

A Methodology for Concurrent Fabrication Process/Cell Library Optimization

Arun N. Lokanathan, Jay B. Brockman
Department of Computer Science and Engineering
University of Notre Dame

John E. Renaud
Department of Aerospace and Mechanical Engineering
University of Notre Dame

Abstract - This paper presents a methodology for concurrently optimizing an IC fabrication process and a standard cell library in order to maximize overall yield. The approach uses the Concurrent Subspace Optimization (CSSO) algorithm, which has been developed for general coupled, multidisciplinary optimization problems. An example is provided showing the application of the algorithm to optimizing a mixed analog-digital library on a CMOS process.

1.0 Introduction

In order to develop high-performance, integrated systems, IC designers are reliant upon having high-performance cell libraries. The development, characterization, and optimization of cell libraries is itself a very complex task, requiring the coordinated efforts of circuit designers and fabrication process engineers. Traditionally, circuit optimization and process optimization have been treated as separate problems. Such an approach has disadvantages in terms of not only design time but also in terms of product performance. Sequentially and iteratively performing circuit design after process design is inappropriate, because of the protracted times involved. Further, as the functional breadth of cells in libraries increases, different cells may want to “push” the process in different directions, and hence, it becomes even more difficult to find a fabrication process that is beneficial to the library as a whole.

In this paper, we present a multidisciplinary optimization approach to concurrent process and cell library development, using the Concurrent Subspace Optimization (CSSO) algorithm. Key features of the CSSO algorithm are that (i) it permits both the circuit and process engineers to simultaneously work towards the common goal of maximizing yield; (ii) permits the process designer to work with an approximate view of the cell library and the circuit designers to work with an approximate view of the process during optimization; and (iii) provides a mechanism for negotiating tradeoffs. We verify the utility of CSSO in concurrent process/library design with a simulated example using a representative sample of analog and digital cells with a CMOS fabrication process.

2.0 Related Work

In the past, work in the area of integrated circuit optimization has focussed predominantly on techniques for local design improvement in either the circuit design discipline or the process design discipline. In fact, circuit optimization is a relatively mature area; a survey of relevant literature can be found in [1]. Statistical circuit optimization that further accounts for process variations has also been studied extensively, for example, in [2], [3] and [4]. Whereas circuit optimization seeks to adjust device geometries, process optimization seeks to adjust process inputs to produce devices with acceptable model parameters [5]. To our knowledge, the issue of setting targets for device model parameters that will optimize a library of circuit cells has not been addressed.

Our approach to concurrent fabrication process/cell library optimization is based upon general techniques for *multidisciplinary design optimization* (MDO). In large part, these techniques were developed for application in the aerospace industry, where large-scale coupled, interdisciplinary optimization problems commonly arise. General references to MDO methodologies may be found in [6] while an overview of the concurrent subspace optimization algorithm, which has been developed by the authors and is used in this work, may be found in [7].

3.0 Problem Formulation

The performances of the circuits (such as speed and power dissipation) in a cell library depend upon the sizes of the transistors in the circuit as well as on the transistor device model parameters. Thus, in order to optimize the aggregate performance of the cell library, and hence systems designed with that library, two routes may be taken: adjust the device geometries of the individual cells, or adjust the fabrication process to achieve better device performance.

The relationship between the process inputs, device model parameters and the circuit performances for the cells in a system may be represented as shown in Fig. 1. The fabrication process maps a vector of process inputs P to the vector of device model parameters D . Since any of cells in the library may be integrated into a single VLSI system that will be manufactured on the fabrication process, all cells are assumed to “see” an identical set of nominal device model parameters D . The vector of performances of the i^{th} circuit

