Device Layer-Aware Analytical Placement for Analog Circuits

Biying Xu
University of Texas at Austin
biying@utexas.edu

Shaolan Li
University of Texas at Austin
slliandy@gmail.com

Chak-Wa Pui
The Chinese University of Hong Kong
cwpui@cse.cuhk.edu.hk

Derong Liu
Cadence Design Systems, Inc.
derong@cadence.com

Linxiao Shen
University of Texas at Austin
lynn.shenlx@utexas.edu

Yibo Lin
University of Texas at Austin
yibolin@utexas.edu

Nan Sun
University of Texas at Austin
nansun@mail.utexas.edu

David Z. Pan
University of Texas at Austin
dpan@ece.utexas.edu

ABSTRACT

The layouts of analog/mixed-signal (AMS) integrated circuits (ICs) are dramatically different from their digital counterparts. AMS circuit layouts usually include a variety of devices, including transistors, capacitors, resistors, and inductors. A complicated AMS IC system with hierarchical structure may also consist of pre-laid out subcircuits. Different types of devices can occupy different manufacturing layers. Therefore, during the layout stage, the devices require co-optimization to achieve high circuit performance. Leveraging the fact that some devices can be built by mutually exclusive layers, they can be carefully designed to overlap each other to effectively reduce the total area and wirelength without degrading the circuit performance. In this paper, we propose an analytical framework to tackle the device layer-aware analog placement problem. Experimental results show that on average the proposed techniques can reduce the total area and half-perimeter wirelength by 9% and 23%, respectively. To verify the routability of the placement results, we also develop an analog global router, which demonstrates that the device layer-aware placement can achieve 18% shorter wirelength during global routing.

ACM Reference Format:
Biying Xu, Shaolan Li, Chak-Wa Pui, Derong Liu, Linxiao Shen, Yibo Lin, Nan Sun, and David Z. Pan. 2019. Device Layer-Aware Analytical Placement for Analog Circuits. In *ISPD '19: 2019 International Symposium on Physical Design, April 14–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3299902.3309751

1 INTRODUCTION

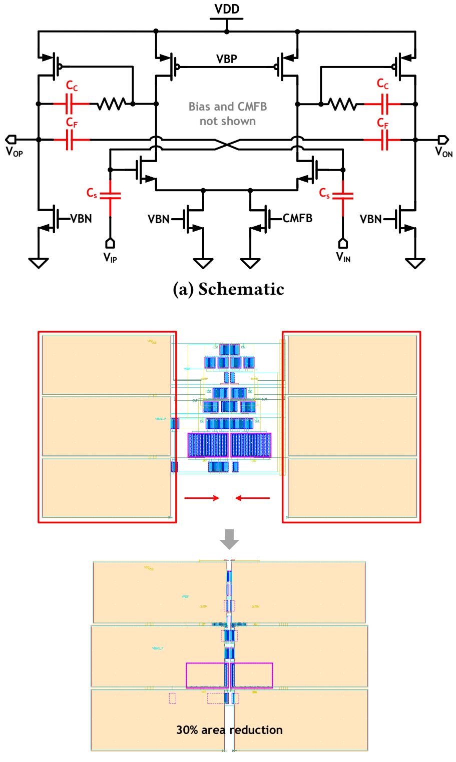
Analog/mixed-signal (AMS) circuits often contain various types of devices, including transistors, resistors, capacitors, and inductors. In a complicated AMS IC system with hierarchical design, there may also exist pre-laid out subcircuits as placement devices (e.g., a pre-laid out comparator in an analog-to-digital data converter system). Different types of devices are built by different manufacturing layers: transistors and resistors can reside only on the substrate and polysilicon layers; high-quality capacitors like metal-oxide-metal capacitors can be directly formed by inter-digitized metal fingers [1]; pre-laid out subcircuits containing different types of devices and the interconnections may occupy substrate, polysilicon, and metal layers. To reduce production cost and routing complexity, it has been a common and desirable practice to create overlapping layouts for the devices occupying mutually exclusive manufacturing layers in the high-performance custom analog designs. For instance, implementation examples of overlapping decoupling capacitors on top of transistors and resistors are reported in [2, 3] to address the area and wirelength overhead.

From the above description, we can see that under most circumstances, it is beneficial to allow the overlap between devices with mutually exclusive layers during analog layout optimization. However, there are some devices that should not overlap other devices even if they occupy mutually exclusive layers, e.g., the devices that are critical and sensitive to coupling. Only those that are insensitive to coupling and reside on mutually exclusive layers should be allowed to overlap each other without degrading the circuit performance. To validate that such overlapping will not induce circuit performance degradation, without losing generality, we provide two widely used analog circuits and their layout examples, with and without overlapping between the insensitive devices on different manufacturing layers in the following.

The capacitive-coupled operational transconductance amplifier (CC-OTA) is a prevalent building block in data converters and sensor interfaces, which is shown in Figure 1a. By overlapping the capacitors with the transistors and resistors as in Figure 1b, the area and wirelength are reduced by 30% and 4%, respectively. Table 1 compares the post-layout-simulated phase margin (stability metric), unity gain bandwidth (speed metric) and loop gain (accuracy metric) between the two layout cases, which shows that the impact on the circuit performance is negligible.

