Pareto Front Analog Layout Placement using Satisfiability Modulo Theories

Sherif M. Saif*, Mohamed Dessouky†, M. Watheq El-Kharashi‡, Hazem Abbas‡ and Salwa Nassar*
*Computer and Systems Department, Electronics Research Institute, Giza, Egypt
Email: sherif_saif,salwa@eri.sci.eg
†Mentor Graphics Egypt, 78, El Nozha St, Heliopolis, Cairo 11361, Egypt
Email: mohamed_dessouky@mentor.com
‡ Computer & Systems Engineering Department, Ain Shams University, Cairo 11517, Egypt
Email: watheq@engr.uvic.ca, h.abbas@ieee.org

***Abstract*—This paper presents an analog layout placement tool with emphasis on Pareto front generation. In order to handle the exploding number of analog physical constraints, a new approach based on the use of a Satisfiability Modulo Theories (SMT) solver is suggested. SMT is an area concerned with checking the satisfiability of logical formulas over one or more theories. SMT is usually well-tuned to solve specific problems. To our knowledge, this is the first effort to use SMT to tackle analog placement. The proposed tool implicitly generates multiple layouts that fulfill the given constraints. Therefore, it gives the user the option to choose from the feasible solutions through specifying an aspect ratio or by selecting the optimum solution from the Pareto front of the generated shape function. In contrast to most of the existing techniques, as the number of physical constraints increases the SMT solver processing time decreases. The proposed system yielded layouts with a competitive area and run time compared to other techniques.**

I. Introduction

Analog layout automation is becoming more important in recent nanometer fabrication nodes. In such processes not only traditional analog constraints like symmetry and matching are mandatory, but also a considerable amount of layout dependent effects can have a disastrous impact on the circuit performance. Achieving rapid and sound layout that complies with all constraints is the only way to efficiently explore such effects.

Many placement techniques have been introduced to automate the analog placement operation including: constructive, combinatorial optimization, and template based techniques: Constructive approaches start with the selection of one module and try to position it, and then proceed with the evolution of the placement solution [1], [2]. On the other hand, combinatorial optimization approaches are either stochastic or deterministic. Some stochastic techniques extract non-quantitative hard or soft constraints in order to be used in the subsequent optimization phase [3]. Other techniques are quantitative [4] where they perform performance-driven optimization and quantitatively evaluate the placement solutions based on estimation models. A deterministic combinatorial optimization approach called Plantage [5] describes the IC hierarchy as a hierarchy tree that represents the required matching, symmetry and proximity constraints, and generates a set of area-optimal placements with different aspect ratios. Finally, template based approaches [6] depend on having template databases that contain analog circuits designed by experienced experts.

The proposed tool uses SMT that combines boolean satisfiability with term-manipulating symbolic systems [7] in order to perform analog placement subject to layout constraints. Microsoft Z3 SMT solver [8] is used to find feasible solutions for the given constraints. In order to do this, the constraints should be translated into equations that can be understood by the Z3 solver. The platform allows the user to choose some solutions based on aspect ratios.

This paper handles analog devices as black boxes of fixed dimensions, where an analog box may represent a MOS, a resistor, a capacitor, or an analog building block such as a differential pair or a current mirror.

This paper is organized as follows: Section II explains the mathematical background of SMT and Z3 solver. Section III presents the structure of the proposed system and Section IV describes its implementation. Section V discusses the results. Finally, Section VI draws some conclusions.

II. Satisfiability Modulo Theories

A. Constraint Satisfaction Problems, SAT and SMT

Solving a Constraint Satisfaction Problem (CSP) [9] implies solving a decision problem by deciding whether this problem has a solution given a set of constraints, or not. Afterwards, a solution has to be identified such that it satisfies the given constraints.

