Integrated Tools for Device Optimization

M. Rudan, M. C. Vecchi and *Zs. M. Kovács*

Dipartimento di Elettronica, Informatica e Sistemistica
Università di Bologna, viale Risorgimento 2, 40136 Bologna, Italy

1. Introduction. The design of semiconductor devices is a complex task that has a strategic role in Integrated Circuits manufacturing. First of all, the performance of systems realized using a specific technology is strongly related to the design of the technological process, which dictates the basic performances of the circuits. On the other hand, the soaring cost of manufacturing equipment gives a strong push to a detailed optimization of the process in order to exploit the available capabilities. The CAD tools available today are playing a key role in order to reduce the cost associated with design optimization of semiconductor devices. In fact, process and device simulators have found a wide acceptance among designers; these tools are providing reliable results that allow them to cut the cost associated with the laboratory experiments required to investigate different tradeoffs. Even though the software environment to support device design has had a steady improvement in the last few years, the tuning of the process parameters is usually left to the process designer who, on his own, has to choose the tradeoff among many objectives in competition with one another. From this point of view, the availability of an integrated synthesis system that supports an optimized design of semiconductor devices is likely to be an useful tool for a timely and high-quality design development.

Based on these premises, we have recently started a research activity which aims at developing an optimization tool based on our two-dimensional device-analysis code **HFIELDS**. The activity is carried out in the frame of an EEC-sponsored Project, **STORM**, in which a number of European groups, both industrial and academic, participate. The main objective of the optimization part (Workpackage) of the Project is the development of an automatic optimizer for process and device design, whose goal is computing optimal values for the process parameters according to given targets and constraints. The implementation and the results achieved as for the device optimization are outlined in the following paragraphs.

2. The Problem. The mathematical model that describes the manufacturing process of semiconductor devices can be formulated in terms of two sets of equations: the first one is associated with the modelling of the technological process, while the other describes the electrical behaviour of semiconductor devices built using these process steps. It is worth mentioning that the final goal of the Project Workpackage referred here is pipelining a process and a device simulator, which will be considered as a single entity in the design procedure. The input of the pipeline is a set of process parameters to be tuned, and the output is a device performance to be achieved. The whole pipeline is controlled by a supervising program that iterates the action of the pipeline until the optimum set of parameters is determined. Since our specific activity has dealt until now with the device simulator, i.e., the second block of the pipeline, the process simulator will be considered here as a black box able to produce the device geometry and dopant distribution. Using the output provided by the black box, the electrical performance of the device is obtained by solving a set of three coupled non-linear PDE's, namely the Poisson equation and the carrier-continuity equations.

The numerical expressions derived by the discretization of the PDE's describing the device behaviour require the solution of systems of non-linear equations in several thousands of unknowns. This implies a high computational cost for each simulation. For the determination of the optimum parameters (which can, in itself, be formalized as a problem of non-linear op-

† M. C. Vecchi and Zs. M. Kovács avail themselves of fellowships provided by SGS-Thomson. This work was supported in part by the EEC under the Project **ESPRIT-2197 (STORM)**, by the Italian National Council of Research under the "Progetto Finalizzato **MADESS**".

timization), an even higher cost is to be expected, and a careful selection of the optimization scheme becomes then mandatory. To face the problem, we have chosen the gradient method, so that the non-linear outer solution related to the optimization is split in a number of linear steps. In this way the inner solution, related to the non-linear set of device equations, is carried out only once for a fixed set of parameters, around which a sensitivity analysis (SA) is then performed by linearization. In this way, the overhead added to the inner solution is small. Among the other advantages of this approach it is worth mentioning that, thanks to linearization, the problems of the geometry and doping variations can be treated separately, and that the designer can easily guess the relative relevance of the parameters. If, in addition, the linearization is carried out analytically, SA needs no extra evaluation of the Jacobian matrix, provides a higher numerical accuracy compared to those methods involving a numerical computation of the derivatives, and further reduces the computational cost. On the other hand, the software development is in the latter case more complex since it requires additional pieces of code to evaluate the analytical derivatives [1].

