Integrated Optimization Capabilities in the VISTA Technology CAD Framework

R. Plasun, *Associate Member, IEEE,* M. Stockinger, *Student Member, IEEE,* and Siegfried Selberherr, *Fellow, IEEE*

***Abstract*—Advanced analysis features implemented in the Vienna Integrated System for TCAD Applications simulation environment are presented. These functionalities support automatic experiment generation (design of experiments), model fitting (response surface methodology), optimization, and calibration. They interact with the core modules of the framework supporting the simulation of the manufacturing process and electrical characterization of semiconductor devices. Two examples demonstrate the efficiency of these framework capabilities. The first one shows the optimization of the electrical characteristics of vertical double-diffused metal–oxide–semiconductor (MOS) field-effect transistors. The second example deals with the optimization of analytical doping profiles of MOS transistors.**

I. Introduction

THE DESIGN and fabrication of smaller and faster semiconductor devices relies on the proper numerical simulation of fabrication processes and electrical characteristics. This field of technology engineering is known as technology computer-aided design (TCAD).

For a complete simulation and characterization of a modern very-large-scale-integration (VLSI) technology, several hundred simulation steps have to be computed. Furthermore, the process steps have different aspects, like geometry manipulations in etching and deposition steps or changes in the doping profiles during implantation and diffusion steps. Grid manipulations can also be necessary between the simulation steps. For these fairly complex tasks the Vienna Integrated System for TCAD Applications (VISTA) [1], [2] has been developed.

To improve the manufacturability of deep submicrometer devices, different variations of process parameters have to be analyzed. For this purpose, high-level analysis functionality has to be supported by a state-of-the-art TCAD framework.

II. Vienna Integrated System for TCAD Applications

The VISTA framework and its modules form a programmable simulation environment for TCAD applications. The framework is based on the VLISP interpreter [3]. This object-oriented LISP language is similar to the popular JAVA [4] programming language with class inheritance and polymorphism. The interpreter provides interfaces to the functions of a number of internal layers. The *operating system layer* encapsulates the operating-system-dependent services, which ensures portability over a wide variety of operating systems and platforms. The *user interface layer* provides interfaces to the MOTIF widget-set [5] and some additional widgets.

The advantage of VLISP compared to Tcl/Tk [6] is the object system and the large number of sophisticated basis classes for data manipulation, user interface generation, and handling asynchronous events.

Manuscript received December 3, 1997; revised April 24, 1998. This work was supported by Christian Doppler Gesellschaft, Vienna, Austria; Austria Mikro Systeme, Unterpremstätten, Austria; Hitachi Device Development Center, Tokyo, Japan; and Siemens Corp., Munich, Germany. This paper was recommended by Associate Editor Z. Yu.

The authors are with the Institute for Microelectronics, Technical University of Vienna, Vienna A-1040 Austria.

Publisher Item Identifier S 0278-0070(98)09364-6.

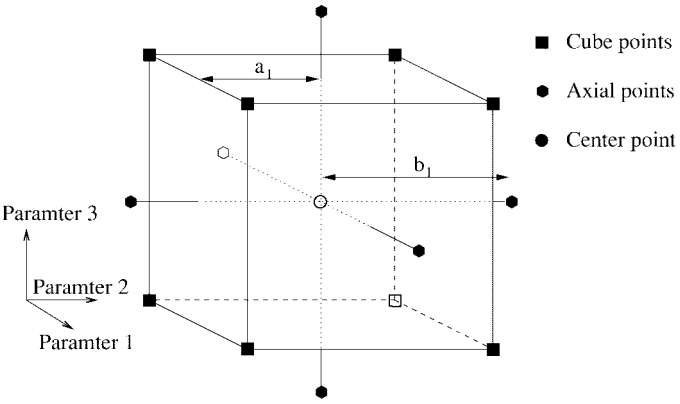

Fig. 1. The example points of a CCC design with three input parameters.

TABLE I
Factors a and b for Central Composite Designs with Full-Factorial Cube Points

	$\frac{b_i}{a_i}$	range_i
CCC	$\sqrt{n}$	$2a_i$
CCI	$\sqrt{n}$	$2b_i$
CCF	1	$2a_i = 2b_i$

TABLE II
Available Experimental Designs

NOM	Nominal Design
SA	Screening Analysis
FUL	Full Factorial Design
RAN	Random Design
DIA	Diagonal Design
GRI	2D - Grid Design
LAT	Latin Hypercube Design
FRA	Fractional Factorial Design
PLA	Plackett-Burman Design
OME	Orthogonal Main Effect Design
SUP	Supplementary Design

