Simulation of Complete VLSI Fabrication Processes with Heterogeneous Simulation Tools

Christoph M. Pichler, *Member, IEEE*, Richard Plasun, Rudolf Strasser, and Siegfried Selberherr, *Fellow, IEEE*

***Abstract*— An integrated environment for the simulation of VLSI fabrication processes is presented. Emphasis is put on automated operation to achieve maximum efficiency in TCAD deployment. Addressing the increasing number and diversity of process steps in state-of-the-art semiconductor fabrication processes, mechanisms have been devised to support the smooth, automatic interaction of heterogeneous simulation tools with multiple data formats in the context of large-scale experiments for global calibration, device optimization, and yield improvement tasks. For maximum versatility, the operation of the environment is either controlled via a graphical user interface, a batch file, or a combination of the two. It is possible to submit predefined analysis tasks for background execution, while still being able to monitor and control operation and to access and view simulation data interactively. Split-lot experiments are performed on workstation clusters in parallel operation, delivering the desired results in the shortest possible time. The TCAD environment presented offers server functionality for running large number of complex simulations. At the same time, it supports the design and seamless integration into the environment of client task applications.**

***Index Terms*— Semiconductor device manufacture, semiconductor device modeling, simulation design automation.**

I. Introduction

SEMICONDUCTOR technology relies essentially on technology CAD (TCAD) support during process development and process optimization. The traditional approach of using simulators for investigating isolated design aspects, e.g., for the determination of the source-drain doping profile in a lightly doped drain (LDD) structure, has been replaced by the computer-aided analysis of complete VLSI fabrication processes [1]. Apart from the increasing availability of more powerful computer systems, three reasons for this TCAD paradigm can be found.

1) Shrinking device dimensions and more tightly integrated circuits lead to increasingly complex dependencies of critical device characteristics on fabrication process parameters.
2) Complex circuitry requires a growing number of interconnect layers and novel device designs that exhibit intricate coupling between topography operations and dopant distributions.
3) Growing competition in semiconductor manufacturing calls for an increased speed of introduction of new products.

Moreover, smaller device structures exhibit new effects, calling for advanced models that require a high degree of specialization on the part of the simulation tool developer, resulting in a multitude of heterogeneous simulators. To be able to deliver accurate results, TCAD users must be able to select the best available tool for a given fabrication process step, and must not be limited to a particular vendor's solution.

A TCAD environment that intends to provide effective support of process and device design in a production setting has to put special emphasis on several key issues.

- *Openness* for the integration of different simulation tools allows to use state-of-art models and tools and to follow technology changes quickly.
- Access to simulators and TCAD tasks in a *black-box* manner ensures a problem-oriented application of TCAD and liberates the user from software engineering issues, data handling procedures, and tool peculiarities.
- A *uniform process representation* makes process flow information independent of individual tools and provides a basis for data exchange with CIM systems.
- *Intuitive operability* and a high degree of *task-level automation* are required to facilitate the application of the TCAD system to large-scale experiments.

A number of TCAD frameworks have been addressing these issues. Silvaco's Virtual Wafer Fab [2] supports the creation of split trees for parallel simulation of independent branches across the network. It does not offer an extension language, confining the set of supported tasks, including design of experiments (DoE), and response-surface model (RSM) generation and optimization. The set of applicable simulators is limited to the vendor's tools. TMA's Work Bench [3] allows to call arbitrary external tools by providing a UNIX-script based interface. The NORMAN system [4] creates response surface models by substituting parameter values in simulator input decks and executing them iteratively. The DoE/Opt framework [5] concerns itself with the design of experiments according to DoE schemes, the generation of response surface models, and with the management of the generated data in a spreadsheet fashion. A built-in optimizer solves optimization tasks with

Manuscript received September 5, 1997; revised September 22, 1998. This work was supported in part by ADEQUAT (JESSI Project BT11), ESPRIT Project 8002; Austria Mikro Systeme, Unterpremstätten, Austria; Christian Doppler Forschungsgesellschaft, Vienna, Austria; Digital Equipment Corporation, Hudson, MA; Hitachi DDC, Tokyo, Japan; National Semiconductors, Santa Clara, CA; Philips NatLab, Eindhoven, Netherlands; and Siemens, Munich, Germany.

