

Hierarchical and Analytical Placement Techniques for High-Performance Analog Circuits

Biyong Xu, Shaolan Li, Xiaoqing Xu, Nan Sun, and David Z. Pan
ECE Department, University of Texas at Austin, Austin, TX, USA
{biyong,slliandy,xiaoqingxu.austin}@utexas.edu,
nansun@mail.utexas.edu, dpan@ece.utexas.edu

ABSTRACT

High-performance analog integrated circuits usually require minimizing critical parasitic loading, which can be modeled by the critical net wire length in the layout stage. In order to reduce post-layout circuit performance degradation, critical net wire length minimization should be considered during placement, in addition to the conventional optimization objectives of total area and half perimeter wire length (HPWL). In this paper, we develop effective hierarchical and analytical techniques for high-performance analog circuits placement, which is a complex problem given its multi-objectives and constraints (e.g. hierarchical symmetric groups). The entire circuit is first partitioned hierarchically in a top-down, critical parasitics aware, hierarchical symmetric constraints and proximity constraints feasible manner, where the placement subproblem for each partition at each level can be solved in reasonable run-time. Then, different placement variants are generated for each partition from bottom up, taking advantage of the computation power of modern multi-core systems with parallelization. To assemble the placement variants of different subpartitions, a Mixed Integer Linear Programming (MILP) formulation is proposed which can simultaneously minimize critical parasitic loading, total area and HPWL, and handle hierarchical symmetric constraints, module variants selection and orientation. Experimental results demonstrate the effectiveness of the proposed techniques.

1. INTRODUCTION

With the expanding market share of emerging applications, including consumer electronics, automotive, and Internet of Things (IoT), the demands in analog and mixed-signal (AMS) integrated circuits (ICs) are becoming higher and higher. The complexity explosion of the design rules and circuit performance requirements in nano-meter IC era also dramatically increases the complexity of their layouts. Hence, it is necessary to have design automation tools for AMS ICs [1].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISPD '17, March 19-22, 2017, Portland, OR, USA

© 2017 ACM. ISBN 978-1-4503-4696-2/17/03...\$15.00

DOI: <http://dx.doi.org/10.1145/3036669.3036678>

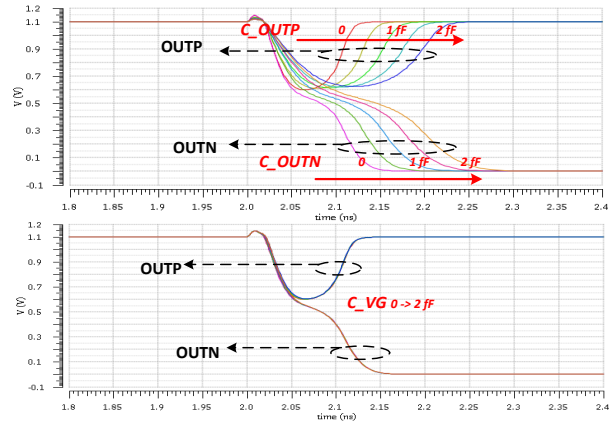


Figure 1: Transient simulations of a comparator. Top: critical parasitics effects; Down: effects of non-critical ones.

1.1 Critical Parasitics in Analog Layout

One key goal and challenge in high-performance analog layout circuits is the minimization of critical parasitics effects on the post-layout circuit performance. Critical parasitics in analog design are the parasitics that would trigger major impacts on key analog performance metrics when they vary. The critical parasitics and their effects on performance are usually identified by the analog circuit designers before starting layout, in order to efficiently minimize the degradation of post-layout performance.

W.l.o.g., we can demonstrate the significance of critical parasitics management through a dynamic latch comparator, as shown in Fig. 2. The parasitic capacitances that are of our interest are drawn, where C_{-*} (e.g. C_{OUTP}) indicates a net's self-capacitance to substrate, and CC_{-*} (e.g. CC_{OUT}) indicates the coupling capacitance between two nets. The speed of a comparator cannot be optimized simply through the minimization of total wire length, which may be over-emphasized by conventional analog placement methodologies. In fact, it is only strongly related to certain capacitances which are called critical parasitic capacitances, while other parasitic capacitances have much weaker or marginal effects on the speed. The critical parasitics identified by the circuit designers are highlighted by the red boxes. We perform simulations to show the difference in the effects caused by the critical parasitics and non-critical ones. Simulation results of the comparator transition waveform are shown in Fig. 1. In the simulation setting, the capacitors are swept

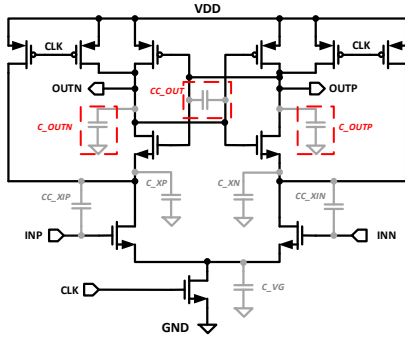


Figure 2: Example comparator circuit.

from 0 to 2 fF, which are typical values of parasitic capacitance in modern processes. It can be clearly seen that the capacitance loading of the outputs (C_{OUTP} and C_{OUTN}) presents major impact (2x increase in delay). On the other hand, other parasitic capacitances (e.g. C_{VG}) have much less impact on the comparator speed. Therefore, in terms of speed, the parasitics on nodes OUP and OUTN are the critical ones, while others can be loosely managed. The wire lengths of OUP and OUTN should be minimized to reduce the self-parasitic capacitances.

