An Automated Flow Integration to Help Analog Layout Design Migration

Jou-Chun Lin, Po-Cheng Pan, Ching-Yu Chin and Hung-Ming Chen
phoebezaq@gmail.com, saxdebreeze@gmail.com, alwaysrain.GR@gmail.com and hmchen@mail.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan

***Abstract*—The development of the computer-aided-design (CAD) tools for digital circuits has been perfected for these years. However, the CAD tools for analog circuits still remains a great deal of challenges. Since the size of transistors scales down as the process technology advances, design migration problem takes place to increase the degree of layout reusing. With previous work such as placement migration and routing preservation tool, further performance boost becomes the next step. We aim at the width of wires that impacts resistance and capacitance of wires so as to improve the performance. We implement a flow, which can further improve the performance, generate the modified layout automatically and pass the verification check, to speed up the analysis process or design flow by adjusting the wire width. We apply greedy heuristic and simulated annealing algorithm in our framework. Our flow can help with the analog layout synthesis flow in more efficient way.**

I. Introduction

The current trend of IC design focuses on system on chip (SoC) design, which represents integrating digital and analog circuits. As the process technology advances, the size of transistors scales down. Migrating the design into another process technology leads to complicated problems due to the difference of limitation of respective technology. In order to decrease the time spending on design flow, design reusing becomes an important issue.

For layout migration, we have the method [1] that handles layout consisting of Pcells (Parametric Cells) [2] for migration-friendly purpose and another method [3] that preserves users routing information to reduce design cycle time and simultaneously produce reasonable performance in migrated design. Building a comprehensive tool for analog design that takes design rule check into account is a goal for CAD tools. Since the tools for placement migration and routing preservation already exist, study the performance becomes feasible based on those tools. We believe the performance might be dominated by wire resistance and capacitance. Therefore, we try to construct a program that is able to adjust the width of wires in layout to get a better performance on design migration framework.

A. Previous Works

Layout reusing problem has been widely studied recently. Placement and routing are two primary issues that most researches focus on. For the placement problem, most methods put emphasis on compaction [4–6] and symmetry island generation [7]. The method, analog layout prototyping for design migration, in [1] preserves layout hierarchy, symmetry and matching structures of the original design with Defer [8]. In addition, it generates multiple placement results so that it is capable of providing feasible solution with multiple layouts.

On the other hand, for layout routing reusing problem, the approach in [3] proposes a routing preservation algorithm based on Constrained Delaunay Triangulation (CDT) [9]. After finding the correlation among layout wires by decomposing routing channels with CDT method, new routing on the target technology can be rebuilt quickly by the routing information, saved in CDT triangles, of the same placement topology of the original technology.

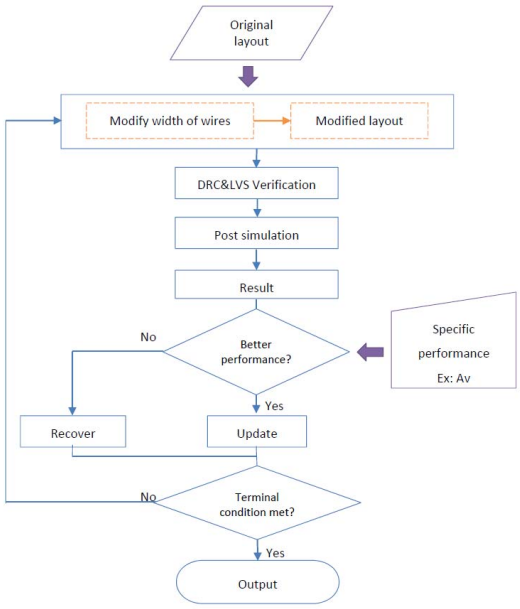

Fig. 1. Overall flow.

