A Performance-Driven Placement Algorithm with Simultaneous Place&Route Optimization for Analog IC's

Juan A. Prieto, Adoración Rueda, José M. Quintana and José L. Huertas

Instituto de Microelectrónica de Sevilla. Centro Nacional de Microelectrónica
Av. Reina Mercedes s/n. 41012 Sevilla. SPAIN. E-mail: jprieto@cnm.us.es

Abstract

This paper presents a performance-driven placement algorithm for automatic layout generation of analog IC's. The main innovations of our approach are essentially: (i) an integrated Place&Route optimization algorithm which is able to provide a realistic measurement of the interconnect parasitics, that is a key issue in performance-driven approaches; and (ii) the simultaneous consideration in the cost function of two levels of symmetries: global symmetry with respect to virtual axes and local symmetry affecting groups of cells. The flexibility and efficiency of the algorithm is mainly due to the use of the same slicing-tree representation for placement and global routing, and to the heuristic algorithm we propose for the global routing estimate. The feasibility of the proposed approach has been demonstrated with several practical examples.

1.- Introduction

The complexity of analog layout derives not only from the diversity of cells and their disparity depending on the concrete application, but also from the influence of the final physical design on the global performance of the circuit. Thus, considerations of symmetry at a cell or wire level, of sensitive and noisy nets, of orientation, etc., which affect the precision of component values and parasitic effects, are as important as those of area and interconnect length. The first approaches towards automatic layout generation of analog circuits accounted for these considerations in the minimization of a cost function with no quantitative evaluation of performance functions [1-3]. Recent research focuses on performance-driven layout methodologies which estimate layout-dependent performance degradations and use them to drive the layout synthesis [4-7]. The sources of performance degradation considered in these approaches include device mismatches and interconnect capacitive/resistive parasitics, and the performance degradations are evaluated using sensitivity information from the parasitic-free design.

Efficient implementation of any performance-driven algorithm requires a precise and low-cost estimate of the interconnect parasitics during the placement phase. An integrated Place&Route optimization algorithm capable of contemplating analog restrictions would be a good solution to this problem. Thus, all the performance constraints which influence the placement are considered simultaneously, consequently easing the task of the posterior router to achieve good routing solutions that cause no additional performance degradation.

This paper presents a solution to the performance-driven placement of analog IC's, which based on slicing structures exhibits the advantages of flat approximations [2, 4, 7], such as representative cost function, device level move set, etc. In addition, it offers a high level of flexibility in the calculation of geometrical properties of the solution, such as well location, cell-abutment, global routing estimate, etc.

The main innovations of the approach, which has been implemented in the tool GELSA, are: (i) the incorporation of an integrated Place&Route optimization algorithm handling analog restrictions which enables a good, low-cost estimate of the global routing for each intermediate solution generated by the placer; and (ii) the simultaneous consideration in the cost function of two levels of symmetries: *global* symmetry with respect to virtual axes, as proposed in [8], and *local* symmetry affecting groups of cells. When cells are basic circuit devices and their possible partitions, the latter avoids the simultaneous Place&Module optimization [5], while enabling exploration of advantageous configurations with interleaving and minimum capacitive loading.

Serious problems are encountered when integrating placement and routing problems in an iterative optimization algorithm. Among these, the most complex is calculation of the routing related to the solutions generated in the placement phase. An optimum solution requires solving an NP-complete problem for each iteration. Our proposal for global routing is a heuristic algorithm, which has been integrated in the optimization process without excessively in-

ED&TC '97 on CD-ROM

creasing the total cost. This heuristics, together with the idea of using the same slicing representation structure for both the placement and global routing problems, makes the algorithm very efficient.

Simulated annealing is the optimization technique used because it is a general and robust method that allows obtaining solutions close to the global optimum. It is also very adequate to deal with analog requirements due to its flexibility in determining complex cost functions without affecting the complexity of the optimization algorithm [2, 4].

The paper is organized as follows. Section 2 gives a general description of the algorithm, highlighting the implementation of analog constraints. Section 3 presents examples of practical placements obtained with the tool, including experimental results. Finally, conclusions are given in Section 4.