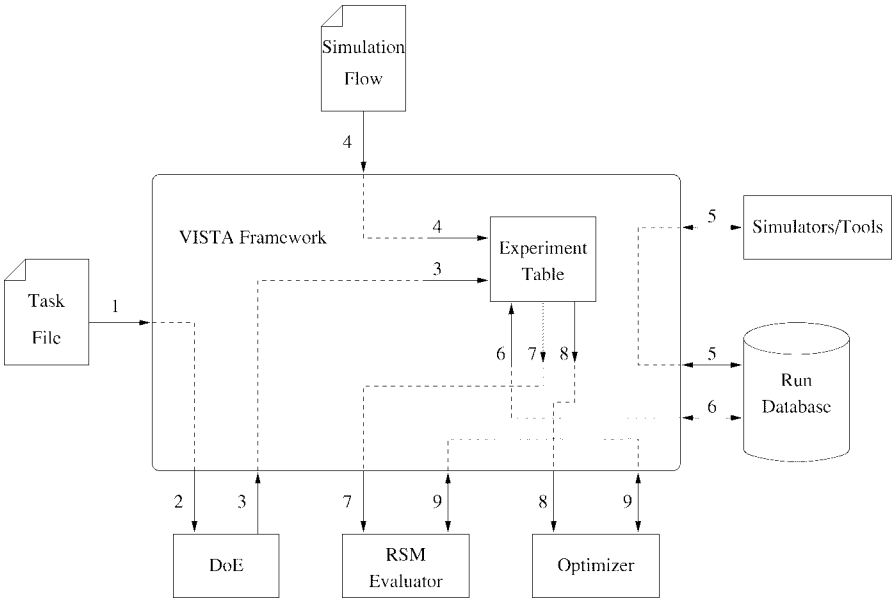

Fig. 2. Structure of a complex optimization task.

In the framework, individual process steps for the manufacturing process of semiconductor devices are stored in a hierarchical *simulation flow description*. In this format, the linear flow can be divided into several blocks (modules), each of which can consist of substeps or subblocks [7]. Furthermore, not only can the manufacturing process be simulated but also electrical characterization of the finished device can be added in this simulation flow description.

As an open framework, a large number of process and device simulators from different vendors, like TSUPREM [8] and MEDICI [9] from TMA and ATHENA [10] from SILVACO, are accessible. Integrated wrapper programs are used to convert the input files to the corresponding data format of the simulator.

It should be emphasized that for the integration of further simulators working with the TIF, SSF, or PIF data formats, the available data converters can be used. For other input formats, a separate wrapper must be supplied.

The evaluation of a *simulation flow* is performed by the *simulation flow controller* (SFC). It checks if the requested evaluation or parts of it are already present in the persistent *run data base*. This prevents the recomputation of existing results so only remaining simulations are scheduled for execution.

The distribution of the workload is done by job farming. For an efficient use of the resources, the framework polls periodically the load of the available hosts and compares it with the allowed maximum number of jobs for automatic load balancing.

The functions of the framework are designed to operate also without the user interface. By defining a small sequence using the VLISP extension language, the framework can be used in a batch-mode fashion.

III. Analysis Functions

For the support of modern integrated technology development and yield improvement, fully automatic TCAD analysis modules with complex features are included and presented in the following sections.

A. Design of Experiments (DoE)

DoE methods [11] are frequently used for automatic generation of experiments.

These methods are also available in other TCAD frameworks, namely, the *Virtual Wafer Fab* [12], the TMA *Work Bench* [13], and the NORMAN framework [14]. Unlike the presented framework, these implementations do not use additional transformations to model the system dependencies (see Section III-C).

Especially central composite designs are useful to explore the input parameter space with a minimum of required experiments. A central composite circumscribed (CCC) design for a three-dimensional control parameter space is shown in Fig. 1.

The design consists of $2n = 6$ *axial points* (n is the number of input parameters), $2^n = 8$ *cube points* (full factorial), and one *center point*. For rotatable designs, the factor b_i/a_i must be $\sqrt{n}$. The differences among the three types of central composite designs [CCC, central composite inscribed (CCI), and central composite face-centered (CCF)] are listed in Table I.

A design is rotatable if the variance of the prediction depends only on the distance from the center of the design and not on the direction.

The rotatability and the small number of necessary experiments make CCC and CCI designs very well suited for estimating the coefficients in a second-order model, as will be explained in the next section.

It would be beyond the scope of this paper to describe all other available designs in the DoE module; the experimental designs available in VISTA [11], [15] are summarized in Table II.

Input data for DoE modules are the minimum and maximum values of the control parameters and optional transformations, which are described in Section III-C.

