The Simplicial Approximation Approach to Design Centering

STEPHEN W. DIRECTOR, SENIOR MEMBER, IEEE, AND GARY D. HACHTEL, MEMBER, IEEE

Abstract—**The basis of a method for designing circuits in the face of parameter uncertainties is described. This method is computationally cheaper than those methods which employ Monte Carlo analysis and nonlinear programming techniques, gives more useful information, and more directly addresses the central problem of design centering. The method, called simplicial approximation, locates and approximates the boundary of the feasible region of an n-dimensional design space with a polyhedron of bounding $(n-1)$-simplices. The design centering problem is solved by determining the location of the center of the maximal hyperellipsoid inscribed within this polyhedron. The axis lengths of this ellipsoid can be used to solve the tolerance assignment problem. In addition, this approximation can be used to estimate the yield by performing an inexpensive Monte Carlo analysis in the parameter space without any need for the usual multitude of circuit simulations.**

I. Introduction

DESIGNING in the face of statistical uncertainty is a pervasive engineering problem, particularly in the field of semiconductor integrated circuits [1]–[6]. Here the values of all circuit components are subject to statistical fluctuations inherent to the thermal, chemical, and optical processes used in this technology. Due to these fluctuations it is likely that some percentage of the total number of circuits manufactured will not meet the desired design specifications. In such a situation design emphasis is placed primarily on obtaining a satisfactory yield, i.e., choosing a nominal design which maximizes the number of circuits which meet specification. We refer to this type of design as *design centering*. In this paper we present a new, simple, efficient method for design centering which is particularly well suited for integrated circuits.

Although the design centering problem is of great importance, relatively little work has been reported on its solution. For the most part, Monte Carlo analysis [4],[5], albeit crude and expensive, has been widely used as an approach to design centering. More recently, Bandler *et al.* [1] have proposed a design centering procedure which is directly related to classical nonlinear programming methods [7]. This approach is based upon selecting, for each individual problem, a scalar objective functional which when minimized, inscribes a hypercube in the "feasible region" R of the n-dimensional parameter space.[1] However, the nonlinear programming approach does not attempt, as does the new procedure described below, to approximate the boundary of the feasible region but only determines points on this boundary.[2] Although the simplicial approximation method has only been used on a few problems, the results obtained suggest that this method will require far fewer circuit simulations (i.e., solutions of the differential-algebraic equations) than the nonlinear programming and Monte Carlo approaches in typical applications. This point is important because the transient simulation of an integrated circuit usually requires a significant amount of computer time. In fact on some of the problems solved using this method, the number of circuit simulations required was about the same as that required for a single unconstrained optimization [7]. On the other hand, the constrained optimization procedure required by the nonlinear programming approach inherently involves a *sequence* of unconstrained optimizations. The constrained nonlinear program is solved by means of a Fiacco–McCormick penalty function method. While this technique is less dependent on the convexity assumption made in the method to be described below, its utility seems to be more suitable for linear circuit applications than the time domain, nonlinear circuit applications which arise in integrated circuit designs.

The proposed method, which we call "simplicial approximation," is based on explicitly approximating the boundary, ∂R, of the feasible region, R, of an n-parameter design space by a polyhedron made up of n-dimensional simplices.[3] Approximation is necessary since ∂R is generally known only in terms of nonlinear inequality constraints which express acceptable circuit performance in terms of voltages and currents which depend implicitly

Manuscript received November 19, 1976; revised February 10, 1977. This work was supported in part by the National Science Foundation under Grant ENG 75-23078.

S. W. Director is with the Department of Electrical Engineering, University of Florida, Gainesville, FL 32611. This work began while he was a Visiting Scientist at IBM T. J. Watson Research Center.

G. D. Hachtel was a Regent's Lecturer at the University of California, Los Angeles, CA. He is now with the Department of Mathematics, University of Denver, Denver, CO 80210.

[1]The feasible region, or region of acceptability, R, is an n-dimensional subspace of the parameter space P which contains all points $p \in P$ which result in circuits that meet the desired specifications.

[2]As will be pointed out later, an explicit approximation of the boundary of the feasible region is useful for solving a variety of other statistical design problems [2], [3], [6] as well as allowing one to obtain a Monte Carlo estimate of the yield at a reduced computational cost.

[3]Butler, in [6], was one of the first to try and approximate the feasible region, however he was interested in graphical displays of the region and therefore approximated only projections of this region onto two dimensional subspaces.

upon the design parameters.[4] We assume that the constraint functions are locally convex, i.e., that the sequence of points generated on ∂R are extreme points of a convex set. The case where this assumption is violated involves mathematical representation of a nonconvex polyhedron, but can still be handled, as we shall discuss in a subsequent report.

We begin our formal discussion in Section II by defining more precisely some of the terms we use, such as yield and design center. A conceptual framework for the simplicial approximation approach to design centering is outlined in Section III. In this section a systematic procedure for initializing and refining the simplicial approximation of ∂R is also given. Furthermore, a theorem is given which describes conditions under which improvement of the simplicial approximation to the boundary ∂R of the feasible region may be assured. In Section IV we describe a scaling strategy which converts the inscription of a hypersphere into that of a hyperellipsoid. We then give the results (Section V) of applying an implementation of the simplicial approximation procedure for determining the boundary ∂R of the feasible region R which arises from a practical integrated circuit design problem. In order to provide some perspective to our work, in Section VI we relate what has been done here to some related work which has been reported in the operations research literature. Finally, in Section VII we discuss some areas of future work.

II. Definitions of Yield and Design Center

Let the network under consideration obey the set of differential-algebraic equations

$$\mathfrak{N}(\boldsymbol{x},\dot{\boldsymbol{x}},\boldsymbol{p},t)=\boldsymbol{0}, \qquad 0\leqslant t\leqslant T \tag{2-1}$$

where $\boldsymbol{x}$ is a vector of node voltages, branch voltages, and branch currents, and $\boldsymbol{p}$ is an n-vector of statistically varying parameters with a joint probability density function $g(\boldsymbol{p})$. Further let P denote the n-dimensional parameter space so that $\boldsymbol{p}\in P$. Thus a network is defined by a nominal set of parameter values $\boldsymbol{p}^0\in P$. In addition a network will be considered acceptable if the solution $\boldsymbol{x}(t,\boldsymbol{p})$ of (2-1) obeys the constraints

$$\Phi_i(\boldsymbol{p})=\int_0^T \phi_i(\boldsymbol{x},\dot{\boldsymbol{x}},\boldsymbol{p},t)\,dt\leqslant 0, \qquad i=1,2,\cdots,n_c. \tag{2-2}$$

Note that the constraints Φ_i depend on $\boldsymbol{p}$ implicitly through the solutions $\boldsymbol{x}(t,\boldsymbol{p})$ of the differential-algebraic equations, as well as explicitly through their role as arguments of the $\phi_i(\boldsymbol{x},\dot{\boldsymbol{x}},\boldsymbol{p},t)$. That is, the specifications $\Phi_i\leqslant 0$ are given in terms of output variables, $\boldsymbol{x}$, i.e., the output space X (e.g., currents, voltages, delay times, etc.) whereas the statistical design takes place in terms of the designable parameters, $\boldsymbol{p}$, i.e., the parameter space, P (e.g., in integrated applications these are diffusion depths, sheet resistivities, and line widths).

