A Technology-Independent Methodology of Placement Generation for Analog Circuit

Wai-chee Wong, Philip C. H. Chan and Wai-on Law*

The Hone Kong University of Science and Technology

*Motorola Semiconductors Hong Kong Ltd.

Abstract

An automatic placement system with emphasis on technology independent methodology and device matching consideration for analog layout design is presented. A novel optimization approach based on circuit partitioning, simulated annealing and branch-and-bound algorithm is proposed to solve the placement problem. The move set used to generate perturbations for annealing is capable of arriving at any topological placement. The branch-and-bound is modified to take circuit performance into consideration. Results of two silicon proven designs generated by the system demonstrate an 8X cycle time reduction as compared to a manual approach.

1. Introduction

In general, layout generation of a circuit is divided into two major stages: placement and routing. Out of these two stages, placement affects the area and performance of a design most critically, particularly for an analog circuit. A poorly placed design always takes up more functional as well as routing area. Even worse, degradation in circuit performance is resulted if devices require matching are not placed properly. Failure to generate a placement satisfying these criteria may lead to costly penalties when the entire layout has to be restructured.

For the importances of placement mentioned above, layout designers must pay extensive attention during placement generation. An automatic layout tool that is capable of relieving the burden on layout designers and shortening design cycle time is therefore necessary.

Jin Xu *et al.* in [1] proposed a cluster refinement method for placing macros and standard cells. The chip area is minimized since a compaction-based approach is adopted. However, the aggressive reduction in die area and the lack of considerations over circuit performance cost a yield-reduction and make this approach not applicable to analog circuits.

Although constructive placement methodologies like min-cut graph partitioning in [2] and schematic position placement in [3] works well for digital designs, but not for their analog counterparts. As compared to digital cells, analog devices are of more arbitrary sizes. Poor area utilization is resulted if the same methodologies are adopted.

In [4], Sutanthavibul *et al.* successfully developed a mixedinteger linear programming formulation for layout problem, which can be solved with standard mathematical software. However, this approach suffers from a long computation time for large circuits as the number of variable and constraint increases. Besides, the objective function in mixed-integer programming must be linear. For this reason it is difficult to take analog constraints into consideration.

Simulated annealing [5] is a probabilistic algorithm widely used as the solving engine in analog placement problem. Analog constraints such as device matching, performance degradation related to parasitic devices and pin locations can be handled by the cost function successfully. In this paper, an automatic placement system which employs a new methodology targeted for an analog design is presented. This placement approach bases on simulated annealing and a modified branch-and-bound algorithm with consideration over the circuit performance. The remaining of this paper is organized as follows: Section 2 gives an overview of the system architecture. Section 3 describes how the system simplifies the design and preserves layout constraints during the placement. The details of the placement algorithm is discussed in Section 4. Experimental results generated by the system are presented in Section 5. Section 6 gives a conclusion of our work.

2. System Overview

The automatic placement system presented in this paper is a design automation tool capable of placing and orienting cells in an analog design at strategic locations. This section introduces the architecture of the system.



Figure 1. Program flow of the analog placement system

The placement tool is comprised of subsystems, each of which handles a particular set of tasks in the design flow being shown in Figure 1.

The inputs to the placement system are a schematic of an optimized circuit and a technology file. Necessary trade-offs between device-counts and functional performance are made on the circuit already. The process-dependent technology file contains design and technology specifications such as design rules and user preference for layout generation.

Devices in the schematic are partitioned into groups, which are annotated by users with relationships provided by the system during device annotation. This step serves to simplify the design and therefore speed up the system. After device annotation, layout of the instances in the design are generated with Devicelevel Editor (DLE)[6], which also set up region for placing cell blocks and places the pins at location according to the schematic.

The device-level placer (DLP) of the analog placement system

places the devices in a group according to its annotated relationship. Once the devices are placed, a protection frame is generated and the whole are replaced by a building block.

The block-level placer (BLP) determines the location and implementation of each block generated from the previous step. Simulated annealing and a modified branch-and-bound algorithm are used together for the search of the optimal placement solution.

After placing the building blocks in a layout in such a way the user-defined cost functions are optimized, the building blocks are replaced by instances generated by DLE again. At last routing tools may be called to complete the rest of a layout generation.

3. Device Annotation

The objective of device annotation is to provide a method that preserves user-defined layout relationships among devices during layout generation. The necessity of annotating layout relationships arises from the fact that process variations do exist. To guarantee that circuit performance and design specifications of a design are satisfied even after fabrication, a number of layout constraints are to be observed during physical design.

