

Performance-driven Routing Methodology with Incremental Placement Refinement for Analog Layout Design

Hao-Yu Chi[†], Han-Chung Chang[‡], Chih-Hsin Yang[‡], Chien-Nan Liu[†] and Jing-Yang Jou[‡]

[†]Institute of Electronics, National Chiao Tung University, Hsinchu City, Taiwan, ROC

[‡]Department of Electrical Engineering, National Central University, Jungli City, Taiwan, ROC
logger1006.ee07g@nctu.edu.tw; hcchang.ee07g@nctu.edu.tw; 18spin@gmail.com; jimmyliu@nctu.edu.tw;
jyjou@ee.ncu.edu.tw

Abstract—Analog layout is often considered as a difficult task because many layout-dependent effects will impact final circuit performance. In the literature, many automation techniques have been proposed for analog placement and routing respectively. However, very few works are able to consider the two steps simultaneously to obtain the best performance and cost after layout. Most of the routing-aware placement techniques optimize the layout results based on an assumed routing result, which may be quite different to the final layout. In this work, we proposed an automatic two-step layout methodology for analog circuits to alleviate the performance loss during layout process. Instead of using a rough routing prediction during placement stage, a crossing-aware global routing technique is first performed to provide an accurate routing resource estimation of the given compact placement. Then, the improved CDL-based layout migration technique is adopted to do a fast adjustment on the placement and routing to reduce the difference between estimation and final layout while keeping the optimality of the given placement. As shown in the experimental results, the proposed methodology is able to improve the accuracy of routing resource estimation thus improving the final layout quality and circuit performance.

I. INTRODUCTION

As semiconductor technology continues to shrink, the number of transistors that an integrated circuit can accommodate is increasing continuously. However, closer device distance and thinner interconnection produce many non-ideal effects, which brings more challenges to the circuit design, especially for analog circuits. Since analog circuits are very sensitive to these non-ideal effects, the circuit performance will get a notable loss if these effects are not handled properly. Due to the limitations and difficulties of analog circuit design, existing analog electronic design automation (EDA) tools are still not powerful enough to support nowadays analog circuits. Although analog layout automation has gained more and more attention in recent years, modern analog EDA tools are still not widely adopted by designers because the layout generated by tools is often hard to meet the performance requirement. As a result, analog design is still a manual and time-consuming task.

In analog designs, circuit performance is often the most important concern to evaluate the layout quality instead of area or wirelength. In the literature, most of the existing analog layout automation works focus on placement or routing individually. In analog placement stage, topological layout constraints, such as symmetry, proximity and common centroid, are often applied to enhance the layout quality. [1-3] proposed simulated annealing (SA) based analog placement algorithm to deal with those layout constraints. [4] proposed analytical approach considering layout dependent effects (LDE) to improve the layout quality. In the routing stage, wire load effects and layout constraints can influence the circuit performance dramatically. [6] proposed a wire load oriented routing algorithm to decrease the performance loss after

routing. [7] proposed length matching and symmetry aware routing algorithm to handle some sensitive nets in analog circuits. However, very few works are able to consider the two steps simultaneously to obtain the best performance and cost after layout.

In addition to satisfy those layout constraints, routability is also an important issue to incorporate the placement and routing. In the previous studies of routability aware analog placement, most of them apply a simple routing model to estimate the routing topology and reserve the routing resource based on the estimation. [11] applied grid-based routing graph with FLUTE [10] for routing estimation. The routing grids that have congestion issue will be expanded by adding some dummy nodes into the placement representation to preserve more routing space. However, it is known that FLUTE is not accurate in Steiner tree construction with obstacles, which may lead to an inaccurate routing resource estimation. [12] applied probabilistic graph-based search on the topological graph of placement. The module size is then expanded if congestion occurs beside the module. However, the expanding area is not accurate enough because the real routes might be quite different to the estimation results. Furthermore, even those approaches are able to achieve overflow free after placement expansion, they cannot guarantee that the routing algorithms can fully utilize the preserved routing space because they have no such information passed from the placement stage. Therefore, some preserved routing space are insufficient or wasted that breaks the optimization results.

In order to optimize placement and routing simultaneously, we propose a performance-driven routing methodology with incremental placement refinement for analog layout design. The main contributions of this work are summarized as follows.