Given a set of constraints (2-2), the parameter space can be separated into two regions: the region of acceptability, R, and the failure region, F. These regions are defined as follows:

$$R\equiv\{\boldsymbol{p}\mid\Phi_i(\boldsymbol{p})\leqslant 0, \quad \text{for all } i\in\{1,2,\cdots,n_c\}\} \tag{2-3}$$

$$F\equiv\{\boldsymbol{p}\mid\Phi_i(\boldsymbol{p})>0, \quad \text{for at least one } i\in\{1,2,\cdots,n_c\}\}. \tag{2-4}$$

In what follows we assume that R is convex. Of particular interest will be the boundary of R denoted by ∂R and defined by

$$\partial R\equiv\{\boldsymbol{p}\mid\Phi_i(\boldsymbol{p})\leqslant 0, \quad \text{for all } i\in\{1,2,\cdots,n_c\}$$

and

$$\Phi_k(\boldsymbol{p})=0, \quad \text{for at least one } k\in\{1,2,\cdots,n_c\}\}. \tag{2-5}$$

We can now formally define *yield* in terms of the probability density function g and the feasible region, R, by the relation

$$Y=\text{prob}\,(\boldsymbol{p}\in R)=\int\int_R\cdots\int g(\boldsymbol{p})\,d\boldsymbol{p} \tag{2-6}$$

where the integration in (2-6) is over the entire feasible region R.

The purpose of design centering can now be stated as that of choosing the nominal values $\boldsymbol{p}^0$ of the designable parameters so that yield Y is maximum for a given distribution $g(\cdot)$. Since in general the distributions are known and fixed, the design centering problem can be defined to have two clearly delineated phases, namely,

a) determining the feasible region R and the design center, and
b) evaluating the yield integral (2-6).

III. Design Centering by Simplicial Approximation

The method of simplicial approximation is based on the approximation of the boundary ∂R of the feasible region R by a polyhedron, i.e., by the union of those portions of a set of n-dimensional hyperplanes which lie inside ∂R or on it. The procedure begins by determining any $m\geqslant n+1$ points $\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_m$ on the boundary ∂R. Usually m is taken to be either $n+1$ or $2n$. (One way to find $2n$ points is by first finding a feasible set of designable parameters inside R and then performing one-dimensional line searches in the positive and negative coordinate directions. See Appendix A for a description of the line search.) The feasible point can be obtained either by designer insight or through the use of an optimization program. Given a set of m points on R, the convex hull (poly-

[4]It is important to recognize that determination of the voltages or currents in a circuit for a given set of design parameters requires a complete circuit simulation, i.e., the solution of a set of nonlinear differential-algebraic equations.

hedron) of these points is then constructed.[5] The convex hull is characterized by the set of m_H inequalities

$$\boldsymbol{c}_k^T\boldsymbol{p} \leqslant b_k, \qquad k=1,2,\cdots,m_H \tag{3-1}$$

where $\boldsymbol{c}_k$ is a unit "outward pointing," vector normal to the kth hyperplane defined by

$$\boldsymbol{c}_k^T\boldsymbol{p} = b_k, \qquad k=1,2,\cdots,m_H \tag{3-2}$$

and b_k is a measure of the distance of the kth hyperplane from the origin. Moreover, the convex hull of the set of points $\boldsymbol{p}_j$, $j=1,2,\cdots,m_H$ approximates ∂R in an *interior* sense. That is, under our assumption of convexity of R, every point interior to these boundary of hyperplanes is also inside ∂R.

Given the first approximation to ∂R we can find a first estimate of the design center by determining the center of the largest hypersphere which can be inscribed inside the polyhedron of m_H hyperplanes described by (3-2). The center of the largest hypersphere which can be inscribed within this polyhedron can be found quite easily using a linear programming approach. First recognize that the distance from a point $\boldsymbol{p}^*$ inside the polytope to the kth hyperplane is

$$d_k = \boldsymbol{c}_k^T\boldsymbol{p}^* - b_k. \tag{3-3}$$

Therefore the center and radius of the largest hypersphere can be found by determining the maximum value of r and the point $\boldsymbol{p}^*$ for which

$$\boldsymbol{c}_k^T\boldsymbol{p}^* + r \leqslant b_k, \qquad k=1,2,\cdots,m_H. \tag{3-4}$$

This linear program is the dual of the following standard (primal) linear programming problem.

Minimize the objective function

$$\Phi = \sum_{k=1}^{m_H} b_k\lambda_k \tag{3-5}$$

subject to the constraints

$$\begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_{m_H} \\ \downarrow & \downarrow & & \downarrow \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \cdot \\ \cdot \\ \lambda_{m_H} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \tag{3-6}$$

and $\lambda_k \geqslant 0$ for all k. The primal form of this linear programming problem is more desirable because the number of unknowns is less than the number of constraints. Any standard linear programming computer subroutine can be used to solve (3-5) and (3-6) and as a byproduct yield the results of the dual problem, i.e., the radius r and center $\boldsymbol{p}^*$ of the largest inscribed hypersphere.

It is important to recognize the fact that the computational effort required to determine the convex hull of a set of boundary points and then inscribe a hypersphere in this convex hull is usually considerably less than the computational effort required to perform a nonlinear transient circuit simulation. In fact an assumption that is made throughout this paper (and experience shows it to be a reasonable assumption) is that the computational cost of performing a circuit simulation far exceeds the computational cost of solving any linear program or of performing any of the matrix-vector manipulations required by our method.

