Efficient Analog Circuit Synthesis with simultaneous Yield and Robustness Optimization

Geert Debyser, Georges Gielen*
Katholieke Universiteit Leuven, Belgium
[geert.debyser, georges.gielen]@esat.kuleuven.ac.be

Abstract

This paper presents an efficient statistical design methodology that allows simultaneous sizing for performance and optimization for yield and robustness of analog circuits.

The starting point of this methodology is a declarative analytical description of the circuit. An equation manipulation program based on constraint satisfaction converts this declarative model into an efficient design plan for optimization based sizing. The efficiency is due to the use of an operating point driven DC formulation, so that the design plan avoids the calculation of simultaneous sets of nonlinear equations. From the same declarative analytical description also a direct symbolic yield estimation plan is generated. The parametric yield is estimated by propagating the spread of the technological variables through the analytical model towards the performance variables of the circuit. The design plan and the yield estimation plan are then combined together in the inner loop of a global optimization routine. The strength of this methodology lies in the low CPU times needed to perform yield estimation compared to the hours of simulation batches with Monte Carlo simulations, while the accuracy is comparable.

I. Introduction

Designing an analog circuit is one thing, producing it is another. Real-life technology parameter variations make the circuits fail for some or all of the specifications if no precautions are taken. The ratio of the number of successful circuits over the number of produced circuits is the total yield. The total yield consists of yield due to production faults and yield due to soft faults. In this paper we concentrate on the yield based on the soft faults, generated by the technology parameter variations: the parametric yield.

The hard way to make the design more robust is to run multiple batch jobs of Monte Carlo simulations in the inner loop of an optimization routine. Because a Monte Carlo simulation consists of typically hundreds of SPICE simulations, the computational effort is so large that only a post-design yield optimization is considered. This is the known design centering method. Starting from the nominal design a local optimization tries to push the performances away from the specification boundaries in order to make the circuit more robust against technology parameter variations [1].

The same is true for analog circuit sizing or synthesis programs [2]. Most of them concentrate on the nominal design only, without considering the process parameter or operating condition variations. Performing hours of simulated annealing for just a nominal design is only half of the game. Only in [3] a first approach towards analog synthesis for manufacturability was presented that combines nominal circuit optimization with variation analysis in an outer optimization loop. The results of the variation analysis are used to change the cost function of the inner circuit optimization by penalizing design solutions that do not meet the specifications over the entire operating range. The approach however is extremely time consuming.

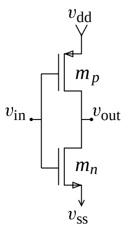
Therefore, in this paper, an alternative approach is presented towards simultaneous circuit sizing for performance and yield/robustness optimization. The approach results in a drastic reduction of the required CPU time, without sacrifying too much accuracy. It is based on the use of symbolic techniques to capture the behavior of a circuit in a declarative model [4]. Constraint statisfaction techniques implemented in the tool DONALD [5] are then used to derive an efficient sizing plan as well as an efficient yield estimation plan. Both plans are then simultaneously evaluated in the inner loop of a global optimization routine. The outcome of the optimization is a circuit design point that fulfills all specifications and that at the same time has pushed away the performances from the specification boundaries under the influence of the yield and/or Cpk measure.

The paper is organized as follows. Section 2 explains the construction of the sizing plan by the DONALD tool starting from the declarative behavioral model of the circuit. Section 3 describes how we used symbolic techniques to construct the yield/robustness estimation plan. Section 4 explains the yield optimization strategy. Section 5 shows experimental results. Section 6 draws some conclusions.

II. Sizing plan

A declarative analytical model of an analog circuit is mainly obtained by 2 sources: by applying symbolic methods on the circuit's graph topology and by hand. A simple illustrative example of the outcome for an inverter is given in Fig. 1, where 7 equations describe the DC operating point.