The current-controlled ring oscillator (CCO), which is commonly seen in phase-locked loops (PLLs), is shown in Figure 2. By strategically sharing the vertical space between the loading capacitors (C_L) and other devices, the compactness of the design is notably

ISPD '19, April 14–17, 2019, San Francisco, CA, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6253-5/19/04…$15.00
https://doi.org/10.1145/3299902.3309751



(a) Schematic

(b) Layouts (top: non-overlap; bottom: overlap)

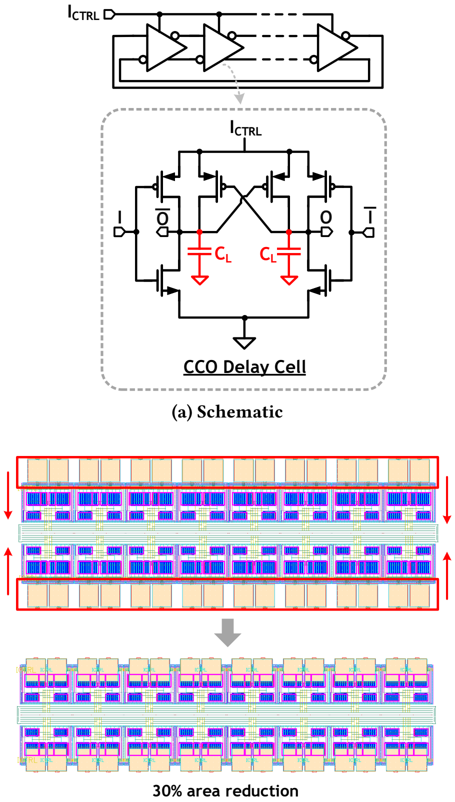
Figure 1: CC-OTA circuit schematic and layout examples.

Table 1: Post-layout simulation results of the CC-OTA circuit.

Layout	Phase Margin (deg.)	Unity Gain Bandwidth (MHz)	Loop Gain (dB)
non-overlap	71.9	103.7	36.3
overlap	71.5	105.4	36.3

Table 2: Post-layout simulation results of the CCO circuit.

Layout	f_{CCO} (kHz)	k_{CCO} (THz/A)
non-overlap	609	0.89
overlap	610	0.9



(a) Schematic

(b) Layouts (top: non-overlap; bottom: overlap)

Figure 2: CCO circuit schematic and layout examples.

improved, leading to 30% area reduction and 20% wirelength decrease. As shown in Table 2, the CCO center frequency (f_{CCO}) and tuning gain (k_{CCO}) remain almost unchanged, indicating that the dynamics of the upper system (e.g., the PLL bandwidth and tuning range) will not be affected.

From the above examples, we know that during layout optimization, overlapping the devices that are insensitive to coupling and reside on mutually exclusive layers can result in area and wirelength benefits without degrading the circuit performance. However, existing analog placement algorithms are still limited to consider complex scenarios and characteristics unique to analog designs. Although there exists previous work considering different cell layers in digital placement [4], none of the prior work on analog placement [5–11] considered the possibility of device bounding box overlapping. This flexibility brought by the device layer-aware layout scheme leads to a dramatically different placement methodology for AMS circuits, which can contribute to better layout quality. Moreover, previous works [12–14] have shown that analytical placement methods in ASICs can also achieve high quality results in both FPGA and analog circuit designs. In this paper, we will address the device layer-aware analog placement problem by using an analytical approach. The main contributions are summarized as follows:

- To the best of our knowledge, this is the first work on analog placement to consider overlapping between the devices that are insensitive to coupling and built on mutually exclusive layers, which offers high flexibility for layout optimization.
- A holistic analytical framework is presented to solve the device layer-aware analog placement problem.
- An analog global router is developed to verify the routability of our device layer-aware placement results.

The rest of this paper is organized as follows: Section 2 gives the definition of the device layer-aware analog placement problem. Section 3 details the algorithms and techniques to solve the problem. Section 4 shows comprehensive sets of experimental results. Finally, Section 5 concludes the paper.

2 PROBLEM DEFINITION

The conventional analog placement problem usually tries to minimize the objectives including the total area and the total half-perimeter wirelength (HPWL). Besides the non-overlapping constraint between each pair of devices, it also needs to satisfy many other constraints, including matching, proximity group, and symmetry constraints [5–11, 15]. The conventional analog circuit placement problem can be stated as follows:

Problem 1 (Analog Placement)**.** Given a netlist and the layout constraints (e.g., symmetry constraints), the analog placement problem is to find a legal placement of the devices satisfying all the given constraints, such that the objectives are optimized, including total area and wirelength.

Compared to conventional analog placement, device layer-aware analog placement allows overlap between certain pairs of devices. Without loss of generality, we categorize the devices in analog circuits into three types:

- *Type I device*: the device built without metal or via layers, and not sensitive to coupling.
- *Type II device*: the device built only with metal and via layers, and not sensitive to coupling.
- *Type III device*: the device occupying not only the metal and via layers but also substrate and polysilicon layers, or the device that is critical and sensitive to coupling.

From the above definitions, we know that Type I devices are allowed to overlap with Type II devices, while Type III devices should not overlap with any other devices. Overlaps between devices of the same type are also considered illegal. The device layer-aware analog circuit placement problem can be stated as follows:

Problem 2 (Device Layer-Aware Analog Placement)**.** The device layer-aware analog placement problem is the analog placement problem where devices built by mutually exclusive manufacturing layers and insensitive to coupling are allowed to overlap each other from the designer specification.

3 DEVICE LAYER-AWARE PLACEMENT

This section presents our method to solve the device layer-aware placement for analog circuits. Figure 3 shows the overall flow of our device layer-aware analytical analog placement engine. Given the analog circuit netlist, the placement constraints (e.g., symmetry constraints), the placement boundary, and the device types from the design specification (i.e., circuit designer will manually specify whether a device is Type I, II, or III, as input to our engine), we first generate the global placement result by optimizing a non-linear objective function using conjugate gradient (CG) method. The next step runs the symmetry-aware and device layer-aware legalization to generate a legal placement solution which honors the result from global placement. Finally, a linear programming (LP) based detailed placement is used to further optimize the wirelength.