- A performance-driven analog layout methodology is proposed with accurate routing resource estimation methodology. Instead of separating placement and routing into individual steps, our methodology is able to improve the accuracy of routing estimation at the placement stage, thus improving the final layout quality and circuit performance.
- A crossing-aware global routing technique under compact placement is proposed. The via requirement can be considered in the early stage to reduce the number of vias and prevent the performance loss. This result is then used to provide accurate routing resource estimation.
- An efficient incremental placement refinement technique is proposed to consider the routability based on the global routing results. Through the combination of Sequence Pair (SP) and Cartesian Detection Line (CDL), a routable placement can be generated without increasing the time complexity of original packing algorithm.

- A migration-based layout construction flow is proposed to generate the final layout rapidly. The optimality of the original placement and global routing can be well preserved to improve the layout quality.

The rest of this paper is organized as follow. The preliminaries and problem formulation are briefly introduced in Section II. The proposed routing estimation technique is provided in Section III. The placement refinement and detailed routing are presented in Section IV. Finally, the experimental results is provided in Section V and the conclusion is provided in Section VI.

II. PRELIMINARIES

In this section, we first introduce the floorplan representation and routing model used in this paper. Then the problem formulation of the performance-driven routing methodology with incremental placement refinement for analog layout design is provided.

A. Sequence Pair

Sequence Pair [16] is a non-slicing topological representation for floorplanning or analog placement. The relative position and topological relationship of modules can be encoded by two sequences in SP, SP₊ and SP₋, kept in horizontal constraint graph (HCG) and vertical constraint graph (VCG). Long common subsequence [13] is often used to efficiently evaluate the placement area constructed by SP. Different from encoding a SP into a placement, [14] proposed a decoding method to convert a placement into a SP. To deal with analog layout constraints in the placement, [15] proposed a symmetry-feasible sequence pair (SFSP) for analog placement to satisfy the symmetry constraint. Furthermore, [5] proposed a current flow and symmetry aware analog placement algorithm with SP. Those placement algorithms are able to satisfy most of the analog layout constraints.

B. Cartesian Detection Line

Chi [17] proposed CDL to record the correlation between the placement and routing in one representation. Given a set of placed modules M with corresponding CDL $L^{CD}(m_i)$. The three major properties of CDL are defined as follows:

Property 1: $L^{CD}(m_i) = \{ l_j^{CD}(m_i) \mid 0 \leq j \leq 7, m_i \in M \}$

Property 2: The $Tar(l_j^{CD}(m_i))$ of each $l_j^{CD}(m_i)$ is the module m_k which the extension meets. If the extension reaches the boundary, $Tar(l_j^{CD}(m_i))$ will be recorded as boundary.

Property 3: For the set $L^{CD}(m_i)$, there is a straight routing segment besides m_i if a net intersects with the pair $\{l_0^{CD}(m_i), l_1^{CD}(m_i)\}$, $\{l_2^{CD}(m_i), l_3^{CD}(m_i)\}$, $\{l_4^{CD}(m_i), l_5^{CD}(m_i)\}$ or $\{l_6^{CD}(m_i), l_7^{CD}(m_i)\}$. Moreover, there must be a routing bend exists besides m_i if a net intersects with the pair $\{l_1^{CD}(m_i), l_2^{CD}(m_i)\}$, $\{l_3^{CD}(m_i), l_4^{CD}(m_i)\}$, $\{l_5^{CD}(m_i), l_6^{CD}(m_i)\}$ or $\{l_7^{CD}(m_i), l_0^{CD}(m_i)\}$.

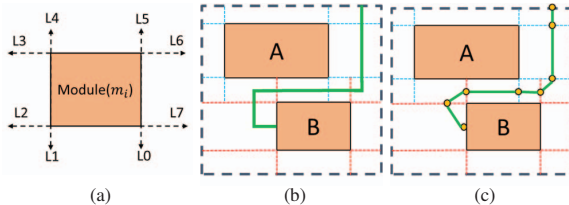


Fig. 1. An example of using CDL in layout migration [17]. (a) feature of CDL (b) the routing and placement in given layout design (c) the routing behavior recorded by CDL.

Property 1 and Property 2 define that each module m_i exists eight edges pointing to four directions by pairs. Property 3 shows that the routing behavior such as bending can be captured by decoding the number of CDLs. CDL model performs a good performance in layout migration problem to record the routing and placement style in the previous layout design. Fig. 1 demonstrates the concept of CDL and the application in the layout migration. The routing information can be recorded on the edge of CDL if there exists some nets crossing over it. The crossing points (yellow points in Fig. 1(c)) are introduced to record the routing information such as routing direction and net width.