2.- General Description of the Algorithm

The main sources of performance deviation in analog circuits are: mismatching, loading and coupling parasitics, and electrical noise. Since their layout-dependent effects are not equally quantizable, they are considered as two different groups of constraints in the placement optimization problem: a) *Topological constraints*, defined as a set of properties related to group of cells (e.g., symmetries, device orientation, proximity). Their goal is to minimize mismatching and the influence of noisy sources. b) Explicit device and interconnect *parasitics evaluation*, whose effect on circuit performance is measured and included into the cost function.

Figure 1 shows the general flow scheme of the optimization algorithm implemented in the program GELSA; it is composed of two parts: 1) A ***Simulated Annealing Optimizer*** to explore the search space. It proposes intermediate solutions as slicing structures (***Perturbation Generator***), and evaluates the quality of the solutions in order to accept or reject them accordingly (***Cost Evaluator***); 2) An ***Expander*** of the solution proposed by the Perturbation Generator. It generates the global routing as well as local symmetry control, cell abutments, and well generation. Details of the different parts follow.

2.1.- Placement Representation and Move Set

Each point in the solution space is represented by a binary slicing tree bounding the N cells of the problem (N-*tree*), and by a factor of orientation&shape θ for each cell. The structure is then made up of this N-*tree* formed by N operands (cells) and N-1 cut operators (horizontal H and vertical V channels), and the set N-θ formed by the N factors θ.

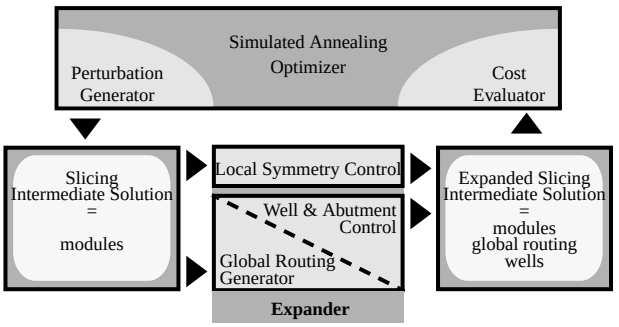


Figure 1: General flow scheme in GELSA

The solution space is explored by a set of moves affecting: a) individual cells and subtrees which represent cell clusters in the N-*tree*, to provide a new cell disposition on the design space, b) the factor θ, to provoke rotation or shape selection of the corresponding cell, and c) the orientation of subtrees in the N-*tree*, to include rotation of clusters of cells represented by the subtree. This rotation affects both the N-*tree* and the N-θ.

The algorithm uses a flat slicing placement representation considering both the N-*tree* and N-θ as annealing variables. The main advantages of slicing representation, versus non-slicing, lie in the easy calculation of local symmetry for clusters (current mirrors, differential pairs, etc.), of global routing (parasitics and area-for-routing estimates), and the flexible control of wells and abutments. Even though in slicing structures the cut operators only determine topological relations between cells, geometry-sharing is not prohibited since these relations can be modified to accommodate abutment. The only appreciable drawback of slicing structures is that some topologies cannot be implemented with slicing based on two cuts; however, this does not invalidate the approach since it could be extended to a non-slicing approach based on other tree representations [9].

2.2.- Integrated Place&Route Algorithm

The *Place&Route* block belongs to the Expander and consists of a *Global Router* generator and a *Well&Abutment Control* block. It is a new version of the *Place&Route* function reported in [10], which has been enhanced to include control of cell abutment and positioning of wells. Figure 2 briefly describes the *Place&Route* function. It works on the slicing-tree generated as intermediate solution to provide: (i) the positions of the cells and terminals over the cut operators as a function of shapes and orientations (*Det_pos(·)* and *Gen_HVE(·)* functions), (ii) the estimate of the global routing at the level of segments occupied in each channel, the maximum occupation density of each channel, and the channel effective widths (*Det_GR(·)* func-

tion), and, (iii) an actualization of both the cell positions and the channel widths to accommodate the routing in order to satisfy geometrical design rules for wells and abutments.