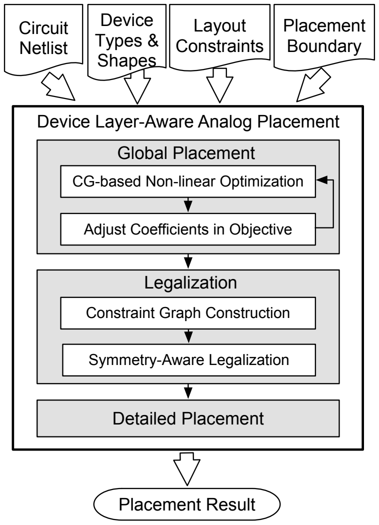

Circuit Netlist
Device Types & Shapes
Layout Constraints
Placement Boundary
Device Layer-Aware Analog Placement
Global Placement
CG-based Non-linear Optimization
Adjust Coefficients in Objective
Legalization
Constraint Graph Construction
Symmetry-Aware Legalization
Detailed Placement
Placement Result

Figure 3: The overall flow of our device layer-aware analytical analog placement engine.

3.1 Global Placement

Our global placement for analog circuits is based on a non-linear global placement framework [14, 16, 17], which simultaneously considers the following: (1) wirelength, (2) device layer-aware overlapping, (3) placement boundary, and (4) symmetry constraints from the design specification. To be specific, it minimizes the objective shown in Equation (1) using unconstrained non-linear conjugate gradient algorithm.

$$Objective = f_{WL} + a \cdot f_{OL} + b \cdot f_{BND} + c \cdot (f^{x}_{SYM} + f^{y}_{SYM}), \quad (1)$$

where f_{WL} is the wirelength of the placement, f_{OL} is the illegal overlap penalty (overlaps between devices on conflicting manufacturing layers are regarded as illegal), f_{BND} and f_{SYM} are the penalties of violating boundary and symmetry constraints, respectively. Our non-linear optimization-based global placement runs iteratively, until the penalties are below the specified thresholds, or the predefined maximum number of iterations is reached. By gradually adjusting the coefficient values of different penalty functions in each iteration, we can get a placement result honoring symmetry and boundary constraints with short wirelength and small illegal overlapping. Log-sum-exponential (LSE) [18] models $\gamma \log \sum_i \exp(x_i/\gamma)$ and $-\gamma \log \sum_i \exp(-x_i/\gamma)$ are used to smooth the $\max_i(x_i)$ and $\min_i(x_i)$ functions in the objective, respectively, where γ is a very small value. Details of each objective function will be explained as follows.

Wirelength is defined as the total HPWL shown in Equation (2).

$$f_{WL} = \sum_{n_k} (\max_{i \in n_k}(x_i) - \min_{i \in n_k}(x_i) + \max_{i \in n_k}(y_i) - \min_{i \in n_k}(y_i)), \quad (2)$$

where device i contains the pins of net n_k and x_i (y_i) is the x (y) coordinate of the center of device i. This definition assumes that the pins are in the center of the device.

In this work, only the overlaps between devices with conflicting manufacturing layers will contribute to the overlap penalty in the objective. Overlaps in global placement are modeled as an area overlap function similar to [19], which is shown in Equation (3).

$$\begin{aligned} f_{OL} &= \sum_{(i,j) \in L} O^{x}_{i,j} \cdot O^{y}_{i,j}, \\ O^{x}_{i,j} &= \max(\min(x_i + w_i - x_j, x_j + w_j - x_i, w_i, w_j), 0), \\ O^{y}_{i,j} &= \max(\min(y_i + h_i - y_j, y_j + h_j - y_i, h_i, h_j), 0), \end{aligned} \quad (3)$$

where L is the set of device pairs whose overlapping are illegal, $O^x_{i,j}$ ($O^y_{i,j}$) is the x (y) directional overlap length, x_i (y_i) and w_i (h_i) are the x (y) coordinate of the lower-left corner and width (height) of device i, respectively.

Besides wirelength and overlapping, symmetry constraints are also considered. Symmetry constraint requires: (1) each symmetric pair of devices within the same symmetric group to be symmetric to each other with respect to the same symmetric axis; (2) the self-symmetric devices to be self-symmetric with respect to the same axis as the symmetric pairs. Hence, the penalty for the violation of symmetry constraint on horizontal direction (with a vertical symmetric axis) is shown in Equation (4), and the vertical one can be calculated similarly.

$$f^x_{SYM} = \sum_{g_k \in G} \Big(\sum_{(i,j)\in g^p_k} ((x_i+x_j-2\cdot x^c_k)^2+(y_i-y_j)^2)+\sum_{i\in g^s_k}(x_i-x^c_k)^2\Big), \tag{4}$$

where G is the set of symmetric groups, g^p_k and g^s_k are the set of symmetric pairs and the set of self-symmetric devices in a symmetric group g_k, respectively, x_i (y_i) is the x (y) coordinate of the center of device i, and x^c_k is the coordinate of the symmetric axis of the symmetric group g_k.

For a given analog design, boundary constraint is usually imposed to control certain circuit area and aspect ratio. In our global placement, given a whitespace ratio and aspect ratio, we can get a desirable placement bounding box for the design. The penalty for the violation of the boundary constraint is shown in Equation (5).

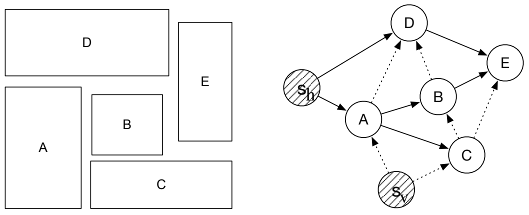
$$f_{BND} = \sum_{i\in D} \begin{aligned}&(\max(x_L - x_i, 0) + \max(x_i + w_i - x_H, 0)\\ &+ \max(y_L - y_i, 0) + \max(y_i + h_i - y_H, 0))\end{aligned}, \tag{5}$$

where D is the set of devices, x_i (y_i) and w_i (h_i) are the x (y) coordinate of the lower-left corner and width (height) of device i, respectively, and x_L (y_L) and x_H (y_H) are the x (y) coordinates of the lower and higher boundaries of the given placement bounding box.