From the above discussion, it can be seen that constraints in analog design are net-specific. Conventional optimization techniques without considering net-specific requirements become suboptimal in optimizing high-performance analog circuits. High-performance analog layout synthesis requires a critical net aware algorithm.

1.2 Related Works

[2] used a branch-and-bound technique in the building block layout problem to consider critical nets with maximum length constraints. Their work considered maximum critical net length constraints for digital circuits rather than minimizing the critical parasitics for analog circuits. [3] proposed to first perform circuit analysis to determine the sensitivity of circuit performance to each parasitic loading, and then optimize performance degradation, among other metrics. However, exhaustive circuit analysis without taking advantage of the designer’s knowledge was time-consuming and could potentially lead to inaccuracy. Proximity constraints have been used to restrict some modules to be placed in close proximity [4–7]. However, they did not impose net-specific requirements, and thus were not enough to minimize critical parasitics. Boundary constraints were applied in [8] for analog placement to reduce wiring parasitics, which were also insufficient because the devices may still be far away even if they are placed on the boundary of a group. [9] considered monotonic current paths constraints to reduce the routing-induced parasitics. Recently, [10] fully separated analog and digital signal paths for noise reduction of AMS circuits. Nevertheless, these heuristics did not minimize critical parasitic loading explicitly, either.

Analog circuit hierarchy has been taken into account during placement previously by [10, 11] which demonstrated the effectiveness of the hierarchical approach. However, neither of them considered critical parasitics explicitly. Meanwhile, topological approaches have been widely used to solve the analog placement problem, including B* tree [4, 5], Corner Block List (CBL) [7], Sequence Pair [12, 13], Slicing

Table 1: Notations

WL, CL	the total HPWL and critical net HPWL.
W, H	the total width and height of the placement.
M	the set of all devices/subpartitions.
$w_i^{(k)}, h_i^{(k)}$	the width and height of the k -th variant of the i -th device/subpartition $M_i^{(k)}$.
x_i, y_i	the horizontal and vertical coordinates of the location of the i -th device/subpartition.
N, N_c	the set of all nets and critical nets.
wl_i	the HPWL of net i .
S	the set of all hierarchical symmetric groups, $S = \{S_1, S_2, \dots, S_m\}$.
L	the placement solution.
A	the summation of the average area over all variants of the devices/subpartitions, i.e. $A = \sum_i ((\sum_{k_i} w_i^{(k_i)} \cdot h_i^{(k_i)}) / k_i)$.

Tree [14], etc. Nonetheless, they require a packing step before wire length can be estimated, while absolute coordinates approaches [11, 15] could provide a more accurate estimation of wire length by construction.

1.3 Our Contributions

Our main contributions are summarized as follows.

- We formulate the high-performance analog circuits placement problem which minimizes the total area, HPWL and critical parasitics simultaneously while accommodating analog placement constraints.
- Since AMS circuits typically have the hierarchical structure, we propose a hierarchical scheme for analog placement which can comprehend the designer’s intent and obtain good circuit performance.
- The proposed hierarchical analog placement algorithm is parallelizable and scalability is demonstrated.
- Experimental results show that circuit performance degradation is reduced by minimizing the critical parasitics while keeping other objectives satisfactory.
- To the best of our knowledge, this is the first work that explicitly minimizes critical parasitics for analog placement.

The rest of this paper is organized as follows: Section 2 gives the problem formulation of the high-performance analog circuits placement problem. Section 3 proposes the hierarchical analog placement framework. Section 4 shows the experimental results. Section 5 concludes the paper.

2. PROBLEM FORMULATION

This section shows the formulation for the high-performance analog IC placement problem. The notations we use are listed in Table 1.

Firstly, we give the optimization objectives of the placement problem for high-performance analog ICs. As discussed in Subsection 1.1, the performance of an analog circuit is strongly affected by the critical parasitics. Several factors could affect parasitic capacitance of a net, e.g. net length, metal overlap with other nets, spacing with other nets running in parallel with it, etc. While the metal overlap and parallel spacing are often hard to control in the placement stage, the critical net wire length can be effectively modeled by its HPWL during placement. Therefore,

the total critical net HPWL can be expressed as $CL = \sum_{net_i \in N_c} wl_i$, and total HPWL can be written as $WL = \sum_{net_i \in N} wl_i$. The high-performance analog placement problem tries to minimize CL in addition to the conventional optimization objectives of WL and the total area A .

Secondly, we discuss how analog placement constraints are considered. Practical analog layout designs typically contain hierarchical structure and symmetric constraints. In a multi-level hierarchical structure, symmetric constraints may apply to subpartitions at each level, thus generating hierarchical symmetric constraints, which we define as follows:

Definition 1 (Hierarchical Symmetric Constraint). A hierarchical symmetric constraint is a placement constraint requiring at least one symmetric group to be symmetric to at least one other symmetric group or component, which forms a new hierarchical symmetric group.

An example hierarchical symmetric constraint is illustrated in Fig. 3. The blue boxes indicate symmetric constraints with horizontal axes (H symmetric), the red boxes indicate those with vertical axes (V symmetric), and the magenta boxes indicate those requiring both horizontal and vertical symmetry (H and V symmetric). For instance, rectangles {1, 2, 3} form an H symmetric group, where 1 and 2 form a symmetric pair and rectangle 3 is self-symmetric with respect to the same axis as the symmetric pair {1, 2}. The V symmetric constraint in this example is a hierarchical symmetric constraint, because it contains the H symmetric groups of {1, 2, 3} and {6, 7, 8} as a hierarchical symmetric pair, and requires the H and V symmetric group of {5, 9, 10} and rectangle 4 to be self-symmetric in the mean time.