Research in [7] handles the placement problem by treating the symmetry pair as a symmetry island which will form a structural template to preserve the desired information. The horizontal and vertical constraint graph will be built by scanning the layout. And the template will be solved by using linear programming [10]. A hierarchical slicing-tree structure is adopted by [11] to provide a solution of placement with different perspective. Besides, [12] take the routability into consideration in the placement stage. As for routing issue, some works [13–15] use maze-style routers to handle symmetry and non-symmetry circuits. Another kind of router, channel router, is proposed in [15, 16] to take care of detailed routing and symmetry issue.

B. Our Contribution

For design migration, previous works implement the placement and routing reusing and retargeting. However, in the aspect of routing, manual routing still takes place during the flow. In general, there is a fixed size block for every functional layout. In order to keep the CAD tool function correctly, reserving a loose bond for blocks is reasonable. Because of

the existence of spare space, it is feasible to adjust the wires width so as to increase the specific performance.

Based on this observation, we propose a flow that can further improve the performance right after the routing has been settled. With target technology design rules, placement, wires and pin connection information, the flow is sufficient to carry out the desired layout automatically. Due to the spare space, the modification, which is normally space-consuming, during our flow is allowed to practice. It is vital that the layout area will be within the fixed area and without impacting the block size. Furthermore, we adopt two methods to maximize the effect on post-simulation result. The two approaches, greedy and simulated annealing [17] algorithm respectively, are used for analysis of the performance in optimization work.

In summary, the flow is able to generate the modified layout automatically. After this flow is constructed, adjusting the width to obtain a new layout for performance analysis becomes an automatic job. The rest of the paper is organized as follows. We propose our flow and basic approaches in Sections II and III. Section IV presents the optimization. Section V shows our results and Section VI concludes this work.

II. Overall Flow

The overall flow is illustrated in Fig. 1. We obtain the layout information of Pcell and wires at the beginning. Then we adjust the width of wires with the method which is either greedy heuristic or simulated annealing, and generate the corresponding new layout. We then extract actual layout information to run DRC and LVS verification. Passing these two verification stands for the correctness of wire width modification. Lastly we use hspice to run post-simulation in the following step and check the result. If having a better performance, the program will keep this layout; otherwise, it will recover from the present layout. We check whether the termination condition of the chosen method is satisfied at the end of the loop; if no, go back to the step that modifying wire width; otherwise, the flow will end by outputting the layout stored.

III. Basic Approaches

A. Adjusting wire width

The overall flow for adjusting wire width is portrayed in Fig. 2. Establish the correlation in horizontal and vertical respectively in each different layer. For every layer, cell area should avoid having routing wires on it because of the possibility of crosstalk and unexpected error taking place. Check the space left for horizontal (vertical) wires on upper and lower (left and right) direction to have the preferred direction and avoid expanding the routing area meaninglessly. Modify the wire width and check whether any congestion occurs. Once congestion happens,detect the volume that should be moved, push the congested wire or cell away to pass DRC verification and execute this step recursively until no congestion exists. However, pushing the wire or cel away directly may destroy the symmetry constraint the layout originally have. If the pushed cell is one of the symmetrical pair, the other part of the symmetrical pair should be also pushed away with the same distance but the contrary direction. Fig. 3 demonstrate an example for conducting the congestion elimination recursively. After the elimination or no congestion occurs during the widening process, check whether there are more nets should be widened, if yes, go back to the step to widen the chosen wire and continue the flow; otherwise, wire width changing flow is finished.

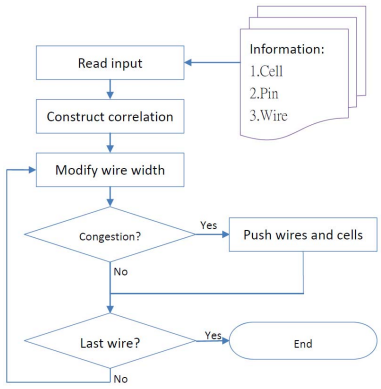

Fig. 2. Flow for adjusting wire width

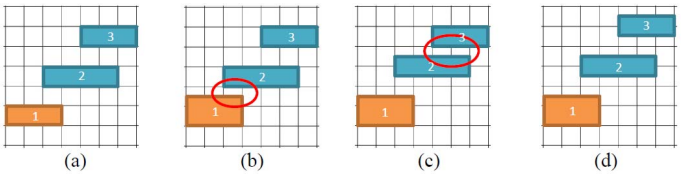

Fig. 3. Example of recursively pushing. (a)original layout. (b)widen wire 1 and congestion occurs. (c)push wire 2 away and another congestion between wire 2 and 3 occurs. (d)push wire 3 away and become the new layout.