The next step in the design centering procedure is to improve the simplicial approximation to ∂R and then update the design center. This step is accomplished by determining which of the m_H "faces" of the polyhedron (i.e., which of the m_H truncated hyperplanes (3-2)), that are tangent to the largest inscribed hypersphere is the "largest." The largest tangential face is of particular interest since it is the one which is most likely the poorest approximation to ∂R. The faces of the polyhedron which are tangent to the largest inscribed hypersphere are those associated with the "zero slack variables" used in solving the linear program (3-5), (3-6). The largest tangential face is defined as the face in which the largest $(n-1)$-dimensional hypersphere may be inscribed.

The procedure for finding the largest hypersphere which can be inscribed in a face of the approximating polyhedron is only a slight variation of the linear programming procedure outlined above for finding the largest hypersphere inscribed in the polyhedron itself. Suppose we wish to find the largest hypersphere inscribed in the jth face of the polyhedron. The center of this hypersphere, denoted by $\boldsymbol{p}_j^*$, must be on the jth hyperplane so that

$$\boldsymbol{c}_j^T\boldsymbol{p}_j^* = b_j. \tag{3-7}$$

Now let $\boldsymbol{c}_j^\perp$ denote a unit vector perpendicular to $\boldsymbol{c}_j$, i.e.,

$$(\boldsymbol{c}_j^\perp)^T\boldsymbol{c}_j = 0. \tag{3-8}$$

Thus $\boldsymbol{c}_j^\perp$ lies in the jth hyperplane and the surface of a hypersphere of radius r_j centered at $\boldsymbol{p}_j^*$ which lies in the jth hyperplane is described by

$$\boldsymbol{p}_j^* + r_j\boldsymbol{c}_j^\perp. \tag{3-9}$$

The largest such hypersphere is the one with as large an r_j as possible subject to the constraint that all points on the surface of this hypersphere lie within the approximating polyhedron, i.e., they satisfy the constraints

$$\boldsymbol{c}_k^T(\boldsymbol{p}_j^* + r_j\boldsymbol{c}_j^\perp) \leqslant b_k, \quad \text{for } k=1,2,\cdots,m_H; \qquad k\neq j \tag{3-10}$$

which can be shown to be equivalent to the constraints

$$\boldsymbol{c}_k^T\boldsymbol{p}_j^* + r_j \sin\phi_{jk} \leqslant b_k \tag{3-11}$$

[5]The convex hull can easily be constructed by considering the hyperplanes defined by combinations of the m boundary points taken n at a time. All hyperplanes which have all boundary points to one side are part of the convex hull. There are more efficient procedures to determine the convex hull, however since this step is performed only once in the entire procedure, efficiency is not of prime importance here.

where ϕ_{jk} is the angle between $\boldsymbol{c}_k$ and $\boldsymbol{c}_j$. Thus the linear program to be solved is to determine $\boldsymbol{p}^*$ and maximize r subject to (3-8) and (3-11). Note that (3-8) can be expressed as

$$\boldsymbol{c}_j^T \boldsymbol{p}^* \leqslant b_j \tag{3-12}$$

and

$$-\boldsymbol{c}_j^T \boldsymbol{p}^* \leqslant -b_j \tag{3-13}$$

so that this linear program is the dual form of the following standard (primal) linear programming problem.

Minimize the objective function

$$\Phi = \sum_{k=1}^{m_H+1} b_k \lambda_k \tag{3-14}$$

subject to the constraints

$$\left[\begin{array}{ccccccc} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_j & \cdots & \boldsymbol{c}_{m_H} & -\boldsymbol{c}_j \\ \downarrow & \downarrow & & \downarrow & & \downarrow & \downarrow \\ \hline \sin\theta_{1j} & \sin\theta_{2j} & \cdots & 0 & \cdots & \sin\theta_{m_H j} & 0 \end{array}\right] \cdot \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \cdot \\ \cdot \\ \cdot \\ \lambda_{m_H+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \tag{3-15}$$

and $\lambda_k \geqslant 0$ for all k. Note that in (3-14) $b_{m_H+1} = -b_j$.

The above procedure is repeated for each tangential face, the largest face being the one with the largest inscribed hypersphere.[6] Once the largest face is determined, a one-dimensional search is made in the outward normal direction, starting from the center of the largest inscribed hypersphere in that face, for a new point on the boundary. This new point is then added to the set of boundary points previously found and a new convex hull is found.[7] Thus our approximation to ∂R has been improved and a new design center can be found. The entire procedure can then be repeated.

In summary, the proposed design centering procedure, illustrated for two dimensions in Fig. 1, is as follows.

[6]In general, for an n-dimensional parameter space, there will be $n+1$ tangential faces. Thus there are about $n+1$ linear programs that must be solved, which is much fewer than if the size of all the faces of the polyhedron had to be found.

[7]Determining the new convex hull at this point requires two steps. First all faces of the original polytope for which the defining inequality (3-1) no longer holds when the new boundary point is considered must be deleted. Next hyperplanes formed by combinations of $(n-1)$ points taken from the set of points which define the deleted faces and the new boundary point are tested to see if all boundary points are on one side. If this test is passed, the hyperplane is part of the convex hull, if not it is ignored.

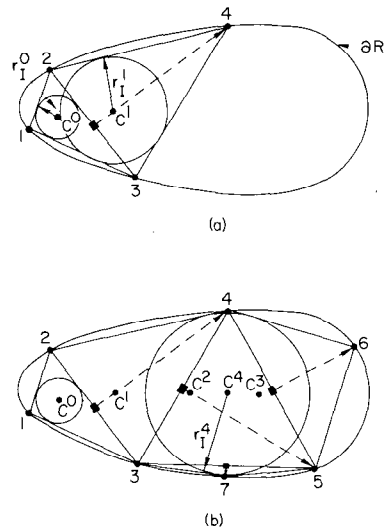

Fig. 1. Illustration of simplicial approximation. (a) Initial polyhedron defined by points (1,2,3) and breaking largest face (2-3). (b) Resulting polyhedron after four inflation iterations.

Design Centering Procedure

a) Determine a set of $m \geqslant n+1$ points on the boundary ∂R.

b) Find the convex hull of these points and use this polyhedron as the initial approximation to ∂R. Set $k=0$.

c) Inscribe (using the linear program (3-5)) the largest hypersphere in this approximating polyhedron and take as the first estimate of the design center the center of this hypersphere.

d) Find (using the linear program (3-14)) the "midpoint" of the "largest" face in the polyhedron which is tangent to the inscribed hypersphere.

e) Find a new boundary point[8] on ∂R by searching (using, for example, the search strategy outlined in Appendix A) along the outward normal of the "largest" face found in step d) extending from the "midpoint" of this face.

f) Inflate the polyhedron by forming the convex hull of all previous points plus the new point generated in step e).

g) Find (using the linear program (3-5)) the center $\boldsymbol{p}_c^k$ and radius r_I^k of the largest hypersphere inscribed in the new polyhedron found in step f). Set $k = k+1$. Go to step e).