C. Problem Formulation and Overall Flow

Fig. 2 illustrates the overall flow of our routing resource estimation and placement refinement. The overall flow can be divided into two major stages. In the first stage, routing resource estimation, it constructs the global routing under a compact placement and estimates the routing resource. The second stage, placement refinement and detailed routing, follows the routing resource estimation and the global route result from the first stage to refine the placement. The detailed routing is then completed rapidly with layout migration technique.

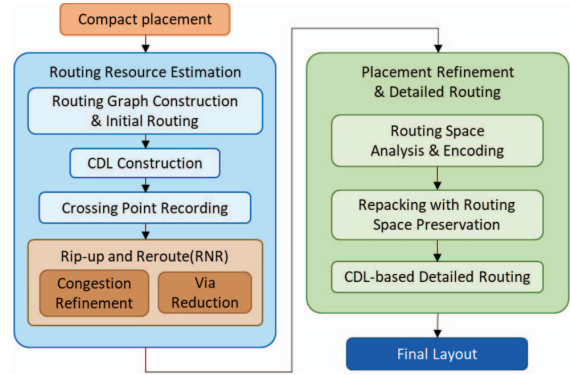


Fig. 2. Overall flow of the proposed analog layout methodology

The problem formulation of this work is defined as follows. Given a netlist, a set of module information and a compact placement, the proposed analog layout methodology will complete the routing task with the placement refinement technique. The proposed methodology will minimize the layout area, wirelength and the number of vias to prevent performance loss and reduce the manufacturing costs.

III. PROPOSED ANALOG ROUTING ESTIMATION FLOW

Given a compact placement, we first construct an edge-based routing graph for the routing estimation problem. After that, we perform an initial routing to evaluate the routing resource and number of vias under the compact placement. Then the CDL routing model will be constructed on the compact placement to keep the correlation between blocks and nets. The crossing points are also created to record the routing information. Finally, the rip-up and reroute with congestion refinement and via reduction is performed to redistribute routing resource and optimize the routing result. The details of each step will be introduced in the following subsections.

A. Routing Graph Construction and Initial Routing

In analog layout design, the routing nets that cross over the active region of devices are forbidden because the analog

devices are sensitive to the noise emerging on the nets. Therefore, devices in analog layout are often treated as obstacle during routing process, and the nets can only be routed between analog devices.

In order to estimate accurate routing resource under the compact placement, finding legal routing tracks are important to preserve sufficient routing resource in the analog placement. In this paper, we apply an edge-based routing graph in our framework. The routing graph construction is illustrated in Fig. 3. Given the compact placement in Fig. 3(a), we analyze the relationship between each block. In this work, the devices in the analog layout are represented as rectangles. We can use the corner coordinates of each block to construct the nodes of the graph. The node and their neighbor nodes that have same y-axis or x-axis can be connected together if the edge connecting these two nodes does not overlap with any devices in the placement. Fig. 3(b) shows the routing graph after construction. The edges in the graph can be regarded as the legal routing tracks.

To increase the accuracy of routing resource estimation, we apply real routing algorithm to find the exact routing path for each net instead of using rough routing estimation model. For the multi-terminal routing problem, minimum spanning tree (MST) and Steiner minimum tree (SMT) are often employed in the routing problem. The SMT generates shorter wirelength than MST. However, there exists a lot of obstacles in the analog layout design. The paths generated from SMT often cross over the obstacles, which requires an alternative routing solution. In this work, we apply Prim's algorithm on MST instead of SMT to construct the routing tree. Then the multi-terminal nets can be decomposed into a set of two-pin nets. After the MST generation, we perform A* search algorithm to find the shortest path for each two-pin net on the routing graph. Fig. 3(c) demonstrates an example for the routing.

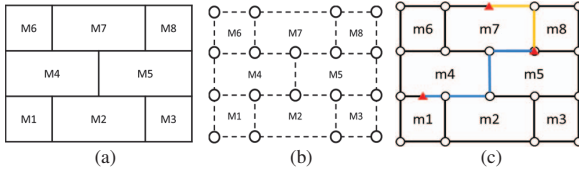


Fig. 3. Illustration of routing graph construction. (a) compact placement (b) edge-based routing graph. (c) MST routing result.

B. CDL Construction and Edge Encoding

In order to record the correlation between blocks and routing behaviors effectively and efficiently, we apply CDL proposed in [17] to record the information. Before recording the initial routing on the CDLs of each block, the conversion of graph node to CDL model is necessary. However, the relationship between nodes and edges is unknown. Hence, we propose an edge encoding algorithm to encode CDL on the routing graph. Fig. 4 illustrates the example of edge encoding in this work. For each edge, it has the source CDL information (*Src*) and the destination CDL information (*Dst*). The *Src* and *Dst* of the edge follow the left-to-right and bottom-to-top rule in the encoding, as shown in Fig. 4.