B. Formula-Based Approach

The method of our original thought is different from the one adopted now. We want to build a program that carry out the outcome with the circuits performance formula as the cost function. The formula will be expressed in a lot of variables, such as transconductance of MOSFET and resistance of nets, and each variable has its own constraint. We use computer to analyze the formula and get the trend analysis. Based on the analysis, we can sort all wires into an order. The wire with higher priority represents that the wire has more influence on the chosen performance, which brings the consequence of getting the most improvement with the least change. We expect the formula will lead us to a result that maybe is not anticipated perfectly but having the right trend to better performance. Using the cost function instead of post-simulation to forecast the result can save lots of time since DRC and LVS verification run only once during the program. For the circuit described in Fig. 4, we obtained its performance formula of gain as follows:

$$Av = g_{m46}(R_{o43} \parallel R_{o37})$$
$$\text{where } R_{o43} = (1 + g_{m43}r_{o43})(r_{o42} + R_{I42I43_1}) + r_{o43},$$
$$R_{o37} = (1 + g_{m37}r_{o37})[R_{I37I38I46_1} + (r_{o46} + R_{I37I38I46_3}) \parallel (r_{o38} + R_{I37I38I46_2})] + r_{o37},$$
$$r_o = \frac{1}{gds},$$
$$g_m = g_m + g_{mb} \text{ and}$$
$$R = resistance\ of\ the\ wire$$

r_o and g_m can be extracted in the spice result. We assume that only resistance of the wires are variables and found out that gain is only influenced by four nets based on the formula and it violated the truth that all nets impact the performance and the four nets are even not the most significant ones. In fact, the bias point is changed after the modification so that the g_m value will also change. Because the number of variables becomes big and the verification and simulation cannot be skipped in this circuit, we need to extend this method.

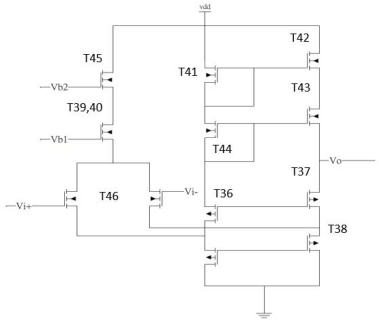

Fig. 4. The schematic of the folded cascode operation amplifier

IV. Optimization

(1)Greedy Heuristic

This algorithm is normally conducted in an intuitive way. Not except for this program, choose the net which has the best performance in every loop. By executing the program, we learn that the change of performance is really tiny. If choosing the wire arbitrarily, the sum of increase of every loop will be miscounted so that the real optimization cannot be done and the condition should be avoided. Therefore, choosing the wire that gets the best result after widening a unit is necessary for optimization.

To be precise, we first run simulations of every single net widening a unit in every loop and choose the one with the best performance according to the result of post-simulation. It is obvious to figure out that this method is time-consuming especially when the number of wires is big. The method of optimization will stop at the loop number reaching its settled target or no better neighbor existing. If both settled target and number of wires are not big, this way will be a reasonably good way.

In addition, we assume that greedy heuristic may lead the outcome to local optima rather than global optima. In order to jump out of the local optimum solution, we change the initial point by widening some wires randomly at the beginning. We execute the program several times under this assumption only to find the performance outcome is inferior to the original way. Despite the fact that no better result is found, it is possible that the number of widened wires for changing initial state is not suitable or changing these wires randomly instead of trying all the possibility results in that the desired initial state is not faced. These possible conditions exist and we remain the expectation that the better initial point may be found in the future.