C. M. Pichler was with the Institute for Microelectronics, Technical University Vienna, A-1040 Vienna, Austria. He is now with National Semiconductor Corporation, Santa Clara, CA 95052 USA (e-mail: pichler@iue.tuwien.ac.at).

R. Plasun, R. Strasser, and S. Selberherr are with with the Institute for Microelectronics, Technical University Vienna, A-1040 Vienna, Austria.

Publisher Item Identifier S 0894-6507(99)01219-1.

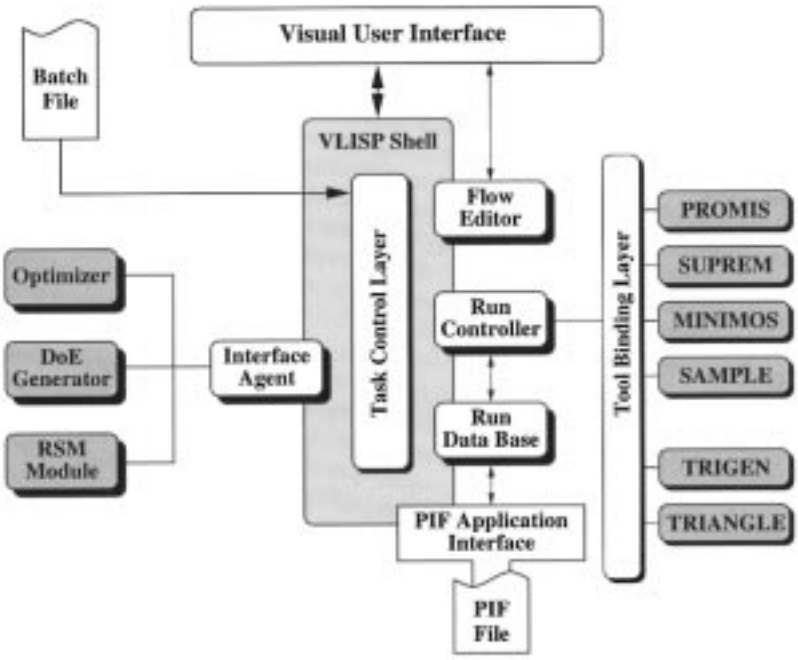

Fig. 1. Basic functional components of the VISTA/SFC environment.

multiple targets and nonlinear constraints. All simulation-related issues must be handled outside of the framework.

Drawing on these results, we have developed the VISTA/SFC TCAD environment. It is based on the VISTA TCAD framework as described previously in [6] and [7], and on the simulation flow control module (SFC) [8], [9]. In this paper, we describe refinements to the architecture, and focus on the new task, batch, and user interface definitions and implementations that enable advanced and highly automated process and device analysis. VISTA/SFC bridges the gap between VISTA's generic versatility and practical requirements for efficient TCAD operation. Experience has shown that the gridding strategies described in [7] do not suffice for complicated structures with highly irregular boundaries. Therefore, all grid-related services have been completely redesigned. Interfaces to external task-level tools have been added, support for batch-mode operation has been improved, and the GUI has been completely redesigned, both conceptually and on the surface.

In the remainder of this paper, a description of the overall architecture and main components of VISTA/SFC is given, followed by more detailed accounts of tool integration strategies, process flow representation, and run control. Finally, the simulation of a complete CMOS process demonstrates the capabilities of the VISTA/SFC both as an integration framework and as a task-level automation environment.

II. Architecture and Components

Fig. 1 gives an overview of the principal functional components of the VISTA/SFC environment, comprising the VLISP interpreter, the run controller, the run data base, the flow editor, the tool binding layer, interface agents for the integration of external task-level tools, and the task control layer.

The VLISP *shell interpreter* [10] forms the basis for the implementation of all other modules. It provides interfaces to the operating system, the graphical user interface (GUI), and the PIF application interface (PAI) [11] to conveniently access simulation data stored in the profile interchange format (PIF) [12].