3. The Analytical Approach to Sensitivity Analysis. The scheme by which SA has been implemented in HFIELDS will be sketched in the following. Using vector notation, the discrete system of device equations is written in the form $\mathbf{f}(\mathbf{x}, \mathbf{w}) = 0$, and the global Newton's scheme used in its solution reads

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \left[\mathbf{J}^{(k)}\right]^{-1} \mathbf{f}^{(k)}, \tag{1}$$

where vector $\mathbf{x}$ is made of N nodal unknowns (electric potential and carrier concentrations), vector $\mathbf{w}$ is made of L nodal parameters, $\mathbf{J}^{(k)}$ is the Jacobian matrix at the kth iterate, and $\mathbf{f}^{(k+1)} = \mathbf{f}(\mathbf{x}^{(k+1)}, \mathbf{w})$, $\mathbf{f}^{(k)} = \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{w})$. The L nodal parameters may be the doping values, the grid-point coordinates, or both. Let the Jacobian matrix at the solution be denoted by $\mathbf{J}$, and let the solution be repeated by changing $\mathbf{w}$. Denoting with $\mathbf{w}'$ the new set of parameters, different from $\mathbf{w}$, let $\mathbf{x}'$ be the corresponding solution, so that $\mathbf{f}(\mathbf{x}', \mathbf{w}') = 0$. Let $\mathbf{w}$ and $\mathbf{x}$ be called the *unperturbed* parameters and solution, respectively, and $\mathbf{w}'$, $\mathbf{x}'$ the *perturbed* ones. If the variations $\delta\mathbf{w} = \mathbf{w}' - \mathbf{w}$, $\delta\mathbf{x} = \mathbf{x}' - \mathbf{x}$ are small enough to allow for a first-order expansion, it is found $\mathbf{f}(\mathbf{x}', \mathbf{w}') = \mathbf{f}(\mathbf{x}, \mathbf{w}) + \mathbf{J}^{(U)}\delta\mathbf{x} + \mathbf{Y}^{(U)}\delta\mathbf{w}$, hence

$$\delta\mathbf{x} = -\left[\mathbf{J}^{(U)}\right]^{-1} \mathbf{Y}^{(U)}\delta\mathbf{w}. \tag{2}$$

In Eq. (2) it is

$$J_{mn}^{(U)} = \left(\frac{\partial f_m}{\partial x_n}\right)^{(U)}, \qquad Y_{ml}^{(U)} = \left(\frac{\partial f_m}{\partial w_l}\right)^{(U)}, \tag{3}$$

and symbol (U) in the partial derivatives indicates that they are evaluated using the unperturbed quantities. Now, it is easily seen that all the quantities at the RHS of $\delta\mathbf{x}$ in (2) are known. In fact, by definition, matrix $\mathbf{J}^{(U)}$ coincides with the Jacobian matrix $\mathbf{J}$ available at the end of the unperturbed solution. Moreover, $\delta\mathbf{w}$ is assigned while the expressions for the elements of $\mathbf{Y}^{(U)}$ can easily be calculated analytically from definition (3). By observing that Eqs. (1) and (2) are similar, it follows that the factorization of $\mathbf{J}^{-1}$ available at the end of the unperturbed solution can be exploited to solve (2), hence the only additional calculation needed to find $\delta\mathbf{x}$ is the evaluation of vector $\mathbf{Y}^{(U)}\delta\mathbf{w}$.

It should also be noticed that vector $\delta\mathbf{w}$ may be given either numerically or analytically. In the first case, it is assumed that a process simulation has been carried out prior to the procedure depicted here, hence the doping variation is calculated directly in terms of variations of process parameters (such as temperature, time, diffusion coefficients, and so on). In the second case, $\delta\mathbf{w}$ is given by means of analytical expressions that, in turn, depend on parameters such as the peak concentration of an implant, the channel length, and so on. When geometry variations are present, a particular grid-generation technique must be used when determining vector $\delta\mathbf{w}$ [2].