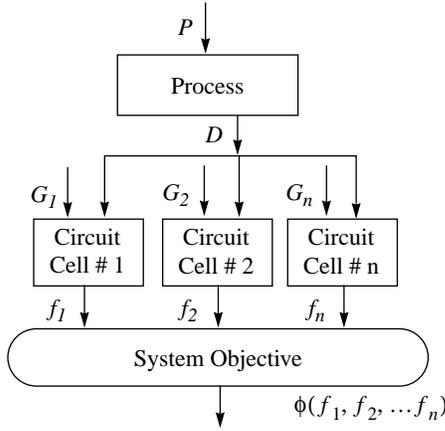


Fig. 1. IC System Performance Computation

cell, f_i , is thus a function of D and the vector of the sizes of the transistors in that circuit, G_i . Optimizing a single circuit cell then involves assuming some value for D and adjusting G_i to improve the desired performances of the cell. In addition to adjusting G_i , a circuit designer can suggest a possible change in D that if feasible, will result in improved performance for that cell. In order to formulate a system objective that considers the performances of multiple cells from a cell library, the vector of performances f_1, f_2, \dots, f_n is transformed to $\phi(f_1, f_2, \dots, f_n)$, which may be as simple as a linear combination or a more involved transformation, in general. We now present an algorithm that enables development of the process as well as the circuit library in a concurrent and coordinated manner.

4.0 Concurrent Subspace Optimization Algorithm

As previously defined, the objective of library optimization is to optimize the vector of performances $\{f_1, f_2, \dots, f_n\}$ over the design variables of the system, $X = \{P, G\}$, where f_i is the performance of the i^{th} cell. Before discussing the concurrent approach to solving this optimization problem, we will first address the limitations of the conventional approach. In the conventional approach, the above formulation would be treated as a single design problem, seeking to adjust all components of the design vector X simultaneously to improve f . The first limitation of this approach is that the dimensionality of the problem would make the optimization impractical. A second problem with treating the problem all-at-once is that the computation of gradients for the system would necessitate employing finite differences on the entire process-circuit system. The third, and most significant practical drawback of the conventional approach is that because the central optimizer makes the design changes, the roles of the circuit and process experts would be reduced from design to just analysis. This is in sharp contrast with how design is undertaken in practice, where both circuit designers and process designers use their own sets of tools and expertise to guide

the design in an independent, yet coordinated, manner.

To overcome these limitations, we have developed an approach called the *concurrent subspace optimization* (CSSO) algorithm as a general means for optimizing complex, multidisciplinary coupled systems. An important aspect of the CSSO algorithm is that each of the disciplines seeks to optimize the same global objective function, which in the case of the process/cell library problem is based on the cell library performance. This means that during the development cycle, fab performance is not only measured against electrical test results, but may also be tied to the circuit yield of the cell library. Furthermore, the cell library is concurrently optimized for a current and accurate reflection of the fabrication process.

The optimization process using the CSSO algorithm is shown in Fig. 2. The algorithm involves performing a system analysis followed by formulating and solving subspace optimization tasks in the different disciplines. In order to allow subspace design to proceed independently, each of the subspaces must have a means for approximating how design changes made in its own domain will affect the value of the global objective. This means that each subspace must be able to efficiently approximate the behavior of the other subspaces to which it is coupled. Once the independent subspaces have suggested their design moves, a coordination mechanism negotiates trade-offs. The process is then repeated until the global system objectives have been met. Hence, instead of treating the design problem as the optimization of a monolithic system, the CSSO algorithm decomposes the optimization into smaller tasks: one task for the process design subspace and one task for each circuit design subspace. Not only are the process and circuit subspace tasks lower in dimension than the original optimization task, but they are capable of being performed concurrently.