The function *Det_GR(·)* implements the heuristics developed for global routing estimate. It works on an extended graph (called HVE graph) which includes the four external borders as additional channels. On this graph and for each net, a set of branches is defined by mapping each terminal to each branch. This group of branches is propagated directly rising the graph avoiding parallel channels until all terminals are connected; those channels susceptible for use in the routing of the net are marked. Since the solution thus generated may result in an inefficient routing, a set of 16 simplification rules have been derived as shortcuts to rise in the hierarchy of the marked graph leading to a minimized routing [10]. Figure 3 illustrates the application and effect of one of the simplification rules. The initial routing solution obtained directly from the slicing tree is shown in Fig. 3a, while the improved solution after applying the shortcut called Rule #2 is in Fig. 3b.

```
Place&Route(j) {  /*j:N-tree (slicing intermediate solution)*/
  P1 = Det_pos(j);       /*determines cell&channel positions*/
  G = Gen_HVE(j);/*extends the slicing tree to HVE graph*/
  I = Det_GR(P1, G);              /*determines global routing*/
  P2 = Update_pos&Det_wells&Det_abutment(I, G);
  /*updates cell positions depending on the global routing*/
  /*determines wells and device abutments*/
}
  /*Output: Expanded Slicing Intermediate Solution*/
```

Figure 2: Place&Route function

To evaluate the quality of the heuristic global router, we have compared the total net length obtained for every intermediate solution during the optimization process with an exhaustive global router. Results showed an average deviation during the cooling process of 7% being both estimations equal at the end of the optimization.

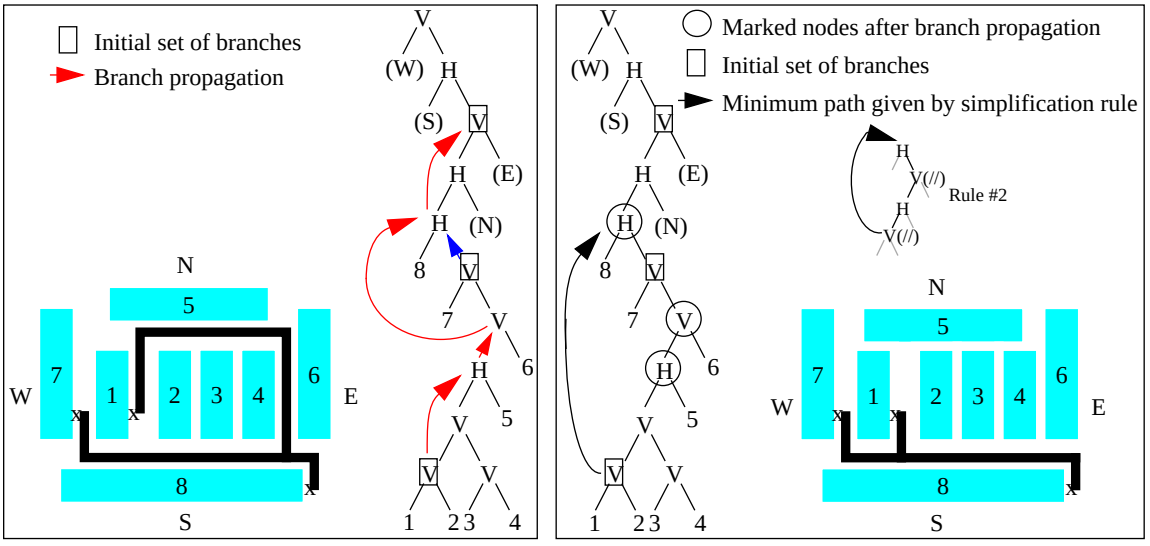
Wells and abutments are accommodated by bottom-up exploration of the slicing tree to determine the contour[1] of each subtree and storing it in the contour stack. The algorithm makes efficient use of the slicing structures, imposing the separations between cells and cell groups as positive values to consider design rules for wells, and as negative values to share device diffusions for abutment.