First, a simple computational geometry is found and a grid is built on it. Then, a differential transformation from the computational space to the real one defines the grid on the device geometry. When the latter is perturbed, a new transformation provides the varied grid in such a way that the node numbering of the unperturbed grid is kept. Thanks to this, the portion of $\delta\mathbf{w}$ related to geometry variations is easily evaluated. To complete the mathematical description, it is worth reminding that each component w_l of $\mathbf{w}$ depends on a number of parameters $\alpha_1, \alpha_2, \ldots$ By then letting, for any α_t, $z_m = \sum_{l=1}^{L} Y_{ml}(\partial w_l/\partial\alpha_t)$, $\mathbf{z} = (z_1, \ldots, z_N)$, Eq. (2) transforms into $\delta\mathbf{x} = -[\mathbf{J}^{(U)}]^{-1}\mathbf{z}\,\delta\alpha_t$. By means of the above procedure the perturbed solution is evaluated via linear operations only. Once the perturbed solution is known, the perturbed current density J_h' and, finally, the perturbed current at the contacts can easily be determined. In fact, the variation in the current at contact γ due to variation $\delta\alpha_t$ will be given by $\delta I_\gamma = I_\gamma' - I_\gamma = \eta_\gamma\delta\alpha_t$, where coefficient η_γ is evaluated in the unperturbed case. The variations δI_γ with respect to each parameter α_t involved in the optimization procedure are stored by **HFIELDS** in a matrix $S_{\gamma t}$, where $\gamma = 1, 2, \ldots$ ranges over the contacts and $t = 1, 2, \ldots$ ranges over the parameters. This matrix is then used to approximate to first order the partial derivatives $\partial I_\gamma/\partial\alpha_t$, which are then fed back by **HFIELDS** to the supervising program, mentioned above, that controls the pipeline.

4. The Object Function. The definition of the object function(s) in building-up an integrated optimization tool requires in itself a careful balance between the algorithmic capabilities of the system and the designers' needs. A designer who wishes to exploit an optimization tool to achieve a specific performance has to face first of all the difficulty of defining in a non-ambiguous way the performance itself. It is convenient here to base the discussion on a typical example, for instance the design of an MOS transistor whose physical characteristics are to be brought as close as possible to some specifications. The definition of the latters may be given in different ways: e.g., the designer could impose *i)* a number of constraints (typically, output conductance, gain, threshold voltage, slope of the $I_D(V_{GS})$ curves in the subthreshold region, short channel effects, etc.), or *ii)* the electrical characteristics directly, such as a number of $I_D(V_{DS}, V_{GS}, V_{BS})$ curves in the practical range of application of the external voltages. From the viewpoint of the optimization procedure, the first way leads to seeking for a constrained minimum of the function to be optimized in the parameter space, the second one leads to seeking for a free minimum. The procedure involving constraints is rather impractical in this context. In fact, most of the constraints lend themselves to an analytical description only when very simple models of the device characteristics are used, while their definition becomes ambiguous in the general case (example: the threshold voltage in MOS devices). Since the bulk of the optimization system is made by general-purpose simulators, the introduction of constraints based on simple models would unnecessary limit the generality of the system itself. For this reason the second approach (imposing the electrical characteristics and seeking for a free minimum) has been chosen, so that the application of SA in the parameters' space can be carried out directly on the full set of PDE's constituting the mathematical model of semiconductor devices. In this way no approximation on the model itself or its parameters is introduced, and the user can modify at will the electrical characteristics constituting the object function without being limited by any set of models incorporated *ad hoc* in the code.

It is worth adding that this approach does not prevent in itself the possibility of extracting a number of significant parameters such as, taking again the MOS device as an example, the threshold voltage, the body factor, and so on. The extraction, in fact, can still be carried out by a postprocessor, using the simpler analytical models mentioned above and without interfering with the optimization process. In this way, it becomes also easier to compare the performance and results of the optimization scheme based on SA with those of different schemes, e.g., the Response Surface Method (RSM) which is also used in the context of the EEC Project.

Finally, the definition of the object function is based on the observation that most of the constraints to be imposed on a design are related to *undesired* effects. In the example of an

MOS transistor, the designer aims at obtaining a device as free as possible of such effects as short-channel effect (non-zero output conductance), dependence of the threshold voltage on the drain voltage, punch-through effect (leaky subthreshold behaviour), etc. These, in fact, are regarded as second-order effects when building-up an analytical model. From this it follows that the object function can be derived by providing an analytical description from which the second-order effects are removed and only the first-order effects are kept.