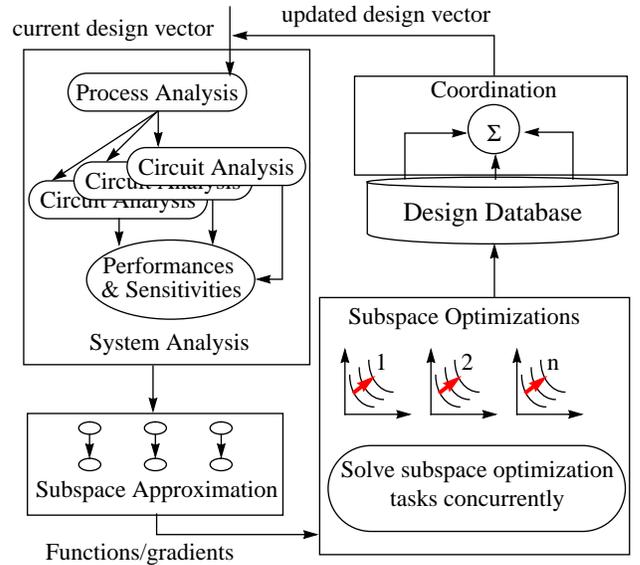


Fig. 2. The Concurrent Subspace Optimization Algorithm

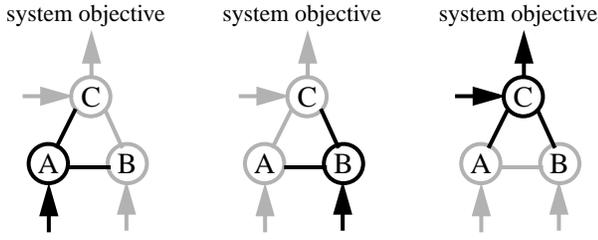


Fig. 3. System approximations as viewed by individual subspaces.

4.1 Subspace Analysis, Approximation & Optimization

In the general CSSO algorithm, as described in the previous section, each of the subspaces must have a way to approximate how local design changes impact the global objective. This step in the CSSO algorithm is referred to as *subspace approximation* and is illustrated in Fig. 3.

During subspace approximation, each subspace computes its own system states or performances using its local *contributing analysis* (a CAD tool such as a process simulator or a circuit simulator). In order to evaluate the system objective, each subspace bases its calculations on approximations developed for each of the other subspaces. Several approximation schemes, including polynomial gradient-based approximations ([7]) and response surfaces based on neural-networks ([8]), have been investigated and their impact on the efficacy of the concurrent optimization methodology have been reported in the past.

For the current application of CSSO to the process/cell library problem, first-order approximations based on gradients were used, in the same manner as [7]. The process designer approximates circuit performances using the gradients of the circuit performances with respect to process inputs. The circuit designers, on the other hand, assume the most current value of device model parameters computed by the process simulator and compute the circuit performances using a circuit simulator. The gradient or total derivatives of the system performances with respect to the design variables of the system may be obtained using subspace-level *local sensitivities* along with the *chain rule of derivatives* or, in general, the *global sensitivity equations* ([9]). In the process/cell library problem, local sensitivities consist of the partial derivatives of the circuit outputs (performances) with respect to circuit inputs (transistor geometries and model parameters) and partial derivatives of process outputs (device model parameters) with respect to process input variables. Local sensitivity computation may be facilitated by mechanisms for efficient sensitivity computation such as direct or adjoint sensitivity computation techniques ([10]) and automatic differentiation ([11]).

Using the local contributing analyses along with the remaining subspace approximations, each subspace performs a local optimization to determine updated values for its own local design variables. That is, the process designer finds new values for the process inputs and each circuit designer finds new transistor geometries, each designer bas-

ing his decision on the impact to the same global objective. Currently, a sequential, quadratic programming approach is employed for the optimization [12].

During each subspace optimization, the design points that are visited are stored in a design database. At the end of the subspace optimizations, this data is used to formulate a system-level optimization task that is solved to trade-off the subspace-optimal design moves in order to make a global design move. This strategy is explained in the next section.

4.2 System-level Coordination

The objective of the coordination mechanism is to arrive at a single combined design vector for the overall system, given the suggested subspace-optimal vectors by the subspace optimizers (process designer and circuit designer). Depending on problem characteristics, different versions of a coordination mechanism may be employed. In the following, we describe two mechanisms for coordination and discuss the classes of problems to which they are applicable.