[5] mentioned the concept of hierarchical symmetric constraints and discussed how they could be handled using hierarchical symmetric feasible B* trees. However, no experiment has been done to demonstrate the effectiveness of this technique for practical analog placement. In this paper, we consider hierarchical symmetric constraints in a hierarchical and analytical placement engine. Suppose M_l and M_r are any 2 devices/subpartitions which form a symmetric pair in a vertical hierarchical symmetric group S_j , and M_m is any self-symmetric device/subpartition in the same S_j . We have $x_l + x_r + w_r = 2 \cdot a_j$, and $2 \cdot x_m + w_m = 2 \cdot a_j$, where a_j is the vertical symmetric axis of S_j . The horizontal hierarchical symmetric constraints can be written similarly. Furthermore, proximity constraints require some devices/subpartitions to be in close proximity, which will be satisfied by construction of the circuit hierarchy in our placement engine. A legal placement also needs to satisfy the non-overlapping constraints which forbid overlap between any devices. Besides, orientation and variants selection will be addressed by our analog placement engine.

Finally, the high-performance analog circuit placement problem can be stated as follows:

Problem 1 (High-Performance Analog Placement).

The high-performance analog placement problem is to find legal device placement/s given the circuit netlist and device variants in different sizes, which simultaneously minimizes critical net wire length, the total wire length and area, while accommodating hierarchical symmetric constraints, proximity constraints, and non-overlapping constraints.

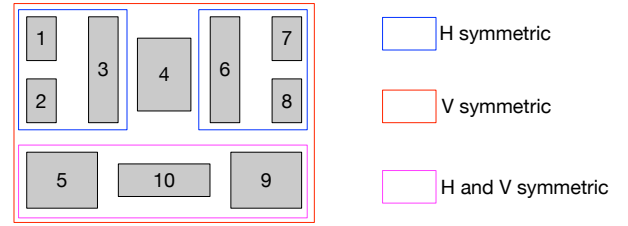


Figure 3: Example hierarchical symmetric constraints.

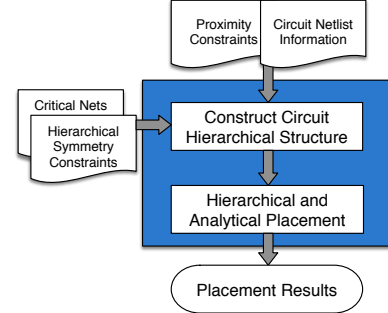


Figure 4: Hierarchical Analytical analog placement flow.

3. HIERARCHICAL PLACEMENT FRAMEWORK

The overall flow of our hierarchical analytical placement algorithm for high-performance analog circuits is shown in Fig. 4. If the circuit hierarchy is provided by the designer, our algorithm takes it as input directly. Otherwise, we apply a critical parasitics aware, symmetric and proximity constraints feasible hierarchical circuit partitioning technique. After the circuit hierarchy is obtained, hierarchical and analytical placement is performed from bottom up. Different placement subproblems at the same level in the circuit hierarchy are solved in parallel. MILP formulation is used to solve the placement subproblems for all subpartitions.

3.1 Hierarchical Circuit Partitioning

Analog circuits are typically organized in a hierarchical manner. The circuit hierarchy input by circuit designers often reflects their expertise and insights, such as which components should be placed in close proximity to avoid process variation induced circuit performance degradation, etc. Also, placing the modules in the way designers partition the circuit would increase the readability of the placement results by the designers. Therefore, our analog placement engine will respect the circuit hierarchy if it is provided, as in [4, 12, 16]. Nevertheless, while the circuit designers have more insights in electrical performance optimization, they may have difficulty optimizing geometrical metrics (e.g. area) and electrical performance simultaneously. Therefore, in addition to being able to take the circuit hierarchy as an input, our analog placement engine will also be able to perform circuit partitioning specific to analog circuits for geometrical and electrical metrics co-optimization.

Although there are many existing well-established circuit partitioning techniques [17–20], they are not directly applicable to analog circuits because of the analog placement constraints. However, we can adapt these algorithms to fol-

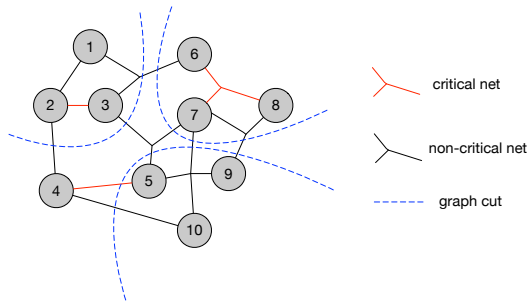


Figure 5: Example hypergraph partitioning.

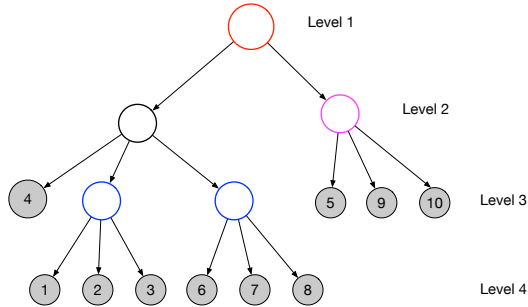


Figure 6: Example circuit hierarchy.

low the following guidelines to make it aware of the parasitic loading and analog placement constraints:

- Modules in a hierarchical symmetric group should be in the same hierarchical partition.
- Modules belonging to a proximity group should be in the same partition (the proximity constraint is satisfied by construction).
- Different criticality of different parasitic capacitances could be reflected by different net weights.