A key step in the above is step e) since it involves a line search along a given direction. Each step in this search involves first solving the circuit equations (2-1) and then evaluating the constraints (2-2) to see which, if any, are violated. The search ends when point $\boldsymbol{p}_B$ is located for which

$$\Phi_j(\boldsymbol{p}_B) = 0, \quad \Phi_i(\boldsymbol{p}_B) \leqslant 0, \qquad \forall i \neq j. \tag{3-16}$$

The above procedure will rapidly generate a sequence of points whose convex hull forms an interior polyhedron,

[8]It can occur that the largest face actually lies in ∂R. We test for this case, and if it occurs, the face is flagged, and never again considered as a candidate for step d). In this case, step d) is repeated using the next largest face.

∂R_I, which approximates ∂R. Step g), the process of inscribing the largest hypersphere in ∂R_I, can be used as a means of monitoring the convergence of $\partial R_I \to \partial R$. If R is convex, as per our assumption, the sequence of centers $\{\boldsymbol{p}_c^k\}$ will converge to the center of the largest hypersphere that can be inscribed in ∂R, and the associated sequence of radii, $\{r_I^k\}$, will converge to the radius of the largest inscribed hypersphere. This sequence is considered to be converged when

$$|r_I^{k+1} - r_I^k| \leqslant \epsilon_R r^k + \epsilon_A \tag{3-17}$$

where ϵ_R and ϵ_A are given relative and absolute convergence parameters.

Convergence is expected on the basis of the following.

Theorem (3-18)

1) Let r_I^k be the radius of the largest hypersphere inscribed in a polyhedron ∂R_I^k formed from $n+1+k$ points on a convex surface ∂R in n-space.

2) Let r^{k+1} be the radius of the largest hypersphere inscribed in a polyhedron ∂R_I^{k+1} formed by applying the inflation step of the design centering procedure to ∂R_I^k, then

$$r_I^{k+1} \geqslant r_I^k. \tag{3-19a}$$

3) Further, if the outward normals to the faces of ∂R_I^k are not parallel and if the inscribed hypersphere is tangent to only $n+1$ faces, then

$$r_I^{k+1} > r_I^k. \tag{3-19b}$$

Note that condition 3) of Theorem (3-18) is not restrictive in practice because if it is violated, a finite number of passes through the design centering procedure will cause condition 3) to be satisfied. For example, if two tangential faces are parallel, breaking the largest face may not allow the sphere to grow. However, upon reapplication of the procedure often enough (no more than n reapplications are necessary), then one of the parallel faces will eventually be broken. Thus we have as a corollary the following.

Corollary

$$r_I^{k+1+l} > r_I^k, \qquad 0 < l \leqslant n. \tag{3-19c}$$

Assume that there are no unsuspected surface pathologies which would cause the simplicial approximation process to converge to some limit *inside* ∂R.[9] Then, it would follow from the above theorem and corollary that the approximating simplex ∂R_I^k would, as $k \to \infty$, fill ∂R like a ball being inflated by an air pump. Theorem (3-18) can be proven by observing that the new polyhedron contains the old polyhedron. The corollary follows from the fact that if two faces are parallel, breaking one of them destroys this relationship.

A word about the computational effort required to use the above procedure is in order. Our basic premise is that this procedure is to be used in situations where the number of statistically varying design parameters is small compared to the number of the algebraic-differential equations that need to be solved in order to determine circuit behavior. This situation is common in the design of integrated circuits where the variations in circuit parameters are related to the variations in a much smaller number of processing parameters. For these cases the dimensionality of the linear programs that are solved in the above procedure is relatively small when compared to the dimensionality of the circuit simulation problem. Thus the time spent in generating and inflating the convex hull is small when compared to the time involved in performing a transient simulation of a large nonlinear network. Based upon the above reasoning the most expensive operation in the above procedure is the search step e) to find a new boundary point. Therefore the search step must be efficient. At this juncture we do not have sufficient data to determine the relationship between the total number of circuit simulations required to perform design centering and the dimensionality of the parameter space. However, since there is no nonlinear programming involved here, we feel that the total number of circuit simulations required would be considerably fewer than that required by some other design centering procedures.

Before leaving this section we note that with each interior approximation ∂R_I^k may be associated a yield

$$Y_I^k = \int \int_{R_I^k} \cdots \int g(\boldsymbol{p}, \boldsymbol{p}^0, \sigma_i)\, d\boldsymbol{p} \tag{3-20}$$

where R_I^k is the volume in n-space bounded by ∂R_I^k, i.e., the kth-interior approximation to R. Note also that since ∂R_I^k consists of interior bounding hyperplanes

$$Y_I^{k-1} < Y_I^k \leqslant Y$$

where Y is computed over the true feasible region R (cf. (3)). That is, the interior simplicial approximation gives a lower bound on the yield. Moreover calculation of this yield requires only generation of random sets of parameter values and determination of the number which fall inside ∂R_I, i.e., only a Monte Carlo analysis on the input space is required; no multitude of circuit simulations is necessary. (For further discussion of these ideas see [20].)

IV. Scaling–Inscribing a Hyperellipsoid

If the design centering procedure outlined in Section III is employed when the region of acceptability is rectangular in shape, the best possible results may not be obtained. For elongated regions of acceptability, it would be more appropriate to determine the design center by inscribing a hyperellipsoid rather than a hypersphere. Fortunately this

[9] Theorem 32 of [11] shows that the above procedure will produce a convergent subsequence as $k \to \infty$. Theorem 33 of [11] shows that ∂R can be approximated by a polyhedron to arbitrary accuracy. However, we have as yet no proof that $\partial R^k \to \partial R$ as $k \to \infty$. This has not been a problem in practice.