Coordination via concatenation

One coordination scheme is to concatenate the subspace-optimal design vectors suggested by the disciplinary design teams to obtain the system-level design vector, as shown in Fig. 4. The greatest advantage of this scheme is its simplicity. In order for the mechanism to be applicable, however, the subspace design vectors cannot share any design variables. As suggested in Fig. 4, the coordinated optimal point obtained via concatenation may lie outside the constraint boundaries and hence be infeasible. Further, it has been our experience that the scheme is robust only for hierarchically coupled systems or systems that have purely feedforward coupling. The mechanism is not generally applicable to non-hierarchical systems even if the subsystems do not share independent design variables [13]. For the system consisting of a fabrication process and a library of cells, the couplings that arise are feedforward, therefore making coordination via concatenation a potential choice for this design problem. The key to being able to apply the mecha-

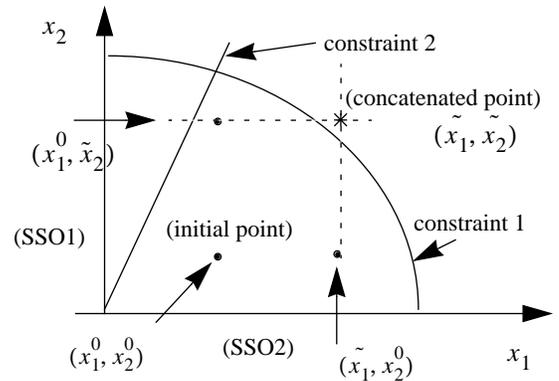


Fig. 4. Coordination via concatenation of variables

nism to the process-circuit problem is in formulating the subspace problems such that they do not share independent design variables. We will discuss such a formulation in Section 4.3.

Coordination by a System-Level Optimization

The idea behind this coordination technique is to formulate and solve an approximation to the original optimization problem ([7], [8]). The contention is that an approximation is required because the original system is so complex that a conventional optimization procedure is inapplicable directly. Hence, smaller subspace optimization tasks are formulated and solved in each discipline using the contributing analyses (process and circuit simulators) of the respective disciplines. In the process of doing so, each subspace produces a sequence of design points, generally in the direction of improving performance for that subsystem. Each design point includes the values of the design vector, the values of the system states that were approximated by that subspace, as well as the values of states that were obtained by executing the local CAD tools. Whereas traditional optimization procedures would discard the intermediate design points, this coordination mechanism requires that the visited design points be archived in a design database. At the end of the subspace optimizations, the coordination mechanism constructs a response surface using the design points visited by the subspace optimizers and solves the system level optimization over the response surface with the set of all design variables. This is illustrated in Fig. 5.

4.3 Process/Cell Library Optimization Formulation

The objective of the concurrent fabrication process/cell library optimization problem is to minimize $\{f_1, f_2, \dots, f_n\}$ over $\{P, G_1, G_2, \dots, G_n\}$, where f_i is the performance of the cell C_i and is a function of $\{P, G_i\}$. Specifically, this design problem is cast as a multi-objective optimization using the goal programming approach [14]. In order to minimize the functions $\{f_1, f_2, \dots, f_n\}$, the following scalar constrained nonlinear optimization problem is solved:

$$\begin{aligned}
 &\text{Minimize} && \gamma \\
 &\text{w.r.t.} && \gamma, P, G_1, G_2, \dots, G_n \\
 &\text{subject to} && f_1(D(P), G_1) - \gamma\alpha_1 \leq \delta_1 \\
 &&& f_2(D(P), G_2) - \gamma\alpha_2 \leq \delta_2 \\
 &&& \dots \\
 &&& f_n(D(P), G_n) - \gamma\alpha_n \leq \delta_n
 \end{aligned} \tag{1}$$

The vector $\{\delta_1, \dots, \delta_n\}$ represents the targets or goals that are set for the performances f_1, \dots, f_n respectively. $\{\alpha_1, \dots, \alpha_n\}$ is a vector of weights used to adjust the relative rigidity with which the targets need to be met. γ is a scalar that is made both an independent design variable as well as the objective function of the transformed optimization problem. The multiple objective functions of the original problem are then transformed into the constraints of the goal programming formulation. Assuming positive weights α_i , as γ is decreased, all of $f_i(D(P), G_i)$, $i = 1 \dots n$ also need to be decreased in order to satisfy their respective constraints in the goal programming problem. This leads to all the functions f_i being simultaneously minimized and traded-off for one another. Consequently, the final solution to the goal programming problem generally yields a design point that tends to occur near the weighted centroid of the acceptability region formed by the target boundary surfaces. Such a solution is particularly desirable from the standpoint of design for manufacturability in the face of statistical process variations [14].