In this work, we adapted the hMetis [21] hypergraph partitioning algorithm by specifying fixed module partitions and setting proper weights for critical nets, with the implementation details clarified in Section 4. The entire netlist is modeled by a hypergraph, which we call the top-level hypergraph, where the placement devices (e.g. transistors) are its vertices and the nets are its hyperedges. It is first partitioned following the high-performance analog circuit partitioning guidelines above, and results in several subpartitions, each of which is a sub-hypergraph of the top-level hypergraph. The *internal hyperedges/nets* of a sub-hypergraph are derived from the hyperedges of the top-level hypergraph that connect only vertices within the sub-hypergraph. The *external hyperedges/nets* are the those connecting different vertices in different sub-hypergraphs. Similarly, each sub-hypergraph of the top-level hypergraph is partitioned following the same guidelines, but now only the internal hyperedges will be considered. The partitioning continues hierarchically until the desired number of placement devices are left in each leaf-level subpartition.

An example hierarchical partitioning of the circuit with hierarchical symmetric constraints and critical nets is shown in

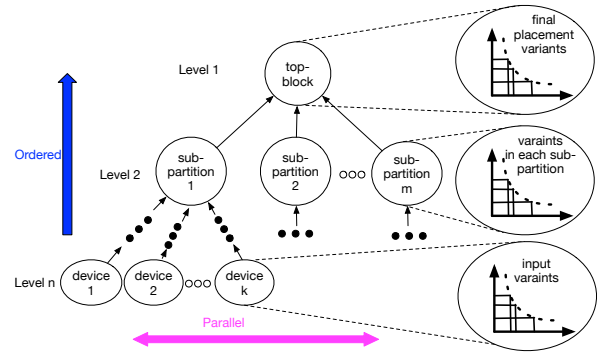


Figure 7: Hierarchical analog placement illustration.

Fig. 5, and the constructed circuit hierarchy from this partitioning is as in Fig. 6. Critical nets of the circuit netlist are colored in red, and the others in black are non-critical nets. This example circuit has hierarchical symmetric constraints as in Fig. 3. While the partitioning algorithm tries to avoid cutting critical nets, it may still do so if much better area balance could be achieved or if other ways of partitioning cannot satisfy the hierarchical symmetric constraints or proximity constraints for the desired number of partitions and desired number of levels in the circuit hierarchy specified by the designers. An example of the resulting circuit hierarchy for the example circuit is shown in Fig. 6. In this case, the partitioning algorithm first separates the H and V symmetric group (in magenta) of {5, 9 10} and others into 2 subpartitions at the second level. The other devices are further partitioned into 3 subpartitions of the H symmetric groups of {1, 2, 3} and {6, 7, 8} and device 4 at the third level. A subpartition may contain a single placement device as a special case.

3.2 Hierarchical and Analytical Placement

Given the circuit hierarchy constructed from the user input hierarchical circuit netlist or from the proposed analog circuit partitioning, our placement algorithm is illustrated in Fig. 7.

The leaf nodes of the circuit hierarchy represent primitive placement devices such as transistors or subcircuits that have been pre-laid out by the designers, and the internal nodes (non-leaf nodes) indicates hierarchical partitions. Each node in the hierarchy contains several *variants*, and exactly one of them will be selected by the placement algorithm. For a leaf node, the variants are inputs from the designers. For example, the variants of a transistor leaf node are different layouts that can be considered electrically equivalent (with the same transistor width and length) but have a different number of fingers and thus different geometrical shapes (with different geometrical width and height). For an internal node, the variants are the placement results for that hierarchical partition, which have different bounding box shapes (with different total widths and heights), different aspect ratios, and different locations, orientations, or selected input variants for the devices. The different variants of an internal node are generated by solving a *placement subproblem* (defined in Subsection 3.2.1) of a subpartition, which then propagate and become inputs to the placement subproblem of its parent node. Different placement subprob-

lems at different levels in the circuit hierarchy are solved orderly from bottom-up, while those at the same level can be solved in parallel (see Subsection 3.2.2). Finally, the different variants contained in the root node are the set of placement results for the top-block.

3.2.1 Solving Placement Subproblems

Since the proximity constraints are satisfied during our construction of the circuit hierarchy, the placement subproblem of a subpartition does not need to handle these constraints. It is formulated as below:

Problem 2 (Placement Subproblem). *Given a set of subpartitions each containing different variants and the external nets (as defined in 3.1) connecting them, the high-performance analog placement subproblem is to find legal placement/s of these subpartitions which simultaneously minimizes the critical net wire length and total wire length within these subpartitions and total area of them, while accommodating hierarchical symmetric constraints and non-overlapping constraints among these subpartitions.*

This is a multiple objectives optimization (MOO) problem, and generally, the optimal values of different objectives are usually not achieved at the same solution. We say that a solution s of a MOO problem dominates another solution \tilde{s} if s has better values for one or more objectives than \tilde{s} and the same values for all other objectives as \tilde{s} . The aim of the high-performance analog IC placement subproblem is to try to obtain the placement solutions that are non-dominated by any other solution in terms of the objectives of critical net wire length, the total HPWL, total width and total height.