2.3.- Control of Symmetries

We distinguish between global and local symmetry constraints. Global constraints define symmetric positioning of the cells with respect to symmetry axes in a similar form as considered in [8]. They allow obtaining layouts with good interconnect-induced parasitics matching, and with devices symmetrically placed with respect to any noisy or heat source. Once the physical positions of the cells are known, their degree of asymmetry with respect to their virtual symmetry axis is reflected in the cost function.

Local symmetry for group of devices is of great importance for precision and matching. These constraints are contemplated by defining a set of properties for cell clusters. The procedures to apply these properties as well as the clusters of symmetries in GELSA are illustrated in Fig. 4a

1. A group of segments on the subtree bounding box.



(a) Non-minimal routing

(b) Minimized routing

Figure 3: Global Routing example

and 4b, respectively. The algorithm represents each cluster by a subtree inside the N-*tree*, modifying it during the optimization to close the corresponding reference subtree that contains the symmetry properties. Figure 4c illustrates the concept of reference subtree, showing the related to a common-centroid cluster. The cost associated to a symmetry cluster is a weighted sum of terms accounting for the degree of orientation of the cluster cells, and, the deviation of the cluster subtree from the reference subtree. The advantage of this approach is that we only require the subtree linking the cells of the cluster, and then the exact cell location on the design surface has not to be calculated.

2.4.- Control of Performance Degradations

The degradation ΔP_i of an *i*-th performance due to parasitics x_j is measured as $\Delta P_i = \Sigma\ (S^i_j\ x_j)$, where S^i_j is the sensitivity of P_i with respect to x_j, defined as the partial derivative of P_i with respect to x_j. Sensitivity values are evaluated by electrical simulation using a perturbation method, and measuring performance variations respect to the parasitic-free design [7]. The parasitics considered are those associated with devices and wires. The device parasitics are estimated by the module generator and in our approach they may only be reduced by abutment, whenever user-specified orientation constraints are not violated. The second are estimated using information provided by the global router. This information consists in the actual length (L_{net}) of each net, the coupled length ($L_{coupled}$), and the number of intersection (N_{cross}) between net pairs. Thus, maximum parasitic capacitances for each net to the substrate and for coupling between net pairs are evaluated using technological parameters and L_{net}, $L_{coupled}$, and N_{cross}. The information to evaluate the parasitic resistances is available as partial net lengths from the global router.

2.5.- Cost Function

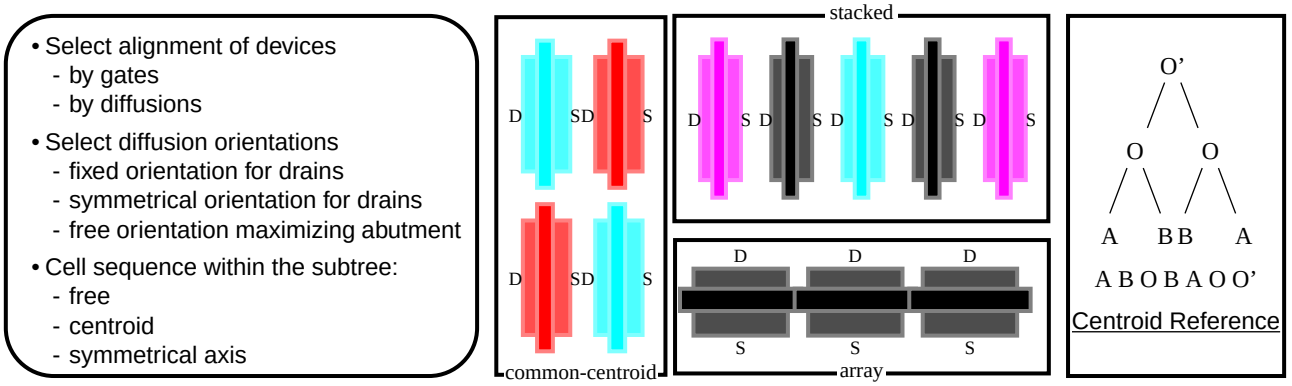
The cost function driving the optimization algorithm is the sum of 11 normalized terms, accounting for: (i) geometrical aspects (global and local area, aspect ratio, number and distribution of wells, abutment, and cell shapes), (ii) global and local symmetries, (iii) routing aspects (total net length, density), and, (iv) performance degradations.

