Template-Based Parasitic-Aware Optimization and Retargeting of Analog and RF Integrated Circuit Layouts *

Nuttorn Jangkrajarng, Lihong Zhang, Sambuddha Bhattacharya,
Nathan Kohagen, C.-J. Richard Shi
University of Washington
Seattle, WA, 98105, USA

ABSTRACT
Parasitic effects are extremely significant for the performance of analog and RF integrated circuits. Although layout retargeting for technology migration or specification update is able to preserve designers' intent, the associated layout parasitics cannot be guaranteed to meet the performance requirements. In this paper, we present a novel algorithm that performs parasitic-aware automatic layout retargeting for analog/RF integrated circuits. Given parasitic resistance/capacitance bounds and matching constraints ensuring desired circuit performance, the algorithm creates a reduced-template-graph from original layouts and adds parasitic constraints. Using a two-dimensional hybrid scheme of graph-based optimization and nonlinear programming, the nonlinear problem is solved effectively and efficiently. The algorithm has successfully retargeted operational amplifiers and an RF low-noise amplifier within minutes of CPU time.

Categories and Subject Descriptors
B.7.2 [**Integrated Circuits**]: Design Aids – *layout*

General Terms
Algorithms, Performance, Design.

Keywords
Analog/RF Integrated Circuits, Layout Automation, Design Reuse, Layout Symmetry, Parasitics.

1. INTRODUCTION
The significant increase of system-on-chip (SoC) designs has intensified the growth of mixed-signal integrated circuits (ICs). Designing the digital portion can be tackled efficiently using modern cell-based tools for synthesis, mapping and physical design. Although the analog part occupies only a small fraction of the total chip area, designers spend an extraordinarily disproportionate amount of time and effort on their analog blocks. Even so, analog circuits are still often responsible for design errors and expensive design iterations. Furthermore, during a fabrication process migration, unlike digital circuits supported by the available intellectual property (IP) in standard-cell based layout automation, analog counterparts have to be manually re-designed. Thus far, analog circuits have been recognized as the design bottleneck for getting SoC products to market due to the limited availability of analog CAD tools.

Significant progress has been made recently in analog circuit optimization tools, which automatically synthesize circuit topologies, device sizes and biasing to meet desired performance specifications [1]. However, for a successful analog/mixed-signal design, the following constraints related to layout intricacies have to be seriously addressed: large variation of the MOS transistor sizes, symmetry requirements, device matching, sensitivity to parasitics, crosstalk, etc [2]. These present a great challenge for analog layout automation.

Malvasi et al. proposed the first fully integrated constraint-driven analog layout system [3]. High-level performance specifications are translated into lower-level bounds on parasitics and geometric parameters. The derived bounds can be used by a set of specialized layout tools to perform stack generation, placement, routing, and compaction. Similarly, in [4], LAYLA was developed to handle the performance and manufacturability issues in the device-level layout automation. However, in those systems, the transformation from constraints to bounds is not straightforward because of the continuous change of floorplan and interconnections. If only the placement and routing tools are used to explore the available constraint space, the computation time would be intolerably high.

Over the years, cell-based automated placement and routing methodologies for analog circuits [4], despite generality, fail to incorporate the expertise of layout designers. Analog IP reuse is inevitable when migrating designs to newer technologies or retargeting designs for another set of specifications. Recently, a template-based parasitic-aware optimization flow has been proposed [5]. The template can be automatically extracted from a coarse-grained layout by cell-based method or from an existing fine-tuned silicon-proven layout hand-crafted by designers. Although the result is promising, the parasitic models used in the optimization are coarse and the complexity of the iterative-steered layout generation with parasitic constraints is extremely high.

ICCAD'06, November 5-9, 2006, San Jose, CA

* This work was supported in part by the National Science Foundation (NSF) Information Technology Research (ITR) Program under Grant 9985507, in part by U.S. Defense Advanced Projects Agency NeoCad Program under Grant 66001-01-1-8920, and in part by a Grant from Conexant Systems. Now Dr. N. Jangkrajarng is with Cadence Design Systems Inc., CA and Dr. S. Bhattacharya is with Synopsys (India) Pvt. Ltd., Bangalore.

This paper proposes a template-based two-dimensional nonlinear optimization algorithm for parasitic-aware layout retargeting. The major contributions of the paper are summarized as follows:

- Parasitic and matching constraints are solved simultaneously in two dimensions using a hybrid algorithm of graph-based optimization and nonlinear programming;
- An accurate model for parasitic resistance and capacitance extraction of parallel structures is applied;
- Parasitic matching effects are thoroughly regarded in the parasitic-aware layout generation;
- Experiments on several analog and RF designs show significant improvement of post-layout circuit performance compared to other algorithms.