3.2 Legalization

After global placement, a placement result with good wirelength, a small number of violations of overlaps, boundary constraint, and symmetry constraint is obtained. To get a placement without any illegal overlapping and violations of symmetry constraints, we first construct the constraint graphs with minimum edges. Then, given the constraint graphs, we legalize the global placement result using LP-based compaction.

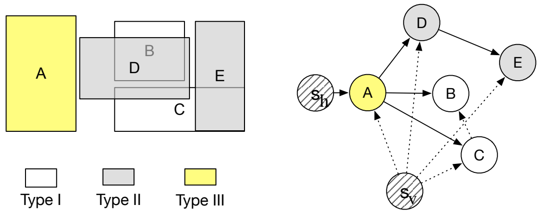
3.2.1 Constraint Graph Construction. Constraint graphs are directed acyclic graphs which impose positional constraints on the devices, including horizontal and vertical constraint graphs. Each node represents a device and an edge between two nodes imposes a positional constraint on the corresponding devices. For example, in the horizontal constraint graph, if there is an edge e_{ij} from node i to node j, device i should be on the left of device j. An example placement and its corresponding constraint graphs are shown in Figure 4. In Figure 4b, the horizontal and vertical constraint graphs are merged into a single graph, where the solid edges represent the edges in the horizontal constraint graph, and the dashed ones are vertical constraint edges. The nodes s_h and s_v are the virtual source nodes in the horizontal and vertical constraint graphs, respectively, which indicate the leftmost and bottommost coordinates of the placement.



(a) placement example (b) the constraint graphs

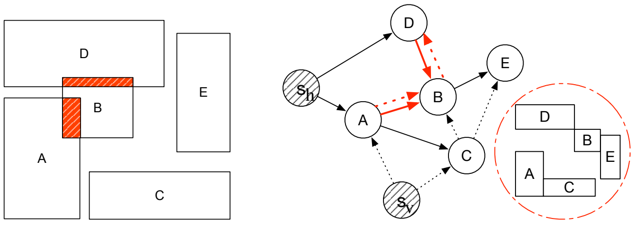
Figure 4: Example placement and constraint graphs.

To represent a legal placement, each pair of devices must have positional relationships in either vertical or horizontal constraint graphs, except between Type I and Type II devices. However, with more edges than necessary, extra constraints may lead to a larger placement area or longer run-time. Therefore, we construct the constraint graphs such that the number of edges is minimized and it can guarantee the placement result to be legal.



(a) (b)

Figure 5: Placement with different device layers and its constraint graphs.

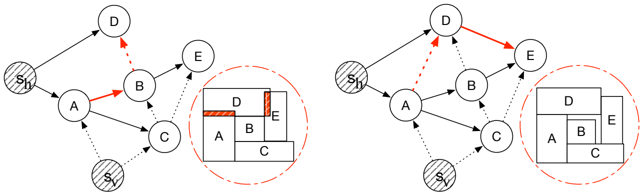


(a) global placement result (b) resulting constraint graphs and legalized placement snapshot

Figure 6: An example of the global placement result, and the constraint graphs resulted from directly applying the plane sweep algorithm as in [20].

First, our irredundant constraint graph construction algorithm is based on the plane sweep algorithm presented in [20]. However, this algorithm does not consider different device layers. To address device layer-awareness, we modify the plane sweep algorithm to be executed in two passes to avoid imposing non-overlapping constraints for the devices on mutually exclusive layers. In the first pass, the inputs are the bounding boxes of the Type I and III devices after global placement, while in the second pass, the inputs are those of the Type II and III devices. Therefore, no constraint edge will be added between Type I and Type II devices. Nevertheless, it will maintain the other necessary constraint edges. A placement example with different device layers and the resulting constraint graphs after running our two-pass procedure are shown in Figure 5, where devices B and C are Type I devices, D and E are Type II devices, and A is a Type III device. In the first pass, edges among devices $\{A, B, C\}$ are added to the constraint graph, while in the second pass the edges among $\{A, D, E\}$ are added.

On the other side, the plane sweep algorithm also encounters problems when the global placement result has illegal overlaps between devices. Figure 6 shows such an example. Directly applying the algorithm to the example global placement result as in Figure 6a will generate the constraint graphs shown in Figure 6b. There are constraint edges in both horizontal and vertical constraint graphs for the device pairs $\{A, B\}$ and $\{B, D\}$, which are highlighted in red. Imposing these positional relationships will over-constrain the legalization and result in a sub-optimal area, as illustrated by the legalized placement snapshot in Figure 6b. To get a more compact placement after legalization, we will remove the extra constraint edges between each pair of devices with illegal overlap by determining their relative position greedily. We choose to spread them in the direction that induces less displacement and decide their relative position. We will only keep the constraint edge corresponding to the chosen positional relationship, while other edges between them will be removed. Continuing our example, Figure 7a shows the resulted constraint graphs after resolving the illegal overlaps, where only the horizontal edge between devices $\{A, B\}$, and the vertical edge between $\{B, D\}$ are kept, which are highlighted in red.



(a) after resolving illegal overlaps

(b) new edges added for the missing postional relationships

Figure 7: The resulting constraints graphs generated by our algorithm after resolving illegal overlaps, and after missing postional relationship detection.