5.0 Example

We now demonstrate the application of the concurrent design methodology discussed above to a specific example of optimizing a CMOS fabrication process targeted to multiple cells in a cell library. Clearly, it is impractical to formulate a design problem that explicitly optimizes all the cells in a given library. Instead a representative subset of the cells that spans the functional breadth of the library should be identified for purposes of formulating and solving the optimization. This is to ensure that the possibly conflicting demands that the cell library is likely to impose on the fabrication process are reflected by the small subset of cells chosen for explicit optimization. With these considerations in mind, a CMOS inverter (a typical CMOS digital logic block) and a CMOS differential pair (a typical CMOS analog circuit) were chosen to be the small but representative subset of cells for explicit optimization. For this example, the emphasis was placed on minimizing power, a typical requirement in modern IC design scenarios.

The requirements of the problem thus translate to finding the optimal set of process inputs (P , representing implant dosages, times and temperatures to a fabrication process/device simulator such as PdfabTM) and device geometries (G_1 , representing transistor sizes for the inverter and G_2 representing transistor sizes for the differ-

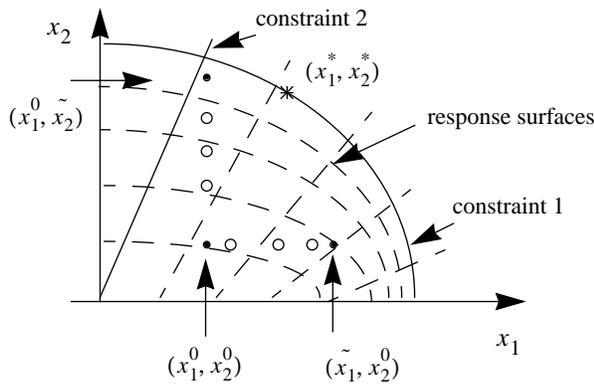


Fig. 5. Coordination via response surface construction and system-level optimization

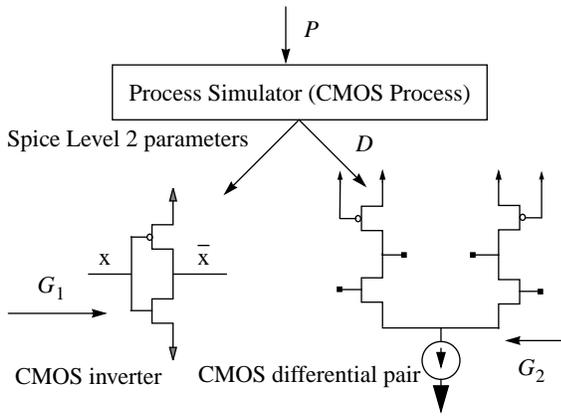


Fig. 6. Cell library optimization example

entail pair). As shown in Fig. 6, the fabrication process simulator maps process inputs to Spice level 2 device model parameters for the NMOS and PMOS FET's. These parameters are used along with the respective netlists for the inverter and the differential pair as inputs to the circuit simulator in order to compute the set of circuit performances.

The specific implementation of the CSSO algorithm as applied to this problem is depicted in Fig. 7. At every outer iteration, a process simulation is performed followed by circuit simulations for each circuit. The process and circuit simulators are both capable of providing not only output values but also their partial sensitivities with respect to their respective inputs. The global sensitivity equations are then