A Boolean satisfiability problem (SAT) [7] is a CSP whose goal is to decide whether a formula can be made true or not, where this formula is usually in a conjunctive normal form (CNF), i.e. an ANDing of ORed terms. For example, in the following formula the solver has to find a satisfying assignment to the Boolean variables v, w, x and y:

$$(v \vee \neg y) \wedge (x \vee w) \tag{1}$$

In this case if $v = 0$, $w = 1$, $y = 0$, and $x = 1$, the above formula becomes satisfied. Hence, this formula is satisfiable because there is at least one valuation which makes the formula true.

978-3-9815370-6-2/DATE16/

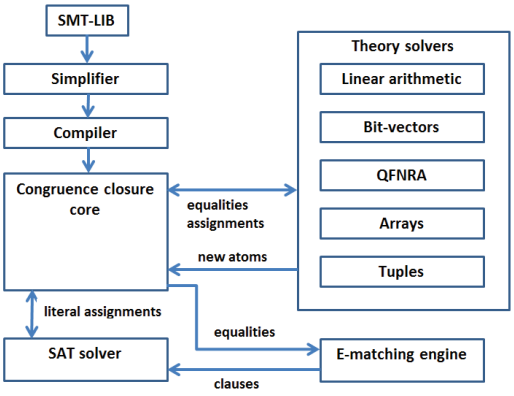

Fig. 1. Overview of Z3 architecture [7].

SAT solvers have many applications in the digital design domain including formal equivalence checking, model checking, formal verification and automatic test pattern generation [10].

SMT extends SAT problems. SMT solvers allow a much richer vocabulary than Boolean operations and variables when creating formulas. It may contain general relations, such as equalities between terms formed from variables and functions [7]. For example in (1), if $v = (cos(a) + b) < 3)$ and $w = (log(a) > 0)$, where a and b are real numbers, therefore it becomes

$$((cos(a) + b) < 3) + \bar{y}) \cdot (x + (log(a) > 0)) \qquad (2)$$

The SMT solver has to find a satisfying assignment to a and b, which are real, and to x and y, which are Boolean variables. So, for example if $a = \pi/2$, $b = 1$, $y = 0$, and $x = 1$, (2) becomes satisfied. Hence, this formula is satisfiable because there is at least one valuation which makes the formula true.

B. Z3 Solver

Z3 solver [8] [11] is an SMT solver that deals with intractable theories [12] and assertions that are represented in Conjunctive Normal Form (CNF) [13]. A theory is defined by a signature that defines the domains, functions, and relations of the theory and a set of interpretations of the relations and functions. Z3 provides capabilities to work with linear real and integer arithmetic.

An overview of the Z3 architecture is shown in Figure 1. The Simplifier module processes the input formulas and applies standard algebraic reduction rules to them. The Compiler module converts the simplified abstract syntax tree representation into a data-structure composed of a set of clauses and congruence-closure nodes. The Congruence Closure Core module receives truth assignments from the SAT solver and propagates the effects of the theory solvers [7]. The theory of quantifier-free linear arithmetic over the reals (QFNRA) is decidable and it is the one used in this application.

III. Proposed System Architecture

The flow diagram depicted in Figure 2 shows the operation of the proposed system. The first module reads the constraints from the user through a text file. The equation generator module generates equations that represent the explicitly specified constraints. Supported constraints in this system are: mirror symmetry, common centroid, proximity, relative placement, row and column constraints. These constraints will be explained in details in Section IV. The equation-generation module also generates equations for two implicit constraints in order to produce a valid output: a non-overlapping constraint to make sure that no two devices overlap, and a constraint that all devices should fit within a provided cell area.

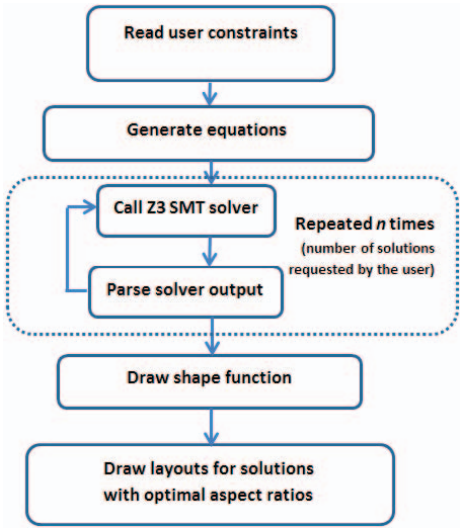

Fig. 2. System flow diagram for analog layout generation using constraints.