However, there may be missing positional relationships from the graphs we obtain in the previous step. For example, removing the vertical constraint edge between devices A and B causes the missing relationship between A and D in Figure 7a. In other words, the positional relationships between these two devices are undefined in both horizontal and vertical directions, which may result in illegal overlaps after compaction, as illustrated by the snapshot in Figure 7a. To add back those missing edges, we first need to identify them. In order to detect the missing positional relationships, a depth-first-search (DFS) based algorithm is used, which is shown in Algorithm 1. First, we use two matrices M_h, M_v to store the relationship between the devices (including the virtual source nodes s_h, s_v with the last indices in the graphs). $M_h[i][j] = 1$ means there is horizontal positional relationship between devices i and j while $M_h[i][j] = 0$ means there is no horizontal positional relationship, and M_v is defined similarly. Then a DFS-based algorithm is used to fill M_h, M_v. After that, we can detect the devices without horizontal (vertical) positional relationship from M_h (M_v). If both horizontal and vertical positional relationships are missing, the pair of devices will be identified as missing positional relationships in the current constraint graphs. For each pair of devices that are not allowed to overlap whose positional relationship is missing, we will add one edge to either the vertical or horizontal constraint graph. To be specific, if the vertical spacing is larger than horizontal spacing between the two devices, we will add an edge to the vertical constraint graph, and vice versa. For example, given the graphs in Figure 7a, our DFS-based algorithm will detect the missing relationships between devices A and D, as well as between D and E. As a result, new edges will be added between them, which are indicated in red in Figure 7b. This will ensure that no illegal overlap exists after compaction, as shown in the snapshot in Figure 7b.

Algorithm 1 Missing Positional Relationships Detection

```
Input: $n$ devices and their vertical and horizontal constraint graphs $G_v, G_h$
Output: Find all the missing relationships of the devices
let $M_h, M_v$ be two $(n+1) \times (n+1)$ Boolean matrices;
let $s_h, s_v$ be the source of $G_h, G_v$ respectively;
DFS($G_h$, $s_h$, $M_h$)
DFS($G_v$, $s_v$, $M_v$)
for i = 0 to $n-1$ do
    for j = $i+1$ to $n-1$ do
        if $\neg(M_h[i][j] \vee M_h[j][i]) \wedge \neg()M_v[i][j] \vee M_v[j][i])$ then
            no positional relationship between devices $i, j$;
        end if
    end for
end for

function DFS($G$, $v$, $M$)
    label $v$ as discovered;
    $M[v][v] = 1$;
    for all edges from $v$ to $w$ in G.adjacentEdges($v$) do
        if vertex $w$ is not labeled as discovered then
            DFS($G$, $w$, $M$)
        end if
        if $M[v]$ is not all 1 then
            for i = 0 to $n-1$ do
                $M[v][i] = M[v][i] \vee M[w][i]$
            end for
        end if
    end for
end function
```

Finally, we will perform transitive edge removal (transitive reduction) on both constraint graphs to remove the redundant edges. To be specific, for each vertex u, we will perform DFS from each vertex v which is the direct descendant of u. Then, for each vertex

v' reachable by v, if edge $e_{uv'}$ exists, it will be removed. The time complexity for this process is $O(|E| \cdot (|E| + |V|))$, where $|E|$ and $|V|$ are the number of edges and nodes in the graph, respectively. After this, the constraint graphs will have the minimum number of edges and can guarantee a legal placement, which is shown in Theorem 1. Due to page limit, the proof is omitted.

Theorem 1. *After our constraint graph construction, the constraint graphs have the minimum number of edges and can guarantee a legal placement.*

3.2.2 Symmetry-Aware Legalization.

After constructing the constraint graphs, we can get a legal compact placement solution using LP in accordance with the constraint graphs. The LP problem can be decomposed into two sub-problems without losing optimality, one for solving the x coordinates to minimize the total width, and another for solving the y coordinates to get the optimal total height. Take the placement with symmetry constraints on the horizontal direction (with vertical symmetric axis) as an example. The x and y coordinates sub-problems can be formulated as in Equations (6) and (7), respectively.

$$\begin{aligned} \text{Minimize} \quad & W \\ \text{Subject to} \quad & 0 \le x_i \le W - w_i, \forall i \in D, \\ & x_i + w_i \le x_j, \forall e_{i,j} \in G_h, \\ & \left.\begin{aligned} & x_i + x_j + w_j = 2 \cdot x_k^c, \forall (i,j) \in g_k^p, \\ & 2 \cdot x_i + w_i = 2 \cdot x_k^c, \forall i \in g_k^s, \end{aligned}\right\} \forall g_k \in G, \end{aligned} \tag{6}$$

$$\begin{aligned} \text{Minimize} \quad & H \\ \text{Subject to} \quad & 0 \le y_i \le H - h_i, \forall i \in D, \\ & y_i + h_i \le y_j, \forall e_{i,j} \in G_v, \\ & y_i = y_j, \forall (i,j) \in g_k^p, \forall g_k \in G, \end{aligned} \tag{7}$$

where W (H) is the total width (height) of the placement, G_h (G_v) is the horizontal (vertical) constraint graph, D is the set of devices, x_i (y_i) is the x (y) coordinate of the lower-left corner of device i, w_i (h_i) is the width (height) of device i, G is the set of symmetric groups, g_k^p and g_k^s are the set of symmetric pairs and the set of self-symmetric devices in a symmetric group g_k respectively, and x_k^c is the coordinate of the symmetric axis of the group g_k. There are two sets of constraints which are topology order (non-overlap) and symmetry constraints. The topology order constraints are from the constraint graphs obtained by the previous section, while the symmetry constraints are directly from the design specification. The y coordinates sub-problem differs from the x coordinates sub-problem in that the x coordinates of a symmetric pair need to be symmetric with respect to an axis (i.e., $x_i + x_j + w_j = 2 \cdot x_k^c$), while their y coordinates need to be the same (i.e., $y_i = y_j$). The LP problems for the symmetry constraints on the vertical direction (with horizontal symmetric axis) can be formulated in a similar way.