Furthermore, there must be some edges that do not exist independently. They are overlapped by two modules. Fig. 4 shows an example that the top edge of module m_2 , which overlaps with the bottom edge of module m_1 , is divided into two segments. For this kind of situation, the overlapping segment is possible to encode the *Src* or *Dst* with more than

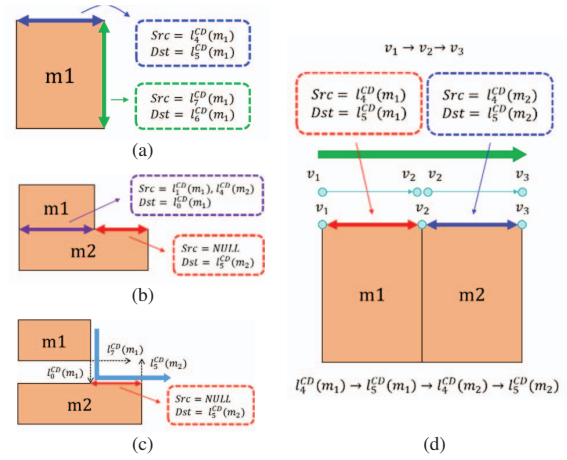


Fig. 4. Illustration of edge encoding (a) Single edge (b) The edge is divided into two sections due to overlap. (c) Example of encoding *NULL* in the edge. (d) Crossing point conversion.

one CDLs, and the *Src* or *Dst* of non-overlapping segment is allowed to be *NULL*. The reason to encode the edge with *NULL* is shown in Fig. 4(c). If there is a routing path going through $l_7^{CD}(m_1)$ and $l_5^{CD}(m_2)$, there must be a routing bend based on the definition of CDL. And the routing path after the routing bend will not intersect with any CDL before it reaches $l_5^{CD}(m_2)$. Therefore, the *Src* of the second segment in the top edge of m_2 should be encoded as *NULL*.

C. Crossing Point Recording

After encoding the initial routing on the CDL model, the crossing points are then created to record the routing information. One net will be separated into several segments based on the connecting order of crossing points. Fig. 4(d) gives an example. A routing path goes through m_1 and m_2 from left to right, passing the vertex v_1 , v_2 and v_3 . This vertex array can be represented by two segments, $v_1 \rightarrow v_2$ and $v_2 \rightarrow v_3$. These two routing segments correspond to two edges in the routing graph. As a result, we can create the crossing points on the CDL model from the edge information based on the routing path direction.

Besides having only one element in *Src* or *Dst*, it is possible to have two elements, as mentioned in Section III.B. Therefore, the order of these crossing points is important because the routing will follow the order of the crossing points to construct the routing path. To determine the order, we need to consider the previous edge e_p that has been added into the routing path and the next edge e_n that is ready to be added. The overlap point of these two edges is called p_o , another point of e_p is called p_p , and the last one is called p_n . Three different situations of the recording orders of crossing points are stated as follows:

- If p_p overlaps with module m_i but does not overlap with module m_j , and p_n overlaps with both m_i and m_j , p_o should create the crossing point on the $l_k^{CD}(m_i)$ first followed by $l_k^{CD}(m_j)$. $\{ 0 \leq k \leq 7 \}$, as illustrated in Fig. 5(a).
- If p_p overlaps with both module m_i and module m_j , and p_n overlaps with m_i but does not overlap with m_j , p_o should create the crossing point on the $l_k^{CD}(m_j)$ first followed by $l_k^{CD}(m_i)$. $\{ 0 \leq k \leq 7 \}$, as illustrated in Fig. 5(b).
- If both $l_k^{CD}(m_i)$ and $l_k^{CD}(m_j)$ exist in p_p and p_n , the recording order of p_o does not matter.

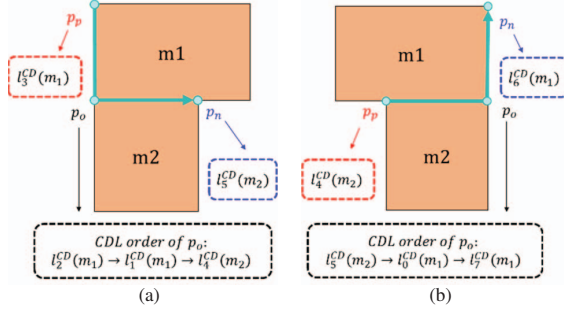


Fig. 5. Illustration of recording the crossing points with an ordered relationship. (a) first situation (b) second situation.