The rest of the paper is organized as follows. In Section 2, analog layout retargeting is reviewed. The parasitic problem definition and constraint generation are discussed in Section 3. Next, the parasitic-optimized layout generation is proposed in Sections 4. In Section 5, we show the experimental results, followed by the conclusion drawn in Section 6.

2. ANALOG/RF LAYOUT RETARGETING

An automatic analog/RF layout retargeting algorithm has been proposed in [6,7]. It addresses fast and reliable layout generation by utilizing intellectual property and expertise embedded in existing layouts, such as topology, floorplan, matching, and connectivity. From a layout, an adaptive symbolic structural template is constructed. Various target layouts can be realized by replacing or imposing additional constraints on the template. This technique is very useful for the modification of design specifications and layout migration between technology processes.

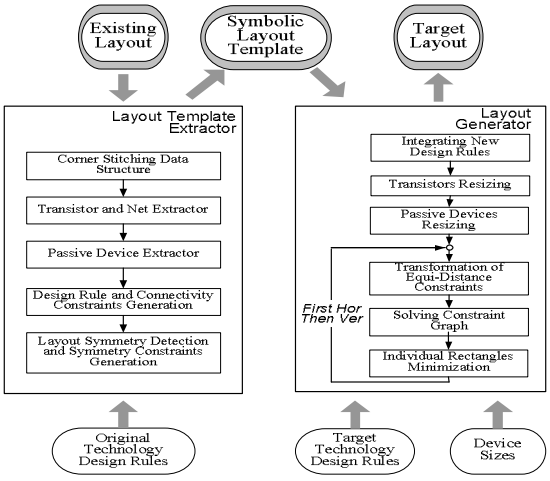


Figure 1. Analog/RF layout retargeting flow.

The detailed steps of this method are shown in Figure 1. The symbolic structural template comprises a set of constraints that are automatically extracted from an original layout. Those constraints include connectivities, design rules, floorplan, device placement topology and matching. The template can be mathematically expressed in an optimization problem as

$$Min\ (x_{rr} - x_{ll}), \tag{1.1}$$

subject to

$$x_i - x_j \geq constraint, \tag{1.2}$$

$$x_i - x_j = constraint, \tag{1.3}$$

$$x_i - x_j = x_k - x_l, \tag{1.4}$$

where x_i is an edge of a layout rectangle, while x_{rr} and x_{ll} are boundaries of the whole layout. For each layout, there will be two templates – a horizontal direction and a vertical direction. Those two templates are solved independently.

With new technology process, device substitution, device sizing, crosstalk-minimization spacing, etc, the template can be updated with both automatically generated and user-imposed constraints. The target layouts are achieved by solving the above linear optimization problem. Since constraints are generally in a form of linear equations of two variables, a fast graph-based longest-path algorithm can be applied to solve for the layout. To incorporate the matching/symmetry constraints, [8] introduced a combination of linear programming and graph-based longest-path algorithm to quickly solve the layout problem.

The algorithm of [6] has been shown to successfully retarget operational amplifiers, a voltage controlled oscillator, and an analog to digital converter. Active and passive devices, along with device parasitics, can also be retargeted using a cell replacing method [6]. However, the above method lacks a scheme for targeting interconnect parasitics, which are very important in high-performance and sensitive analog/RF layouts.

3. INTERCONNECT PARASITICS

3.1 Parasitic Problem Definition

Performance of an analog or RF circuit is dictated by each design topology and device sizing. In addition, its layout parasitics (both device and interconnect) affect the silicon performance.

Interconnect parasitics include wire resistance, wire-substrate capacitance, and coupling (or crosstalk) capacitance between wires. For a Manhattan-style interconnect, those parasitics for each rectangle (of *len* and *wid*) can be calculated as

$$R = \rho_{sh} \bullet (len / wid), \tag{2.1}$$

$$C_{sub} = c_a \bullet (len \bullet wid) + c_{sw} \bullet (2 \bullet len), \tag{2.2}$$

$$C_{coup} = c_c \bullet (len / dist), \tag{2.3}$$

where ρ_{sh} is sheet resistance per unit length, c_a is substrate capacitance per unit area, c_{sw} is sidewall substrate capacitance per unit length, c_c is coupling capacitance per unit length.

In order to ensure desired circuit performance, two criteria are required for all interconnect parasitics. First, for sensitive nets in the design, resistance and capacitance values need to be restricted within certain bounds. Second, the parasitics of some nets must be closely matched with those of other nets. This is significant in some symmetric structures, such as a differential pair.