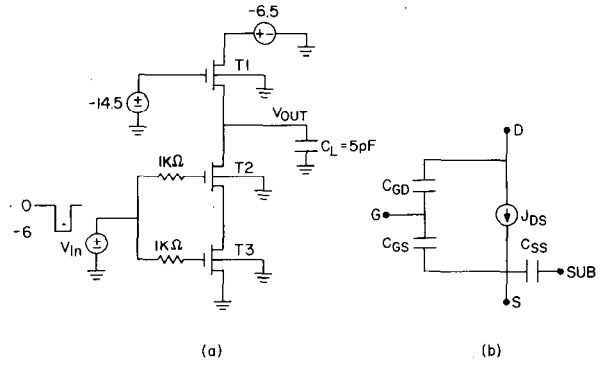

Fig. 2. (a) Two input NAND gate used in the sample. (b) FET model used: $J_{DS} = G_m(V_{GS} - V_T)^2 W/L$ above pinch-off and $J_{DS} = G_m V_{DS}(V_{GS} - V_T - V_{DS}/2)W/L$ below pinch-off, where W is width and L is length of device. Capacitances are functions of W and L: $C_{GS} = (L + \alpha)W\beta$; $C_{GO} = \alpha W\beta$; and $C_{SS} = (W/\gamma + \delta)\epsilon$; where $\alpha, \beta, \gamma, \delta$, and ϵ are constants.

result is easily obtained by merely scaling the parameters prior to employing the design centering procedure. One scheme for scaling, which has proven to be successful, is based upon using an estimate of the spreads in possible parameter values. This technique is employed in our computer implementation and is described below.

Assume that the mth step of the design centering procedure has been carried out and that $n+m+1$ boundary points $\{\boldsymbol{p}_k\}$, $k = 1,2,\cdots,n+m+1$ have been found. The lower and upper bounds of each parameter are now determined

$$p_i^L = \min_k \{p_{ik}\}$$

and

$$p_i^U = \max_k \{p_{ik}\}$$

for $i = 1,2,\cdots,n$ where p_{ik} is the ith coordinate of the kth-boundary point. The scale factor for the ith component of the parameter vector is given by

$$S_i = p_i^U - p_i^L, \qquad i = 1,2,\cdots,n$$

and a scale matrix is defined

$$\boldsymbol{S} = \text{diag}\,\{S_1, S_2, \cdots, S_n\}.$$

A scaled set of boundary points are now defined

$$\hat{\boldsymbol{p}}_k = \boldsymbol{S}^{-1}\boldsymbol{p}_k, \qquad k = 1,2,\cdots,n$$

and steps c) and d) of the design centering procedure can be carried out as before using the set of points $\{\hat{\boldsymbol{p}}_k\}$. Observe though that the radius of the largest hypersphere, $\hat{r}$, found in step c) can be unscaled to give the "radii," or axis lengths, of a hyperellipsoid

$$r_i = S_i \hat{r}, \qquad i = 1,2,\cdots,n$$

which can be thought of as being inscribed in the unscaled polyhedron defined by the points $\{p_k\}$. Similarly, if $\hat{\boldsymbol{C}}$ denotes the design center of the scaled problem found in step c), the design center of the unscaled problem is

$$\boldsymbol{C} = \boldsymbol{S}\hat{\boldsymbol{C}}.$$

This interpretation of $\boldsymbol{C}$ as the center and the r_i as the radii of a hyperellipsoid comes about by recognizing that in the scaled problem the largest hypersphere of radius r and center $\hat{\boldsymbol{C}}$ is characterized by the locus of points $\hat{\boldsymbol{p}}$ for which

$$\sum_{i=1}^{n} (\hat{p}_i - \hat{C}_i)^2 = r^2 \tag{4-1}$$

or

$$\sum_{i=1}^{n} \left(\frac{p_i - C_i}{S_i r} \right)^2 = 1. \tag{4-2}$$

In terms of the unscaled space (4-2) is the equation of a hyperellipsoid. Note that r_i is *not* the distance between the design center $\boldsymbol{C}$ and the ith face of the polyhedron (as it would have been without scaling) but the distance from the design center to the surface of the hyperellipsoid along the ith axis.

Clearly the results of the above procedure are dependent upon the scaling. It is suggested that once the design centering procedure has converged, the lower and upper bounds on the parameters be redetermined, considering now all the boundary points of ∂R_I^{m+1}, and the scaling adjusted. The design centering procedure should then be repeated to ensure that a good nominal point had been found.[10]

V. Example

To illustrate the design centering algorithm, consider the three transistor NAND gate circuit of Fig. 2. In order that we may graphically illustrate the steps in the procedure, we allow only three designable parameters: the width W_1 of the load device $T1$, the width W_{23} of the

[10]Note that all boundary points previously found as well as the approximating polyhedron can be rescaled and used as the starting approximation here. Thus, in general, little additional work will be required to repeat the design centering procedure.

TABLE I
INITIAL BOUNDARY POINTS

W_1	W_{23}	V_T	A	T_D	V_o
12.85	230.0	1.2	2500	68.5	-.6
8.28	230.0	1.2	2442	110.0	-.395
12.0	231.1	1.2	2500	73.8	-.555
12.0	180.7	1.2	1989	70.0	-.7
12.0	230.0	2.	2490	78.7	-.631
12.0	230.0	1.	2390	72.8	-.543

input devices $T2$ and $T3$ (both $T2$ and $T3$ have the same width), and the threshold voltage, V_T, of the three devices (the same for all three). The lengths of all three devices are held fixed. Therefore we have two geometric parameters and one processing related variable to adjust.

We have three constraints which must be satisfied in order for the circuit to be considered acceptable: the total area A, occupied by the three devices must be less than 2500 mils^2; the delay time, T_D (i.e., the length of time between which the input passes through -3 V and the output falls through -3.25 V) must be less than 110 ns, and the output voltage V_0, be greater than -0.7 V when the gate is on. In addition there are *box* constraints on the parameters

$$5.0 \leqslant W_1 \leqslant 50.0 \quad \text{(microns)}$$
$$50.0 \leqslant W_{23} \leqslant 250.0 \quad \text{(microns)}$$

and

$$1.0 \leqslant V_T \leqslant 2.0 \quad \text{(volts)}.$$

The feasible starting point was $W_1 = 12$, $W_{23} = 230$, and $V_T = 1.2$. For this set of parameter values, $A = 2490$, $T_p = 73.8$, and $V_0 = -0.558$.

The first step in the procedure is to find 6 points on the boundary of the feasible region by searching in the positive and negative coordinate directions. The six points found and the associated constraint values are given in Table I. The constraints which are satisfied exactly are framed. In two cases the box constraints on V_T were the limiting constraints.