Afterwards, the proposed system calls the Z3 solver. The solver processes the generated equations and decides whether there is a model (a solution) for these equations. The system parses the output of the solver in order to get the coordinates of the valid placement solutions, if there are any. This step is repeated n number of times, where n is a user choice that limits the number of solutions at each aspect ratio and at each area overhead step as will be explained in Section IV.

The system translates valid solutions to coordinates and generates a shape function that represents all aspect ratios of the generated solutions with the Pareto front of the optimization problem [14]. The user can select the "optimum" solutions from the Pareto front graph. The system will send to the layout editor only those selected layouts.

IV. System Implementation

In this section we discuss the implementation of the system and the realization of the mathematical formulas corresponding to the constraints that the system accepts.

The circuit is represented by groups of modules, such that there are constraints within each group and constraints between different groups. The group identification and the description of the constraints are specified through an input text file.

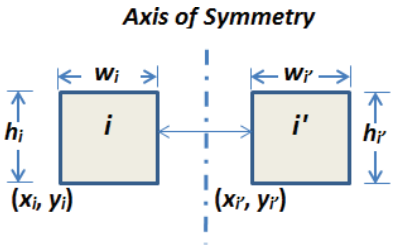

Fig. 3. Devices having mirror symmetry and aligned to the center.

A. Equation Generation

The equation-generation module, depicted in Figure 2, uses the obtained constraints to generate the corresponding equations that can be used by the Z3 solver. The constraint equations will be presented in the following where for any two modules i and $i^{'}$, the lower left coordinate of module i is represented by (x_i, y_i) and that of module $i^{'}$ is represented by $(x_{i'}, y_{i'})$. The width and height of modules i and $i^{'}$ are w_i, h_i, $w_{i'}$ and $h_{i'}$, respectively.

1) Mirror Symmetry: Symmetry constraints serve geometric and electrical reasons [5] and can be classified into: mirror-symmetry, perfect-symmetry, and self-symmetry [1].

Mirror symmetry is measured relative to some vertical axis as shown in Figure 3. If module i' is to be placed symmetrically to another module i, mirror symmetry constraints will be represented by (3) and (4) [5]:

$$\forall_{i,i'} \left(\frac{1}{2} \left(x_i + \frac{w_i}{2} + x_{i'} + \frac{w_{i'}}{2} \right) = x_{sym} \right) \tag{3}$$

$$\forall_i \left(y_i + \frac{h_i}{2} = y_{i'} + \frac{h_{i'}}{2} \right) \tag{4}$$

where the x-coordinate of the symmetry axis is x_{sym}.

2) Common Centroid: Common centroid constraint [15] is usually applied to a subcircuit of a current mirror or a differential pair to reduce process-induced mismatches among the devices. Common centroid constraints can be formulated by calculating the Center Of Gravity (COG) of a module which is defined by (5) [5]:

$$A = \begin{bmatrix} x_{COG}(i) = x_i + \frac{w_i}{2} \\ y_{COG}(i) = y_i + \frac{h_i}{2} \end{bmatrix} \tag{5}$$

For groups of modules, a COG is defined according to (6) as [5]:

$$X_{COG}(G) = \frac{\sum_{i \in modulesOf(G)} w_i.h_i.X_{COG}(i)}{\sum_{i \in modulesOf(G)} w_i.h_i} \tag{6}$$

For two groups A and B, the common centroid constraint dictates that the centers of gravity of the two groups coincide, or:

$$X_{COG}(A) = X_{COG}(B) \tag{7}$$

3) Proximity:

a) Proximity Within One Group: If a group of devices should fall in proximity, then for every two devices i and i' in this group (8) and (9) should be satisfied.