3.3 Detailed Placement

After legalization, we can get a legal compact placement solution. In the detailed placement stage, we will use LP to further optimize the wirelength for the given legal placement. The formulations are similar to Equations (6) and (7) in the legalization step, except that the placement boundaries are fixed to the optimal total width and height obtained from legalization to ensure that the placement remains compact, and the optimization objective becomes minimizing wirelength, which is shown in Equation (8), where W^* and H^* are the optimal total width and height obtained from legalization.

$$\begin{aligned} \text{Minimize} \quad & Wirelength \\ \text{Subject to} \quad & 0 \le x_i \le W^* - w_i, \forall i \in D, \\ & x_i + w_i \le x_j, \forall e_{i,j} \in G_h, \\ & 0 \le y_i \le H^* - h_i, \forall i \in D, \\ & y_i + h_i \le y_j, \forall e_{i,j} \in G_v, \\ & \left.\begin{aligned} & x_i + x_j + w_j = 2 \cdot x_k^c, \forall (i,j) \in g_k^p, \\ & 2 \cdot x_i + w_i = 2 \cdot x_k^c, \forall i \in g_k^s, \\ & y_i = y_j, \forall (i,j) \in g_k^p, \end{aligned}\right\} \forall g_k \in G, \end{aligned} \tag{8}$$

4 EXPERIMENTAL RESULTS

All algorithms are implemented in C++ and all experiments are performed on a Linux machine with 3.4GHz Intel(R) core and 32GB memory. WNLIB [21] is used as the unconstrained non-linear optimization solver for the global placement. All benchmark circuits used are from experienced analog circuit designers, reflecting the real analog design complexities. Benchmark 1 is an operational amplifier (opamp), which can be widely used to implement gain, filtering, buffering, and voltage regulation. Benchmark 2 is a g_m-C integrator, which can be found in low-power continuous-time filters. Benchmark 3 is a continuous-time delta-sigma modulator (CTDSM). CTDSMs are oversampling ADCs that leverage noise-shaping. As specified by the circuit designer, all capacitors in the circuit benchmarks are allowed to overlap the transistors and resistors without degrading the circuit performance. The devices in benchmark 1 and 2 include Type I and Type II devices. Benchmark 3 contains not only Type I and Type II devices, but also Type III devices. The benchmark circuit information is summarized in Table 3. In the rest of this section, Section 4.1 compares the placement results with and without device layer-awareness, Section 4.2 shows the routability of the placement, and Section 4.3 demonstrates the effectiveness of the analytical framework by comparing with the baseline algorithm.

Table 3: Benchmark circuits information.

Index	Design	#Devices	#Type I	#Type II	#Type III	#Nets
1	opamp	46	42	4	0	29
2	g_m-C integrator	15	13	2	0	9
3	CTDSM	21	6	2	13	27

4.1 Effects of Device Layer-Awareness

Table 4 compares the placement results with and without device layer-awareness in our analytical analog placement framework, where "NLP" means the placement results without device layer-awareness, and "Device layer-aware NLP" means the placement results with device layer-awareness. We can see that area and HPWL benefits are consistently achieved (on average 9% and 23% reduction, respectively) when we co-optimize the devices occupying different layers during placement. The run-time is comparable with and without device layer-awareness. Figure 8 shows the placement results with and without device layer-awareness for the opamp circuit, where the Type I and II devices are indicated in pink and blue, respectively. The figure visualizes and demonstrates that our device layer-aware placement is effective in improving the layout quality.

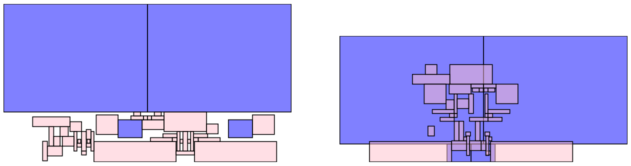

(a) without device layer-awareness (b) with device layer-awareness

Figure 8: Placement results of the opamp circuit.

4.2 Routability Verification

4.2.1 Analog Global Routing. Since the Type II and Type III devices occupy metal and via layers, they will create routing blockages and lead to fewer available routing resources. From this perspective, routing congestion and the resulting wirelength might be a concern. To validate the routability of the proposed device layer-aware layout scheme, a maze routing based analog global router is developed.

Maze routing is a classic and efficient routing strategy widely used in routers [22–24]. Although it is time consuming compared to other routing methods, the result quality is good due to its optimality for two-pin net. Since there exists a limited number of nets in a constrained area for typical analog circuits, it is preferable to utilize maze routing for designing wire connections. To save the search space, we introduce a detour ratio defining how much detour is allowed. If a legal solution cannot be acquired within the bounding regions defined by the ratio, it will be relaxed to a larger value to cover more routing resources. By constraining the region to be explored, the run-time of our global routing can be controlled efficiently. Meanwhile, to handle the specific characteristics brought by analog circuits, we extend the traditional maze routing by introducing the adjustments as follows.

It is known that the symmetric nets in analog circuits should conform to the stringent topological symmetry constraints. Thus we prefer to route the symmetric nets with higher priorities than the others in our sequential global routing algorithm. Also, as the topologies for two symmetric nets have to satisfy the symmetric constraints, we will treat the symmetric nets as one routing object, similar to [25, 26]. Hence, we consider the blockages confronted by both nets to generate the feasible route.