The 7 equations are presented in a bipartite graph form (see



$$\text{eq\_idsp}: I_{\mathrm{dsp}} = \frac{KP_p}{2}\frac{W_p}{L_p}(V_{\mathrm{gsp}} - V_{\mathrm{Tp}})^2(1 + \lambda_p V_{\mathrm{dsp}})$$

$$\text{eq\_idsn}: I_{\mathrm{dsn}} = \frac{KP_n}{2}\frac{W_n}{L_n}(V_{\mathrm{gsn}} - V_{\mathrm{Tn}})^2(1 + \lambda_n V_{\mathrm{dsn}})$$

$$\text{eq\_vgsp}: V_{\mathrm{gsp}} = V_{\mathrm{in}} - V_{\mathrm{dd}}$$

$$\text{eq\_vgsn}: V_{\mathrm{gsn}} = V_{\mathrm{in}} - V_{\mathrm{ss}}$$

$$\text{eq\_vdsp}: V_{\mathrm{dsp}} = V_{\mathrm{out}} - V_{\mathrm{dd}}$$

$$\text{eq\_vdsn}: V_{\mathrm{dsn}} = V_{\mathrm{out}} - V_{\mathrm{ss}}$$

$$\text{KCL}: I_{\mathrm{dsn}} + I_{\mathrm{dsp}} = 0$$

Fig. 1. Inverter and its simplified behavioral DC model.

* research associate of the Belgian Fund of Scientific Research

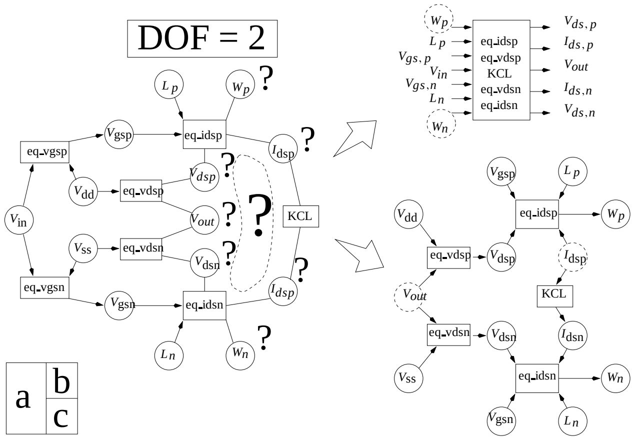


Fig. 2. Depending on the input variables, DONALD has to solve a 5x5 cluster or 2 simple equations.

Fig. 2a), where the square boxes represent the equations and the circles represent the variables. We assume that the user can give a value for V_{dd}, V_{ss}, V_{in} (typically half-way V_{dd} and V_{ss}). L_n and L_p can be chosen to be minimal size. The equation manipulation tool DONALD propagates these inputs throught the graph, which is then partially directed. The traditional SPICE-like inputs (choosing W_p and W_n, see Fig. 2b), lets DONALD simultaneously solve a set of 5 (possibly nonlinear) equations with a nonlinear root solver. In this case convergence cannot be guaranteed and the solution also depends on the starting point. The operating point driven DC formulation [6], however, constrains node voltages and branch currents (choosing V_{nout} and I_{dsp}, as in Fig. 2c). The DONALD program only has to solve 2 one-dimensional nonlinear equations to obtain W_n and W_p, which can be executed much faster. The computational effort is drastically reduced since, for a circuit with N nodes and M MOS-transistors, solving M one-dimensional nonlinear equations is of the order $O(M\ \text{Root}(1))$ (where $\text{Root}(x)$ is the effort to solve a cluster of size x). A simulator needs an effort of the order $O(\text{Root}(N))$, with a risk of divergence. This approach has already been used for optimization based nominal sizing [6], [7]. In this paper we extend this to yield optimization.

III. Yield/robustness estimation plan

Our approach replaces the Monte Carlo simulations with a direct yield estimation technique. We start with a statistical transistor model, which gives us a reduced set of quasi-independent technology parameters $\boldsymbol{\theta}$. Then we calculate the nominal design point and then we calculate the variances of all performance parameters $\mathbf{y}$ with respect to the reduced set of technology parameters $\boldsymbol{\theta}$. Using these variances, we construct an efficient representation of "yield" based on C_p/C_{pk} indices. Both the sizing plan and the yield estimation plan are then placed in the inner loop of the optimization routine as explained in section 4. The flow diagram in Fig. 3 with the setup for the yield estimation plan is explained in the following subsections (x stands for designable parameters and e stands for simulator variables).