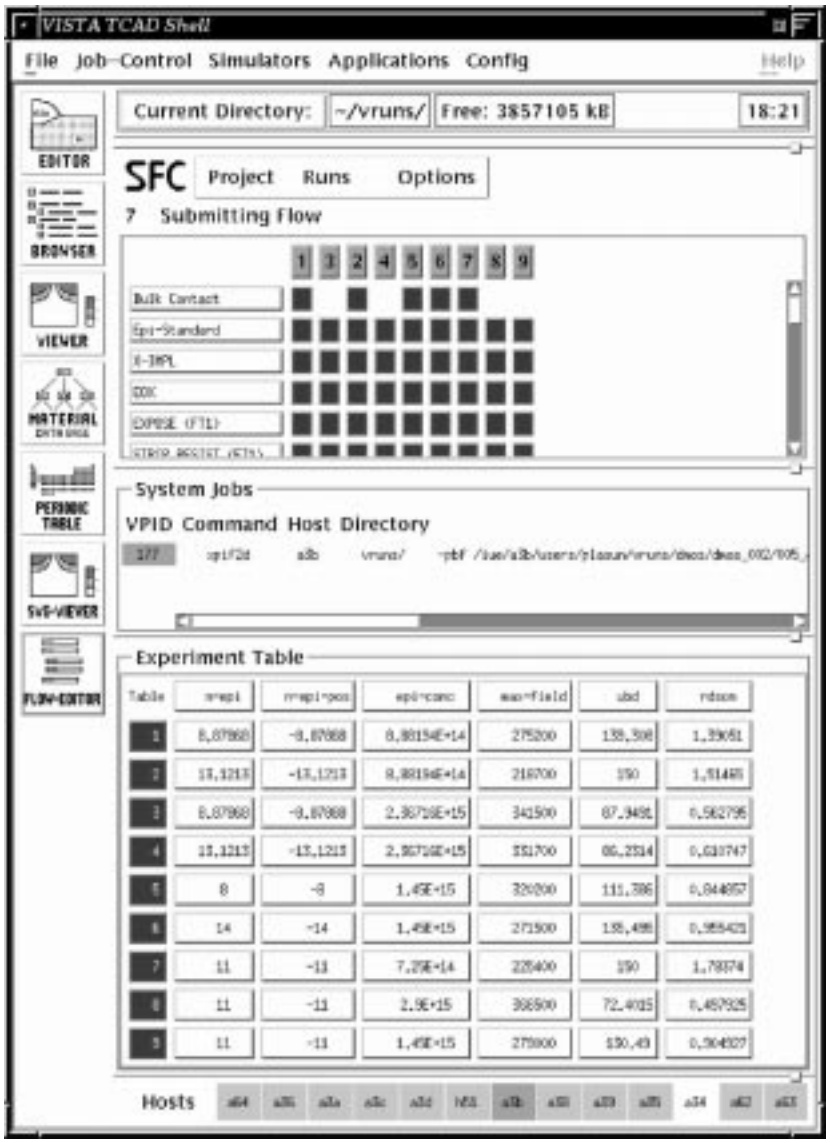

Fig. 3. Graphical user interface of the VISTA TCAD framework.

B. Response Surface Methodology (RSM)

RSM [16] is a technique to create mathematical models for the relationship between one or more responses and a set of input variables. The most widely used model functions are polynoms of second order

$$F(x_1, \cdots, x_n) = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} \sum_{j=i}^{n} a_{ij} x_i x_j$$

where n denotes the number of input parameters.

The factors for this analytical function can be calculated by a least square estimation. For both input (x) and output (F) parameters, transformations can be specified, so additional knowledge about the system behavior can be included.

RSM methods are also used in other TCAD tools, namely, DEBORA [17], the TMA framework, and the SILVACO framework. Unlike the presented RSM module, they do not use transformations for the input and output parameters. These tools are able to generate surfaces with approximately the same fit error as the presented RSM tool, but they need polynoms of higher order due to the lack of a transformation feature. Therefore, for solving the least square problem, they need a higher number of sample points.

C. Transformations

To model the system behavior accurately, both the DoE and RSM modules make use of transformations of the parameter space to linearize the dependence of the output variables on the transformed input parameters. Subdivision of the parameter space as well as fitting of the response surfaces takes place in the transformed space.

For each input parameter, a transformation function can be selected from a set of transformations. This list consists of logarithmic [18], square root, inverse [19], exponential, and other special transformations [20].

If the transformation function requires parameters (*transformation parameters*), these parameters either may be specified explicitly—for example, in the case where a physical formula has been established—or may be determined from a set of

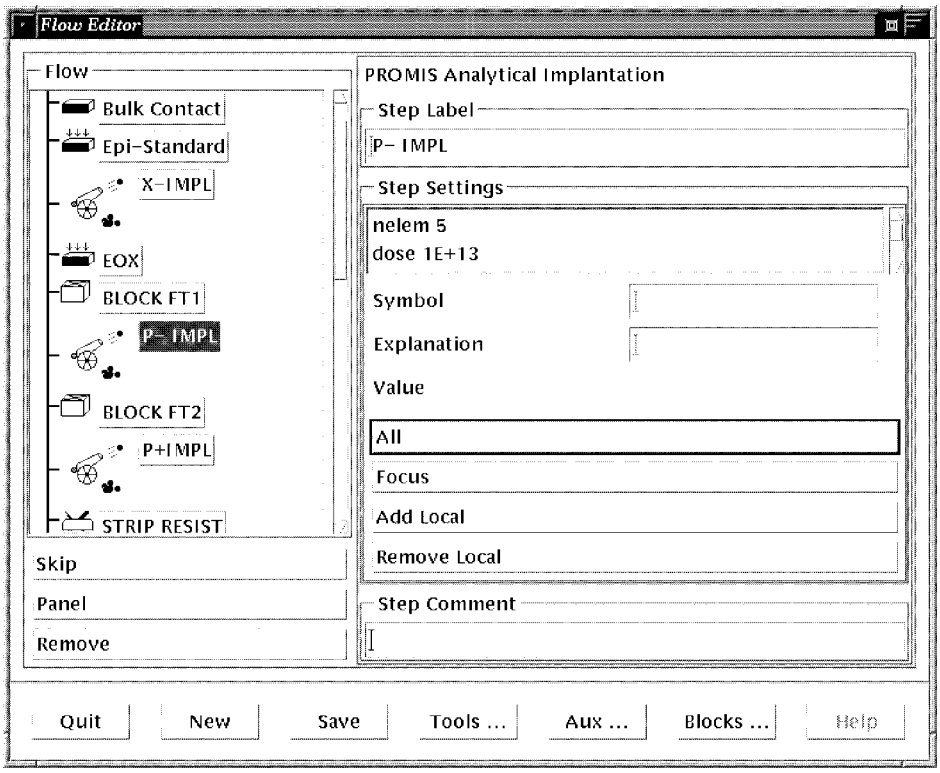

Fig. 4. The simulation flow of the VDMOS transistor shown in the *simulation flow editor*.