(2)Simulated Annealing

Since the block outline is fixed, the maximum change, the sum of all wires width widening, should be also settled to assure that the new layout will not exceed the block size. To implement this principle, we can give all wires of the layout the maximum change in average. We choose two wires randomly and increase ones widening width while decreasing anothers width with the same degree to generate the neighbor. We then check whether the result is an acceptably good by the current temperature with the formula of simulated annealing, if yes, record and update the layout; otherwise, recover the layout. This algorithm will stop at the termination temperature is reached and the result of this optimization is generated by outputting the layout directly. The detailed formula is as follows: The possibility P to accept a solution:

TABLE I
Performance comparison by formula-based approach

choice	power dissipation(uW)	gain	bandwidth(MHz)	phase(deg)
$R_{I37I38I46_1}$	118.6962	**146.6002**	107.0622	57.8295
$R_{I37I38I46_2}$	118.6962	**146.6346**	107.0686	57.8301
$R_{I37I38I46_3}$	118.6961	**146.6545**	107.0644	57.8300
R_{I42I43_1}	118.6961	**146.6625**	107.0619	57.8307
Greedily choose	118.6972	**146.7278**	107.0642	57.8293

$$P = \begin{cases} 1, if \Delta > 0 \\ e^{\frac{\Delta}{T}}, if \Delta \leq 0 \end{cases}$$

where Δ = new performance value old performance value
T= temperature for annealing schedule

V. Experimental Result

Our work is implemented in the C++ programming language on a 2.4GHz Intel Xeon Quad Core machine under the Linux operation system with 32G memory. This experiment is performed on different topology, which is generated by design migration framework, on a folded-cascode operational amplifier. Some wires are changed from metal 1 to metal 3 to avoid the interference of inner connection of MOSFET.

The following sections will show the wire adjustment in a folded-cascode operation amplifier, the layout of the schematic depicted in Fig. 4, with different approaches. The experimental layout, which is shown in Fig. 5(a), is generated by [3]. Each wire of the layout uses the default width without any modification. The result of formula-based approach is demonstrated in the first section. Then we introduce the result of simulated annealing algorithm so as to choose the best result to compare with the result of greedy heuristic. The result of greedy heuristic is illustrated in the last section. The DRC and LVS verifications are very time-consuming and cannot be skipped in order to assure correctness of post-simulation result. These verifications occupy most of the time spent on the whole flow. Each approach takes about 8 to 12 hours respectively.

A. Formula-Based Approach

We widen the wire by 90nm in the formula listed in section III-B respectively. We apply the greedy heuristic to find the result with the same widening degree. From the observation, the performance of gain-improve case has a greater gain improvement for widening by greedy heuristic than by any wire demonstrated in the formula. The comparison shows that the formula is not specific enough to predict the trend. The simulation result is listed in Table I. As a result, we extend this approach for further experiment.

B. Simulated-Annealing Approach

Every wire is set to be two times width as the default value and the constraint of the new layout is set in the meantime. Then we apply simulated annealing algorithm and the results are in Table II. We can find that the target performance are very similar after the adjustment. Despite the difference is very small, we can still see that we are on the right direction: the gain performance is getting bigger.

C. Greedy Heuristic Approach

We use the 30nm as the widening unit, and simulate every wire in one step. The best performance of all simulations is selected in every step. After increasing the same quantity

TABLE II
PERFORMANCE COMPARISON BY SIMULATED ANNEALING ALGORITHM

	SA iteration	power dissipation(uW)	gain	bandwidth(MHz)	phase(deg)
placement 1	Origin	118.6961	**146.6519**	107.0623	57.8301
	1	118.7002	**146.9896**	108.2500	58.2828
	2	118.7002	**146.9877**	107.8379	58.1545
	3	118.7002	**146.9904**	108.1904	58.2501
	SA iteration	**power dissipation(uW)**	**gain**	**bandwidth(MHz)**	**phase(deg)**
placement 2	Origin	118.6664	**142.4455**	108.0662	58.2766
	1	118.6736	**143.1437**	108.3192	58.4342
	2	118.6736	**143.1438**	108.2518	58.4035
	3	118.6736	**143.1433**	108.1384	58.3671