Therefore, the layout retargeting with interconnect parasitic consideration involves a template-based optimization problem specified in equations (1.1)-(1.4) with additional nonlinear constraints due to resistance and capacitance bounds in equations (3.1)-(3.2) and matching requirements in equations (3.3)-(3.4)

$$b_n \leq \sum_i \rho_{sh} \frac{len}{wid} + R_{MFT} + R_{cont} \leq b_x', \tag{3.1}$$

$$b_n \leq \sum_i c_a \cdot len \cdot wid + \sum_i c_{sw} \cdot 2 \cdot len \leq b_x', \tag{3.2}$$

$$\sum_i \rho_{sh} \frac{len}{wid} + R_{MFT} + R_{cont} = \sum_j \rho_{sh} \frac{len}{wid} + R_{MFT} + R_{cont}, \tag{3.3}$$

$$\sum_i c_a \cdot len \cdot wid + \sum_i c_{sw} \cdot 2 \cdot len = \sum_{ji} c_a \cdot len \cdot wid + \sum_j c_{sw} \cdot 2 \cdot len, \tag{3.4}$$

where b_n and b_x are the minimum and maximum parasitic resistance/capacitance bounds of sensitive nets, R_{MFT} is a net resistance contributed by multi-finger transistors, and R_{cont} is a net resistance contributed by contact rows.

Clearly due to the nonlinear constraints, the conventional graph-based retargeting method is not applicable to this extended problem. Additionally, the equations (3.1)-(3.4) relate horizontal and vertical variables in the same constraint equation/inequality. Hence, a two-dimensional framework is required. A modified algorithm to quickly solve this problem is presented in section 4.

3.2 Sensitivity-based Parasitic Bounds and Matching

The parasitic sensitive nets can be identified as in [3]. Parasitic bounds for each sensitive net can be extracted based on a set of simulations. In addition, for certain matching nets, interconnect parasitics must be equal and within their bounds. The interconnect parasitic and matching constraints would keep the layout optimization within an acceptable part of the design space.

3.3 Interconnect Modeling

For any two-terminal interconnect in a layout, net resistance and net capacitance are simply the total sum of each individual wire rectangle resistance and capacitance respectively. Multi-terminal nets require intricate computations for accurate parasitic resistance and capacitance. However, the resultant constraint equations are too complex to be imposed on the retargeting algorithm. Thus, any net, which has more than two terminals (i.e., parallel structures), is split into several two-terminal sub-nets that share the split-box, as depicted in Figure 2. Each split-net parasitics are then calculated and retargeted separately.

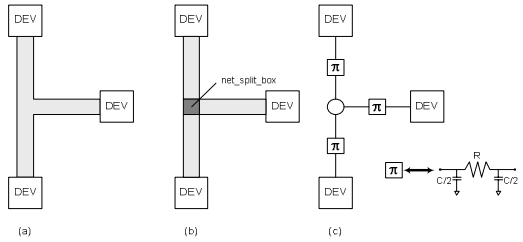


Figure 2. (a) 3-terminal net, (b) splitting into three 2-terminal nets, (c) representation of parasitics with π model.

For each rectangle segment of interconnects, a basic π model or T model can be applied to represent parasitic resistance and capacitance. In this paper, we use a RC π model to represent net parasitic resistance and capacitance as shown in Figure 2.

3.4 Multi-finger Transistor Approximation

Figure 3 shows a layout of a multi-finger transistor (MFT). The device generation [7] considers device-related performance constraints, as well as device parasitics in diffusion and gate rectangles that construct transistors. Therefore, the other port rectangles must be included in interconnect parasitic calculation. Using the idea of split net in Section 3.3 for the junctions of each finger connection will inflate the number of the constrained nets. To avoid that, port resistance and capacitance approximation for multi-finger transistors must be devised.

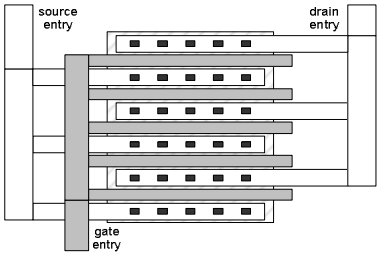


Figure 3. Multi-finger transistor with source/drain/gate connection.

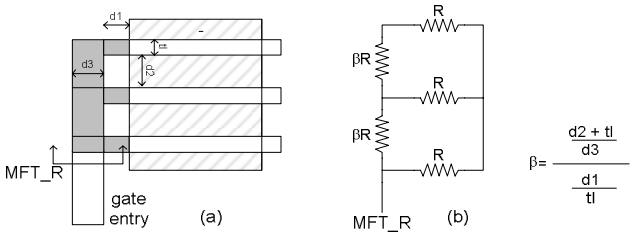


Figure 4. (a) Polysilicon of multi-finger transistor, and (b) its equivalent resistance circuit.