sample points automatically. Additionally, it is possible to automatically select the best out of a given set of transformation functions for a given set of sample points. This is done by computing several response surfaces where any possible variation of available transformations is applied to all input parameters. The selection criteria for the best set of transformations is determined by the minimum fit error of the sample points calculated by the total sum of squares of the residuals. For all these surfaces, the fit error of the sample points (the sum of squares) is used to select the best combination of transformations.

D. Optimization

For optimizing device performance parameters over a given input variable space, a constrained optimizer has been integrated. The use of a constrained optimization algorithm is important to prevent possible unphysical optima. The integrated nonlinear constrained optimizer uses an augmented Lagrangian method [21]. It minimizes the target function, which can be assembled out of input and output values. The gradient is calculated by evaluating adaptive finite differences. With this gradient-based optimizer, a local optimum can be found. To find global optimum solutions for general problems, algorithms like genetic algorithms have to be used, usually requiring an even larger number of evaluations. Choosing different initial parameters for the optimization is a simple method to provide further evidence of an optimal point.

For calibration and parameter fitting tasks, an optimizer based on the Levenberg–Marquardt algorithm [22] is also integrated in the framework.

For optimization tasks, *Virtual Wafer Fab* and *Work Bench* have a least square algorithm available, which is designed to work together with polynomial response surfaces. In the NORMAN environment, a constrained optimizer is implemented.

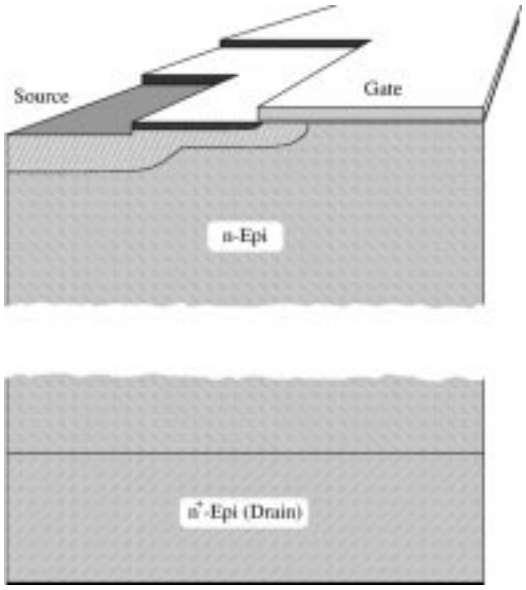

Fig. 5. Structure of the VDMOS transistor (source metallization and gate oxide are not shown).

E. Example of a Complex Task

In the VISTA framework, a complex optimization task consists of several analysis functions described above. Fig. 2 shows an example with DoE, RSM, and optimization steps. The numbers in the figure indicate the actions that have to be done by the framework listed below.

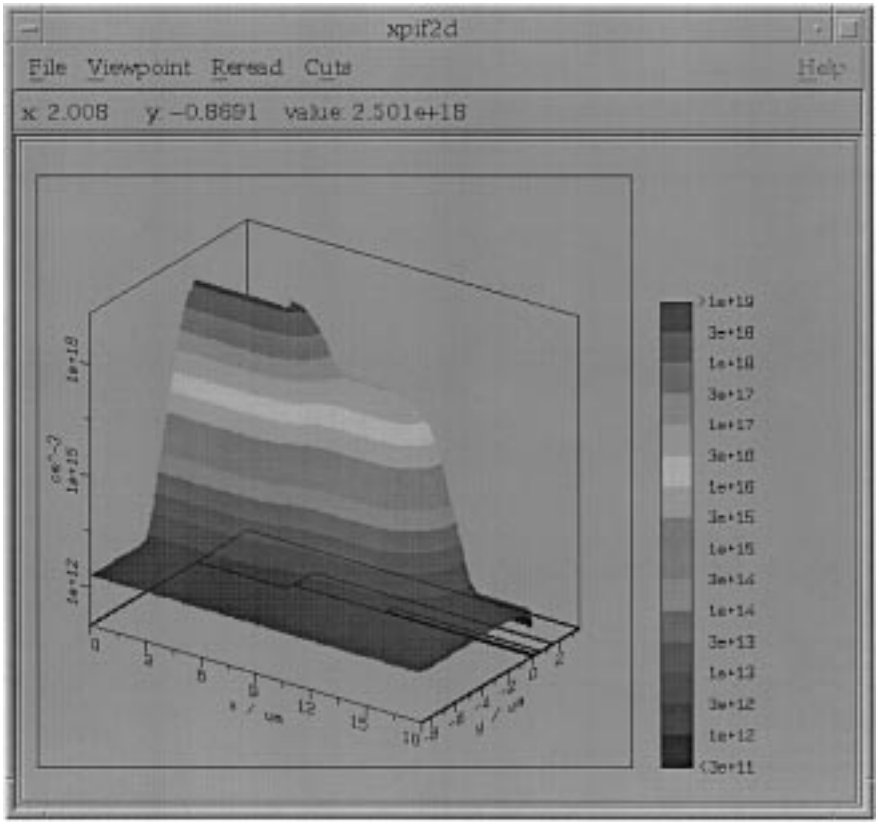

Fig. 6. Net-doping concentration (cm^{-3}) of the final VDMOS transistor.