5. Results. The approach outlined above has been followed in the implementation of the optimization procedure in `HFIELDS`. The simulator has been fully interfaced with `PROFILE` (developed at the Technische Universiteit Delft). The latter code determines the set of parameters α_1, α_2, ... that minimize $\Phi(\alpha_1, \alpha_2, \ldots) = (1/N_B) \sum_{\nu=1}^{N_B} [(\Psi_\nu - \Theta_\nu)/\Lambda_\nu]^2$, where $\Psi_\nu = \Psi(V_\nu, \alpha_1, \alpha_2, \ldots)$, $\Theta_\nu = \Theta(V_\nu)$ are the current calculated by the simulator and the object function, respectively, Λ_ν is a suitable weight, N_B is the number of bias points, and (for an MOS transistor) $V_\nu = [V_{DS}, V_{GS}, V_{BS}]_\nu$ is the ν-th set of applied voltages. The tests presented here have been based on an n-channel MOS transistor, with $t_{ox} = 10$ nm and $W = 1$ μm. The source and drain diffusions have been modeled as gaussian profiles with $N_{max} = 10^{20}$ cm^{-3} and 0.7 lateral penetration, this resulting in an effective channel length $L_{eff} = 2$ μm and junction depth of about 150 nm. A gaussian, p-type channel implant has been added with $r_0 = 0$. As mentioned before, the characteristics (goal of the optimization) have been derived from a simplified analytical model. The parameters in the latter are $t_{ox} = 10$ nm, $W = 1$ μm, $L_{eff} = 2$ μm, $V_{thr} = 0.5$ V, $\gamma = 0.3$ V$^{1/2}$, $\phi_F = 0.3$ V, and $\eta = 0$ (the last quantity is the short-channel coefficient). The possible parameters for the optimization were the peak concentration N_0 and the standard deviation σ of the implant, and the substrate doping N_A. The optimizations have been run using either one or two of the above parameters, and have been repeated using the relative and the absolute mean square error at a time (Fig. 1).

References.

[1.] A. Gnudi, P. Ciampolini, R. Guerrieri, M. Rudan, G. Baccarani, *Sensitivity Analysis for Device Design*, IEEE Trans. on CAD, vol. CAD-6, no. 5, 1987, pp. 879–885

[2.] Zsolt M. V.-Kovács, M. Rudan, *Boundary-Fitted Coordinate Generation for Device Analysis on Composite and Complicated Geometries*, IEEE Trans. on CAD, Special Issue on NUPAD-III, in press

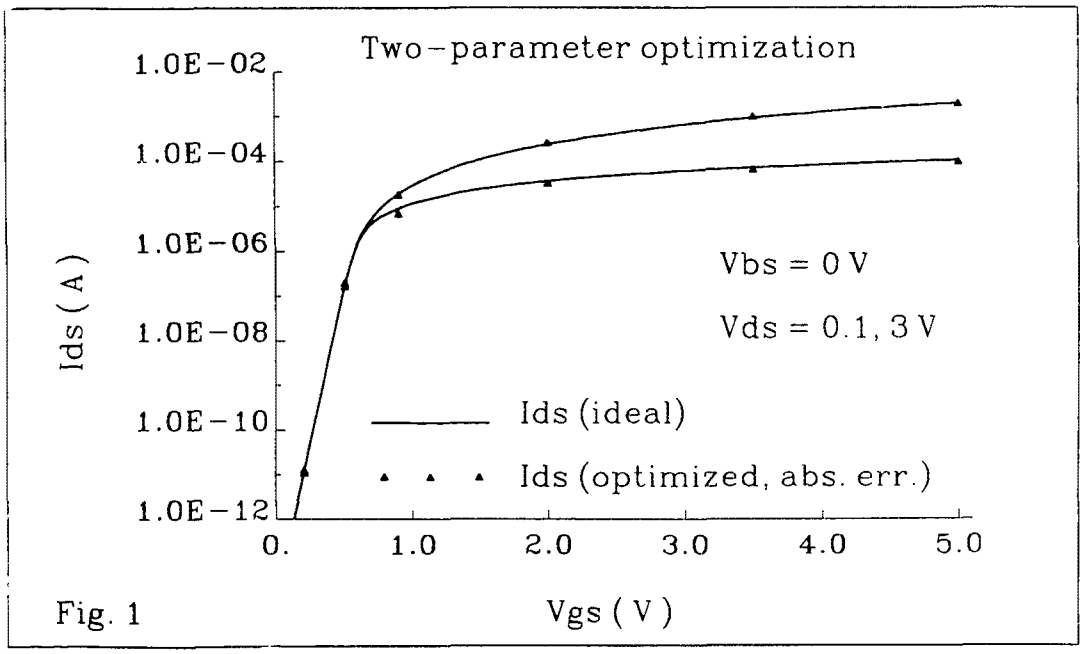

Fig. 1