For the polysilicon gate in Figure 4, parasitic resistance and capacitance can be calculated based on the known geometries of the multi-finger transistor, when $0.5 \le \beta \le 5.0$, which is typical in analog/RF layouts. The resistance equation (4.1) is derived from a transistor structure with three fingers. Since resistance values do not diverge much when there are more fingers, the equation (4.1) can be applied for all cases fairly accurately. The substrate capacitance equation (4.2) is also calculated from the known multi-finger transistor geometries.

$$R = \rho_{sh} \frac{d1}{tl} \frac{\beta^2 + 3\beta + 1}{\beta^2 + 4\beta + 3}, \quad (4.1)$$

$$C = C_a [\, m \bullet d1 \bullet tl + m \bullet d2 \bullet d3 + m \bullet d3 \bullet tl - d2 \bullet d3] + C_{sw} [\, 2 \bullet m \bullet d2 - 2 \bullet d2 + m \bullet tl + 2 \bullet m \bullet d1], \quad (4.2)$$

where m is the number of fingers.

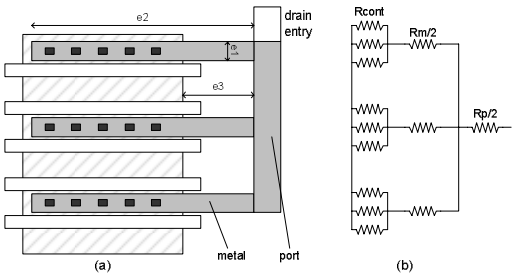


Figure 5. Drain/Source of multi-finger transistor, and its equivalent resistance circuit.

For the metal drain/source in Figure 5, we approximate the resistance of each metal rectangle, as well as the resistance of each port, to be half of the corresponding metal rectangle resistance. This assumption is valid based on the fact that there are an equal number of connections to the transistor diffusion on either side of the midpoint. From the simplified parallel resistance model shown in Figure 5b, the entry-resistance and entry-capacitance can be calculated in equations (5.1) and (5.2),

$$R = \frac{2}{m+1}\left(\frac{e2}{e1}\rho_{metal} + \frac{R_{cont}}{\#cont}\right), \quad (5.1)$$

$$C = c_a \bullet e1 \bullet e3 \bullet (m+1)/2 + c_{sw} \bullet e3 \bullet (m+1), \quad (5.2)$$

where m is the number of fingers.

4. PARASITIC-OPTIMIZED ANALOG/RF LAYOUT RETARGETING

With the additional constraint requirements in equations (3.1)-(3.4), the retargeting tool in Figure 1 must be updated to include a nonlinear constraint solver, as well as the capability to solve both the horizontal and the vertical directions simultaneously. We will explain the detailed implementation in this section.

4.1 Parasitic-Aware Layout Template Extraction

The layout template extraction is similar to the one shown in Fig. 1, with the addition of a parasitic extractor. The parasitic extractor will mark all sensitive nets with their maximum and/or minimum resistance and capacitance bounds, and matching information – if applicable. If any sensitive net has more than 2 terminals, split-boxes are inserted, and thus, nets are split into sub-nets.

After that, the parasitic extractor will search for either gate-connected or diffusion-connected multi-finger transistors on both ends. If a multi-finger transistor is found, resistance and capacitance values will be approximated and subtracted from the boundary values, then rectangles constructing those transistors will be excluded from the next step.

A list of rectangles constructing each sub-net is then created. Current direction in each rectangle is determined based on [10]. If vias or contacts are found along the path, their resistance and capacitance values will be calculated and subtracted from the boundary values. All of this information, along with electrical constants from the target technology process, will be used to construct parasitic constraints in equations (3.1)-(3.4).

4.2 Parasitic-Aware Layout Generation

Figure 6 shows an updated two-dimensional layout generation flow. This method can solve for the layout retargeting in a single pass, as opposed to the method in [5] requiring three nested iterative loops and optimization, which may not converge.

The integration of new design rules and device sizes, as well as the post-processing with the longest-path algorithm for both directions are similar to [6]. Due to the nonlinear and two-dimension nature of parasitic constraints, after the symbolic template is updated, a powerful nonlinear optimization package is required to solve the problem. Nevertheless, running a full nonlinear optimization requires a lot of processing time. Here, the idea of a reduced-graph, extended from [9], is used to construct an equivalent graph with less nodes and arcs.

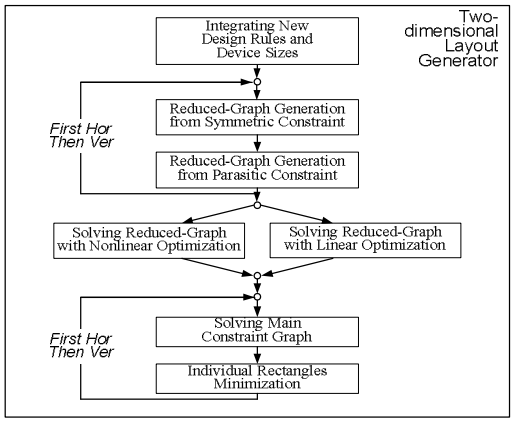


Figure 6. Two-dimensional layout generation flow.