1) Read the specification of the analysis task from the file. It consists of the reference of the stored simulation flow, the definition of control and response parameters with their location in the flow, and additional information like default values and ranges. The sequence of analysis steps [steps 2)–9)] is also read from this file.
2) Prepare the input for the DoE module (ranges of the control variables and the specified design) and start the DoE program.
3) Collect the evaluated result of the DoE module and add the control parameter values of the new experiments to the *experiment table*.
4) Load the referenced process flow. The *evaluable entity* links the set of controls from the *experiment table*, and the run controller generates the simulation split tree. An *evaluable entity* (*Eve*) is an object that encapsulates different kinds of evaluations, like the simulation of a process flow and extraction of the results or the evaluation of a point on a response surface.
5) The experiments are simulated using the integrated simulators and tools. Simulation results are stored in the persistent *run data base*.
6) All responses are collected from the finished runs and added to the *experiment table*.
7) After all experiments have been carried out, the complete data of the *experiment table* are given to the RSM module for evaluation.
8) The ranges, start values, and target functions are passed to the optimizer.
9) The optimizer requests evaluations from the RSM module and receives the calculated target function until an optimum is found.

During execution of this sequence, no user interaction is necessary, but the progress can be tracked on the graphical user interface. Alternatively, the whole optimization task can be executed as a batch job.

IV. Graphical User Interface

The graphical user interface of the VISTA framework enables the user to access the framework functions in an intuitive way. It supports the configuration of the simulation environment, definition and execution of simulation flows, and examination of the simulated data. The user interface (Fig. 3) consists of several parts.

The topmost window contains the interface of the SFC. The process steps of the selected simulation flow are listed on the left side. The split tree of simulation runs in the current project provides direct access to all data of the computed steps and gives a quick summary of the activity states of the simulations.

The *system jobs* window shows all active and queued system jobs started on behalf of the *run controller*.

The *hosts* window displays the busy state of all network hosts used for submitting system jobs. Different colors indicate the status of the enabled hosts.

```
;; process flow
(Project :flow-file #"~/vwork/flows/dmos.sfe")

;; creates an evaluation object
(Flow-Eve)

;; definition of the control variables
(Eve-Define-Control
   "n-epi"      "Epi-Standard"  "thickness")
(Eve-Define-Control
   "n-epi-pos"  "Bulk Contact"  "y-offset"
   :eval-expr '(* -1 n-epi))
(Eve-Define-Control
   "epi-conc"   "Epi-Standard"  "dope-conc")

;; assign values to the control variables
(Eve-Set-Control "n-epi" 10.5 :min 8. :max 14.)
(Eve-Set-Control "epi-conc" 1.45e15
   :min 0.725e15   :max 2.9e15    :trans 'log)

;; definition of the response variables
(Eve-Define-Response
   "max-field" "MMNT-breakdown"  "max-filed"
   :eval-expr '(apply 'max max-field))
(Eve-Define-Response
   "ubd"       "MMNT-breakdown"  "iv-data"
   :eval-expr '(when ubd (calc-breakdown ubd)))
(Eve-Define-Response
   "rdson"     "MMNT-rdson"      "iv-data"
   :eval-expr '(when rdson (calc-rdson rdson)))

;; optimization tasks
(sequence
 '(Eve-Doe 'CCI)
 '(Rsm-Eve "rsm-eve" :skip '("n-epi-pos"))
 '(Eve-Optimize "target"
   :eve "rsm-eve"
   :range-eve sfc::current-eve
   :eval-exp '(+ rdson
                 (* 100 (if (plusp (- 110 ubd))
                            (- 110 ubd) 0)))))
```

Fig. 7. Listing of the VLISP program, which solves the optimization problem of the VDMOS transistor example.

The *experiment table* window contains a spreadsheet representation of all simulation runs and control and response variables defined in the process flow. Task-level tools like DoE or RSM generation are directly accessible.

For the definition and editing of hierarchical simulation flows, the *simulation flow editor* (Fig. 4) uses a tree-like representation for maneuvering through the flow. In the right part of the window, parameters of the simulation steps can be specified.

V. Applications

A. Optimization with a Complete Process Simulation

In this section it will be shown how the electrical parameters of a vertical double-diffused metal-oxide-semiconductor field-effect transistor (VDMOS) can be improved by our high-level analysis functionality.

The structure of the VDMOS transistor in Fig. 5 has the advantage of a lower on-resistance and a smaller lateral size compared with a lateral DMOS transistor. Due to the high breakdown voltage of up to 100 V, these transistors are commonly used as power devices, for example, in automotive electronics. Fig. 6 shows the net-doping concentration of the final VDMOS transistor.