TABLE III
PERFORMANCE COMPARISON BY GREEDY HEURISTIC

	Type	power dissipation(uW)	gain	bandwidth(MHz)	phase(deg)
placement 1	Origin	118.6961	**146.6519**	107.0623	57.8301
	greedy	118.7002	**146.9921**	108.1770	58.2554
	SA	118.7002	**146.9904**	108.1904	58.2501
	Type	**power dissipation(uW)**	**gain**	**bandwidth(MHz)**	**phase(deg)**
placement 2	Origin	118.6664	**142.4455**	108.0662	58.2766
	greedy	118.6736	**143.1447**	108.2594	58.4040
	SA	118.6736	**143.1438**	108.2518	58.4035

with simulated annealing algorithm, we record the result and compare it with best result in previous table. At the end, we use the results, which have the same quantity change of width, to do the comparison. In Table III, it shows that the performance of greedy heuristic always gets better results. The comparison of layout is shown in Fig. 5. In the gain performance, the results of SA cases can get closer to the results of greedy heuristic, but never better than the results of greedy heuristic.

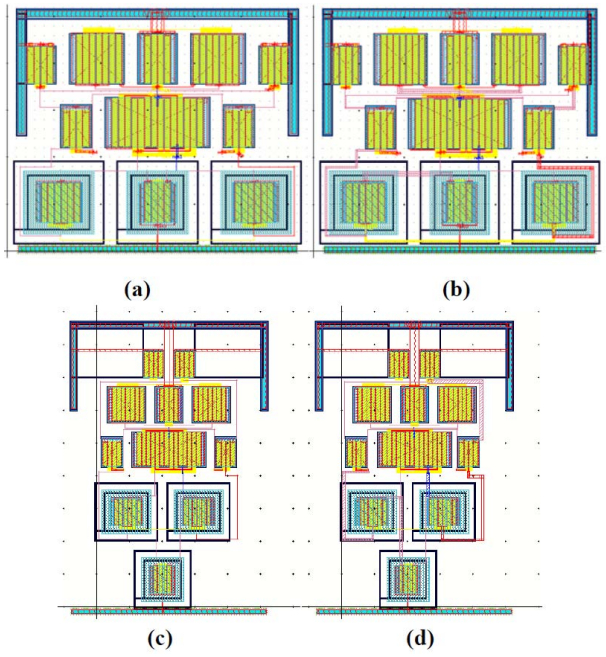

Fig. 5. (a) Original layout for placement 1 (b) Layout for placement 1 after greedy heuristic algorithm (c) Original layout for placement 2 (d) Layout for placement 2 after greedy heuristic algorithm

D. Greedy Heuristic Approach

We choose bandwidth as the target performance. The result is shown in Table IV. The flow stops at 61 and 50 step for two placement respectively because no better neighbor exists. Therefore, greedy heuristic does not get better in bandwidth-aimed performance, but simulated annealing cases result better in bandwidth-aimed performance. Overall, the experiment results on bandwidth have improved.