4.2.2 Congestion Analysis. In our global routing settings, the grid size and routing capacity inside each grid are determined by the metal pitch defined in Process Design Kit (PDK). The number of metal layers used for routing is set to 6, according to the common practice of manual analog layout. This is because our benchmark circuits usually serve as buildings blocks for a larger system, and the higher metal layers can be used for interconnection across different building blocks. The Type II devices in our benchmarks occupy metal layers 3 to 6, while the Type III devices occupy the substrate, polysilicon, and metal layers 1 to 3. The grids occupied by these devices will be marked as routing blockages on corresponding layers. After running global routing, the results are listed in Table 6, which verify the routability of our device layer-aware analog placement solutions. An average of 18% wirelength improvement is observed, compared with the global routing results on the non-overlapping placement solutions. The run-time of the global routing is very fast (i.e., less than 0.5 seconds), and it is not listed in the table.

4.3 Effectiveness of the Analytical Framework

We further compare our analytical analog placement framework with the Mixed-Integer Linear Programming (MILP) based analog placement engine in [11]. Since none of the previous work considered the overlaps between devices on mutually exclusive manufacturing layers during the placement stage, the comparisons are done in the setting without device layer-awareness. Table 5 shows the comparisons between the results of our analytical framework (NLP) and the baseline algorithm (MILP). In the table, all the metrics are normalized with respect to the results set NLP. The set of results MILP-Q is generated by running the MILP-based placement until it reaches comparable quality as the results set NLP. Another set of results MILP-R is generated by specifying the same (or similar) run-time as the results set NLP, or when the MILP-based algorithm begins to find a solution. Comparing the results sets MILP-Q and NLP, our analytical framework generates comparable (or better) results with the run-time speedup of above 18x. On the other hand, comparing the results sets MILP-R and NLP, we can see that our analytical framework is able to achieve better total placement area and HPWL given the similar run-time (18% area reduction and 13% HPWL improvement on average). Therefore, we can conclude that even when we disable the capability of device layers-awareness, our results are still better than the baseline MILP-based algorithm. Note that for the circuits with a small number of placement devices (e.g., g_m-C integrator and CTDSM circuits), the MILP-based algorithm is effective in achieving good quality in reasonable run-time. However, for medium or large size circuits (e.g., opamp), our analytical framework is much more efficient than [11] to achieve good quality in a short run-time, demonstrating the scalability of our algorithm.

5 CONCLUSION

In this paper, we have presented a device layer-aware analog placement. Different from the prior work where non-overlapping constraints were imposed on every pair of devices, we strategically overlap the devices which reside on mutually exclusive manufacturing layers and are insensitive to coupling, so that the total area and wirelength can be effectively reduced without degrading the circuit performance. We propose an analytical framework to tackle the device layer-aware analog placement problem. We also develop an analog global router to verify the routability of the device layer-aware analog placement solutions. Experimental results show that the proposed techniques can improve the total area and HPWL by 9% and 23%, respectively, and can also achieve an average of 18% reduction on the wirelength after global routing.

Table 4: Results of our analytical analog placement framework (NLP).

Design	NLP						Device layer-aware NLP					
	area (μm^2)		HPWL (μm)		run-time (s)		area (μm^2)		HPWL (μm)		run-time (s)	
	actual	norm.	actual	norm.	actual	norm.	actual	norm.	actual	norm.	actual	norm.
opamp	2973	1	753	1	17.1	1	2370	0.80	498	0.66	10.9	0.64
g_m-C integrator	182	1	73	1	1.2	1	175	0.96	60	0.83	1.2	1.00
CTDSM	57455	1	3129	1	6.5	1	56060	0.98	2580	0.82	6.5	1.00
average		1		1		1		0.91		0.77		0.88

Table 5: Results of MILP-based placement with quality matching (MILP-Q), and run-time matching (MILP-R) our framework without device layer-awareness (NLP).

Design	NLP						MILP-Q						MILP-R					
	area (μm^2)		HPWL (μm)		run-time (s)		area (μm^2)		HPWL (μm)		run-time (s)		area (μm^2)		HPWL (μm)		run-time (s)	
	actual	norm.	actual	norm.	actual	norm.	actual	norm.	actual	norm.	actual	norm.	actual	norm.	actual	norm.	actual	norm.
opamp	2973	1	753	1	17.1	1	3295	1.11	714	0.95	607.2	35.57	4181	1.41	927	1.23	20.2	1.19
g_m-C integrator	182	1	73	1	1.2	1	187	1.03	69	0.95	20.7	17.21	184	1.01	77	1.06	5.7	4.73
CTDSM	57455	1	3129	1	6.5	1	57960	1.01	3611	1.15	20.5	3.16	64472	1.12	3443	1.10	10.5	1.62
average		1		1		1		1.05		1.02		18.65		1.18		1.13		2.51

Table 6: Global routing (GR) results for the proposed analytical placement framework (NLP).

wirelength (μm)	NLP + GR		Device layer-aware NLP + GR	
	actual	norm.	actual	norm.
opamp	839	1	617	0.74
g_m-C integrator	89	1	79	0.89
CTDSM	3591	1	3034	0.84
average		1		0.82

ACKNOWLEDGMENT

This work is supported in part by the NSF under Grant No. 1527320, No. 1704758, and the DARPA ERI IDEA program.

REFERENCES

[1] R. Aparicio and A. Hajimiri, "A noise-shifting differential colpitts vco," *IEEE Journal Solid-State Circuits*, vol. 37, no. 12, pp. 1728–1736, 2002.

[2] R. Kapusta, J. Shen, S. Decker, H. Li, E. Ibaragi, and H. Zhu, "A 14b 80 ms/s sar adc with 73.6 db sndr in 65 nm cmos," *IEEE Journal Solid-State Circuits*, vol. 48, no. 12, pp. 3059–3066, 2013.