To solve each placement subproblem, first, a list of initial widths $\{W_0^{(1)}, W_0^{(2)}, \dots, W_0^{(r)}\}$ are calculated based on the desired initial aspect ratios and total area by GETINITIALWIDTHS (Alg. 1), where $AR_0^{(i)} = (W_0^{(i)})/(H_0^{(i)})$, $i = 1, \dots, r$ are the input initial aspect ratios by the designers. The normalization factors can be obtained in different ways, e.g. by $H_0^{(i)} = (W_0^{(i)})/(AR_0^{(i)})$, and $WL_0^{(i)} = (H_0^{(i)} + W_0^{(i)})/2 \cdot n$, where n is the number of nets.

Then, three approaches with different optimization flavors towards different objectives are explored which are shown in DIFFERENTMOOFLAVORS (Alg. 2), each varying the general MILP problem as shown below, where total width/height boundary constraints specify the placement boundaries:

$$\begin{aligned} \min_L \quad & \alpha \cdot \frac{H}{H_0} + \beta \cdot \frac{W}{W_0} + \theta \cdot \frac{WL + \gamma \cdot CL}{WL_0} \\ \text{s.t.} \quad & \text{hierarchical symmetric constraints} \\ & \text{non-overlapping constraints} \\ & \text{total width/height boundary constraints} \end{aligned}$$

- SEQUENTIALMIN finds a solution on the Pareto Front by sequentially minimizing H , W and the weighted sum of WL and CL . First, it minimizes H given a specific W_0 with MINHEIGHTMILP by setting β and θ to 0 in the general MILP, and the width boundary to W_0 . \tilde{H} indicates the resulting optimal height, and \tilde{L}_1 represents the placement result of this step. Then, it minimizes W given the obtained optimal height \tilde{H} with MINWIDTHMILP by setting α and θ to 0, and the width and height boundaries to W_0 and \tilde{H} . \tilde{W} indicates the resulting optimal width, and \tilde{L}_2 represents

Algorithm 1 GetInitialWidths

```

1: procedure GETINITIALWIDTHS( $A, AR_0^{(1)}, \dots, AR_0^{(r)}$ )
2:    $\tilde{W}_0 \leftarrow \text{sqr}t(A)$ 
3:   for all  $i$  do
4:      $W_0^{(i)} \leftarrow \text{sqr}t(AR_0^{(i)}) \cdot \tilde{W}_0$ 
5:   end for
6:   return  $\{W_0^{(1)}, W_0^{(2)}, \dots, W_0^{(r)}\}$ 
7: end procedure

```

Algorithm 2 DifferentMOOFlavors

```

1: procedure SEQUENTIALMIN( $W_0, M, S$ )
2:    $\tilde{H}, \tilde{L}_1 \leftarrow \text{MINHEIGHTMILP}(W_0, M, S)$ 
3:    $\tilde{W}, \tilde{L}_2 \leftarrow \text{MINWIDTHMILP}(\tilde{H}, M, S)$ 
4:    $\tilde{L} \leftarrow \text{MINWLCLMILP}(\tilde{W}, \tilde{H}, M, S)$ 
5:   return  $\tilde{L}$ 
6: end procedure
7: procedure FIXEDAREAMIN( $W_0, H_0, A_m, M, S$ )
8:    $W_m \leftarrow \text{sqr}t(\frac{A_m}{A}) \cdot W_0$ 
9:    $H_m \leftarrow \text{sqr}t(\frac{A_m}{A}) \cdot H_0$ 
10:   $\tilde{L} \leftarrow \text{MINWLCLMILP}(W_m, H_m, M, S)$ 
11:  return  $\tilde{L}$ 
12: end procedure
13: procedure WEIGHTEDSUMMIN( $W_0, H_0, WL_0, M, S$ )
14:   $\tilde{L} \leftarrow \text{MINWSMILP}(W_0, H_0, WL_0, M, S)$ 
15:  return  $\tilde{L}$ 
16: end procedure

```

the placement result of this step. Finally, it tries to minimize the weighted sum of WL and CL with MINWLCLMILP by setting α and β to 0, and the height and width boundaries to the optimal height and width \tilde{H} and \tilde{W} obtained from the previous 2 steps, respectively.

- FIXEDAREAMIN tries to minimize the weighted sum of WL and CL given maximum area A_m , by setting α and β to 0, and the width and height boundaries to W_m and H_m which are calculated as the maximum total width and height if the initial aspect ratio is maintained, respectively.
- WEIGHTEDSUMMIN uses MinWSMILP which is identical to the general MILP. In this approach, the placement boundaries can be tuned in order to get the desirable placement results.

3.2.2 Parallelization

In our algorithm, when solving the placement subproblem of a subpartition, the locations of the other components outside of the subpartition have not been determined. Therefore, we will ignore the interconnections between the components inside and outside of the subpartition of concern, and the placement subproblems of different subpartitions at the same level in the circuit hierarchy can be regarded as “independent” by our algorithm. Moreover, the placement subproblems to generate different variants with different aspect ratios for the same subpartition do not depend on the results of each other. Hence, the proposed algorithm is well-suitable for parallelization, which can take advantage of the

computation power of modern multi-core systems. In the ideal case, the fully parallelized version of our algorithm finishes in wall time proportional to the number of levels in the circuit hierarchy, assuming the circuit is partitioned such that each subproblem at each level can be solved in reasonable amount of time. In reality, the available computation resource may not allow for full parallelism, thus perfect run-time scaling may not be achieved.