Figure 7a shows the original graph, consisting of 8 nodes and 11 arcs. In the reduced-graph, only the nodes that are related to boundaries, symmetry constraints, and parasitic constraints (marked grey in the figure) are included. These are called core-nodes. In order to maintain the property of the graph, arcs between those nodes must be calculated. This is carried out by running a longest-path algorithm from every core-node to all other core-nodes in the graph. Figure 7b shows the reduced-graph of Figure 7a, which has only 4 nodes and 5 arcs.

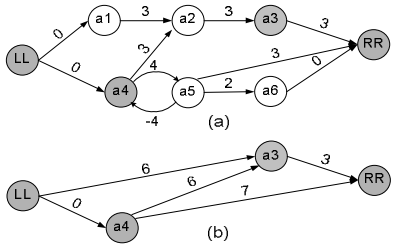


Figure 7. (a) Original graph, (b) reduced-graph.

The reduced-arcs of both vertical and horizontal directions are created and converted to equation form separately. Combining both equation sets with the parasitic and the symmetry constraints, a reduced-size optimization problem can be solved. In this work, we have implemented two optimization methods. The linear optimization discussed in Section 4.3 is faster, but lacks accuracy due to linearization of parasitic constraints. The nonlinear optimization described in Section 4.4 runs slower, but can obtain very accurate results.

For parasitics, mapping the solution of the reduced-graph back to the original graph is performed by enforcing exact-size constraints in equation (1.3) for both width and length of all parasitic-related rectangles. For symmetry, exact-size constraints for distance between two device edges in the main graph are added. Then the main graph can be solved with the longest-path algorithm.

4.3 Reduced-Graph Solved with Linear Optimization

Using the linear optimization, the parasitic constraints of resistance and capacitance in equations (2.1)-(2.3) have to be linearized before adding to equations (3.1)-(3.4). The first order Taylor Series expansion is used in the linearization. The series is expanded at the solution found before adding parasitic constraints.

Equations (6.1)-(6.2) describe the Taylor Series expansion of resistance and capacitance of one interconnect rectangle,

$$R \approx \frac{\rho \Delta x_0}{\Delta y_0} + \frac{\rho}{\Delta y_0}[x_r] + \frac{-\rho}{\Delta y_0}[x_l] + \frac{-\rho \Delta x_0}{\Delta y_0^2}[y_r] + \frac{\rho \Delta x_0}{\Delta y_0^2}[y_l], \quad (6.1)$$

$$C_{sub} \approx [-c_a \Delta x_0 \Delta y_0] + [c_a \Delta y_0 + 2c_{sw}][x_r] + [-c_a \Delta y_0 - 2c_{sw}][x_l], \quad (6.2)$$
$$+ [c_a \Delta x_0][y_r] + [-c_a \Delta x_0][y_l]$$

where x_r, x_l, y_r, and y_l are right, left, top, and bottom position of a rectangle. And $\Delta x_0 = (x_{r0} - x_{l0})$, $\Delta y_0 = (y_{r0} - y_{l0})$, where x_{r0}, x_{l0}, y_{r0}, and y_{l0} are the nominal values of that rectangle obtained from the pre-processing longest-path solving without parasitics.

4.4 Reduced-Graph Solved with Nonlinear Optimization

To achieve accurate solutions, the reduced-graph is solved with a nonlinear optimization approach, which applies a primal-dual interior-point algorithm with a filter line-search method featuring strong global and local convergence properties [11]. Therefore, no linearization is required for the consideration of parasitics. This approach shows superior quality compared to linear optimization without significantly degrading the computation efficiency.

Moreover, to ease the optimization and speed up the search process, we have developed a two-phase scheme for the parasitic-aware layout retargeting. First, a parasitic-free compaction (i.e., only considering size and symmetry constraints) is executed using a fast hybrid algorithm of graph-based optimization and linear programming [6]. Then, a parasitic-aware compaction (i.e., considering parasitic and matching constraints besides size and symmetry constraints) is solved using the hybrid algorithm of graph-based optimization and nonlinear programming. The result in the first phase will be taken as the start point of the second phase. In this way, the search effort might be focused on the adjacent region around the feasible optimal solution satisfying size and symmetry constraints. Our experimental results show that the execution time was significantly reduced due to the application of this two-phase scheme.

5. EXPERIMENTAL RESULTS

The algorithm has been implemented in C/C++ and integrated into the retargeting tool of [6]. In this section, we shall present the results of performance-driven layout retargeting on distinct analog/RF circuits, including a two-stage Miller-compensated operational amplifier (opamp) depicted in Figure 8, a single-ended folded cascade opamp depicted in Figure 9, and a double-ended low-noise amplifier (LNA) depicted in Figure 10. The first two opamps were designed initially in 0.25µm CMOS technology and retargeted to 0.18µm process with new specifications, while the LNA was designed in 0.18µm MITLL SOI technology process and retargeted to new specifications.