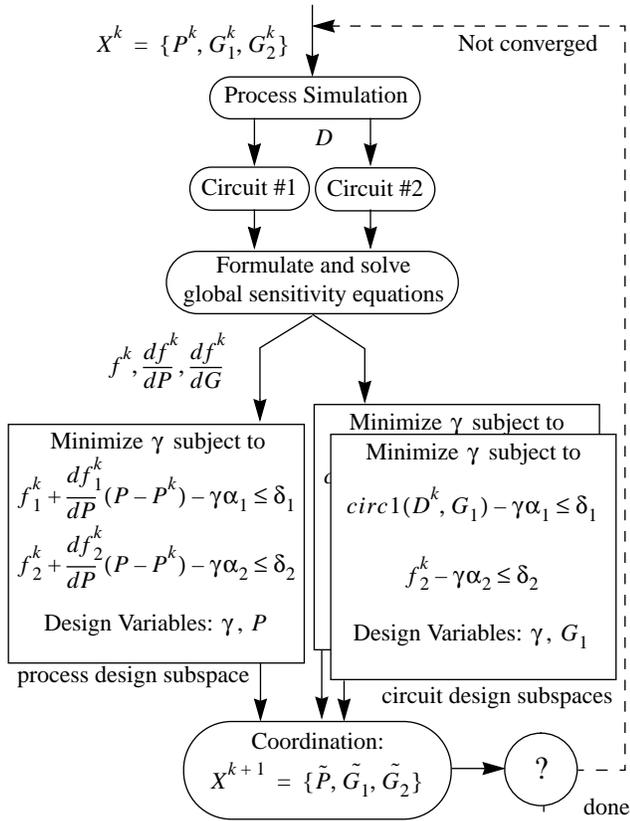


Fig. 7. Application of Concurrent Methodology to Design Example

formulated (using the partial sensitivities) and solved to obtain total gradients.

As can be seen from the Fig. 7, in the process subspace optimization or design task, the current gradient information is used to build first order approximations of the circuit performances thereby relieving the process designer of having to run expensive circuit simulations. In the circuit subspaces, circuit designers adjust the respective transistor geometries to optimize performances, assuming constant values (updated at every outer iteration) for device model parameters. In each of the subspaces, the design variables of that subspace are adjusted to solve the goal programming formulations of Eq. 1 that minimize their respective scalar goal attainment factors. At the end of the subspace optimizations, the physical design variables are concatenated to yield the system-level optimal design vector.

6.0 Results

The concurrent subspace optimization algorithm was applied to the problem of optimizing the CMOS inverter and the CMOS differential amplifier with respect to circuit geometries and CMOS fabrication process inputs. The goal of the optimization was to minimize power dissipation for both circuits. Because the purpose of the CSSO algorithm is to find a process that is maximally beneficial to the library as a whole, an appropriate way to evaluate the results is by comparing cell performances when they are optimized both separately and concurrently. In other words, the results of three separate optimizations should be compared:

- find the optimal geometries and performance for both circuits, given that the fabrication process has been optimized for the inverter only
- find the optimal geometries and performance for both circuits, given that the fabrication process has been optimized for the differential amplifier only
- find the optimal geometries and performance for both circuits, given that the fabrication process is optimized for both circuits concurrently using the CSSO algorithm

Table 1 shows the results obtained from each of these three optimization experiments. For clarity, the table indi-

TABLE 1. Sample Optimal Model Parameters

	NFET v_{t0}	NFET K_n	Inverter power	Diff-pair power	Total power
	V	A/V ²	mW	mW	mW
Optimized for inverter	1.11	5.75e-5	0.839	2.615	3.454
Optimized for diff-pair	0.22	17.25e-5	1.784	2.104	3.888
CSSO	0.65	5.75e-5	0.917	2.446	3.363

cates the values of just two of the Spice level-2 device model parameters, NFET threshold voltage and NFET pro-

cess gain, for the optimal designs. Observe (from the first two rows of the table), that a combination of process inputs that produces an optimal inverter does not produce an optimal differential pair and vice versa. By setting weights and targets on the performances of either circuit in the formulation of Eq. 1 and applying the CSSO algorithm to solve the formulation, the performance of one circuit is traded off for the performance of the other to obtain the system level optimal design. Again, a key contribution of the methodology is that during the design flow, the optimization tasks in the process discipline and the two circuit disciplines are decoupled and are capable of proceeding concurrently.

7.0 Conclusions

The problem of refining a semiconductor fabrication process to enhance the performance of a library of circuit cells has been formulated as a *multidisciplinary optimization* (MDO). A conventional optimization procedure is inapplicable directly to the problem because (i) the problem is very complex in that analyzing the system for performances involves *sequentially* performing expensive process and circuit simulations, and (ii) a conventional procedure restricts subspace design freedom. We have demonstrated the utility of applying the *Concurrent Subspace Optimization* (CSSO) algorithm as a solution strategy that overcomes both the above shortcomings of a conventional methodology.