A. *Statistical transistor model*

The default transistor models have to be replaced by statistical models in order to take the correlations of the technology parameters into account. This is necessary to estimate the yield in a statistically correct way. The statistical model describes all technological variables as a function of only 7 quasi-uncorrelated input variables, which are TOX, NSUB$_n$, NSUB$_p$, CJ$_n$, CJ$_p$, LD$_n$ and RSH$_n$. A method for deriving such a statistical transistor model has been discussed in [8]. The Monte Carlo routine perturbes only these 7 parameters and extracts performances from the SPICE output to construct the yield estimation figures based on a pass/fail mechanism. This way of calculating the yield is very costly: a large number (e.g. 300) of SPICE simulations have to be executed. Our symbolic yield estimator also starts from the same 7 parameters $\boldsymbol{\theta}$.

B. *Direct yield estimation*

As the pass/fail measure used in Monte Carlo simulation is very rough and gives little or no information about which performance does not meet its specification, another quality measure is preferred. The Taguchi quality measure gives a much better idea of the quality of a circuit, because it takes absolute variability and bias to the target spec into account:

$$M_{\text{TAG}} = var\{\mathbf{y}(x)\} + (\bar{\mathbf{y}} - \mathbf{Spec}^T)^2 \qquad (1)$$

To ease the calculation of M_{TAG}, two capability indices were introduced in [9]: the capability *potential* index C_p:

$$C_p = \frac{\text{Spec}^U - \text{Spec}^L}{6\sigma_y} \qquad (2)$$

and the capability *performance* index C_{pk}:

$$C_{pk} = \min\{\frac{\text{Spec}^U - \bar{y}}{3\sigma_y}, \frac{\bar{y} - \text{Spec}^L}{3\sigma_y}\} \qquad (3)$$

The first index represents the variability, the second index the bias. As can be noticed from formulae (2,3), these indices strongly depend on the variances of the performances. The variances of the 7 technological parameters are propagated through the computational path by means of sensitivities. In case θ_1, θ_2, ... $\theta_{n=7}$ are not correlated, the variability of the performances can be written as follows:

$$\sigma_{y_i} = \sqrt{Var(y_i)} \approx \sqrt{\sum_{j=1}^{n=7} (S_{\theta_j}^{y_i})^2 \sigma_{\theta_j}^2} \qquad (4)$$

(assuming that $\sigma_{\theta_j}/\bar{\theta}_j$ is sufficiently small).

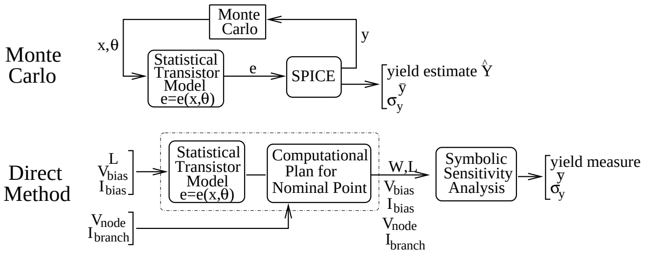

Monte Carlo
Monte Carlo
x,θ
y
Statistical Transistor Model e=e(x,θ)
e
SPICE
yield estimate Ŷ
y
σ_y
Direct Method
L
V_bias
I_bias
Statistical Transistor Model e=e(x,θ)
Computational Plan for Nominal Point
W,L
V_bias
I_bias
V_node
I_branch
Symbolic Sensitivity Analysis
yield measure
y
σ_y
V_node
I_branch

Fig. 3. Yield estimation methods. Monte Carlo and SPICE versus direct method.

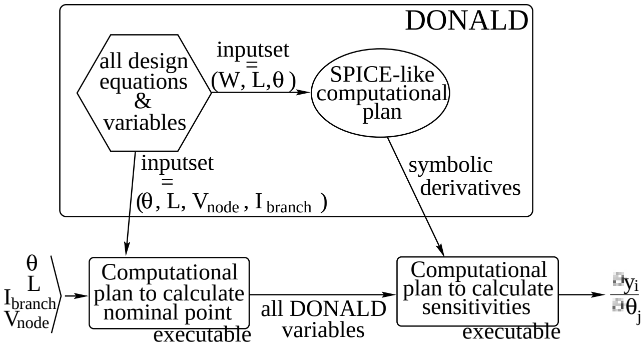

DONALD
all design equations & variables
inputset = (W, L,θ)
SPICE-like computational plan
inputset = (θ, L, V_node , I_branch)
symbolic derivatives
θ
L
I_branch
V_node
Computational plan to calculate nominal point
executable
all DONALD variables
Computational plan to calculate sensitivities
executable
∂y_i / ∂θ_j

Fig. 4. Creation and use of direct yield estimation model.