$$\forall_i \left(x_i + w_i = x_{i'} \ OR \ x_{i'} + w_{i'} = x_i \right) \tag{8}$$

$$\forall_i \left(y_i + h_i = y_{i'} \ OR \ y_{i'} + h_{i'} = y_i \right) \tag{9}$$

b) Proximity Among Different Groups: Three cases are defined to realize proximity constraints among different groups:

- proximity: i,$i^{'}$
  Any two blocks from the two groups have a common border or part of a border. This is implemented by ORing (8) and (9) for all blocks of the two groups.
- abut groups: i,$i^{'}$
  At least two blocks one from each group are adjacent side-by-side. This is implemented by ORing (8) for all blocks of two groups.
- ahead groups: i,$i^{'}$
  At least two blocks one from each group have a common side where this is either the upper side or the lower side of the block. This is implemented by ORing (9) for all blocks of two groups.

4) Relative Placement: Due to certain floorplanning constraints, it is sometimes required to place a group of devices in a certain relative placement with respect to others.

This can be one of the following:

- West: Module i should be to the left of module $i^{'}$.

$$\forall_i \left(x_i + w_i < x_{i'} \right) \tag{10}$$

- East: Module i should be to the right of module $i^{'}$.

$$\forall_i \left(x_{i'} + w_{i'} < x_i \right) \tag{11}$$

- South: Module i should be below module $i^{'}$.

$$\forall_i \left(y_i + h_i < y_{i'} \right) \tag{12}$$

- North: Module i should be above module $i^{'}$.

$$\forall_i \left(y_{i'} + h_{i'} < y_i \right) \tag{13}$$

5) Tiling: In some cases layouts need to be arranged in uniform rows, uniform columns, or both uniform rows and columns.

There are three variations of the row constraint.

1) Devices aligned to the bottom of the row: In this case the lower side of all the devices that have the row constraint should be on the same y-coordinate. Therefore any two devices i and j should satisfy:

$$y_i = y_i^{'} \tag{14}$$

2) Devices aligned to the top. In this case:

$$y_i + h_i = y_i^{'} + h_i^{'} \tag{15}$$

3) Devices aligned to the center. In this case:

$$y_i + \frac{h_i}{2} = y_i^{'} + \frac{h_i^{'}}{2} \tag{16}$$

Similarly the column constraint can be implemented by using x instead of y and w instead of h in equations (14)–(16).

B. Shape Function and Optimal Area

A shape function is an ordered set of shapes. Each shape represents a placement with a different aspect ratio. The Pareto front of a cell is defined as the lower area bound of all possible rectangles of the cell [16], [17].

The platform calculates an estimate for the "exact area", by summing up all the areas of the modules inside the given circuit. The program reads an argument "overhead" from the user input text file and according to this argument it calculates the "expected area" according to the following equation.

$$expected\_area = exact\_area \ * \ overhead \tag{17}$$

The "overhead" should be greater than one.

Since the aspect ratio is defined by the following equation:

$$AR = \frac{W}{L} \tag{18}$$

where W and L are the top cell width and height, respectively. Therefore, using equations (17) and (18), W and L of the top cell can be calculated using the following equations:

$$W = sqrt(expected\_area \ * \ AR) \tag{19}$$

$$L = sqrt(\frac{expected\_area}{AR}) \tag{20}$$

The system allows the user to specify the maximum user-acceptable overhead. In this case the system will start with this overhead and tries to get a solution. If it fails, it asks the user to provide a new "overhead". On the other hand, if it succeeds, it tries a smaller overhead according to a default reduction step value that can be modified by the user. The system keeps reducing the overhead until it fails to generate a solution. Hence, reaching the Pareto front of the layout.

With each aspect ratio that has a successful solution on the Pareto front, the system works to get n number of solutions, where n is a user choice. The system achieves this by calling the Z3 solver n number of times, each time with a different seed. Changing the seed makes the solver search for a different solution.