D. Rip-up and Reroute

To reduce the parasitic effects generated from routing and to avoid the overhead on layout area expansion caused by the routing resource, we apply rip-up and reroute with congestion refinement and via reduction to decrease the wire load and reduce the area expansion. In this subsection, we first provide the definition of congestion in this work and the crossing detection methodology. Then the details of the rip-up and reroute will be introduced.

1) Congestion Detection in the Compact Placement

Because the global routing is applied on a compact placement in this work, the problem of routing congestion is different to the conventional congestion issue. In this work, the routing congestion will be treated as a penalty on expanding area for more routing resource. Even in the compact placement, there still exists some of the whitespaces. Some of the white spaces are enough to route the net without paying extra penalty on total area. Therefore, we classify the routing edges into two categories, inner edge and white edge, which represent the edge with zero width and the edge with non-zero width respectively. Fig. 6 gives an example of white edge and inner edge. The inner edges will have larger congestion penalty than white edges because the area will be expanded if some nets route on the inner edge.

2) Crossing Detection through CDL

To decrease the parasitic effects generated from the wirelength and vias, the net crossing issue needs to be considered in global routing stage with via reduction techniques. Therefore, this paper also considers the net crossing issue to enhance the accuracy on estimating routing resource. On the CDL model, we can use the encoding number to recognize the crossing. An example is illustrated in Fig. 7. There is a net going through $l_6^{CD}(m_2)$ and $l_7^{CD}(m_1)$ in the beginning. When traversing to e_n , the direction of e_p and e_n

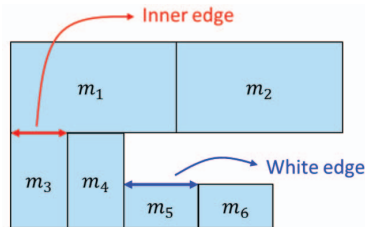


Fig. 6. The illustration of inner edge and white edge. If the net routes through the inner edge, the modules need to be shifted to remain some spacing for the net, which might expand the layout area. In contrast, the layout area will not be expanded if the net routes through the white edge which has enough spacing.

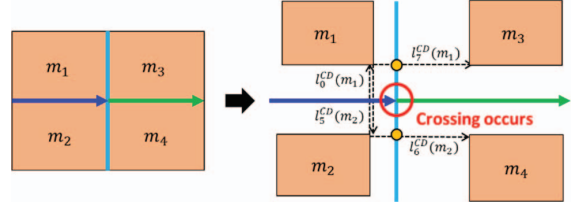


Fig. 7. Illustration of crossing detection through CDL

are the same, and the Dst of e_p has two CDLs. These factors will incur the crossing detection. Afterwards, $l_5^{CD}(m_2)$ will check $l_6^{CD}(m_2)$ and $l_0^{CD}(m_1)$ will check $l_7^{CD}(m_1)$. If both of their corresponding CDLs have the same net information, a crossing occurs if we add the edge e_n on the routing graph.

3) Rip-up and Reroute with Congestion Detection and Via Reduction

After detecting possible congestion and net crossing, we apply rip-up and reroute to consider the area penalty and via reduction simultaneously. To evaluate the routing quality, those two effects are added into the cost function of routing engine. Different from the original A* search algorithm used in the initial routing stage, we redefine term $g(n)$ as $g'(n)$ in our cost function to consider the congestion and crossing penalty. Two user-defined parameters, α and β are provided for designers to decide the priority on layout area and via reduction respectively. The $g'(n)$ is defined as follow:

$$g'(n) = (h(n) - h(n-1)) \times (\alpha \times cg + \beta \times cs) \quad (1)$$

where cg represents the congestion, cs represents the crossing and $h(n) - h(n-1)$ represents the edge length. If cs is higher than cg , the algorithm tends to reduce the number of vias to prevent the performance loss. In contrast, the layout will keep more compact if the cg is higher than the cs .

In the rip-up and reroute process, we sort all the nets in the decreasing order based on the initial cost of each net. Then, the net with the highest cost is picked up to do the rip-up and reroute with the modified A* search algorithm. If the new cost is greater or equal to the initial cost during the rip-up and reroute, the new path will be removed. Furthermore, we also set a wirelength constraint for each net to restrict solution spaces. In our experiment, the wirelength constraint is 1.5 times longer than the original wirelength to avoid the excessive wirelength due to the pursuit of minimum cost. The rip-up and reroute will be terminated if there is no net that can further reduce the routing cost.