An extension has been made to DONALD (see Fig. 4), so that it also builds a computational plan to calculate all sensitivities from the output variables w.r.t. the input variables. The sensitivities have to be known for *every* operating point x. Our Jacobians are in symbolic form, so the Jacobian updating for every x is relatively cheap.

This computational path is derived from the same set of equations used to determine the nominal point, but W and L must be chosen as input variables. This time however, we do *not* solve the set of nonlinear equations, but we reuse the values previously obtained in the nominal point calculation (see Fig. 4). Symbolic derivatives from all equations are automatically derived. Out of this a computational path results, which is a chain of one-dimensional and more-dimensional subsystems of equations. The local sensitivities are calculated using Jacobians in symbolic form. The global sensitivity matrix S^y_θ with elements $S^{y_i}_{\theta_j} = \partial y_i / \partial \theta_j$, is calculated by applying the chain rule according to the computational plan:

$$S^y_\theta = S^y_{z_n} S^{z_n}_{z_{n-1}} \cdots S^{z_1}_\theta \tag{5}$$

where z_k are the internal variables along the calculation path.

The calculation of the local sensitivities is complicated but straightforward. First, at each square subsystem with equations $f_1, \ldots, f_n$, a distinction is made between those variables of the subsystem that have been solved by the subsystem, and those variables that are solved by previous subsystems. The former are called the *output variables* $\mathbf{z^{out}} = (\mathbf{z^{out}_1}, \ldots, \mathbf{z^{out}_q})^\mathbf{T}$ of the subsystem, the others are called the *input variables* $\mathbf{z^{in}} = (z^{in}_1, \ldots, z^{in}_p)^T$. For each square subsystem the change $\delta\mathbf{z^{out}}$ is then calculated of the output variables with respect to a unit change $\delta\mathbf{z^{in}}$ of the input variables. The local sensitivity $S^{z^{out}_i}_{z^{in}_j} = \delta z^{out}_i$ is solved from the following system of linear equations:

$$\nabla\mathbf{F}^{out}\delta\mathbf{z^{out}} = -\nabla\mathbf{F^{in}}\delta\mathbf{z^{in}} \tag{6}$$

where $\delta\mathbf{z^{in}} = \mathbf{1}$ and

$$\nabla\mathbf{F}^{out} = \begin{bmatrix} \frac{\partial f_1}{\partial z^{out}_1} & \cdots & \frac{\partial f_1}{\partial z^{out}_q} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial z^{out}_1} & \cdots & \frac{\partial f_n}{\partial z^{out}_q} \end{bmatrix} \tag{7}$$

and

$$\nabla\mathbf{F}^{in} = \begin{bmatrix} \frac{\partial f_1}{\partial z^{in}_1} & \cdots & \frac{\partial f_1}{\partial z^{in}_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial z^{in}_1} & \cdots & \frac{\partial f_n}{\partial z^{in}_p} \end{bmatrix} \tag{8}$$

where all partial derivatives are in symbolic form. By multiplying local sensitivity values along the computational path between two variables r and s, a *global sensitivity* value $S^r_s = \partial r / \partial s$ can be calculated.

IV. Yield Optimization

We can now perform a nominal point analysis and to calculate all sensitivities and hence the C_p/C_{pk} values. The step towards simultaneous nominal and statistical optimization is small by combining the approach of Fig. 4 in a optimization loop. The optimization is done stepwise, as can be seen in Fig. 5, where the solution is gradually narrowed by enforcing more and more constraints on the design [4].

The initial solvability space is narrowed by adding the following sequence of constraints to the design plan. The manufacturability space Ω_M is the set of circuits that can be produced within a given technology. The operationality space Ω_O contains the circuits whose transistors are in the correct operating region. The functionality space Ω_F contains the designs that fulfill the design requirements (such as first order behavior for an OTA). The applicability space Ω_A contains the circuits that fulfill all specifications. The robustness space Ω_R contains the designs that take all performance variations into account by retreating from the design space boundaries. Once all specifications fulfilled, there is still room left for trade-off between different performance parameters.