Aspects related to the definition of the cost function terms have been studied to maximize the independence between the control parameters and the problem. Cost function behavior was evaluated using a Level II Factorial Analysis of Experiments. Each term in the cost function is expressed as

$$T_j = k + \sum_i C_i \cdot W_i \qquad (1)$$

where W_i is a boolean variable representing the activation (W_i=1) or no activation (W_i=0) of the ith-term of the cost function, and C_i are the parameters which when evaluated by the Factorial Analysis will indicate the relative importance of each term and its collateral effects on the other terms. The *k* parameter measures the offset of the cost associated to each term. Negative values of C_i mean that W_i help to minimize the corresponding T_j, while positive values indicate degradations of the optimization.

Table I shows the results obtained applying this analysis to 640 different placements of the Fully Differential Folded Cascode OTA in Fig. 5. The cost function terms that were studied are indicated in the first column (T_{arel}: total relative area, T_{alocal}: local area, T_{bound}: solution bound, T_{nets}: total net length, T_{rests}: qualitative routing restrictions, T_{dens}: routing densities, T_{wells}: sizing of wells). Data in diagonal table cells indicate the relative importance of each term, giving us information on how to recalculate the weight set controlling the optimization. Data on the other cells show the interactions between terms. For instance, from the last column we can see how the activation of the minimization



(a) Properties (b) Local symmetry groups (c) Reference subtree

Figure 4: Local symmetry control

of the distribution and number of wells term in the cost function (W_{wells}=1), minimizes the cost of its own term, T_{wells}, because the corresponding C_i value is -22, while degrades the treatment of the constraints associated to the routing, T_{rests}, because of C_i=+11.

We have applied this analysis to different circuits obtaining similar results after a normalization.

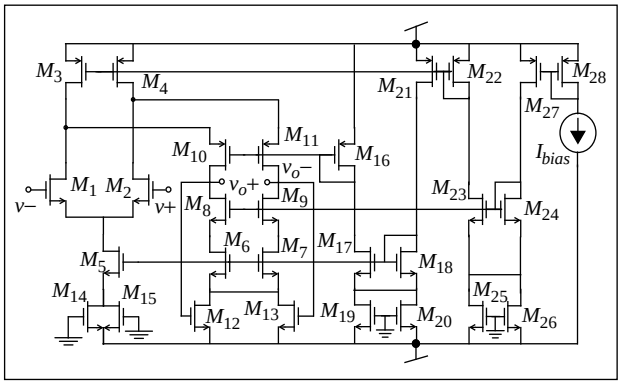


Figure 5: OTA schematic

Table I: Coefficients C_i from a Factorial Analysis of a Fully Differential Folded Cascode OTA

Weight Terms	*k*	W_{arel}	W_{alocal}	W_{bound}	W_{nets}	W_{rests}	W_{dens}	W_{wells}
T_{arel}	675	-14	-5	-11	-9	+7		
T_{alocal}	317	-2	-11	-2		+2	+2	
T_{bound}	472	-9	-5	-14	-9	+11	+4	
T_{nets}	296	-8		-6	-13	+5		
T_{rests}	408				-13	-74		+11
T_{dens}	480	-9		-6	-11		-33	
T_{wells}	195	-2				+3		-22