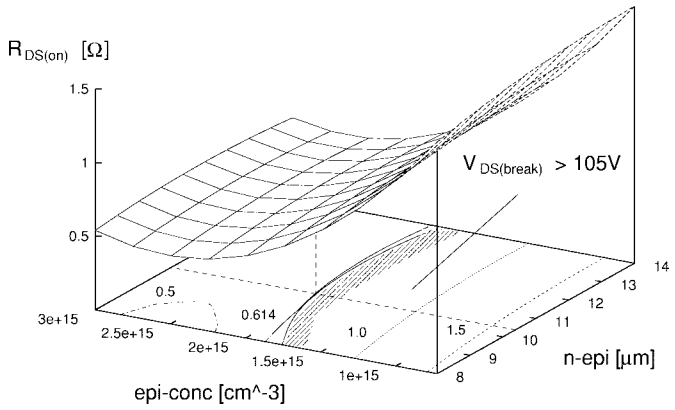

Fig. 8. Response surface of the target function.

The two control variables are the epitaxial doping value and the thickness of the epi-layer. The extracted responses are the on-resistance `rds-on` and the breakdown voltage `ubd`. The goal of this optimization is to minimize the on-resistance but keep the breakdown voltage above 105 V.

Fig. 4 shows the simulation flow of the device in the simulation flow editor. For the simulation of the process fabrication the programs SKETCH, ETCH [23], and PROMIS-Implant [24], the commercial tool TSUPREM [8], and, for the electrical characterization, MINIMOS-NT [25] were used.

The optimization problem is defined by a short VLISP program listed in Fig. 7.

In the first part of the program, a *Flow-Eve* with its related flow file is defined. The function `Eve-Define-Control` defines an internal name of a control variable (first argument) from a process step and from a parameter of this step (second and third argument). Derived controls can be declared using the additional key `:eval-expr`, which is done in this case for the control `n-epi-pos`—the bottom position of the epi-layer—denoting the negative value of the epi-layer thickness (`n-epi`).

`Eve-Set-Control` assigns default values, ranges, and transformations for the internal control variables. The used transformations are logarithmic for epitaxial doping and linear for the thickness of the epi-layer. `Eve-Define-Response` defines a control variable and has the same arguments as `Eve-Define-Control`.

The second part of the program describes the optimization task like that given in Fig. 2. `Eve-Doe` generates a CCI experimental design with the two independent control variables. The resulting nine runs were simulated on a workstation cluster, scheduled by the *run controller*.

After all steps are finished and the responses are extracted, the `Rsm-Eve` builds the responses surface. `Eve-Optimize` starts the optimization using the evaluations of the response surface.

The functions `calc-breakdown` and `calc-rdson` calculate the specified values from the `iv-data` structure.

The minimum value of the on-resistance was found on the constraint where the breakdown voltage reaches the lower limit of 105 V. Fig. 8 shows the target function (on-resistance) versus the parameter space of the control variables (epi-doping concentration and epi-layer thickness). Thus, the optimum

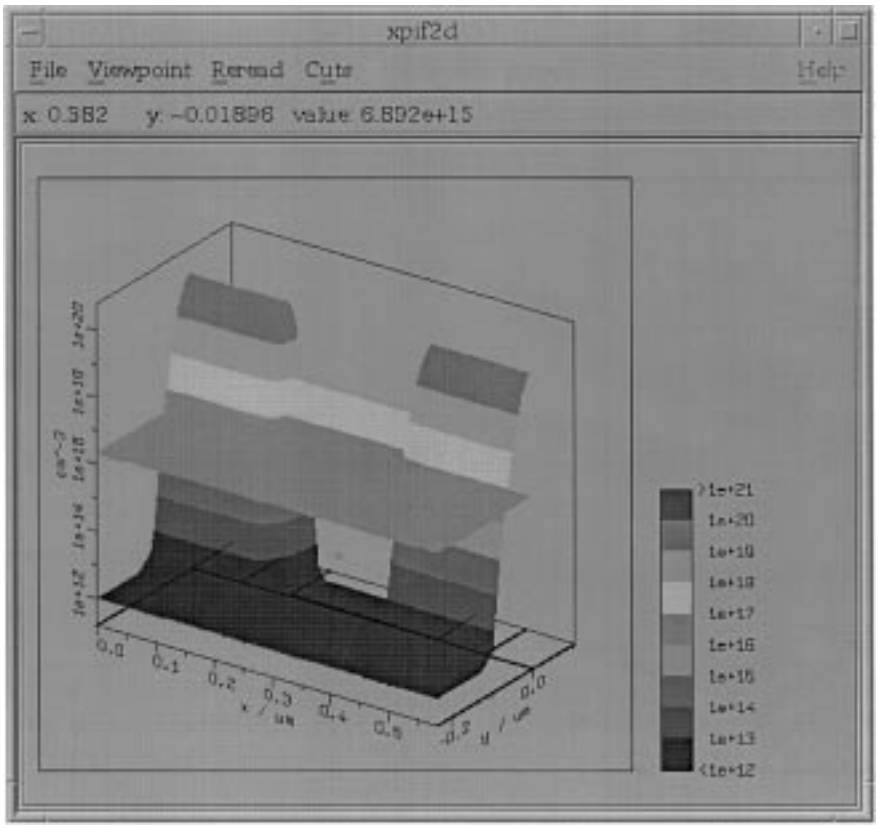

Fig. 9. Analytical acceptor and donor doping profile of the optimized MOS transistor.

process parameters were found with `n-epi` $= 10.37\ \mu$m, `epi-conc` $= 1.9\,10^{15}$ cm^{-3}, and $R_{DS(\mathrm{on})} = 0.614\ \Omega$ (Fig. 8).