IV. PLACEMENT REFINEMENT AND DETAILED ROUTING

To preserve sufficient routing resource in the placement, we transform the routing resource recorded in CDL model into SP and repack a routability-aware placement. In the proposed flow, this can be done easily because the routing resource in CDL model can be added on the edge of SP as a weight. During repacking with HCG and VCG, the position of modules will be shifted automatically by adding the routing resource recorded on the edge.

A. Routing Space Analysis and Encoding

According to the CDL definitions, the CDL for each module will point to the neighbor modules. This property is similar to the HCG and VCG of SP that record the topological relationship with neighbor modules. Therefore, the routing

resource on each CDL can be kept in the corresponding edges of HCG and VCG for preserving routing resource. However, there exists more than one CDLs that can be mapped to the edges of HCG and VCG. To guarantee that the placement after expansion has sufficient routing resource, we will find out the maximum routing resource on CDLs that map to the same edge and record the value on the edge. Fig. 8 and Fig. 9 demonstrate an example of routing space analysis and edge encoding of HCG and VCG. In Fig. 9, m_4 and m_6 have the vertical constraint. It requires two tracks for the red net and green net between m_4 and m_6 . However, $l_1^{CD}(m_4)$ only record one net. Therefore, the CDL $l_0^{CD}(m_4)$ has to record two nets to provide sufficient routing resource.

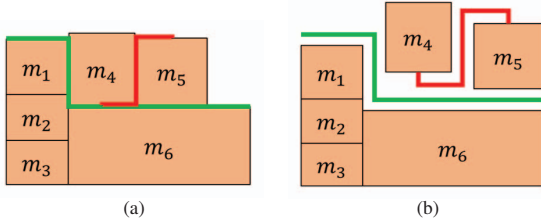


Fig. 8. Illustration of repacking the sequence pair with routability consideration. (a) Compact placement with two routing nets estimation result (b) Repacking the placement considering routing spaces for two nets.

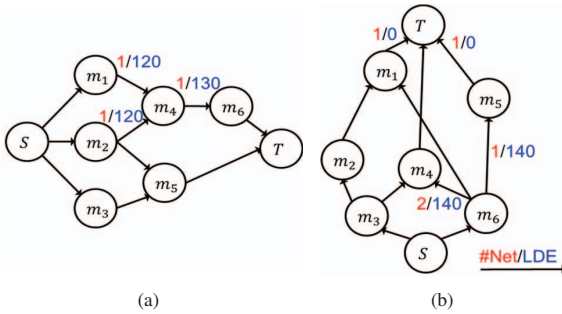


Fig. 9. Illustration of weighted HCG and VCG. (a) HCG (b) VCG

B. Repacking with Routing Space Preservation and LDE Consideration

In order to prevent the performance loss from LDE effects, the spacing between devices is sometimes larger than the minimum spacing in design rules. As demonstrated in Fig. 9, the required spacing from LDE effects can also be recorded on HCG and VCG, which is similar to routability consideration. After the minimum spacing for LDE constraints is obtained from [18] or other similar works, the spacing between devices is determined as the maximum value between routing resources and LDE. While deciding the module location using SP packing algorithm, the edge weights are added as additional distance from previous placed module. The overall time complexity for packing is still as efficient as previous one because those edge weights do not change the computation flow.

C. CDL-based Detailed Routing

Because all the routing information are kept in the CDL model, the detailed routing of the proposed methodology can be completed by a quick routing migration. Therefore, we invoke the routing reconstruction method proposed in [17] to complete the routing without complicated detail routing algorithm. Jog insertion and peak error elimination are also applied to refine the routing quality.

V. EXPERIMENTAL RESULTS

The proposed routing resource estimation and placement refinement algorithm are implemented with C++ on the Linux platform to perform the following experiments on the workstation with Intel Xeon 2.4GHz CPU and 32GB memory. The layout is generated on Synopsys Laker [19] with the TCL script file generated from our program. To further compare the circuit performance, the post-layout simulation results with HSPICE are also provided.

To demonstrate the effectiveness of our methodology, we implement [11] and compare with the proposed approach. The commercial tool, Laker, is applied to complete the detailed routing task for [11]. Because [11] first applies simulated annealing to generate a compact placement then expands the placement later. It is weird to compare the accuracy of routing estimation model on different placement. To be a fair comparison, we skip the SA process in [11] and use the same compact placement as input for two methodologies. The folded cascode operational amplifier (OPAMP) and variable-gained amplifier (VGA) are selected as the test case in our experiment. OPAMP and VGA are designed in TSMC 90nm technology and UMC 65nm technology respectively. TABLE I shows the design information of the test circuits.