A sketch of the initial approximating simplex defined by these boundary points is shown in Fig. 3. The x axis corresponds to $W_1/4.6$, the y axis corresponds to V_T, and the z axis corresponds to $W_{23}/50.3$. The scaling was determined by the initial spread in parameter values as determined by the coordinate searches above. The axes emanate from the "center" of the body. (Note that this figure represents a perspective drawing and does not necessarily display the actual relative size of each face.) The first estimate of the design center, based upon the first approximating simplex, is $W_1 = 11.4$, $W_{23} = 219$, and $V_T = 1.32$. The radii (i.e., axis) of the largest inscribed ellipsoid (i.e., unscaled sphere radius) is 1.1 for W_1, 11.7 for W_{23}, and 0.23 for V_T. In other words, given this design center, acceptable circuits would be obtained for a deviation of 9.6 percent in W_1 if W_{23} and V_T were kept fixed, a 5.3-percent deviation in W_{23} if W_1 and V_T were kept fixed and a 17.4 percent deviation in V_T if W_1 and W_{23} were kept fixed. Note that since we are inscribing a hyperellipsoid in the region of acceptability, and not a hypercube whose sides are twice the maximum deviation, we are not guaranteed of acceptable circuits when two or more parameters deviate maximally from the nominal.

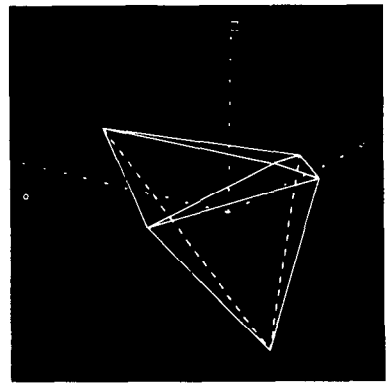

Fig. 3. Initial (scaled) approximating polyhedron to region of acceptability for NAND gate example. Axes eminate from estimate of design center.

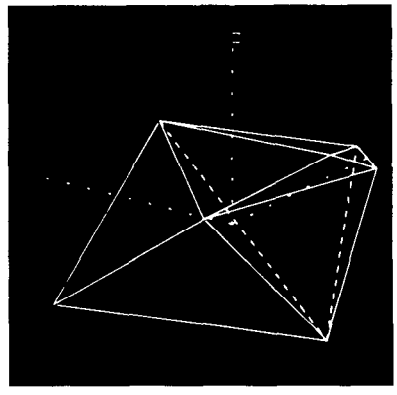

Fig. 4. Resulting polyhedron when largest face of polyhedron in Fig. 3 was broken.

The next step involves searching for a new point on the boundary starting from the center of the largest face and proceeding in an outward normal direction. A new polyhedron is then constructed using this new point and a new estimate of the design center made. Fig. 4 shows the new polyhedral approximation. This procedure is repeated until no increase in the size of the inscribed hypersphere is

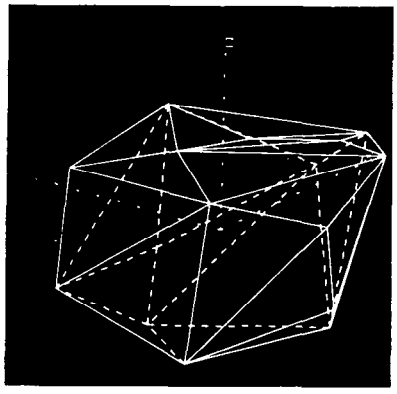

Fig. 5. Final polyhedral approximation.

possible. The final approximating polyhedron is shown in Fig. 5. The final design center was $W_1 = 10.6$, $W_{23} = 209$, and $V_T = 1.47$; and the radius of the inscribed hyperellipsoid was 2.2 for W_1, 24 for W_{23} and 0.47 for V_T. Thus acceptable circuits would be formed for a 21 percent change in W_1 if W_{23} and V_T were kept fixed, an 11.5 percent change in W_{23} if W_1 and V_T were kept fixed, and a 32 percent change in V_T if W_1 and W_{23} were kept fixed. Therefore we see that the design centering procedure increased the allowable tolerances of the designable parameters by a factor of two over the value quoted above for the first computed design center (which was itself an improvement over the initial feasible point).

VI. Relationship of Simplicial Approximation to Existing Operations Research Literature

In two obvious respects, the design centering problem itself, and the simplicial approximation method proposed for solving it, are generally related to various problems and algorithms [11]–[19] in operations research because

a) the problem requires design *optimization* as well as simulation, and
b) the basic approach does, as do most methods in operations research, employ the pervasively useful properties of the simplex method [16].

Because some of this pertinent literature may not be familiar to those who wish to apply the above method and because this literature helps put what we have done in perspective, we briefly review it here.

In its statistical aspect the design centering problem is tangentially related to the so-called chance constrained programming problem [12], [18] which has, in some special cases, been successfully attacked. Two major distinctions exist between previous work in this area, and the problem and methods of this paper, namely,

1) the feasible region, R, is not known in advance but must be implicitly determined, and
2) the constraints implicitly determining R are not only nonlinear, but are functionals of the solutions of a (usually large) set of differential-algebraic equations.

Because of these distinctions, what integrated circuit designers consider a large, realistic problem differs from the conventional definitions of operations research. Whereas in operations research a large scale problem may contain hundreds of thousands of (linear inequality) constraints and variables, in integrated circuit design situations we are typically concerned with only a few design variables and a small number of constraints. In the latter case, it is mainly the size of the underlying system of nonlinear differential-algebraic equations, often of very high (10^3–10^4) order, which describes the circuit that determines the practical realism of the problem.

It is, partially, this dichotomy which persuaded us to attempt what was known to be possible (cf. [11, theorems 32 and 33]) but has, apparently, heretofore been considered too large (cf. [13, p. 35, ¶3]) to consider, that is, the approximation in n-space of the boundary ∂R of an arbitrary convex set R by an inductively constructed polyhedron (cf. [15, ch. 5] for undeveloped ideas which might lead to alternative constructions.) The procedure given above for such "simplicial approximation" contains various features which appear to be novel. However, the operations research literature does contain many possibilities which may speed up the proposed algorithm. These alternate implementations need further investigation. Of particular relevance in this respect is the so-called "decomposition" principle of Dantzig and Wolfe [17]. Much work in this apparently fruitful direction will be necessary before the full economic impact of the proposed method can be properly evaluated. However, we emphasize again that the major portion of the computational cost associated with the above procedure is due to the circuit simulations required to find new boundary points and not in the solution of the associated linear programs.

VII. Conclusions

To summarize, we have described a method, called the simplicial approximation method, which locates with repeated line searches, an increasingly dense set of points on the boundary ∂R of the feasible region R. These points are used to define a polyhedron of interior bounding hyperplanes for the region R. At each stage of approximation, the design center is (after appropriate scaling) the center of the maximal hypersphere inscribed in the polyhedron. The approximation is refined by searching for the intersection, with the true boundary ∂R, of the outward normal line emanating from the "midpoint" of the "largest" of the faces of the polyhedron which are tangent to the maximal inscribed hypersphere.

