing the proposed algorithm with the results of [15], [16]. Shaded cells show locations where diffusion breaks must be inserted.

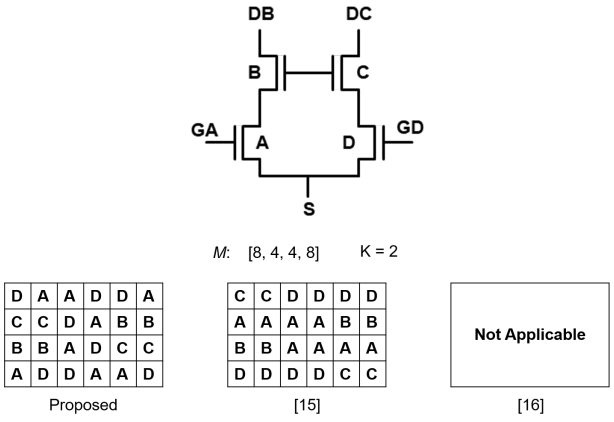


Figure 9: Cascoded differential pair testcase: Comparing the proposed algorithm with the results of [15], [16].

Testcase	Systematic variations			Diffusion sharing			LOD			Parasitic mismatch			EM		
	Our work	[15]	[16]	Our work	[15]	[16]	Our work	[15]	[16]	Our work	[15]	[16]	Our work	[15]	[16]
Fig. 7(b)	✓	✓	✓	✓	✓	✓	✓	×	×	✓	×	×	✓	×	×
Fig. 7(c)	✓	✓	✓	✓	✓	×	✓	×	×	✓	×	×	✓	×	×
Fig. 7(d)	✓	✓	✓	✓	✓	✓	✓	×	×	✓	×	×	✓	×	×
Fig. 8	✓	✓	✓	✓	✓	×	✓	×	×	✓	×	×	✓	×	×
Fig. 9	✓	✓	NA	✓	✓	NA	✓	×	NA	✓	×	NA	✓	×	NA

Figure 10: Comparison of cell level testcases for the proposed CC algorithm.

therefore liable to face reliability issues for these high-current analog circuits in advanced FinFET technologies. In all figures, diffusion breaks are shown by shaded cells.

• Fig. 7(b) shows the case where $M = [2, 2, 4, 10], K = 2$. For this case, all three layouts are successful in maximizing *diffusion sharing*. However, devices A and B, which have an equal number of unit cells, will see *LDE* and *parasitic* mismatch for the placements using [15], [16], as illustrated in Fig. 7(b). These will result in current mismatch, which cause the current ratio to deviate from its nominal value.

• Fig. 7(c) shows placements for inputs $M = [2, 2, 4, 8], K = 1.3$. In this case, the placement generated using our approach and that in [15] maximize *diffusion sharing*, whereas the placement using [16] has *diffusion breaks* between devices B and C. This will result in an increase in area and parasitics at the corresponding nodes. Moreover, the placement using [15] results in a *parasitic mismatch* between devices A and B, and the placement using [16] has *parasitic* and *LDE* mismatch, and will consequently see a current ratio mismatch.

• In Fig. 7(d), with $M = [4, 4, 8, 8], K = 1.3$, all three placements maximize *diffusion sharing*. However, the placement generated using [15] has *parasitic* and *LDE* mismatch, especially between the devices with the same number of unit cells. In [16] *parasitic* mismatch between devices A, B, C and D exists.

Fig. 8 shows a **five device CMB** testcase and its placements generated using the proposed technique and the algorithms in [15], [16]. The placements using [15], [16] both have *LDE* and *parasitic mismatch*, and *EM* is not taken into account. In our approach, *LDE* mismatch is minimized further by adding one column of unit cells corresponding to dummy transistors on the left and right sides, and *parasitic mismatch* is intentionally minimized during routing. Moreover, the placement generated using [16] does not maximimize *diffusion sharing*, and has diffusion breaks.

Finally, Fig. 9 shows placements of a **cascoded differential pair**, obtained by employing our proposed approach and that used in [15]. The algorithm presented in [16] is specific to current mirrors, and is not applicable for this testcase. For this circuit, the absolute and mismatch parasitic resistances are critical as they impact the effective transconductance (G_m) and input offset. We consider this by limiting the value of resistive parasitics. As before, diffusion sharing remains important as it reduces output capacitance. EM considerations are particularly important for this circuit as it draws high currents.

Fig. 10 summarizes the quality of placements of the testcases generated using our approach and those presented in [15], [16]. We can see that our algorithm results in the best overall placements.