4. EXPERIMENTAL RESULTS

Table 2: Benchmark circuits

Circuit	#Mod.	#Sym. Mod.	Mod. Area	#Nets	#Crit. Nets
comparator	15	14	-	14	2
ring sampler slice	102	32	-	57	4
xerox	10	0	19.35	203	16
apte	9	8	46.56	97	-
hp	11	8	8.83	83	-
ami33	33	6	1.16	123	-
ami49	49	4	35.45	408	-

We implemented the hierarchical analytical placement algorithms for high-performance analog ICs in C++ and all experiments were performed on a Linux machine with 2 8-core CPUs (2.9GHz Intel(R) E5-2690) and 192GB memory. Gurobi [22] is adopted as our MILP solver. When circuit partitioning is performed, the same parameters in hMetis are used except the number of levels in the circuit hierarchy, the number of partitions, hyperedge weights, and fixed components. The number of levels and partitions are tuned to balance run-time and placement quality. The hyperedge weight reflects the net criticality, and the components in the same hierarchical symmetric group or proximity group are fixed in the same hierarchical partition accordingly.

Table 2 lists the benchmark circuits information used in our experiments, which include real analog circuits and MCNC benchmark circuits. If not otherwise specified, the units for the real analog circuits is in μm and μm^2 , and those of the MCNC benchmark circuits are in mm and mm^2 . The columns in the table indicate the total number of modules, the number of modules that belong to any symmetric group, the sum of the area of all modules, the total number of nets, and the number of critical nets, respectively. The comparator circuit is of small size and a slice of a ring sampler circuit is of medium to large size. Since in real analog circuits the transistors can have multiple input variants with different numbers of fingers and different area, we do not calculate the total area of all the placement devices for those circuits. Both of the real analog designs are intended to achieve high performance, so the parasitic capacitances of the critical nets need to be minimized to reduce post-layout circuit performance degradation. For completeness, we also run experiments on the MCNC benchmark circuits used by other previous works on analog placement to compare results.

4.1 Critical Parasitics Minimization

4.1.1 Comparator Circuit

Table 3 includes the experimental results for the comparator circuit with and without critical parasitics minimization. Different rows represent different variants generated from

different initial aspect ratios. It can be seen that the proposed techniques consistently reduce the critical net wire length CL for all variants with different aspect ratios. While the resulting WL slightly increases, this metric is not crucial for the high-performance analog IC placement problem, as the simulation results in Subsection 1.1 show that the parasitics of non-critical nets have a marginal impact on the circuit performance. Hence, CL has a much more significant effect than WL when shooting for high circuit performance. Two example placement results which have the same area and the same aspect ratio but different critical net wire lengths are shown in Fig. 8. In this figure, the rectangular regions filled with pink represents different placement devices. The bounding boxes for critical nets are highlighted in red, while those for non-critical nets are indicated in blue. We can see clearly that the placement considering critical parasitics minimization yields smaller bounding boxes for the critical nets than the other one. This shows that even with the same area and aspect ratio, we can get better critical parasitics results using the proposed placement techniques. Meanwhile, symmetry is also observed in the resulting layouts.

4.1.2 Ring Sampler Slice Circuit

We extracted the HSPICE format hierarchical netlist of the slice of ring sampler circuit from the analog schematic design environment, and takes the file as input and constructs the circuit hierarchical structure. We ran the parallelized hierarchical placement algorithm for it. Two example placement results are shown in Fig. 9, with the symmetric groups highlighted in yellow and the critical net bounding boxes in red. Hierarchical symmetric groups can also be observed in the results, i.e. some symmetric groups are symmetric to other symmetric groups. The first variant has a smaller area and slightly longer CL than the second one, while the latter achieves better CL but is less compact in terms of total area. Our algorithm generates several non-dominated placements so that designers can choose from them according to their trade-offs.

After obtaining the critical net lengths and the metal to substrate capacitance parameters from the target process technology files, we are able to estimate the critical parasitic loading of each critical net, and do the schematic-level circuit performance simulation with these estimated parasitic capacitances injected to the corresponding critical nets. Non-critical net parasitics are not injected since their effects are marginal and can be ignored for estimation purpose. We compare our placement results with the manual layout by experienced designers using the same performance simulation method, except that the critical net half perimeter wire lengths are measured from the manual layout. Unit capacitance per μm for minimum wire width we used to do simulation is $0.111\text{fF}/\mu\text{m}$. Table 4 shows the comparisons of the simulation results for our second variant and the manual layout. Note that since the manual layout is post-routing, it is natural that it will have longer CL than our placement result. In the table, K_{vco} is the voltage-controlled oscillator (VCO) gain, which determines the loop gain, and has a direct impact on the signal to noise ratio (SNR). Smaller critical parasitics can reduce the degradation of K_{vco} , maintaining a good SNR. I_{bias} is the VCO bias current sampled at VCO frequency of 110MHz. As the critical parasitics loading increases, power increases in order to maintain the

same center frequency. Overall, the simulation results show that it is compelling to minimize critical parasitics in the layout synthesis of high-performance analog circuits, and demonstrate the merits of this work.

Table 3: Comparisons with and without minimizing critical parasitics of comparator circuit

Size			w/o considering critical parasitics		Considering critical parasitics	
W	H	Area	WL	CL	WL	CL
4.44	11.04	49.02	465.4	53.8	466.8	52.4
5.7	7.52	42.86	500.6	113.2	505.8	97.2
6.38	7.38	47.08	514.6	130.2	515.8	129.6
6.8	6.58	44.74	541.5	110.4	551.1	105.6
7.48	6	44.88	427.6	84.2	472.2	52.4

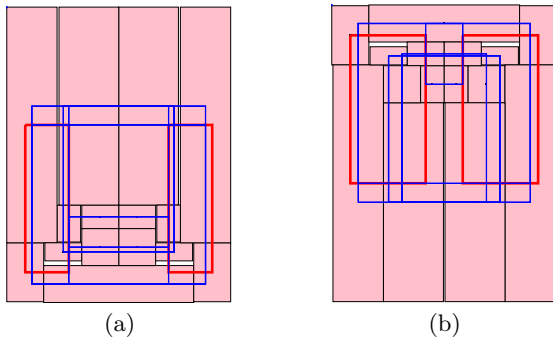


Figure 8: Placement results of comparator circuit: (a) considers critical nets (b) does not consider critical nets.