The parasitics that affect circuit performance are indicated in the circuit schematics. The bounds on the parasitic resistance (R) and capacitance (C) of interconnect π model obtained with the sensitivity-analysis-based approach [8] are listed in Tables 1-3. The process also detected the required matching constraints between interconnects, which would be preserved during the performance-driven retargeting.

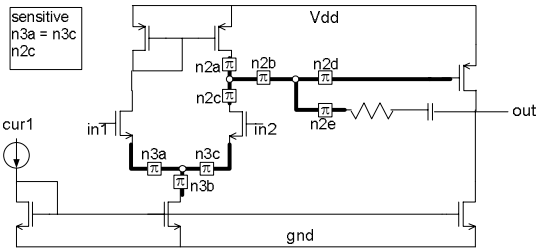


Figure 8. Two-stage Miller-compensated opamp.

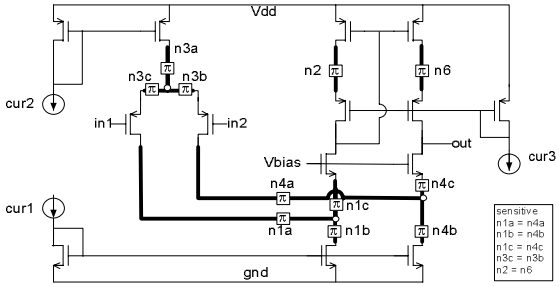


Figure 9. Folded-cascode opamp.

First, the layout retargeting was performed without performance consideration (i.e., no parasitics regarded) (WOP). Only layout floorplan, symmetry, and device sizes were modified in this retargeting. Then, the performance-driven layout retargeting was performed on the same design, via the linear programming with linearization (LPL) and the nonlinear optimization (NL) algorithms. All results were then compared to verify the effectiveness of the novel retargeting method.

After each retargeting, parasitic values for each method were extracted and compared in Tables 1-3. Here, the resistance values include metal, contact, and gate polysilicon resistances.

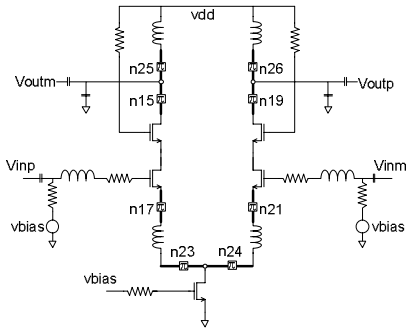


Figure 10. Double-ended low noise amplifier.

Table 1. Parasitic bounds for two-stage opamp obtained from the sensitivity-analysis-based approach, and parasitic values extracted from the layouts obtained by WOP, LPL, and NL.

	Res (Ω)	Bounds	WOP	LPL	NL
Non matching	R3a	0.45	6.82	3.59	0.45
	R3c	0.45	5.13	3.99	0.45
	R2c	19.00	3.86	3.94	3.94
Matching	R3a	4.10	-	3.74	3.98
	R3c	4.10	-	3.99	3.98
	R2c	19.00	-	3.94	3.94

Table 2. Parasitic bounds for cascode opamp obtained from the sensitivity-analysis-based approach, and parasitic values extracted from the layouts obtained by WOP, LPL, and NL.

	Res (Ω)	Bounds	WOP	LPL	NL
Non matching	R1b	4.20	2.91	2.91	2.91
	R4b	4.20	3.08	2.93	3.08
	R1c	5.00	1.00	1.57	1.57
	R4c	5.00	0.83	1.55	1.39
	R3b	0.80	5.84	3.21	0.80
	R3c	0.80	6.58	3.89	0.79
	R2	4.20	8.37	5.30	4.20
	R6	4.20	7.42	5.10	4.14
Matching	R1b	300.00	-	2.91	3.07
	R4b	300.00	-	2.92	3.07
	R1c	300.00	-	1.57	1.52
	R4c	300.00	-	1.55	1.52
	R3b	130.00	-	4.07	4.20
	R3c	130.00	-	4.33	4.20
	R2	110.00	-	5.30	4.14
	R6	110.00	-	5.10	4.14

In Table 1, the results of the two-stage opamp undoubtedly show that retargeting without parasitics consideration (i.e., WOP) failed to meet both matching and bound requirements. The parasitic-aware retargeting with linear programming (i.e., LPL) also failed to meet a very tight bound, due to miscalculation from the linearization. However, the nonlinear optimization parasitic-aware retargeting (i.e., NL) clearly met all stringent parasitic requirements, but with a very large area from interconnects expansion to reduce resistances (shown in Table 4). When the matching was imposed on simulations and retargetings, the parasitic bounds were significantly relaxed. Both linear-programming (LPL) and nonlinear optimization (NL) results satisfied the bounds. But matching accuracy experienced some loss due to the linearization of LPL. The results of the folded-cascode opamp in Table 2 exhibit the same observation as well.