V. Results

A. Benchmark

The system was used to generate the layout for the folded cascode operational amplifier described in [5].

The system was asked to generate 20,000 layouts to plot a shape function that is depicted in Figure 4. The platform generated 20,000 layouts in 2202 seconds with an average of 0.1 seconds/layout. It was noted that as the number of added constraints increases the Z3 processing item decreases. Figure 5 shows one of the generated layouts.

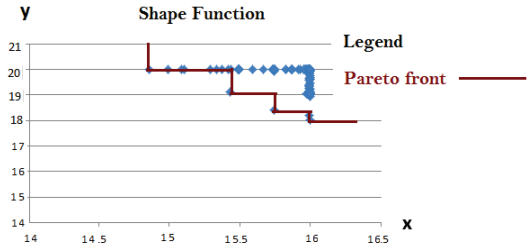

Fig. 4. Shape function of the folded cascode amplifier.

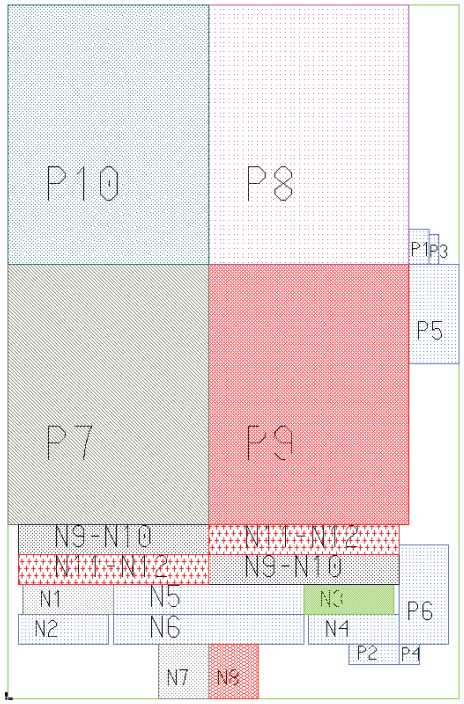

Fig. 5. Folded cascode op amp layout.

TABLE I
Comparison of area usage and runtimes for different approaches, based on folded cascode operational amplifier.

Approach	Area (% of total area)	Time (sec)
Plantage [5]	121	4
Proposed SMT-based method	115	0.1

Table I compares the numerical results of the approach with another approach, Plantage [5]. In this case it is compared with Plantage because the latter approach proved to be competitive from a speed perspective when compared to other techniques. Our proposed methodology achieves a better area utilization in less time. It should be noted that this experiment was run on a different hardware. Using FLOPs calculation the SMT-based method is only ten times faster than Plantage.