We have constructed, and verified by experiment, an algorithm based on the process described above. We have also been able to show that the radius of the maximal hypersphere is monotone increasing as the approximation is refined. This has enabled us to compute a monotone increasing lower bound on the yield, i.e.,

$$Y^k < Y^{k+1} < Y$$

where Y is the true maximum available yield and Y^k is the available yield for the kth approximation to ∂R.

It has been suggested by Wolfe [8] that an exterior set of supporting hyperplanes could be constructed with a procedure essentially dual to the procedure presented in Section III. In the corresponding "exterior" approximating polyhedron, ∂R_E^k, the location of a new boundary point on ∂R would be followed by a gradient evaluation, yielding a supporting hyperplane containing the new point. All extreme points of ∂R_E^k which are outside the new supporting hyperplane would then be deleted from ∂R_E^k, and the convex hull of the new set of extreme points would constitute ∂R_E^{k+1}. Ideas related to "exterior" simplicial approximation have been discussed in [13]–[15]. The cutting plane method of Kelley, [14], is quite similar in method, although different in purpose.

Also significant in the list of future research possibilities are the establishment of conditions for, and rates of, convergence and of procedures for handling nonconvex, locally convex, and/or quasi-convex cases.

In the meantime, it is important to note that the described implementation is based on a straightforward interfacing of code for the new algorithm with standard programs for (small scale) LP problems and for network simulation. More specifically, the method of "interior" simplicial approximation (unlike the nonlinear programming method of [1]) requires *no* gradient evaluations. Thus any readily available conventional network simulator, e.g., SPICE, [9], or ASTAP-II, [10], can be employed to provide prospective users with immediate access to design centering via interior simplicial approximation.

Appendix A–The One-Dimensional Search Strategy

Consider the n_c constraint functions $\Phi_i(\boldsymbol{p})$, $i = 1,2,\cdots,n_c$ of the n-parameter vector $\boldsymbol{p}$. For any feasible point $\boldsymbol{p}^0$

$$\Phi_i(\boldsymbol{p}^0) \leqslant 0, \qquad i = 1,2,\cdots,n_c.$$

A point on the boundary of the feasible region, $\boldsymbol{p}^*$, is defined as a point for which

$$\Phi_j(\boldsymbol{p}^*) = 0$$

and

$$\Phi_i(\boldsymbol{p}^*) \leqslant 0, \qquad i = 1,2,\cdots,n_c; \qquad i \neq j$$

for some $j \in [1,m]$. During the design centering procedure we wish to find a point on the boundary of the feasible region starting from some feasible point $\boldsymbol{p}^0$ and proceeding in a given direction $\boldsymbol{s}$. More specifically, we wish to determine the step size α for which

$$\Phi_j(\boldsymbol{p}^0 + \alpha \boldsymbol{s}) = 0$$

and

$$\Phi_i(\boldsymbol{p}^0 + \alpha \boldsymbol{s}) \leqslant 0, \qquad i = 1,2,\cdots,n_c; \qquad i \neq j$$

for some j. The one-dimensional search procedure which has been implemented is based on the secant method and proceeds as follows.

1) Evaluate $\Phi_i(\boldsymbol{p}^0)$, $i = 1,2,\cdots,m$. Define $C_i^0 = C_i^l = \Phi_i$ and $\alpha^0 = \alpha^l = 0$.

2) Determine the smallest α such that the point $(\boldsymbol{p}^0 + \alpha \boldsymbol{s})$ is on one of the box constraints and evaluate $\Phi_i(\boldsymbol{p}^0 + \alpha \boldsymbol{s})$. If $\Phi_i(\boldsymbol{p}^0 + \alpha \boldsymbol{s}) \leqslant 0$ for all i then the box constraint is active and the point $\boldsymbol{p}^0 + \alpha \boldsymbol{s}$ is on the boundary, otherwise define $\alpha^N = \alpha^u = \alpha$ and $C_i^N = C_i^u = \Phi_i$ for all $\Phi_i > 0$. Let I denote the set of indices $\{i\}$ for which the constraints are violated.

3) If $(\alpha^u - \alpha^l)s_i > \Delta p_k$ (where Δp_k is the minimum allowable change in p_k) for any $k = 1,2,\cdots,n$, then proceed, if not then stop and take current value of $\boldsymbol{p}^0 + \alpha \boldsymbol{s}$ as boundary point.

4) Calculate the value of α given by

$$\alpha \rightarrow \min_{i \in I} \left\{ \alpha^N - C_i^N (\alpha^N - \alpha^0)/(C_i^N - C_i^0) \right\}$$

If $\alpha < \alpha^l$ then set α to α^l, or if $\alpha > \alpha^u$ then set α to α^u. Then set α^0 to α^N, C_i^0 to C_i^N, $i \in I$, and α^N to α.

5) Evaluate $\Phi_i(\boldsymbol{p}^0 + \alpha^N \boldsymbol{s})$, $i \in I$, and set C_i^N to Φ_i, $i \in I$. If $\Phi_i < 0$ for all $i \in I$ set α^l to α^N and C_i^l to Φ_i and repeat from step 3). If there is at least one $i \in I$ for which $\Phi_i > 0$, than set α^u to α^N and $C_i^u < \Phi_i$ for all $i \in I$ and repeat from step 3).

Observe that step 4) is essentially the secant method for determining α. Moreover, at every step in the procedure there is an upper and lower bound on the step size α. When the interval, $[\alpha^l, \alpha^u]$ gets small enough, the procedure terminates.

Acknowledgment

We wish to thank R. Brayton, H. Crowder, F. Gustavson, A. Wilson, and P. Wolfe for many helpful discussions. A special debt is owed to G. Geortzel for sharing outstanding unpublished work in related areas.