Table 3. Parasitic bounds for the LNA obtained from the sensitivity-analysis-based approach, and parasitic values extracted from the layouts obtained by WOP, LPL, and NL. Note that LPL and NL resistance bounds were not different between non-matching and matching.

	Res (Ω)	Bounds	WOP	LPL	NL
Matching and Non Matching	R26	3.75	13.38	13.74	3.72
	R25	3.75	13.34	13.89	3.75
	R19	4.00	14.11	10.36	4.01
	R15	4.00	17.57	10.21	4.01
	R21	2.00	173.51	166.89	2.00
	R17	2.00	171.71	166.33	2.00
	R23	1.50	5.27	4.19	1.50
	R24	1.50	6.43	4.08	1.50

The simulation results of opamps are reported in Table 4. The results visibly confirm that the retargeting with the performance-driven nonlinear optimization (NL) can significantly improve the layout performance. The layouts from WOP and LPL failed to meet the design specification. Both layouts of NL-nm (i.e., NL with non-matching) and NL-m (i.e., NL with matching) fulfilled the specifications, while the layout from NL-m occupied less area than NL-nm – due to less rigorous resistance bounds. Layouts of opamps with NL-m method are shown in Figures 11 and 12.

Table 4. Performance of the retargeted layouts obtained by WOP, LPL, and NL (-nm: no maching, -m: matching).

		Gains (dB)	BW (MHz)	PM (°)	GM (dB)	Area (μm^2)
Two-stage opamp	Spec.	60.0	100.0	90.0	10.0	
	WOP	67.0	481.9	31.4	8.6	2904
	LPL-nm	59.1	101.8	90.4	17.0	3107
	LPL-m	61.2	103.4	90.4	16.9	3056
	NL-nm	64.1	106.7	90.5	17.2	9652
	NL-m	63.9	105.1	90.5	16.7	3084
Folded-cascode opamp	Spec	60.0	60.0	60.0	10.0	
	WOP	58.6	62.8	56.9	9.0	2308
	LPL-nm	58.9	63.2	61.1	10.2	2386
	LPL-m	60.0	63.3	61.1	10.2	2360
	NL-nm	60.6	63.7	60.3	10.4	5632
	NL-m	60.6	63.3	61.0	10.2	2320

Table 5. Performance of the retargeted LNA layouts obtained by WOP, LPL, and NL.

		Gain (dB)	Noise Figure (dB)	S11 (dB)	IIP3 (dB)	Area (mm^2)
5.6 GHz	Spec.	10.00	2.00	-15.00	-9.00	
	Ideal (R=0)	15.62	1.52	-16.01	-9.26	0.616
	WOP	0.43	8.94	-2.56	3.83	0.616
	LPL	-0.61	8.58	-2.18	6.17	0.630
	NL	11.93	1.98	-20.03	-8.76	0.647

In Table 3, for the LNA, the results from WOP clearly failed to meet the bound requirements. The LPL also did not meet the bounds, due to the linearization with Taylor Series expansion. Since the WOP values (used as midpoint for the series expansion) were far away from the bound requirements, the Taylor Series approximation was completely inaccurate. Only the result from NL can meet the bound requirements for the LNA.

Table 6. Statistics of the distinct approaches on cascade opamp and low-noise amplifier circuits.

	Cascode Opamp				Low Noise Amplifier			
	WOP	LPL	NL1	NL2	WOP	LPL	NL1	NL2
#Nodes	1103				3065			
#constraints	10390				77390			
Template Extraction Time (sec)	6.9	6.9	6.9	6.9	162	162	162	162
Layout Gen. (w/ parasitics) Time (sec)	4.1	10.2	48.7	18.3	99.2	138.1	205.4	174.0
Parasitic Solver Time (sec)	-	1.5	38.1	8.5	-	3.2	62.3	34.6
#Parasitic tiles	38				32			

Retargeted LNA layouts from WOP and NL are shown in Figure 13. Table 5 reports the simulation results of the original and all targeted LNA results. With the ideal parasitic resistance (i.e., zero resistance), the LNA can meet all performance thresholds. However, the WOP retargeting minimized all interconnect widths down to their design-rule minimum-width requirements. Thus, the parasitic resistances were enormously increased, resulting in a non-functioning LNA. When parasitics were considered, each

interconnect width was expanded in order to reduce the resistances to meet the requirement. The most visible interconnect widening can be seen in the two middle nets in Figure 13b. The simulation results for NL could meet all the specifications.

Table 6 reports the statistics on the number of constraint graph node, number of design-rules, symmetry and parasitic arcs in the constraint graph for the cascade opamp and the low-noise amplifier using distinct algorithms. Note that NL1 stands for the simple nonlinear optimization algorithm, while NL2 indicates the nonlinear optimization with the two-phase scheme described in Section 4.4. Clearly, it shows the significant execution-time reduction of NL2 compared to NL1. For all the designs, the layouts are generated within six minutes of CPU time.