TABLE I. CIRCUIT INFORMATION

Circuit	# Blocks	# Nets	# Pins
OPAMP	11	16	37
VGA	35	147	119

A. Layout Comparison

We first compare the difference between the routing resource estimation stage and detailed routing. [11] used B*-tree to represent the analog placement and introduced dummy nodes to preserve the spacing for each net. They adopted FLUTE to guide their routing resource estimation process and iteratively adjusted the dummy nodes to reduce the congestion in the placement. Therefore, we assume that the wirelength calculated by FLUTE is their estimation result. In our work, the estimation result is obtained from real global routing. Using different estimation model, the placement expansion will become quite different thus resulting in different area. Because there is no global routing information from [11], we use Laker auto route to complete detailed routing for the result in [11] after placement expansion. In contrast, the proposed flow applies CDL-based layout construction framework [19] to preserve the global routing results.

TABLE II shows the layout area (Area), wirelength of routing resource estimation (ER) and wirelength of detailed routing (DR) for two methodologies in OPAMP and VGA respectively. Because [11] and Laker work individually, the routing resource preserved by [11] cannot be well utilized by Laker. So the wirelength difference (Diff) and total area of [11]

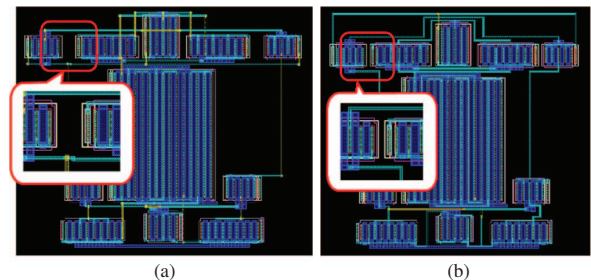


Fig. 10. Illustration of layout comparison. (a) [11] with Laker. (b) This work

are larger than that of our work. This shows that the proposed routing resource estimation methodology and placement refinement technique are effective on estimating accurate routing resource for analog placement. Fig. 10 provides the layout comparison between [11] with Laker and this work. In the enlarged region, the preserved routing resource between two devices is not used by the detailed router in the layout of [11], which leads to extra area waste. In contrast, our layout does not preserve routing space between these two devices due to accurate routing prediction. The runtime of both methodologies are less than 1 second even though extra placement refinement is required in the proposed approach. It shows that our approach will not incur too much additional overhead on layout generation.

TABLE II. LAYOUT COMPARISON BETWEEN [11] AND THIS WORK

Test case	Layout	Area (μm^2)	ER (μm)	DR (μm)	Diff (%)
OPAMP	[11]	480.34	183.64	203.84	+11.0
	This work	440.98	220.43	215.66	-2.2
VGA	[11]	9752.1	1662.1	2178.3	+23.7
	This work	9707.3	2288.3	2263.2	-0.9

B. Performance Comparison

We also provide the performance comparison between the layouts generated from different approaches. The performance metrics of OPAMP and VGA are gain (A_v), gain bandwidth (BW), phase margin (PM) and power consumption (Power). The post-layout simulation results of two test cases are shown in TABLE III. In the performance comparison of OPAMP, the gain of two methodologies have similar performance. However, in the frequency domain, the performance of our work is better than that of [11] with the same power consumption. Similar to the OPAMP result, our work has better performance in gain and bandwidth than that of [11], even the power consumption is slightly higher than that of [11] in the result of VGA. This shows that the proposed analog layout automation technique can generate a layout with better post-layout performance than previous works.

TABLE III. PERFORMANCE COMPARISON BETWEEN [11] AND THIS WORK

Test case	Layout	A_v (dB)	BW (MHz)	PM (deg)	Power (μW)
OPAMP	[11]	45.12	116.50	57.13	112.4
	This work	45.08	119.63	60.37	112.4
VGA	[11]	17.9	6.8	92.8	546.6
	This work	18.8	7.4	88.4	573.8

VI. CONCLUSIONS

In this paper, a routing methodology is proposed to provide accurate routing resource under compact placement. A placement refinement technique is also developed to efficiently adjust the placement considering routability. Different from the previous works that do the routability-aware placement and detailed routing independently, we can feedback an accurate routing information to the placement, and use layout migration technique to perform the detailed routing on the adjusted placement. This may reduce the difference between global routing and detailed routing in traditional approach to avoid unnecessary waste or expansion of routing space. In addition, routing congestion and via reduction are also considered in our work to reduce the performance loss after layout. As shown in the experiment, the difference of global routing and detailed routing of this work can be reduced to only 1%, which is able to keep the

optimal routing resource. Therefore, the proposed approach not only has better circuit performance, but also has smaller layout area than previous works.