3.1 Device Mismatch

Pelgrom et al. [7] related variance of a process parameter *P*, the device area *WL* and separation distance *D* between the devices by:

$$\sigma^2(P) = \frac{a_p}{WI} + s_p^2 D^2 \qquad (\text{eq. 1})$$

where a_p and s_p are process-dependent constants. As it is implied in the equation, the process variation, and therefore mismatch, is minimized as the separation distance *D* approaches zero. Thus, better device matching can be achieved by employing some layout techniques.

3.2 Types of Device Relationship

Three novel relationships are provided by this system. We define each relationship as follows:

- 1. *Unidirectional*: Devices annotated with unidirectional relation are placed in close proximity and in identical orientation, which is determined by the system. For a group of devices to be placed in a unidirectional fashion, they must be of the same cell category and either their lengths or widths are identical. The system rejects an annotation if the requirements mentioned above are not satisfied.
- 2. *Matching*: Four matching configurations are defined in our placement system. The system determines the configuration for an annotated group according to the number of devices in it. If the group of devices does not fall into one of the defined configurations, or if the devices in the group are not of the same cell category, the annotation is rejected.

In addition to placing devices in close proximity, devices that requires critical matching must be placed near the center of the die, where minimum stress gradient is experienced. To take this into consideration, designers are required to assign a *criticality* to each matching group. The value of *criticality* ranges from zero to unity and a value close to unity is assigned to the group which requires maximum matching. The details of implementation is presented in Section 4.

3. *Sharing*: In layout design, it is always more economical to increase the packing density through device-merging. To reduce die area, devices which share common P or N-type region can be merged into a single device structure. However, if every P or N-type region is allowed to be shared, a large

number of constraints must be considered [8]-[9], which in turn leads to intensive calculation during optimization. The choice of the sharing layers also depends on the targeted process and technology. For the BiCMOS library we are using as a test case, the buried layer and epi layer are selected for sharing.

3.3 Device Level Placement

Once the devices in the schematic are annotated, the layout of all instances in the design are generated using DLE. Devices from the same group are placed according to the corresponding relationship. The placed groups are replaced by blocks then. DLP also replaces every single un-annotated device by a block, after individual buried and epi layers are generated for it. During the subsequent operations, devices inside each block are fixed in their orientations and relative positions.

The merit of the device annotation and device level placement approach is that it is totally technology independent. Unlike the procedural module generators [10], parameterized cell generators from any library can be called by the annotation program and DLP. Device annotation therefore reduces the effort required for library development and simplifies the work of process migration.

4. Block Level Placement



Figure 2. Flow chart of Automatic Placement Algorithm

The block level placer searches for the optimal placement and implementation of each block generated from device annotation with the placement algorithm presented in this section. The algorithm consists of two phases: an outer annealing loop and an inner branch-and-bound search, as shown in Figure 2. In the first phase, the relative position of blocks are determined by simulated annealing. In the second phase, the implementation for each block is chosen using the modified branch-and-bound algorithm. The two phases are driven by different evaluation functions. Before we go into the details of the implementation of the algorithm, a short review on the placement representation is presented.

4.1 Placement Representation

A placement can be described by a horizontal and vertical polar graph G(U,E) and H(V,F) [11]. A vertex u in the horizontal polar graph G(U,E) denotes a vertical line segment on the placement. An arc e connects two vertices u_i and u_j if they are the left and right edges of a rectangular block in the placement. This arc is weighted with the width of such rectangle. The graph H(V,F) is

defined similarly by considering the horizontal line segments instead of the vertical line segment.

4.2 Determination of Topological Placement

The relative position of blocks in a layout is determined by simulated annealing [6]. Simulated annealing is a widely used iterative heuristic for solving several combinatorial optimization problems, including the placement problem.

Move Set

Placement solutions are represented by the polar dual and the perturbation of a given placement solution amounts to incurring some changes to its corresponding polar dual. The move set used in the placement system consists of three basic local moves and one global move. They are defined as follows:



Figure 3. (a) Serial-to-parallel and parallel-to-serial (b) vertex splitting and merging (c) vertex shifting

- 1. *Serial-to-parallel*: Any two adjacent edges in series are randomly chosen and transformed to a pair of parallel edges. A parallel-to-serial is performed on its dual. SeeFigure 3(a);
- Vertex splitting: An edge is chosen at random and its sink, V, is spited into V and V*. A vertex merging operation is performed on its dual, as shown in Figure 3(b);
- Vertex shifting: An edge is chosen at random first. All the other edges destined at its source, V_j, are shifted to its sink, V_j. Another vertex shifting is also performed on its dual. See Figure 3(c).
- 4. *Swap*: this is the only non-local move in the move set. Any two edges in the graphs are selected and interchanged. The result is that two blocks in the layout interchange their positions.