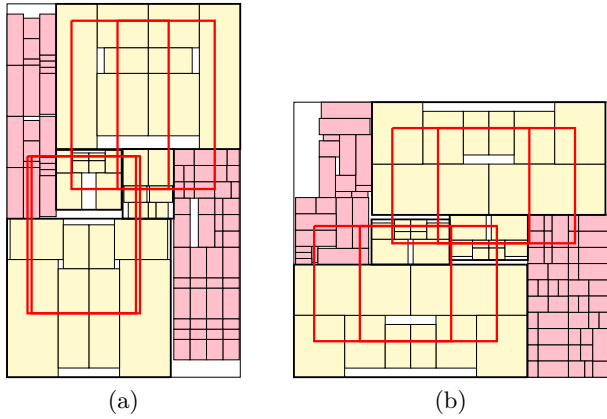


Figure 9: Placement results of a slice of ring sampler circuit: (a) variant 1 (b) variant 2.

4.2 Comparisons on Different MOO Flavors

This set of experiments was run using the xerox circuit, both without and with critical parasitics consideration, whose results are shown in Table 5 and Table 6. Different input aspect ratios are used, and placements are run using the 3 MOO flavors for the same amount of time (e.g. 100s) for each initial aspect ratio. For SEQUENTIALMIN, the run-time

Table 4: Simulation results of our placement and the manual layout of ring sampler circuit

Layout	CL (μm)	K_{vco} (THz/A)	I_{bias} (μA)	SNR (dB)	Finish time
ours	19.88	2.418	38.7	72.6	1243s
manual	43.44	2.35	40	72	1 month

* Our CL was based on placement results, and that of the manual layout were extracted from post-routing layout.

is accumulated from its first to the last step. When considering critical parasitics, the critical net weight γ is set to a high weight (e.g. 20x higher than non-critical ones). For FIXEDAREAMIN, the fixed area is set such that the maximum white space is 0.3.

Table 5 lists the placement results of 2 example initial aspect ratios without considering critical parasitics. Since critical nets are not assigned higher weights than the non-critical ones, comparing CL is not meaningful in this case. Therefore we only compare area and total wire length WL as highlighted. The results indicate that from SEQUENTIALMIN, WEIGHTEDSUMMIN to FIXEDAREAMIN approach we get placements with increasing area but decreasing WL.

On the other hand, when we consider critical parasitics and the critical net weight is high enough, it means CL is our primary focus and WL has less importance. Therefore, in this circumstance we only compare the total area and critical net wire length CL as highlighted in Table 6. SEQUENTIALMIN results in the most compact placements in terms of area, FIXEDAREAMIN leads to better CL at the expense of area, while WEIGHTEDSUMMIN realizes the trade-off between them.

Table 5: Comparisons of different MOO flavors w/o considering critical parasitics (run-time of each flavor: 100s).

Initialized with:	Aspect ratio 1			Aspect ratio 2		
	Area (mm^2)	WL (mm)	CL (mm)	Area (mm^2)	WL (mm)	CL (mm)
Sequential	19.8	646.5	52.6	20	760.1	61.1
Weighted Sum	21.9	626.7	45.9	21.8	748	72.2
Fixed Area	24.3	600.2	49.9	25.4	634.2	51.1

Table 6: Comparisons of different MOO flavors considering critical parasitics (run-time of each flavor: 100s).

Initialized with:	Aspect ratio 1			Aspect ratio 2		
	Area (mm^2)	WL (mm)	CL (mm)	Area (mm^2)	WL (mm)	CL (mm)
Sequential	19.8	648.5	53	20.2	586.4	41.3
Weighted Sum	21.9	690.1	49.5	23.5	579.4	32
Fix Area	26.5	769	45	24.2	517.3	30

4.3 Comparisons with Previous Work

In this subsection, we compare the placement results of our proposed techniques with the state-of-the-art analog placement work [5]. For larger benchmarks (ami33 and ami49), we run our hierarchical circuit partitioning algorithm on them. Parameters including the number of levels in the circuit hierarchy, the number of partitions in different levels, and the number of variants kept in each subpartition are determined according to the trade-off between optimization quality and efficiency. W.l.o.g., this set of experiments is

run using the SEQUENTIALMIN approach to solve the placement subproblems. Comparisons are shown in Table 7. We do not compare HPWL results for apte and hp circuits, because there might be some difference in the way they calculated HPWL for these 2 benchmarks per our discussion with the authors of [5] which makes the two numbers incomparable, and their detailed results and the executable of their program were not obtainable. The results demonstrate that our algorithm achieves better total area and HPWL results with tolerable run-time overhead.

Table 7: Comparisons with state-of-the-art analog placement work.