[3] L. Kull, T. Toifl, M. Schmatz, P. A. Francese, C. Menolfi, M. Brandli, M. Kossel, T. Morf, T. M. Andersen, and Y. Leblebici, "A 3.1 mw 8b 1.2 gs/s single-channel asynchronous sar adc with alternate comparators for enhanced speed in 32 nm digital soi cmos," *IEEE Journal Solid-State Circuits*, vol. 48, no. 12, pp. 3049–3058, 2013.

[4] Y. Lin, B. Yu, B. Xu, and D. Z. Pan, "Triple patterning aware detailed placement toward zero cross-row middle-of-line conflict," *IEEE TCAD*, vol. 36, no. 7, pp. 1140–1152, 2017.

[5] M. Strasser, M. Eick, H. Gräb, U. Schlichtmann, and F. M. Johannes, "Deterministic analog circuit placement using hierarchically bounded enumeration and enhanced shape functions," in *Proc. ICCAD*, 2008, pp. 306–313.

[6] P.-H. Lin, Y.-W. Chang, and S.-C. Lin, "Analog placement based on symmetry-island formulation," *IEEE TCAD*, vol. 28, no. 6, pp. 791–804, 2009.

[7] P.-Y. Chou, H.-C. Ou, and Y.-W. Chang, "Heterogeneous b*-trees for analog placement with symmetry and regularity considerations," in *Proc. ICCAD*, 2011, pp. 512–516.

[8] Q. Ma, L. Xiao, Y.-C. Tam, and E. F. Young, "Simultaneous handling of symmetry, common centroid, and general placement constraints," *IEEE TCAD*, vol. 30, no. 1, pp. 85–95, 2011.

[9] H.-C. Ou, H.-C. C. Chien, and Y.-W. Chang, "Simultaneous analog placement and routing with current flow and current density considerations," in *Proc. DAC*, 2013, p. 5.

[10] P.-H. Wu, M. P.-H. Lin, T.-C. Chen, C.-F. Yeh, T.-Y. Ho, and B.-D. Liu, "Exploring feasibilities of symmetry islands and monotonic current paths in slicing trees for analog placement," *IEEE TCAD*, vol. 33, no. 6, pp. 879–892, 2014.

[11] B. Xu, S. Li, X. Xu, N. Sun, and D. Z. Pan, "Hierarchical and analytical placement techniques for high-performance analog circuits," in *Proc. ISPD*, 2017, pp. 55–62.

[12] G. Chen, C.-W. Pui, W.-K. Chow, K.-C. Lam, J. Kuang, E. F. Young, and B. Yu, "Ripplefpga: Routability-driven simultaneous packing and placement for modern fpgas," *IEEE TCAD*, vol. 37, no. 10, pp. 2022–2035, 2018.

[13] C.-W. Pui, G. Chen, Y. Ma, E. F. Young, and B. Yu, "Clock-aware ultrascale fpga placement with machine learning routability prediction," in *Proc. ICCAD*, 2017, pp. 929–936.

[14] H.-C. Ou, K.-H. Tseng, J.-Y. Liu, I.-P. Wu, and Y.-W. Chang, "Layout-dependent effects-aware analytical analog placement," *IEEE TCAD*, vol. 35, no. 8, pp. 1243–1254, 2016.

[15] B. Xu, B. Basaran, M. Su, and D. Z. Pan, "Analog placement constraint extraction and exploration with the application to layout retargeting," in *Proc. ISPD*, 2018, pp. 98–105.

[16] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang, "Ntuplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints," *IEEE TCAD*, vol. 27, no. 7, pp. 1228–1240, 2008.

[17] H.-C. Ou, K.-H. Tseng, J.-Y. Liu, I. Wu, Y.-W. Chang *et al.*, "Layout-dependent-effects-aware analytical analog placement," in *Proc. DAC*, 2015, p. 189.

[18] W. Naylor, "Non-linear optimization system and method for wire length and delay optimization for an automatic electric circuit placer," *US Patent No. 6301693*, 2001.

[19] S. Kuwabara, Y. Kohira, and Y. Takashima, "An effective overlap removable objective for analytical placement," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 96, no. 6, pp. 1348–1356, 2013.

[20] J. Doenhardt and T. Lengauer, "Algorithmic aspects of one-dimensional layout compaction," *IEEE TCAD*, vol. 6, no. 5, pp. 863–878, 1987.

[21] W. Naylor and B. Chapman, "Wnlib," http://www.willnaylor.com/wnlib.html.

[22] G. Chen, C.-W. Pui, H. Li, J. Chen, B. Jiang, and E. F. Young, "Detailed routing by sparse grid graph and minimum-area-captured path search," in *Proc. ASPDAC*, 2019.

[23] F.-Y. Chang, R.-S. Tsay, W.-K. Mak, and S.-H. Chen, "Mana: A shortest path maze algorithm under separation and minimum length nanometer rules," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 10, pp. 1557–1568, 2013.

[24] W.-H. Liu, W.-C. Kao, Y.-L. Li, and K.-Y. Chao, "Nctu-gr 2.0: Multithreaded collision-aware global routing with bounded-length maze routing," *IEEE Transactions on computer-aided design of integrated circuits and systems*, vol. 32, no. 5, pp. 709–722, 2013.

[25] D. Liu, V. Livramento, S. Chowdhury, D. Ding, H. Vo, A. Sharma, and D. Z. Pan, "Streak: synergistic topology generation and route synthesis for on-chip performance-critical signal groups," in *Proc. DAC*, 2017, pp. 1–6.

[26] D. Liu, B. Yu, V. Livramento, S. Chowdhury, D. Ding, H. Vo, A. Sharma, and D. Z. Pan, "Synergistic topology generation and route synthesis for on-chip performance-critical signal groups," *IEEE TCAD*, 2018.