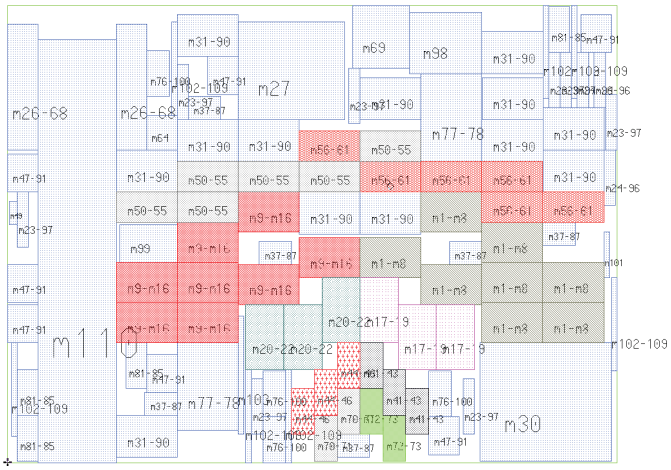

Fig. 6. Layout 1 of industrial circuit lnamixbias_2p4g

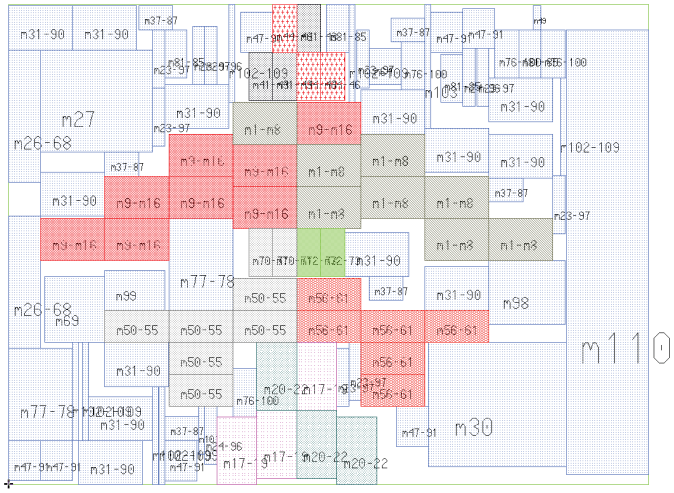

Fig. 7. Layout 2 of industrial circuit lnamixbias_2p4g

B. Industrial Circuits

The system was run on an industrial circuit "lnamixbias_2p4g" [18]. "lnamixbias_2p4g" consists of 110 cells with five symmetry groups (eight pairs, three pairs, three pairs, six pairs, and two pairs) represented by:

- Eight symmetric pairs arranged into group:1 and group:2. The devices are identified by m1-8 and m9-16.
- Three symmetric pairs arranged into group:3 and group:4. The devices are identified by m41-43 and m44-46.
- Three symmetric pairs arranged into group:5 and group:6. The devices are identified by m17-19 and m20-22.
- Six symmetric pairs arranged into group:7 and group:8. The devices are identified by m50-55 and m56-61.
- Two symmetric pairs arranged group:9 and group:10. The devices are identified by m70-71 and m72-73.

Some of the generated layouts are shown in Figure 6, and Figure 7.

In a normal run on this circuit given these five symmetry groups, the system required 55 minutes to generate a layout solution. The system was run again after adding a proximity constraint to the symmetry groups. The average Z3 processing time became: 8 minutes and 45 seconds.

TABLE II
COMPARISON OF AREA USAGE AND RUNTIMES FOR DIFFERENT APPROACHES, BASED ON AN INDUSTRIAL CIRCUIT LNAMIXBIAS_2P4G. INITIAL COMPARISON WAS MADE BY STRASSER *et al.* [5] FOR TWO CIRCUITS AND REUSED HERE WITH ONE CIRCUIT IN ADDITION TO THE RESULTS OF OUR PROPOSED METHOD.

Approach	Area (% of total area)	Time (sec)
Sequence Pair [19]	110.43	2824
Segment Tree with Segment Tree [20]	109.35	726
Sequence Pair and Linear Programming [21]	108.59	3252
Sequence Pair with Dummy Nodes [22]	113.50	227
Symmetry Islands [23]	105.72	43
Sequence Pair with	109	480
Johnson's Priority Queue [24]		
Plantage [5]	107.68	387
Proposed SMT-based method with dimension 80 x 60	116	3300
Proposed SMT-based method with dimension 130 x 90	282	525

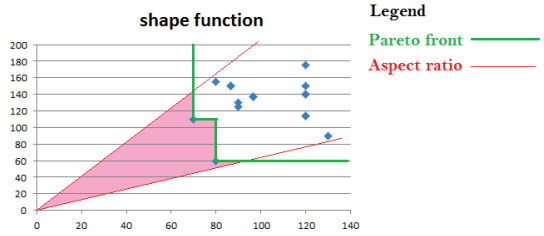

Fig. 8. Shape function for lnamix_bias_2p4g circuit with range.

Comparison of area usage and runtimes for different approaches, based on an industrial circuit lnamixbias_2p4g is shown in Table II. The SMT experiments were run on a different hardware compared to the other listed techniques. With FLOPs calculation the two SMT experiment are slower than other non-SMT approaches listed in the table.