Bench- marks	[5]			This work				
	Area	HPWL	Time (s)	Area	change	HPWL	change	Time (s)
apte	47.9	*	3	47.08	-1.72%	297.12	-	6
hp	10.1	*	16	9.57	-5.25%	74.38	-	32
ami33	1.29	47.23	39	1.26	-2.36%	45.05	-4.62%	348
ami49	41.32	769.99	96	39.52	-4.35%	763.93	-0.79%	559

5. CONCLUSION

In the paper, we propose hierarchical and analytical placement techniques for high-performance analog ICs. The circuit hierarchical structure is either obtained from the designers' input or with the proposed critical parasitic loading aware, hierarchical symmetric constraints and proximity constraints feasible hierarchical circuit partitioning, followed by a hierarchical and parallelized placement algorithm for high-performance analog circuits. An MILP formulation is proposed to solve the placement subproblem for each subpartition, which minimizes critical parasitic loading, the total area and HPWL simultaneously, and handles hierarchical symmetric constraints, orientations and variants selection at the same time. Experimental results demonstrate that our proposed techniques are able to obtain analog placement results with high circuit performance in reasonable run-time.

Acknowledgment

This work is supported by the National Science Foundation under Grant No. 1527320.

6. REFERENCES

- [1] Mark Po-Hung Lin, Yao-Wen Chang, and Chih-Ming Hung. Recent research development and new challenges in analog layout synthesis. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 617–622, 2016.
- [2] Hidetoshi Onodera, Yo Taniguchi, and Keikichi Tamaru. Branch-and-bound placement for building block layout. In *ACM/IEEE Design Automation Conference (DAC)*, pages 433–439, 1991.
- [3] Koen Lampaert, Georges Gielen, and Willy M Sansen. A performance-driven placement tool for analog integrated circuits. *IEEE J. Solid-State Circuits*, 30(7):773–780, 1995.
- [4] Martin Strasser, Michael Eick, Helmut Gräß, Ulf Schlichtmann, and Frank M Johannes. Deterministic analog circuit placement using hierarchically bounded enumeration and enhanced shape functions. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 306–313, 2008.
- [5] Po-Hung Lin, Yao-Wen Chang, and Shyh-Chang Lin. Analog placement based on symmetry-island formulation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 28(6):791–804, 2009.
- [6] Hui-Fang Tsao, Pang-Yen Chou, Shih-Lun Huang, Yao-Wen Chang, Mark Po-Hung Lin, Duan-Ping Chen, and Dick Liu. A corner stitching compliant b*-tree representation and its applications to analog placement. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 507–511, 2011.
- [7] Qiang Ma, Linfu Xiao, Yiu-Cheong Tam, and Evangeline FY Young. Simultaneous handling of symmetry, common centroid, and general placement constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 30(1):85–95, 2011.
- [8] Cheng-Wu Lin, Jai-Ming Lin, Chun-Po Huang, and Soon-Jyh Chang. Performance-driven analog placement considering boundary constraint. In *ACM/IEEE Design Automation Conference (DAC)*, pages 292–297, 2010.
- [9] Po-Hsun Wu, Mark Po-Hung Lin, Yang-Ru Chen, Bing-Shiun Chou, Tung-Chieh Chen, Tsung-Yi Ho, and Bin-Da Liu. Performance-driven analog placement considering monotonic current paths. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 613–619, 2012.
- [10] Mark Po-Hung Lin, Po-Hsun Chang, Shuenn-Yuh Lee, and Helmut E Graeb. Demixgen: Deterministic mixed-signal layout generation with separated analog and digital signal paths. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 35(8):1229–1242, 2016.
- [11] Ricardo Martins, Nuno Lourenço, and Nuno Horta. Multi-objective optimization of analog integrated circuit placement hierarchy in absolute coordinates. *Expert Systems with Applications*, 42(23):9137–9151, 2015.
- [12] Po-Hung Lin and Shyh-Chang Lin. Analog placement based on hierarchical module clustering. In *ACM/IEEE Design Automation Conference (DAC)*, pages 50–55, 2008.
- [13] Mark Po-Hung Lin, Hongbo Zhang, Martin DF Wong, and Yao-Wen Chang. Thermal-driven analog placement considering device matching. pages 593–598, 2009.
- [14] Po-Hsun Wu, Mark Po-Hung Lin, Tung-Chieh Chen, Ching-Feng Yeh, Tsung-Yi Ho, and Bin-Da Liu. Exploring feasibilities of symmetry islands and monotonic current paths in slicing trees for analog placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 33(6):879–892, 2014.
- [15] Hung-Chih Ou, Kai-Han Tseng, Jhao-Yan Liu, I-Peng Wu, and Yao-Wen Chang. Layout-dependent effects-aware analytical analog placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 35(8):1243–1254, 2016.
- [16] Takashi Nojima, Xiaoke Zhu, Yasuhiro Takashima, Shigetoshi Nakatake, and Yoji Kajitani. Multi-level placement with circuit schema based clustering in analog ic layouts. In *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 406–411, 2004.
- [17] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *ACM/IEEE Design Automation Conference (DAC)*, pages 175–181, 1982.
- [18] L. A. Sanchis. Multiple-way network partitioning. *IEEE Trans. Comput.*, 38:62–81, January 1989.
- [19] Jianhua Li, Laleh Behjat, and Jie Huang. An effective clustering algorithm for mixed-size placement. In *International Symposium on Physical Design (ISPD)*, pages 111–118, 2007.
- [20] Jackey Z. Yan, Chris Chu, and Wai-Kei Mak. Safechoice: a novel clustering algorithm for wirelength-driven placement. In *International Symposium on Physical Design (ISPD)*, pages 185–192, 2010.
- [21] George Karypis and Vipin Kumar. Multilevel k-way hypergraph partitioning. In *ACM/IEEE Design Automation Conference (DAC)*, pages 343–348, 1999.
- [22] Gurobi. GUROBI. <http://www.gurobi.com/html/academic.html>, 2014.