3.- Results

GELSA has been implemented in C requiring about 15000 lines of code. Its operation is illustrated herein with some examples. The first shows the flexibility of the placement algorithm to deal with user-specified topological constraints. The circuit is the fully differential folded-cascode CMOS OTA in Fig. 5 whose 28 transistors were split into 40 devices constituting the basic cells for the placer. Different local symmetries were imposed to be controlled directly by the placer (common-centroid for the input differential pair and 5 stacked structures for current mirrors), and minimization of the capacitive coupling parasitics between 7 pairs of nets was required. The resulting layout, T1, is shown in Fig. 6a (detailed routing was performed manually, using the wire locations given by the global routing generator). All the symmetry and parasitic constraints were met and the area, total routing length, and number of wells indicated in the figure were obtained. The circuit was integrated and several samples were characterized. A manually laid-out circuit with the same cell shapes was also integrated. Table II gives the simulated and experimental results measured for these circuits. Nominal results in the first column correspond to the simulated wire-parasitics-free design. In the second column, results for the GELSA generated layout T1 are shown. The fourth and fifth columns give the experimental measurements for T1 and the manual layout. In another example, the same OTA is considered but GELSA must search for an optimum solution which minimizes the unity-gain-bandwidth (GBW) deviation due to the 104 more relevant capacitive parasitics. The resulting placement, T2, is shown in Fig. 6b, and the predicted GBW deviation was 0.9 MHz. The circuit was simulated with HSPICE using the parasitics estimated by GELSA and the obtained results are indicated in column labelled T2 in Table II. Both placements took less than 250s of CPU on a SUN SPARC 10 workstation.

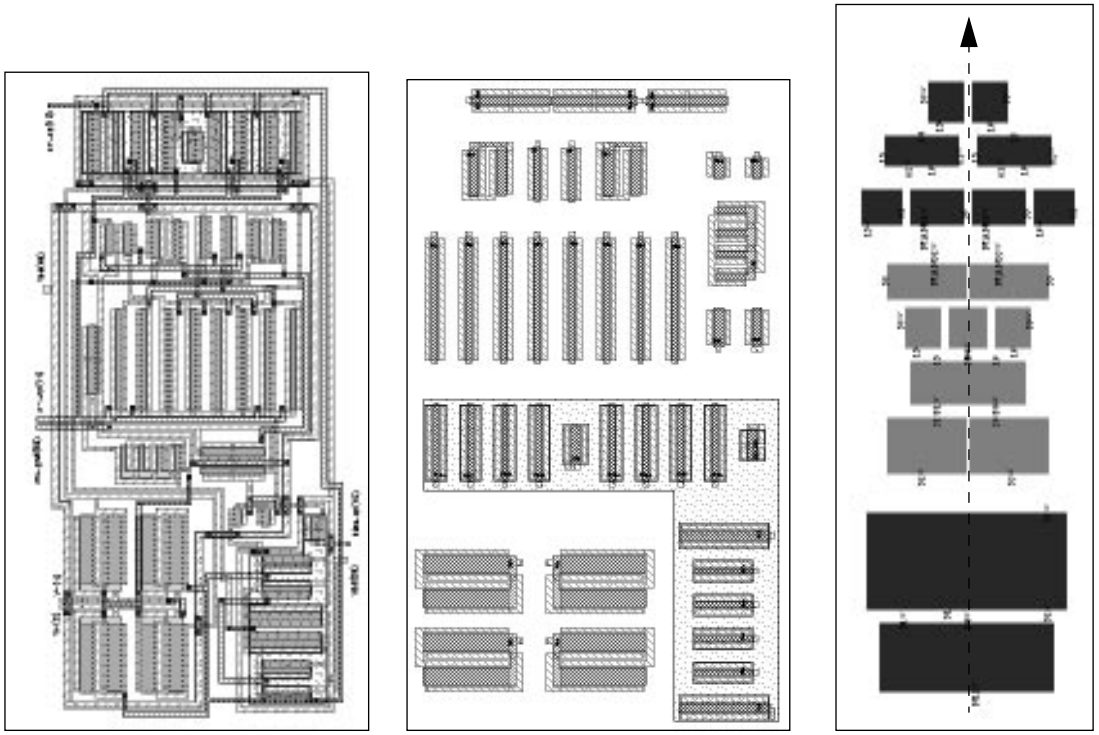
The third example is the comparator *"comp"* in [2], which serves us to illustrate the handling of global symmetry. It is one of the circuits produced by the integration of GELSA as the placer in the ALSYN system [3]. Circuit partition and module shape generation were carried out by the tools in ALSYN and the placement was performed by GELSA. Figure 6c shows one of the resulting layouts where the global symmetry axis is drawn. It required 175s of CPU on a SUN SPARC 10 workstation.