B. Post-layout simulation results

We also perform post-layout simulation for these testcases using a commercial 12nm FinFET process and list the results in Table 1. For the CMB testcases, the first two rows present the current ratios without and with considering LDEs: this analysis isolates LDE mismatch from parasitic mismatch in placement and routing.

Table I: Comparison of post-layout simulations results for two CMBs and a cascoded differential pair (DP) with and without LDE (w/o LDE)

Specification	Proposed	[15]	[16]
CMB1 [Schematic: Current ratio (CR)= 2: 2: 4: 8]			
CR (w/o LDE)	2: 2: 4: 7.98	2: 2:01: 4.02: 8.02	2: 1.98: 3.98: 7.96
CR (with LDE)	2: 2: 4: 7.98	2: 2.01: 3.67: 7.66	2: 1.99: 3.99: 7.60
Max deviation (%)	-0.25	-4.25	-5.0
Max IR drop (mV)	1.8	3.7	4.0
CMB2 [Schematic: Current ratio (CR) = 4: 4: 4: 10: 10]			
CR (w/o LDE)	4:4:4:9.97:9.97	4:3.94:3.95:10.06:10.12	4:4:4.05:10.06:10.05
CR (with LDE)	4:3.97:3.97:9.98:9.98	4:3.64:3.61:9.39:9.48	4:4.02:4.11:9.61:10.10
Max deviation (%)	-0.75	-9.75	-3.90
Max IR drop (mV)	5.7	18.5	19.5
Cascoded DP [Schematic: G_m = 825uA/V; Offset = 0V; Output capacitance (C) = 0.8fF]			
G_m(uA/V) (w/o LDE)	943	921	NA
G_m(uA/V) (with LDE)	809	790	NA
Offset (uV) (w/o LDE)	8	850	NA
Offset (uV) (with LDE)	7	897	NA
C (fF) (w/o LDE)	1.15	1.27	NA
C (fF) (with LDE)	1.17	1.27	NA

The current ratios with LDEs show the impact of variations due to LDEs. The table also highlights the maximum percentage deviation in current ratio compared to the ideal value and the maximum IR drop at the source node in the layouts: here again, it can be seen that our approach outperforms prior methods.

CMB1 corresponds to the four-device CMB testcase in Fig. 7(c) for our approach and the methods in [15], [16]. Our solutions uses three parallel wires at the source node to minimize parasitics and one column of unit cells corresponding to dummy transistors on the left and right sides. Our work achieves a current ratio that is close to the ideal value with and without LDE considerations.The maximum deviation of the current ratio from its ideal value for a device in the analog cell is tabulated. In this case, the actual ratio for device D is 7.98 which is -0.25% of the ideal value, 8. For the layouts generated using the algorithms in [15], [16], the maximum deviations are -4.25% and -5.0%, respectively. We observe that the maximum IR drop using our approach is low due to the quality of our routing solution: 1.8 mV compared to the other techniques, where the drops are 3.7 mV and 4.0 mV, respectively.

CMB2 was laid out with the placements shown in Fig. 8. Our work uses five parallel wires at the source node and one column of unit cells on the left and right sides, corresponding to a dummy transistor. The maximum deviation achieved is significantly less in our work, -0.75%, compared to [15], [16] where it is -9.75% and -3.9%, respectively. The maximum IR-drop is also $3\times$ lower.

For the cascoded differential pair (DP) in Fig. 9, we compare the input-referred offset, G_m, and output capacitance. While our G_m and output capacitance are in a similar range as [15], the offset is greatly improved. Moreover, the offset remains robust in the present of LDEs, unlike [15], which sees significant degradation.

The runtime of our CC placement and routing algorithms for CMB1, CMB2, and cascoded DP are 1.27ms, 3.79ms, and 3.36ms, respectively. This has been measured on a RedHat system with an Intel(R) Xeon(R) Silver 4114 CPU @2.20 GHz and 20 cores.

V. Conclusion

We have presented generalized constructive common-centroid placement and routing algorithms for transistor arrays or analog cells e.g., differential pair, current mirror bank, cascoded cell, etc. The placement algorithm maximize the diffusion sharing as well as optimize the layout pattern considering layout dependent effects (LDEs). The proposed routing algorithm handles constraints such as parasitics matching, current density in interconnects (to avoid EM and resistive degradation), which are critical to analog circuits in FinFET technologies. We have validated the method on a set of analog cells and demonstrated improvements over prior methods.

REFERENCES

[1] S. Borkar, *et al.*, "Parameter variations and impact on circuits and microarchitecture," in *Proc. DAC*, pp. 338–342, 2003.

[2] K. Qian, *Variability modeling and statistical parameter extraction for CMOS devices*. PhD thesis, University of California–Berkeley, Berkeley, CA, 2015.