REFERENCES

- [1] P.-H. Lin, Y.-W. Chang, and S.-C. Lin, "Analog placement based on symmetry-island formulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 6, pp. 791–804, Jun. 2009.
- [2] Q. Ma, L. Xiao, Y. Tam, and E. F. Y. Young, "Simultaneous handling of symmetry, common centroid, and general placement constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 1, pp. 85–95, Jan. 2011.
- [3] 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 Transactions on ComputerAided Design of Integrated Circuits and Systems*, vol. 33, no. 6, pp. 879–892, Jun. 2014.
- [4] H.-C. Ou, K.-H. Tseng, J.-Y. Liu, I.-P. Wu, Y.-W. Chang, "Layout-dependnet Effects-aware Analytical Analog Placement" *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 8 pp. 1243–1254, Aug. 2016.
- [5] A. Patyal, P.-C. Pan, A. K. A. H.-M. Chen, and W.-Z. Chen, "Exploring Multiple Analog Placements with Partial-Monotonic Current Paths and Symmetry Constraints using PCP-SP", *IEEE Transactions on ComputerAided Design of Integrated Circuits and Systems*, early access, Feb. 2020.
- [6] H.-Y. Chi, C.-N. J. Liu, H.-M. Chen, "Wire Load Oriented Analog Routing with Matching Constraints" *ACM Transactions on Design Automation of Electronics Systems*, vol. 25, no. 6, Aug. 2020.
- [7] H.-C. Ou, H.-C. C. Chien, and Y.-W. Chang, "Non-uniform multilevel analog routing with matching constraints." in *IEEE/ACM Design Automation Conference*, 2012.
- [8] M. Ozdal and R. Hentschke, "Algorithms for maze routing with exact matching constraints." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 1 pp. 101–112, Jan. 2014.
- [9] M. Torabi and L. Zhang, "Electromigration- and parasitic-aware ILP-based analog router." *IEEE Transactions on Very Large Scale Integration System*, vol. 26, no. 10, pp. 1854–1867, Jun. 2018.
- [10] C. Chu and Y.-C. Wong, "FLUTE: Fast lookup table based rectilinear Steiner minimal tree algorithm for VLSI design," *IEEE Transactions on ComputerAided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 70–83, Jan. 2008.
- [11] C.-W. Lin, C. Lu, J.-M. Lin and S.-J. Chang, "Routability-driven placement algorithm for analog integrated circuits," in *International Symposium on Physical Design*, NY, United States, 2012.
- [12] H. Zhou, C.-W. Sham, H. Yao, "Revisiting Routability-Driven Placement for Analog and Mixed-Signal Circuits" *ACM Transactions on Design Autoamtion of Electronics Systems*, vol. 23, no. 2, Sep. 2017.
- [13] X. Tang, R. Tian, and D.-F. Wong, "Fast Evaluation of Sequence Pair in Block Placement by Longest Common Subsequence Computation," in *IEEE Design, Automation, and Test in Europe*, 2000.
- [14] C. Kodama, K. Fujiyoshi and T. Koga, "A Novel Encoding Method into Sequence-pair," in *IEEE International Symposium on Circuits and Systems*, pp. 329-332, 2004.
- [15] K. Krishnamoorthy, S. C. Maruvada, and F. Balasa, "Topological placement with multiple symmetry groups of devices for analog layout design," in *IEEE International Symposium on Circuits and Systems*, May 2007, pp. 2032–2035.
- [16] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Vlsi module placement based on rectangle-packing by the sequence-pair," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 12, pp. 1518–1524, Dec. 1996.
- [17] H.-Y. Chi, Z.-J. Lin, C.-H. Hung, C.-N. Liu and H.-M. Chen, "Achieving Routing Integrity in Analog Layout Migration via Cartesian Detection Lines," in *IEEE/ACM International Conference on Computer-Aided Design*, 2019.
- [18] X. Li, Z. Ye, Y. Tan, and Y. Wang, "A two-dimensional analysis method on STI-aware layout-dependent stress effect," *IEEE Transactions on Electron Devices*, vol. 59, no.11, pp. 2964 - 2972, 2012.
- [19] Laker™ from Synopsys, <http://www.synopsys.com>