Table II: Simulated and measured characteristics of the CMOS OTA

	Simulated			*Measured*	
	Nominal	*T1*	*T2*	*T1*	*Manual*
GainDC (dB)	78.4	78.4	78.4	75.4	72.1
GBW (MHz)	27.1	25.1	26.2	21.3	21.7
Voffset (mV)	0	0	0	0.76	7.2

4.- Conclusions

A new performance-driven placement tool for automatic layout generation of analog IC's has been presented. Control of the effects inducing performance degradation of the circuit can be set in two different ways: qualitatively, e.g., designer-supplied constraints for symmetry, orientation, coupling, etc., or quantitatively as performance degradation limits to be met. The approach herein proposed exhibits the same advantages of flat approximations, offering in addition more flexibility in the calculation of geometrical properties of the solution. Moreover, as the complexity of circuit increases our slicing approach will facilitate the hi-



(a) 179x379 μm^2, L=5512µm, 2 wells

(b) 228x366 μm^2, L=5098µm, 1 well

(c) 100x241 μm^2, L=1381µm, 1 well

Figure 6: CMOS OTA placements: a) T1 (topological constraints) and b) T2 (performance-driven). c) Comparator "comp" placement. (Note: each plot has different scale)

erarchical treatment required by a high number of devices. Practical examples proved the flexibility and efficiency of our approach.

5.- References

[1] J. Rijmenants, J. Litsios, T. Schwarz, and M. Degrauwe, "ILAC: An Automated Layout Tool for Analog CMOS Circuits", *IEEE J. Solid-State Circuits*, vol. 24, no. 2, pp. 417-425, April 1989.

[2] J. Cohn, D. Garrod, R. Rutenbar, and L. R. Carley, "KOAN/ ANAGRAM II: New Tools for Device-Level Analog Placement and Routing", en *IEEE J. Solid-State Circuits*, vol. 26, no. 3, pp. 330-342, March 1991.

[3] V. Meyer zu Bexten, C. Moraga, R. Klinke, W. Brockherde, and K. Hess, "ALSYN: Flexible rule-based layout synthesis for analog IC's", *IEEE J. Solid-State Circuits*, vol. 28, no. 3, pp. 261-268, March 1993.

[4] E. Charbon, E. Malavasi, U. Choudhury, A. Casotto,and A. Sangiovanni-Vincentelli, "A Constraint-Driven Placement Methodology for Analog Integrated Circuits", *Proc. IEEE CICC*, pp. 28.2.1-28.2.4, May 1992.

[5] E. Charbon, E. Malavasi, D. Pandini, and A. Sangiovanni-Vincentelli, "Simultaneous Placement and Module Optimization of Analog IC's", *Proc. 31st IEEE DAC*, pp. 31-35, 1994.

[6] U. Choudhury, and A. Sangiovanni-Vincentelli, "Automatic Generation of Parasitic Constraints for Performance-Constrained Physical Design of Analog Circuits", *IEEE Trans. on CAD*, vol. 12, no. 2, Feb. 1993.

[7] K. Lampaert, G. Gielen, and W. M. Sansen, "A Performance-Driven Placement Tool for Analog Integrated Circuits", *IEEE J. Solid-State Circuits*, vol. 30, no. 7, pp. 773-780, July 1995.

[8] E. Malavasi, E. Charbon, G. Jusuf, R. Totaro and A. Sangiovanni-Vicentelli, "Virtual Symmetry Axes for the Layout of Analog IC's", in *Proc. 2nd ICVC*, Seoul, Korea, pp. 195-198, Oct. 1991.

[9] W. M. Dai, and E. S. Kuh, "Simultaneous Floor Planning and Global Routing for Hierarchical Buiding-Block Layout", *IEEE Trans. on Computer-Aided Design*, vol. CAD-6, no. 5, 1987.

[10] J. A. Prieto, J. M. Quintana, A. Rueda, and J. L. Huertas, "An algorithm for the Place-and-Route problem in the layout of analog circuits", *Proc. ISCAS 94*, vol. 1, pp. 491-494, June 1994.