[3] A. Tchikoulaeva, *et al.*, "ACLV degradation: Root cause analysis and effective monitoring strategy," in *Proc. SPIE*, vol. 7028, pp. 373–382, 2008.

[4] D. J. Frank, *et al.*, "Monte Carlo modeling of threshold variation due to dopant fluctuations," in *Proc. Int. Symp. on VLSI Circ.*, pp. 171–172, 1999.

[5] P. Oldiges, *et al.*, "Modeling line edge roughness effects in sub 100 nanometer gate length devices," in *Proc. SISPAD*, pp. 131–134, 2000.

[6] M. J. Pelgrom, *et al.*, "Matching properties of MOS transistors," *IEEE J. Solid-St. Circ.*, vol. 24, no. 5, pp. 1433–1439, 1989.

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

[8] M. Dessouky and D. Sayed, "Automatic generation of common-centroid capacitor arrays with arbitrary capacitor ratio," in *Proc. DATE*, pp. 576–580, 2002.

[9] C. W. Lin, *et al.*, "Mismatch-aware common-centroid placement for arbitrary-ratio capacitor arrays considering dummy capacitors," *IEEE T. Comput. Aid. D.*, vol. 31, no. 12, pp. 1789–1802, 2012.

[10] F. Burcea, *et al.*, "A new chessboard placement and sizing method for capacitors in a charge-scaling DAC by worst-case analysis of nonlinearity," *IEEE T. Comput. Aid. D.*, vol. 35, no. 9, pp. 1397–1410, 2015.

[11] P.-Y. Chou, *et al.*, "Matched-routing common-centroid 3-D MOM capacitors for low-power data converters," *IEEE T. VLSI Syst*, vol. 25, no. 8, pp. 2234–2247, 2017.

[12] C. C. McAndrew, "Layout symmetries: Quantification and application to cancel nonlinear process gradients," *IEEE T. Comput. Aid. D.*, vol. 36, no. 1, pp. 1–14, 2016.

[13] V. Borisov, *et al.*, "A novel approach for automatic common-centroid pattern generation," in *Proc. SMACD*, pp. 1–4, 2017.

[14] M. P.-H. Lin, *et al.*, "Thermal-driven analog placement considering device matching," *IEEE T. Comput. Aid. D.*, vol. 30, no. 3, pp. 325–336, 2011.

[15] D. Long, *et al.*, "Optimal two-dimension common centroid layout generation for MOS transistors unit-circuit," in *Proc. ISCAS*, pp. 2999–3002, 2005.

[16] P. H. Wu, *et al.*, "Parasitic-aware common-centroid FinFET placement and routing for current-ratio matching," *ACM T. Des. Automat. El.*, vol. 21, no. 3, p. 39, 2016.

[17] T. Dhar, *et al.*, "ALIGN: A system for automating analog layout," *IEEE Des. Test*, vol. 38, no. 2, pp. 8–18, 2020.

[18] S. Müler and L. Schüler, "GeoStat-Framework/GSTools," April 14 2020. https://doi.org/10.5281/zenodo.1313628.

[19] A. K. Sharma, *et al.*, "Common-centroid layouts for analog circuits: Advantages and limitations," in *Proc. DATE*, 2021.

[20] J. V. Faricelli, "Layout-dependent proximity effects in deep nanoscale CMOS," in *Proc. CICC*, pp. 1–8, 2010.

[21] M. G. Bardon, *et al.*, "Layout-induced stress effects in 14nm & 10nm FinFETs and their impact on performance," in *Proc. IEEE Symp. on VLSI Technology*, pp. T114–T115, 2013.

[22] T. Hook, *et al.*, "Lateral ion implant straggle and mask proximity effect," *IEEE Trans. Electron Devices*, vol. 50, no. 9, pp. 1946–1951, 2003.

[23] K. W. Su, *et al.*, "A scaleable model for STI mechanical stress effect on layout dependence of MOS electrical characteristics," in *Proc. CICC*, pp. 245–248, 2003.

[24] P. G. Drennan, *et al.*, "Implications of proximity effects for analog design," in *Proc. CICC*, pp. 169–176, 2006.

[25] S. Yang, *et al.*, "10nm high performance mobile SoC design and technology co-developed for performance, power, and area scaling," in *Proc. Int. Symp. on VLSI Circ.*, pp. T70–T71, 2017.

[26] P. R. Kinget, "Device mismatch and tradeoffs in the design of analog circuits," *IEEE J. Solid-St. Circ.*, vol. 40, no. 6, pp. 1212–1224, 2005.