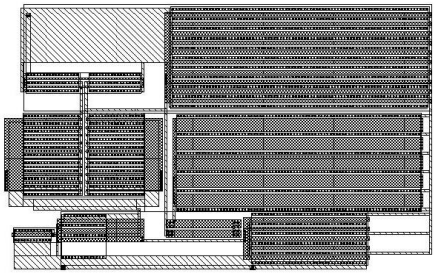


Figure 11. Retargeted layout of the two-stage opamp generated by NL.

6. CONCLUSION

In this paper, we presented a performance-driven parasitic-aware and matching-aware retargeting algorithm using nonlinear optimization. The complexity of the algorithm is significantly lower than previous work enabling the performance-driven retargeting to be performed within minutes of CPU time. The layout quality of several analog and RF circuits has been improved by taking into account the device and interconnect parasitics and device matching.

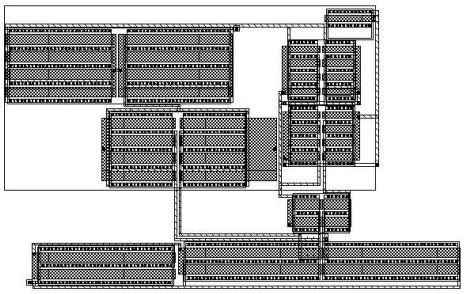
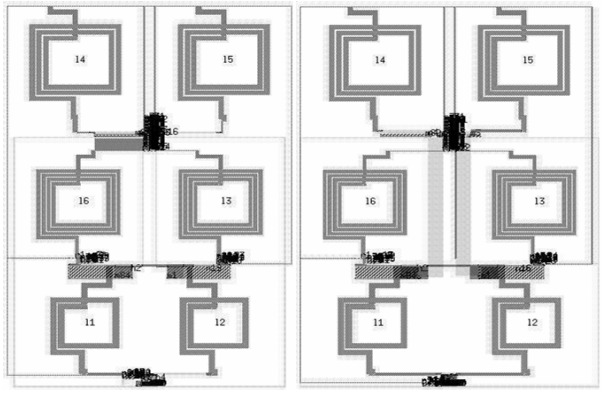


Figure 12. Retargeted layout of the cascode opamp generated by NL.

7. REFERENCES

[1] T. Dastidar, P. Chakrabarti, and P. Ray, "A synthesis System for Analog Circuits Based on Evolutionary Search and Topological Reuse," *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 2, pp. 211 - 224, Apr 2005.



(a) (b)

Figure 13. Retargeted layouts of the LNA generated (a) by WOP, and (b) by NL.

[2] G. Gielen and R. A. Rutenbar, "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits," *Proc. of the IEEE*, vol. 88, pp. 1825-1852, Dec 2000.

[3] E. Malavasi, E. Charbon, E. Felt, and A. Sangiovanni-Vincentelli, "Automation of IC Layout with Analog Constraints," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 923-942, Aug 1996.

[4] K. Lampaert, G. Gielen, and W. Sansen, Analog Layout Generation for Performance and Manufacturability, Boston: Kluwer Academic Publishers, 1999.

[5] S. Bhattacharya, N. Jangkrajarng, and C.-J. R. Shi, "Template-Driven Parasitic-Aware Optimization of Analog Integrated Circuit Layouts," *Proc. IEEE/ACM Design Automation Conference*, pp. 644-647, Jun 2005.

[6] N. Jangkrajarng, S. Bhattacharya, R. Hartono, and C.-J. R. Shi, "IPRAIL – Intellectual Property Reuse-based Analog IC Layout Automation", *Integration – The VLSI Journal*, pp. 237-262, Nov 2003.

[7] N. Jangkrajarng, S. Bhattacharya, R. Hartono, and C.-J. R. Shi, "Multiple Specification Radio Frequency Integrated Circuit Design with Automatic Template-driven Layout Retargeting", *Proc. IEEE ASP Design Automation Conference*, pp. 394-399, Jan 2004.

[8] U. Choudhury and A. Sangiovanni-Vincentelli, "Use of Performance Sensitivies in Routing Analog Circuits", *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 348-351, May 1990.

[9] R. Okuda, T. Sato, H. Onodera, and K. Tamaru, "An Efficient Algorithm for Layout Compaction Problem with Symmetry Constraints", *Proc. IEEE/ACM Int. Conf. on CAD*, pp.148-151, Nov 1989.

[10] K.-W. Chiang, "Resistance Extraction and Resistance Calculation in GOALIE2", *Proc. IEEE/ACM Design Automation Conference*, pp. 682-685, Jun 1989.

[11] A. Wachter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming", *Mathematical Programming*, volume 106(1), pp. 25-57, April 2005.