For the calculation of the response surface, nine runs with 28 simulation steps have been executed, and the optimizer needed about 250 evaluations of this response surface.

B. Optimization of Analytical Doping Profiles

As MOSFET's are scaled to the submicrometer regime, the doping profile becomes increasingly important in affecting their performance. Profile optimization can be done either on the process level or on the profile-abstraction level, which means that instead of varying certain process parameters, analytical models are used to describe the doping profile and the model parameters are optimized. This method provides higher speed because no process simulation steps are required. The demand from the semiconductor industry for such methods has increased rapidly in the past few years, offering the possibility to fit simulated doping profiles to measured device data (*inverse modeling* [26], [27]).

In this section, the doping profile for a specified device structure is optimized to achieve certain performance improvements. Analytical doping profiles are generated by a template-based device generator. Parameters are global technology values like the geometry and doping level of a layer. These parameters are used as control parameters for the optimization process.

In this example, an NMOS device with 0.25-μm geometry gate length and 5-nm gate-oxide thickness is optimized for 1.5-V supply voltage. The analytical doping profile consists of a retrograde well and a channel implant, both generated with Gauß-functions. Control parameters are the depth, the deviation, and the doping levels of the channel implant and the retrograde profile. As the doping levels are varying in a large range, a logarithmic transformation is used.

The optimization target is to achieve maximum on-current. Without a constraint, this optimization would result in a decrease of the threshold voltage only. The off-current would be drastically increased because of its exponential dependence on threshold voltage in the subthreshold region [28]. A large off-current leads to high system standby power, which does not meet the requirements for future MOSFET technology [29]. Therefore, the off-current is entered as a constraint for the optimization process and kept at a constant value. Two device simulation steps have to be performed to extract the two current values. To increase the speed of the optimization, these two steps are done in parallel.

A uniformly doped device is used as the initial device. The on-current is improved drastically by the presented optimization process. Together with the flexible framework features, tasks like reverse engineering of doping profiles can be realized.

Because of the larger parameter space compared to the previous example, the simulation flow consisting of one generation step, and the electrical characterization, direct optimization is performed.

Fig. 9 shows the analytical acceptor and donor doping profile of the optimized transistor with the source and drain regions and the channel implant.

The on-current of the optimized transistor ($I_{\rm on} = 2.98\,10^{-4}$ A) was increased by 20% compared with a uniformly doped device ($I_{\rm on} = 2.49\,10^{-4}$ A) with the same off-current $I_{\rm off}$.

It took about 284 evaluations of the simulation flow, including the finite differences for the gradient calculation.

VI. Conclusion

Several complex tasks in TCAD engineering have to be supported by a state-of-the-art framework. We have shown that the flexible design of the VISTA framework makes it possible to integrate modules for design of experiments, response surface methodology, and optimization. These modules can be used with the existing simulation flows with no additional overhead.

We have demonstrated in the example of the vertical DMOS transistor and the NMOS transistor that the VISTA framework is well suited for the complex task of optimizing device parameters.

We intend to use this framework for a fully two-dimensional doping profile optimization with an even larger number of parameters using optimization targets that also include dynamic device characteristics.

References

[1] S. Halama, F. Fasching, C. Fischer, H. Kosina, E. Leitner, C. Pichler, H. Pimingstorfer, H. Puchner, G. Rieger, G. Schrom, T. Simlinger, M. Stiftinger, H. Stippel, E. Strasser, W. Tuppa, K. Wimmer, and S. Selberherr, "The Viennese integrated system for technology CAD applications," in *Technology CAD Systems*, F. Fasching, S. Halama, and S. Selberherr, Eds. Wien, Austria: Springer, 1993, pp. 197–236.

[2] C. Pichler, R. Plasun, R. Strasser, and S. Selberherr. (May 1997). High-level TCAD task representation and automation. *IEEE J. Technol. Computer-Aided Design*. [Online]. Available WWW: http://www.ieee.org/journal/tcad/accepted/pichler-may97.

[3] *VISTA Documentation 1.3-1, VLISP Manual*, Institut für Mikroelektronik, Technische Universität Wien, Austria, Jan. 1996.

[4] K. Arnold and J. Gosling, *The Java Programming Language*. Reading, MA: Addison-Wesley, 1996.

[5] D. Heller and P. Ferguson, *Motif Programming Manual*. Sebastopol, CA: O'Reilly, 1994, vol. 6A.

[6] B. Welch, *Practical Programming in Tcl and Tk*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[7] C. Pichler, "Integrated semiconductor technology analysis," Ph.D. dissertation, Technische Universität Wien, Austria, Mar. 1997.

[8] *TMA TSUPREM-4, Two-Dimensional Process Simulation Program, Version 6.2*, Technology Modeling Associates, Inc., Palo Alto, CA, June 1995.

[9] *TMA MEDICI, Two-Dimensional Device Simulation Program, Version 2.0*, Technology Modeling Associates, Inc., Palo Alto, CA, Sept. 1994.