The move set described above is generic, and any topological placement is achievable from any initial placement using it.

Solution Evaluation

The quality of the placement solution at each stage is evaluated by a cost function. The cost function used in the placement system is a weighted sum of four components:

$$C(l) = \omega_a f_a(l) + \omega_{wl} f_{wl}(l) + \omega_{\beta} f_{\beta}(l) + \omega_m f_m(l) \quad (\text{eq. 2})$$

where $\omega_a, \omega_{wl}, \omega_{\beta}$ and ω_m are user-defined weighting coefficients, $f_a(l), f_{wl}(l)$ and $f_{\beta}(l)$ are the area, the total wirelength estimated using Manhattan geometry and the aspect ratio of the intermediate placement respectively.

The last component $f_m(l)$ in the cost function is used to drive matching devices to be placed near the center of the chip in order to take full advantage of the small stress gradient there. If *M* is the set of devices annotated to be matching, \bar{x} and \bar{y} are the x and ycoordinate of the center of the die, $f_m(l)$ is formulated as follows:

$$f_m(l) = \sum_{d \in M} c_d \sqrt{(x_d - \bar{x})^2 + (y_d - \bar{y})^2}$$
 (eq. 3)

where x_d and y_d are the x and y-coordinate of an element *d* in the set *M*. The criticality c_d is assigned to each matching block during device annotation.

4.3 Determination of Building Block Implementation

After a successful move is applied by annealing, a modified branch-and-bound algorithm [12] is carried out on the resulting placement to determine the physical implementation of every building block in a layout. Instead of using the area as the only way for placement evaluation, we modify the algorithm such that a cost function is accepted.

Types of Building Block

Variants of each block are determined before executing the algorithm. In this system blocks fall into three categories:

- 1. *Block of annotated devices*: Inside a block, devices are rigid in their locations and orientations. Only one possible implementation is available.
- Block of un-annotated bipolar devices: As a bipolar device is rigid, two variants are available if rotation is allowed and one variant is available otherwise. The user determines if all the bipolar devices are fixed in orientation.
- 3. Block of un-annotated MOSFET or passive devices: Up to three different implementations are generated by a variant generator. The variant generator determines the possible combinations of the layout parameters and creates the variants which are electrically equivalent.

Cost Function

An extra component, the *density E*, is added to the cost function as shown below. It is defined as the sum of distances between each block and the center of final layout. This term drives the white spaces to the boundary of the layout and it is evaluated only if the area of the intermediate placement equals the minimum area.

$$\begin{array}{c} \mbox{cost_function}(A, A_{min}, E_{min}, b_i) \left\{ \\ & \mbox{if } (A < A_{min}) \left\{ \\ & A_{min} = A \\ & \mbox{assign length to block } b_i \\ \right\} \mbox{else if } (A = = A_{min}) \left\{ \\ & \mbox{find density E} \\ & \mbox{if } (E < E_{min}) \left\{ \\ & E_{min} = E \\ & \mbox{assign length to block } b_i \right\} \\ \end{array} \right\}$$

In general, the modified branch-and-bound algorithm can be driven by any cost function. However, increasing the number of terms in cost function increases the computation complexity as well. To minimize the computation time per iteration, it is necessary to assign priority to each component in the cost function. A term in the cost function is evaluated only if components of higher priority fail to give a good estimation of a particular placement solution.

5. Results

The system described in this paper was implemented using the SKILL language for device annotation and variant generation in Cadence environment. The algorithm described in Figure 2 is implemented in C language to increase computation efficiency.

The program was tested with a number of practical circuits. Owing to the space limitation, only two of the silicon proven circuits, a bandgap reference (BR) and a temperature sensing amplifier(TSA), are presented in this section. The placements generated by the system are routed with IC Craftsman[13] and they are shown in Figure 4 and Figure 5.

Table 1 reports the size of the two test circuits in term of the number of instances and nets and the performance of the system. The CPU time is the run time required for executing the placement algorithm on a Ultra Sparc workstation.

Table 2 compares the quality of the layouts generated manually and automatically for TSA. The cycle time for automatic placement includes device annotation and generation, devicelevel placement and block-level placement. From Table 2, we observe an 8X cycle time reduction using the placement system reported. The quality of the placement is comparable to a manual design in terms of die area and wirelength.