TABLE IV
BANDWIDTH-AIMED PERFORMANCE COMPARISON

	Type	power dissipation(uW)	gain	bandwidth(MHz)	phase(deg)
placement 1	Origin	118.6961	146.6519	**107.0623**	57.8301
	greedy	118.6971	146.5944	**108.6630**	58.3751
	SA	118.6966	146.5760	**108.6632**	58.3762
	Type	**power dissipation(uW)**	**gain**	**bandwidth(MHz)**	**phase(deg)**
placement 2	Origin	118.6664	142.4455	**108.0662**	58.2766
	greedy	118.6669	142.2578	**108.6590**	58.4872
	SA	118.6669	142.2129	**108.7176**	58.4949

VI. CONCLUSIONS

In this paper, we propose a flow to control the width of wires during the design migration process. It is capable of generating the modified layout automatically which simplifies performance analysis process. The applied circuit in our work is enhanced on its performance via wire width optimization. The appropriate circuit performance is dominated by the resistance and capacitance of the wire, which fits our algorithm to strengthen the performance.Furthermore, we believe that the construction of this flow is more significant if the widen strategy is applied on circuit which is more sensitive on wire parasitics.

REFERENCES

[1] Yi-Peng Weng, Hung-Ming Chen, Tung-Chieh Chen, Po-Cheng Pan, Chien-Hung Chen, and Wei-Zen Chen. Fast analog layout prototyping for nanometer design migration. In *Proceedings of the International Conference on Computer-Aided Cesign*, ICCAD, pages 517–522, 2011.

[2] *OpenAccess Release 2.2 Standard*. Slilcon Integration Initiaive Inc., 2005.

[3] C.-Y. Chin, P.-C. Pan, H.-M. Chen, T.-C. Chen, and J.-C. Lin. Efficient analog layout prototyping by layout reuse with routing preservation. In *Proceedings of the International Conference on Computer-Aided Cesign*, ICCAD, pages 40–47, 2013.

[4] S. Bhattacharya, R. Hartono N. Jangkrajarng, and C.-J. R. Shi. Correctby-construction layout-centric retargeting of large analog designs. In *Proceedings of the Design Automation Conference*, pages 139–144, 2004.

[5] S. Hammouda, H. Said, M. Dessouky, M. Tawfik, Q. Nguyen, W. Badawy, H. Abbas, and H. Shahein. Chameleon art: a nonoptimization based analog design migration framework. In *Proceedings of the Design Automation Conference*, pages 885–888, 2006.

[6] S. Bhattacharya, N. Jangkrajarng, and C.-J. R. Shi. Multilevel symmetry-constraint generation for retargeting large analog layouts. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):945–960, Jun. 2006.

[7] 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*, 28 (6):791–804, Jun. 2009.

[8] J. Z. Yan and C. Chu. Defer: Deferred decision making enabled fixedoutline floorplanning algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29 (3):367–381, Mar. 2010.

[9] J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22:21–74, May. 2002.

[10] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 1984.

[11] P.-H. Wu, M. P.-H. Lin, Y.-R. Chen, B.-S. Chou, T.-C. Chen, T.-Y. Ho, and B.-D. Liu. Performance-drive analog placement considering monotonic current paths. In *Proceedings of the International Conference on Computer-Aided Cesign*, pages 613–619, 2012.

[12] C.-W. Lin, C.-C. Lu, J.-M. Lin, and S.-J. Chang. Routability-driven placement algorithm for analog integrated circuits. In *International Symposium on Physical Design*, pages 71–78, 2012.

[13] J. Cohn, D. Garrod, R. Rutenbar, and L. Carley. Koan/anagram ii: new tools for device-level analog placement and routing. *IEEE J. Solid-State Circuits*, 26(3):330–342, Mar. 1991.

[14] L. Xiao, E. Young, X. He, and K. Pun. Practical placement and routing techniques for analog circuit designs. In *Proceedings of the International Conference on Computer-Aided Cesign*, pages 675–679, 2010.

[15] E. Malavasi, E. Charbon, E. Felt, and A. Sangiovanni-Vincentelli. Automation of ic layout with analog constraints. *IEEE Transactions Computer-Aided Design Integrated Circuits Systems*, 15(8):923–942, Aug. 1996.

[16] U. Choudhury and A. Sangiovanni-Vincentelli. Constraint-based channel routing for analog and mixed analog/digital circuits. *IEEE Transactions Computer-Aided Design Integrated Circuits Systems*, 12(4):497–510, Apr. 1993.

[17] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.