References

[1] J. W. Bandler, P. C. Liu, and H. Tromp, "Practical design centering, tolerancing and tuning," in *Proceedings of the 1975 IEEE International Symposium on Circuits and Systems*, Apr. 1975, pp. 206–209.
[2] J. F. Pinel and K. A. Roberts, "Tolerance assignment in linear networks using nonlinear programming," *IEEE Trans. on Circuit Theory*, vol. CT-19, pp. 475–479, Sept. 1972.
[3] A. R. Thorbjornsen and S. W. Director, "Computer-aided tolerance assignment for linear circuits with correlated elements," *IEEE Trans. on Circuit Theory*, vol. CT-20, pp. 518–524, Sept. 1973.
[4] B. J. Karafin, "An efficient computational procedure for estimating circuit yield," Ph.D. dissertation, University of Penn., 1974.
[5] N. J. Elias, "New statistical methods for assigning device tolerances," in *Proceedings of the 1975 IEEE Symposium on Circuits and Systems*, pp. 329–332, Apr. 1975.
[6] E. M. Butler, "Realistic design using large change sensitivities and performance contours," *IEEE Trans. on Circuit Theory*, vol. CT-18, pp. 58–59, Jan. 1971.
[7] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *Computer Journal*, vol. 6, pp. 163–168, 1963.
[8] P. Wolfe, private communication.
[9] L. W. Nagel, "SPICE 2: A computer program to simulate semiconductor circuits," Ph.D. dissertation, Univ. of Calif., May 1975.
[10] *Advanced Statistical Circuit Analysis Program (ASTAP) Program Reference Manual*, SH20-1118-0, IBM Corporation.
[11] H. Eggleston, *Convexity*. London, England: 1958, Cambridge University Press.
[12] A. Charnes and W. Cooper, "Chance constrained programming," *Management Science*, vol. 6 pp. 73–80, 1959.
[13] A. M. Geoffrion, Ed., *Perspectives on Optimization*. Reading, MA: Addison-Wesley, 1972.

[14] J. E. Kelley, "The cutting plane method for solving convex programs," *J. Soc. Indus. Appl. Math*, vol. 8, pp. 703–712, 1960.
[15] B. Grunbaum, *Convex Polytopes*. New York: Wiley-Interscience, 1967.
[16] G. Dantzig, *Linear Programming and Extensions*. Princeton, NJ: Princeton University Press, 1963.
[17] G. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, no. 1, pp. 101–111, 1960.
[18] L. Cooper and D. Steinberg, *Methods and Applications of Linear Programming*. Philadelphia, PA: W. B. Saunders, 1974.
[19] D. Luenberger, *Introduction to Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1973.
[20] S. W. Director and G. D. Hachtel, "Yield estimation using simplicial approximation," in *Proceedings of the 1977 IEEE Symposium on Circuits and Systems*.

✚

Stephen W. Director (S'65–M'69–SM'75) received the B. S. degree in engineering science from the State University of New York at Stony Brook, Stony Brook, NY, in 1965, and the M. S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1967 and 1968, respectively.

In 1968 he joined the University of Florida, Gainesville, where he is currently a Professor in the Department of Electrical Engineering. From September 1974 to August 1975 he was a Visiting Scientist in the Mathematical Sciences Department at IBM's T. J. Watson Research Center, Yorktown Heights, NY. His research interests are in the area of computer-aided network design and numerical methods and he has been a Consultant to industry in this area. He has served as a member of the Administrative Committee of the IEEE Circuits and Systems Society, as Chairman of the CAS Technical Committee on Computer-Aided Network Design (CANDE), as an Associate Editor of the *IEEE Transactions on Circuits and Systems*, and as a Guest Editor of this Transactions for a Special Issue on Computer-Aided Design. In addition to writing numerous technical articles, he has coauthored the text *Introduction to System Theory* (McGraw-Hill), edited the volume *Computer-Aided Circuit Design: Simulation and Optimization* (Dowden, Hutchinson, and Ross), and authored the text *Circuit Theory–A Computational Approach* (Wiley).

Dr. Director is a member of Sigma Xi and Eta Kappa Nu. In 1970 he received the Best Paper Award from the IEEE Circuits and Systems Society and in 1975 he received the Frederick Emmons Terman Award as the outstanding young Electrical Engineering Educator from the American Society of Engineering Education.

✚

Gary D. Hachtel (S'62–M'65) received the B.Sc. degree from the California Institute of Technology, Pasadena, CA, in 1959 and the Ph.D. degree from the University of California, Berkeley, in 1965, both in electrical engineering.

From 1961 to 1964 he worked in the Berkeley Integrated Circuits Laboratory. In 1965 he joined the IBM T. J. Watson Research Center, Yorktown Heights, NY, as Research Staff Member in the departments of Applied Research and Mathematical Sciences. He has taught at University of California, Berkeley, NYU, New York, NY, and UCLA, Los Angeles, CA, where he was Regents Lecturer for 1974. In 1975 he was with IBM Office Products Division, Boulder, CO. He was Associate Editor of the *International Journal for Numerical Methods in Engineering* and is now Associate Editor of the IEEE TRANSACTIONS IN CIRCUITS AND SYSTEMS. He is presently with the Department of Mathematics, University of Denver, Denver, CO 80210. His current research is on simplicial approximation, sparse matrices, finite elements, and solar cell modeling.

Dr. Hachtel received an IBM Outstanding Contribution Award for work on analysis and design of integrated circuit transistors via numerical solution of partial differential equations in 1968, and an Outstanding Invention Award for the Tableau Approach Design System in 1971. In 1972 he received the Best Paper Award from the IEEE Circuits and Systems Society.

A 2-D FIR Filter Structure Derived from the Chebyshev Recursion

JAMES H. McCLELLAN, MEMBER, IEEE, AND DAVID S. K. CHAN

***Abstract*—A new structure is proposed for the implementation of 2-D FIR digital filters designed by transformations of 1-D filters. This structure requires a minimum number of multiplications and uses directly as coefficients the impulse response samples of the 1-D prototype filter. The roundoff noise of the new structure is analyzed in detail and is shown to be superior to that of previous implementations for all examples studied.**

Manuscript received September 10, 1976; revised March 9, 1977. This work was supported in part by the National Science Foundation under Grant ENG71-02319-A02 and in part by the Joint Services Electronics Program under Grant DAAB07-75-C-1346. Mr. Chan's work also was supported by R.L.E. Industrial Fellowship.

The authors are with the Research Laboratory of Electronics, Massachusetts Laboratory of Technology, Cambridge, MA 02139.

I. INTRODUCTION

RECENT WORK by Mecklenbräuker and Mersereau [1] has uncovered a very efficient implementation for two-dimensional (2-D) FIR digital filters designed via a transformation technique first discussed in [2] and extended in [3]. The design technique starts with a 1-D zero phase filter and produces the 2-D filter by making a change of variables. The new implementation actually realizes this two stage process and thereby saves many multiplies in the final structure. Three alternative forms of the implementation were proposed: the direct trans-