First, the CSSO algorithm enables the fabrication process designers to estimate and improve circuit performances even while circuit designers are *concurrently* designing the circuits for that process. Second, the CSSO algorithm casts the original problem as a number of temporarily decoupled and independent subtasks that are assigned to subdiscipline design teams. The algorithm does not enforce a mechanism for design changes at the subspace level. Consequently, design moves can be made by discipline experts who may use experience and heuristics in addition to the local CAD tools to guide the design changes. By formulating the problem as a goal programming problem, the performances of multiple circuit cells could be traded off to meet system level targets. An example involving the optimization of a CMOS digital block and a CMOS analog circuit that imposed conflicting demands on the fabrication process has been used to illustrate the application of the concurrent methodology to fabrication process/cell library design.

Acknowledgments

This effort was supported in part by the 1996 ACM/IEEE Design Automation Scholarship, NASA Research Grant NAG-1-1561, NSF Research Grant DMI93-08083 and Motorola Inc., Austin, TX.

References

- [1] R. K. Brayton, G. D. Hachtel, and A. L. Sangiovanni-Vincentelli, "A Survey of Optimization Techniques for integrated circuit design", *Proceedings of the IEEE*, vol. 69, pp. 1334-1362, October 1981.
- [2] A. Dharchoudhary, and S. M. Kang, "Worst-Case Analysis and Optimization of VLSI Circuit Performances", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 4, April 1995.
- [3] K. J. Antreich, H. E. Graeb, and C. U. Weiser, "Circuit Analysis and Optimization Driven by Worst-Case Distances", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 1, January 1994.
- [4] S. W. Director, and W. Maly, "*Statistical Approach to VLSI*", Advances in CAD for VLSI, vol. 8, North-Holland, 1994.
- [5] K. K. Low, and S. W. Director, "A New Methodology for Design Centering of IC Fabrication Processes", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, no. 7, July 1991.
- [6] R. J. Balling, and J. Sobieszczanski-Sobieski, "Optimization of Coupled Systems: A Critical Overview of Approaches", In *Proceedings of The AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Panama City, FL, September 7-9, 1994.
- [7] J. E. Renaud, and G. A. Gabriele, "Improved Coordination in Non-Hierarchic System Optimization", *AIAA Journal*, vol. 31, num. 12, Editor-In-Chief, George W. Sutton, Published by the American Institute of Aeronautics and Astronautics, Washington, DC, pp. 2367-2373. December 1993.
- [8] J. E. Renaud, R. S. Sellar, S. M. Batill, and P. Kar, "Design Driven Coordination Procedure for Concurrent Subspace Optimization in MDO", *Proceedings of the 35th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, April 1994.
- [9] J. Sobieszczanski-Sobieski, "On the Sensitivity of Complex, Internally Coupled Systems", In *Proceedings of the 29th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference*, Williamsburg, VA, April 1988.
- [10] S.W. Director and R.A. Rohrer, "The Generalized Adjoint Network and Network Sensitivities", *IEEE Transactions on Circuit Theory*, vol. 16, num. 3, August 1969, pp. 318-323
- [11] P. Feldmann, R. Melville, and S. Moinian, "Automatic Differentiation in Circuit Simulation and Device Modeling", *Proceedings of IEEE Conference on CAD*, pp. 248-253. IEEE, 1992.
- [12] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, "*Nonlinear Programming: Theory and Algorithms*", Second Edition, John Wiley & Sons, Inc., 1993.
- [13] A. N. Lokanathan, J. B. Brockman, and J. E. Renaud, "A Multidisciplinary Approach to Integrated Circuit Design", *Proceedings of Concurrent Engineering 95: Research and Applications*, August 1995.
- [14] R. K. Brayton, S. W. Director, G. D. Hachtel, L. Vidigal, "A New Algorithm for Statistical Circuit Design Based on Quasi-Newton Methods and Function Splitting", *IEEE Transactions on Circuits and Systems*, vol. CAS-26, pp. 784-794, 1979.