The consumed time decreased as the number of input constraints increased. The large number of cells with low number of constraints made the system consume a larger amount of time because the search space is not narrowed.

When the circuit is allowed to make many runs with different cell limits, the shape function of these generated layouts, the aspect ratio range, as well as the Pareto front of are shown in Figure 8. Only feasible solutions are shown. Each circuit dimension may contain more than one solution.

VI. CONCLUSION

In this paper the problem of automating placement for analog ICs was discussed and a solution using SMT was proposed.

The system is implemented in C++ and Unix shell scripts. It uses SMT Z3 solver to resolve the analog constraints.

The proposed system uses the "shape function" and Pareto front concepts in order to allow the user to select optimum solutions.

With input circuits that have few geometries and many constraints which is typical for analog circuits such as the folded cascode opamp, the proposed system was able to over perform other systems that were proposed in the literature from the perspective of speed while preserving the quality of the output. The proposed system generated many layouts, each of them consumed an average of 0.1 seconds. This is a much higher speed if compared to the other techniques compared in [5]. On the other hand, when the circuit has many geometries and few constraints such as the presented industrial circuit, our proposed system took more time than other approaches and provided a competitive compaction ratio.

For future work, this system can be expanded in order to handle more constraints and more options. In addition, a generation of a topological layout representation would be investigated to enhance the speed of the solver in the case of many geometries with few constraints. In this case cell positions are relatively specified based on topological relations between cells instead of their coordinates. In addition, the possibility of having more than one layout with different dimensions and aspect ratios for each analog device should be incorporated.

REFERENCES

[1] F. Balasa, "Device-Level Topological Placement with Symmetry Constraints," Analog Layout Synthesis - A Survey of Topological Approaches H. Graeb (ed.) Springer Verlag, New York, ISBN 978-1-419-6931-6, (2011) 1921.

[2] E. Malavasi, J.L. Ganley, and E. Charbon. Quick placement with geometric constraints. In Custom Integrated Circuits Conference, 1997., Proceedings of the IEEE 1997, pages 561–564, May 1997. doi: 10.1109/CICC.1997.606689.

[3] J. Cohn, D. Garrod, R. Rutenbar, and L. Carley. Koan/anagram II: new tools for devicelevel analog layout. IEEE Journal of Solid-State Circuits, SC-26(3):330–342, March 1991.

[4] K. Lampaert, G. Gielen, and W.M. Sansen. A performance-driven placement tool for analog integrated circuits. Solid-State Circuits, IEEE Journal of, 30(7):773–780, Jul 1995. ISSN 0018-9200. doi: 10.1109/4.391116.

[5] M. Strasser, M. Eick, H. Graeb, U. Schlichtmann, and F. M. Johannes. "Deterministic Analog Circuit Placement using Hierarchically Bounded Enumeration and Enhanced Shape Functions". In IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 306-313, November 2008.

[6] M. Elshawy, M. Dessouky, S. Saif, S. Mansour, and E. Petrus. Multi-device layout templates for nanometer analog design. In Design Test Symposium (IDT), 2014 9th International, pages 83–88, Dec 2014. doi: 10.1109/IDT.2014.7038592.

[7] L. de Moura and N. Bjϕrner. Satisfiability modulo theories: An appetizer. In Formal Methods: Foundations and Applications, 12th Brazilian Symposium on Formal Methods, SBMF 2009, Gramado, Brazil, August 1921, 2009, Revised Selected Papers, volume 5902 of Lecture Notes in Computer Science, pages 2336. Springer, 2009.

[8] Microsoft. Z3 solver, 2015. URL http://research.microsoft.com/projects/z3.

[9] Vipin Kumar. Algorithms for constraint-satisfaction problems: A survey. AI Mag., 13(1):32–44, April 1992. ISSN 0738-4602. URL http://dl.acm.org/citation.cfm?id=140370.140377.