With each of these spaces inequality functions can be associated, which in their turn correspond to penalty functions h_j:

$$h_j = \begin{cases} \leq 0 & \text{if inequality is fulfilled} \\ > 0 & \text{else} \end{cases}$$

The subdivision of the solvability region Ω_S is reached by adding a weighted sum of penalties to the total cost function of the optimization routine. To assure a *sequential design space pruning process*, the weights W are chosen as follows: the steering function for robustness gets the reference weight W_R. The penalties of the applicability space Ω_A get weights of $10W_R$. The weights belonging to the other spaces are $10^2 W_R$ for Ω_F,

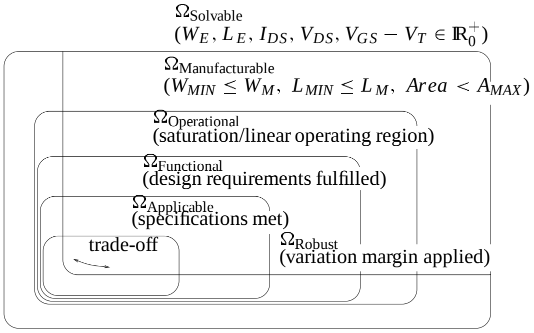

Ω_{Solvable}
$(W_E, L_E, I_{DS}, V_{DS}, V_{GS} - V_T \in \mathbb{R}^+_0)$
$\Omega_{\text{Manufacturable}}$
$(W_{MIN} \leq W_M,\ L_{MIN} \leq L_M,\ Area < A_{MAX})$
$\Omega_{\text{Operational}}$
(saturation/linear operating region)
$\Omega_{\text{Functional}}$
(design requirements fulfilled)
$\Omega_{\text{Applicable}}$
(specifications met)
trade-off
Ω_{Robust}
(variation margin applied)

Fig. 5. Spaces encountered in optimization based sizing of analog circuits.

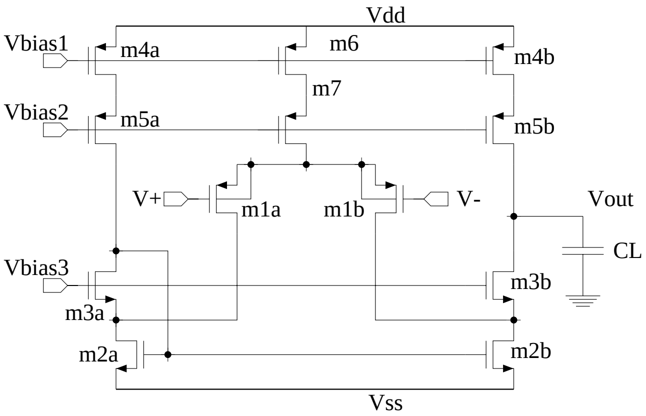


Fig. 6. Schematic of current buffer OTA.

$10^3 W_R$ for Ω_O and, $10^4 W_R$ for Ω_M. This way the optimizer tackles the sizing problem in a stepwise way. First it assures the manufacturability, then the operationality, the functionality and the applicability. In the final stage the yield estimation model is activated and the cost belonging to the not-yet optimal yield and variability is added to the global cost function of the optimization routine. We chose to leave out the yield estimation till the optimizer was in the applicability space Ω_A, because a yield estimation in an earlier stage of the optimization is computational expensive and doesn't change the problem that much.

The quality measure for robustness used in the cost function is:

$$\Phi_i^{\pm}(x) = C_{p,i} \pm \lambda \frac{\frac{\text{Spec}_i^U + \text{Spec}_i^L}{2} - \overline{y}_i}{3\sigma_{y_i}} \tag{9}$$

$$\Phi_i(x) = \min\{\Phi_i^+(x), \Phi_i^-(x)\} \tag{10}$$

The weight factor λ acts as a penalty term on the bias, with respect to target Spec_i^T. If $\lambda = 0$ then $\Phi_i = C_{p,i}$, if $\lambda = 1$ then $\Phi_i = C_{pk,i}$. A value of $\lambda = 0.8$ has been chosen experimentally. The optimization problem to be solved is then

$$\Phi(x) = \sum_{\substack{x \in R_x \\ j = \Omega_M, \Omega_O, \Omega_F, \Omega_A}} W_j h_j(x) + \max_{x \in R_x} \min_i \{\Phi_i(x)\} + \alpha f(\text{trade-off}) \tag{11}$$

where R_x is a hyperbox of constraints.