6. Conclusions

An automatic placement tool bases on a new methodology targeted for analog designs is reported. On one hand, ingenuity and creativity of the layout engineer are preserved. On the other hand, it provides flexibility in process migration through device annotation. Moreover, the tool generates layout applicable to analog circuit by considering device matching during device annotation and optimization. Results from Section 6 illustrates the capability of the placement system in generating placement with an 8X cycle time reduction.

Future work includes the improvement to placement quality and algorithm efficiency by differentiating the cooling schedules of each parameter temperature and the cost function temperature. Analog placement and routing will be integrated into a single system, in which a layout is evaluated by a cost function in term of area and circuit performance after actual routing is performed.

Acknowledgment

The first author is supported by Motorola postgraduate internship. The second author is supported by RGE Earmarked Grants HKUST 547/94E and HKUST 6025/97E.

The first author wishes to thank Dr. W. H. Ki from the Hong Kong University of Science and Technology, Bernard Fung and other designers from Motorola Semiconductors HK Ltd. for their valuable suggestions during the preparation of this work.

References

- [1] J. Xu, P. N. Guo, C. K. Cheng, "Cluster Refinement for Block Placement," in *Proc. 34th ACM / IEEE DAC*, Jun 1997.
- [2] U. Lauther, "A min-cut placement algorithm for general cell assemblies based on a graph representation," in *Proc. 16th ACM/ IEEE DAC*, 1979, pp. 1-10.
- [3] S. W. Mehranfar, "A Technology-Independent Approach to Custom Analog Cell Generation," in *IEEE JSSC*, Vol. 26, No. 3, March 1991, pp. 386-393.
- [4] S. Sutanthavibul, E. Shargowitz, J.B. Rosen, "An Analytical Approach to Floorplan Design and Optimization," in *IEEE Trans. on CAD*, Vol. 10, No. 6, Jun 1991, pp. 761-769.
- [5] D. F. Wong, H. W. Leong, C. L. Liu, Simulated annealing for VLSI design, Kluwer Academic Publishers, Boston, 1988.
- [6] Device-Level Editor (DLE) and Device-Level Router (DLR) Help, Cadence Design System, Inc., 1995
- [7] M. J. M. Pelgrom, A. C. J. Duinmaiger, A. P. G. Welbers, "Matching properties of MOS transistors," in *IEEE JSSC*, Vol. 24, No. 5, Oct 1989, pp. 1433-1439.
- [8] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, L. R. Carley, "KOAN/ ANAGRAM II: New Tools for Device Level Analog Placement and Routing," in *IEEE JSSC*, Vol. 26, No. 3, Mar 1991, pp. 330-342.
- [9] E. Charbon, E. Malavasi, U. Choudhury, A. Casotto, A. Sangiovanni-Vincentelli, "A Constraint-Driven Placement Methodology for Analog Integrated Circuits," in *Proc. IEEE 1992 Custom Integrated Circuits Conf.*, May 1992, pp. 766-771.
- [10] D. J. Chen, J. C. Lee, B. J. Sheu, "SLAM: A Smart Analog Module Layout Generator for Mixed Analog-Digital VLSI Design," in *Proc.*

1989 IEEE International Conf. on Computer Design: VLSI in Computers and Processors, 1989, pp. 24-27.

- [11] S. Wimer, I. Koren, I. Cederbaum, "Floorplans, planar graphs and layouts," in *IEEE Trans. on Circuits and Systems*, Vol. 35, March 1988, pp. 267-278.
- [12] S. Wimer, I. Koren, I. Cederbaum, "Optimal Aspect Ratios of Building Blocks in VLSI," in *IEEE Trans. on CAD*, Vol. 8, No. 2, Feb. 1989, pp. 139-145.
- [13] *IC Craftsman Design Language Reference*, Cooper & Chyan Technology, Inc., 1995.

Table 1: Summary of Circuits Parameters

Parameters	BR	TSA
Number of devices	41	108
Number of nets	17	68
CPU Time (sec)	66	393
Number of moves attempted	5901	50001

Table 2: Comparison of Automatic and Manual Placement for TSA

Parameters	Automatic	Manual
Die area (µm ²)	702055.9	723290.8
Wirelength (µm)	59831.4	58223.9
Cycle Time	30 minutes	4 hours



Figure 4. Layout of a bandgap reference



Figure 5. Layout of the TSA