[10] Marques-Silva, J., "Practical applications of Boolean Satisfiability," in Discrete Event Systems, 2008. WODES 2008. 9th International Workshop on , vol., no., pp.74-80, 28-30 May 2008 doi: 10.1109/WODES.2008.4605925

[11] codeplex. Z3. Open source at:, 2015. URL http://z3.codeplex.com.

[12] Fred J. Hickernell and Henryk Woniakowski. tractability of multivariate integration for periodic functions. J. Complexity, 17(4), 1999.

[13] Zhaohui Fu; Malik, S., "Extracting Logic Circuit Structure from Conjunctive Normal Form Descriptions," in VLSI Design, 2007. Held jointly with 6th International Conference on Embedded Systems., 20th International Conference on , vol., no., pp.37–42, 6–10 Jan. 2007. doi: 10.1109/VLSID.2007.81

[14] A. Toro-Frias, R. Castro-Lopez, E. Roca, and F.V. Fernandez. Layout-aware pareto fronts of electronic circuits. In Circuit Theory and Design (ECCTD), 2011 20th European Conference on, pages 345–348, Aug 2011. doi: 10.1109/ECCTD.2011.6043357.

[15] Linfu Xiao and E.F.Y. Young. Analog placement with common centroid and 1-d symmetry constraints. In Design Automation Conference, 2009. ASP-DAC 2009. Asia and South Pacific, pages 353–360, Jan 2009. doi: 10.1109/ASPDAC.2009.4796506.

[16] A. A. Szepieniec and R. Otten. The genealogical approach to the layout problem. Proc. of the 17th Design Automation Conference, 1980, pages 535–542, 1980.

[17] Ralph H.J.M. Otten. Automatic floorplan design. In Proceedings of the 19th Design Automation Conference, DAC '82, pages 261–267, Piscataway, NJ, USA, 1982. IEEE Press. ISBN 0-89791-020-6. URL http://dl.acm.org/citation.cfm?id=800263.809216.

[18] S. Koda, C. Kodama, and K. Fujiyoshi. Linear programming-based cell placement with symmetry constraints for analog ic layout. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 26(4):659–668, April 2007. ISSN 0278-0070. doi: 10.1109/TCAD.2007.891365.

[19] F. Balasa and K. Lampaert. Symmetry within the sequence-pair representation in the context of placement for analog design. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 19(7):721–731, Jul 2000. ISSN 0278-0070. doi: 10.1109/43.851988.

[20] Florin Balasa and Koen Lampaert. Module placement for analog layout using the sequence-pair representation. In Proceedings of the 36th Annual ACM/IEEE Design Automation Conference, DAC '99, pages 274–279, New York, NY, USA, 1999. ACM. ISBN 1-58113-109-7. doi: 10.1145/309847.309930. URL http://doi.acm.org/10.1145/309847.309930.

[21] Shinichi Kouda, Chikaaki Kodama, and Kunihiro Fujiyoshi. Improved method of cell placement with symmetry constraints for analog ic layout design. In Proceedings of the 2006 International Symposium on Physical Design, ISPD '06, pages 192–199, New York, NY, USA, 2006. ACM. ISBN 1-59593-299-2.doi: 10.1145/1123008.1123050. URL http://doi.acm.org/10.1145/1123008.1123050.

[22] Yiu-Cheong Tam, E.F.Y. Young, and C. Chu. Analog placement with symmetry and other placement constraints. In Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference on, pages 349–354, Nov 2006. doi: 10.1109/ICCAD.2006.320057.

[23] Po-Hung Lin and Shyh-Chang Lin. Analog placement based on novel symmetry-island formulation. In Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE, pages 465–470, June 2007.

[24] K. Krishnamoorthy, S.C. Maruvada, and F. Balasa. Topological placement with multiple symmetry groups of devices for analog layout design. In Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on, pages 2032–2035, May 2007. doi:10.1109/ISCAS.2007.378437.