V. Experimental Results

The circuit is a CMOS current buffer OTA (see Fig. 6) in a 0.7u CMOS process. An arbitrary starting point (formulated as a DC operating point) has been chosen for the optimizer. Then a simultaneous sizing and optimization is performed (in 2h15' time) with both the sizing plan and the yield/robustness estimation plan (see Fig. 4) in the inner loop of of the annealing routine. In the resulting optimized point a Monte Carlo is run for verification.

The yield/robustness estimation plan is extremely fast (10s) compared to Monte Carlo (300 samples: 2h20' on a Sparc I). Such an optimization with Monte Carlo in the inner loop would take approximately 20 days! The first two columns of Table 1 contain the mean and the variance of the performance variables, which come out of the yield/robustness estimation model. The last two columns in Table 1 give the estimate mean and variance of the performances from the Monte Carlo run.

TABLE I

FINAL RESULTS OF YIELD OPTIMIZATION.

Specs	After optimization			
	Yield Model		Monte Carlo	
	$\overline{y}$	σ_y	$\overline{y}$	σ_y
GBW>100MHz	165	23.1	172	21.4
A_{v0}>60dB	78.0	1.68	78.1	1.74
PM>60°	60.2	0.2	63.6	0.4
OR>3.0V	3.1	0.05	3.27	0.058
V_{off}<5mV	4.8	0.14	3.5	0.7
I_{tot}<3.0mA	2.6	0.4	2.6	0.56

VI. Conclusions

A new statistical design method using symbolic techniques has been presented. A direct yield estimation model has automatically been derived from the whole set of symbolic design equations by constructing a computational plan to calculate symbolically the sensitivities of the performances w.r.t. the technology parameters. By propagating the variances of 7 quasi-independent technology parameters the quality measures C_p and C_{pk} for the performances were obtained. This yield model was directly used in the inner loop of a yield optimization routine to perform simultaneous nominal and yield/robustness optimization. The experimental results show that the model is accurate enough to steer the optimization in the direction of a "better" design in a much faster way than using a simulator. Research is still to be continued to improve the yield model.

Acknowledgments

The authors acknowledge the financial support for parts of this work from the Flemish IWT under the MEDEA SADE project.

References

[1] K. Antreich, H. Graeb, C. Wieser, "Circuit analysis and optimization driven by worst-case distances," IEEE Transactions on Computer-Aided Design, Vol. 13, No. 1, pp. 57-71, January 1994.

[2] C. Borchers, L. Hedrich, E. Barke, "Equation-based behavioral model generation for nonlinear circuits," proc. IEEE/ACM DAC, pp. 236-239, 1996.

[3] T. Mukherjee, L.R. Carley, "Rapid yield estimation as a computer aid for analog circuit design," IEEE Journal of Solid-State Circuits, Vol. 26, No. 3, pp. 291-199, March 1991.

[4] F. Leyn, W. Daems, G. Gielen, W. Sansen, "A behavioral signal path modeling methodology for qualitative insight in and efficient sizing of CMOS opamps," Proc. IEEE ICCAD, pp. 374-381, Nov. 1997.

[5] K. Swings, W. Sansen, G. Gielen, "DONALD : An intelligent analog IC design system based on manipulation of design equations," Proc. IEEE CICC, pp. 8.6.1-4, May 1990.

[6] F. Leyn, G. Gielen, W. Sansen, "An efficient DC root solving algorithm with guaranteed convergence for analog integrated CMOS circuits," submitted to ICCAD 1998.

[7] G. Gielen, G. Debyser, et al., "Use of symbolic analysis in analog circuit synthesis," Proc. IEEE ISCAS, pp. 2205-2208, May 1995.

[8] J. Chen, M. Styblinski, "A systematic approach of statistical modeling and its applications to CMOS circuits," Proc. IEEE ISCAS, pp. 1805-1808, May 1993.

[9] M. Styblinski, S. Aftab, "IC variability minimization using a new Cp and Cpk based variability/performance measure," Proc. IEEE ISCAS, May-June 1994.