[10] *ATHENA: 2D Process Simulation Framework*, Silvaco, user's manual, 1993.

[11] T. Lorenzen and V. Anderson, *Design of Experiments*. Berlin, Germany: Marcel Dekker, 1991.

[12] *VWF Interactive Tools User's Manual*, 1st ed., Silvaco International, Santa Clara, CA, Mar. 1994.

[13] *TMA WorkBench Version 2.0 User's Manual*, Technology Modeling Associates, Inc., Sunnyvale, CA, Mar. 1996.

[14] R. Booth, L. Dupas, and R. Cartuyvels, *NORMAN User's Manual*, IMEC, Leuven, Belgium, Feb. 1993.

[15] J. Cornell, *How to Apply Response Surface Methodology*. Milwaukee, WI: American Society for Quality Control, 2nd ed., 1990.

[16] G. Box and N. Draper, *Empirical Model-Building and Response Surfaces*. New York: Wiley, 1987.

[17] R. Cartuyvels and L. Dupas, *DEBORA User's Manual*, IMEC, Leuven, Belgium, May 1993.

[18] G. Gaston and A. Walton, "The integration of simulation and response surface methodology for the optimization of IC processes," *IEEE Trans. Semiconduct. Manufact*, vol. 7, no. 1, pp. 22–33, 1994.

[19] D. Boning and P. Mozumder, "DOE/Opt: A system for design of experiments, response surface modeling, and optimization using process and device simulation," *IEEE Trans. Semiconduct. Manufact.*, vol. 7, no. 2, pp. 233–244, 1994.

[20] R. Cartuyvels, R. Booth, S. Kubicek, L. Dupas, and K. De Meyer, "A powerful TCAD system including advanced RSM techniques for various engineering optimization problems," in *Simulation of Semiconductor Devices and Processes*, S. Selberherr, H. Stippel, and E. Strasser, Eds. Wien, Austria: Springer, 1993, vol. 5, pp. 29–32.

[21] P. Gill, W. Murray, and M. Wright, *Practical Optimization*. New York: Academic, 1995.

[22] N. Khalil, J. Faricelli, C. Huang, and S. Selberherr, "Two-dimensional dopant profiling of submicron MOSFET's using nonlinear least squares inverse modeling," in *Proc. 3rd Int. Workshop Measurement and Characterization of Ultra-Shallow Doping Profiles in Semiconductors*, J. Ehrstein, R. Mathur, and G. McGuire, Eds., Mar. 1995, pp. 6.1–6.9.

[23] E. Strasser and S. Selberherr, "Algorithms and models for cellular based topography simulation," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 9, pp. 1104–1114, 1995.

[24] H. Stippel and S. Selberherr, "Three dimensional Monte Carlo simulation of ion implantation with octree based point location," in *Proc. Int. Workshop VLSI Process and Device Modeling*, Nara, Japan, 1993, pp. 122–123.

[25] T. Simlinger, H. Kosina, M. Rottinger, and S. Selberherr, "MINIMOS-NT: A generic simulator for complex semiconductor devices," in *25th European Solid State Device Research Conference*, H. de Graaff and H. van Kranenburg, Eds. Gif-sur-Yvette Cedex, France: Editions Frontieres, 1995, pp. 83–86.

[26] Z. Lee, M. McIlrath, and D. Antoniadis, "Inverse modeling of MOSFET's using I–V characteristics in the subthreshold region," in *Proc. Int. Electron Devices Meeting*, 1997, pp. 683–686.

[27] A. Das, D. Newmark, I. Clejan, M. Foisy, M. Sharma, S. Venkatesan, S. Veeraraghavan, V. Misra, B. Gadepally, and L. Parrillo, "An advanced MOSFET design approach and a calibration methodology using inverse modeling that accurately predicts device characteristics," in *Proc. Int. Electron Devices Meeting*, 1997, pp. 687–690.

[28] Y. Tsividis, *Operation and Modeling of the MOS Transistor*. New York: McGraw-Hill, 1987.

[29] P. Voorde, "MOSFET scaling into the future," *Hewlett-Packard J.*, vol. 48, no. 4, pp. 96–100, Aug. 1997.

R. Plasun (S'95–A'96) was born in Vienna, Austria, in 1969. He received the Diplomingenieur degree from the Technical University of Vienna in 1994. He currently is pursuing the doctoral degree at the Institut für Mikroelektronik, Technical University of Vienna.

From October to December 1996, he was with the Advanced Technology Group at National Semiconductor Corp., Santa Clara, CA, where he concentrated on tool binding and process optimization aspects. His scientific interests include linear and nonlinear optimization, technology CAD framework aspects, and software development.

M. Stockinger (S'98) was born in Vienna, Austria, in 1970. He received the Diplomingenieur degree from the Technical University of Vienna in 1996. He currently is pursuing the doctoral degree at the Institut für Mikroelektronik, Technical University of Vienna.

He worked on an ASIC-design project at the Institut für Allgemeine Elektrotechnik und Elektronik for six months. His scientific interests include device and circuit simulation, circuit design, solid-state device technology, and low-power technology.

Siegfried Selberherr (M'79–SM'84–F'93), for a photograph and biography, see p. 572 of the July 1998 issue of this TRANSACTIONS.