The *flow editor* offers an intuitive and convenient visual interface for writing process flows. It supports the definition of process flows in a hierarchical and modular manner in terms of tool-independent process statements as well as explicit tool statements.

Process flow information is interpreted by the *run controller*, which together with the *run data base* forms the core components for the management of iterative and parallel split-lot experiments. The run controller takes care of the detection of splits, of scheduling multiple runs in parallel operation on workstation clusters, and offers a number of operation modes to facilitate development and debugging of both processes and simulation tools. The run data base stores and retrieves simulation output data and extracted data of any format, with the PIF format being used as primary exchange format for wafer data.

The *task control* layer takes care of controlling all activities initiated via the GUI, the ASCII interface, or a batch file. It is implemented as a set of dedicated VLISP functions and establishes object-oriented interfaces for all task-level services.

Task-level tools are interfaced with the task control layer by interface agents that establish communication channels with concurrent modules like optimizers, design-of-experiment (DoE) tools, response-surface-modeling (RSM) modules. While being run as clients of the task control layer, they also operate as servers for more complex applications. For instance, a parameter extractor for a circuit simulator needs as input a set of device characteristics for different devices. It

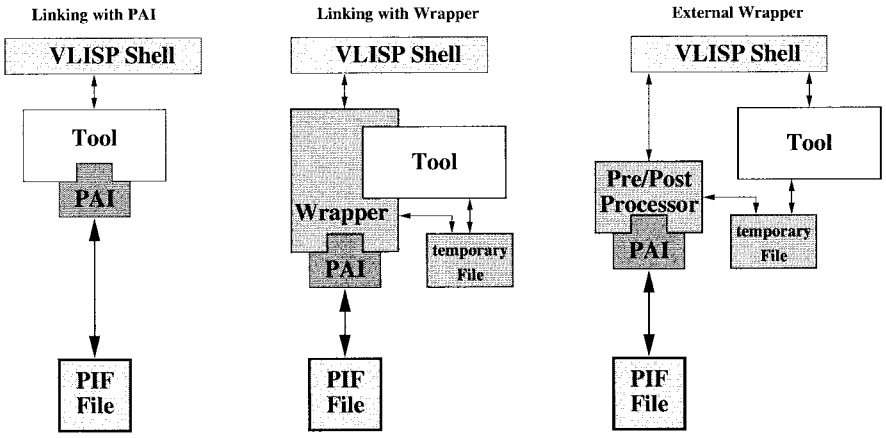

Fig. 2. Data level integration schemes.

then uses different strategies to adjust the model parameters to accurately reflect the simulated device characteristics. The circuit simulator reads a netlist and the extracted parameters to simulate the electrical circuit. In VISTA/SFC, the parameter extractor operates as a client of the run controller, while at the same time acting as a server for the circuit simulator.

III. Tool Integration

Fast technology changes have resulted in the development of specialized simulation tools, targeting particular simulation tasks. To optimally model a process technology, it is of great importance to use the tool best suited for each process step instead of confining oneself to a particular product's or vendor's solutions.

The integration of separate tool requires coupling on the *data level* and on the *control level*. Furthermore, on the *presentation level* a user interface has to be provided to allow editing of tool settings in a comfortable and intuitive way.

A. Data Level

On the data level, the exchange and conversion of tool input and output data between syntactically and possibly semantically different data representations has to be provided. VISTA/SFC uses the profile interchange format (PIF) [12] as the primary data format for representing wafer data and for exchanging simulation data between tools. PIF data are accessed using the PIF Application Interface (PAI) [11], one of VISTA's core components. Tools are either linked with the PAI or use external wrappers to convert data to the PIF format (Fig. 2). The second approach is often preferable as no changes of tool code are necessary and tool upgrading is simplified.

A single common data format is highly desirable for the integration of heterogeneous simulation tools. However, enforcing the conversion to a common data format after each simulation step causes a significant overhead in computing time and a loss in numerical stability. If two subsequently executed simulators share a common data format, intermediate conversions can be skipped, which leads to an increase in speed and numerical accuracy. Therefore, VISTA/SFC supports native tool formats[1] as well, with conversions taking place automatically when required by a subsequent simulation step or by a postprocessor.

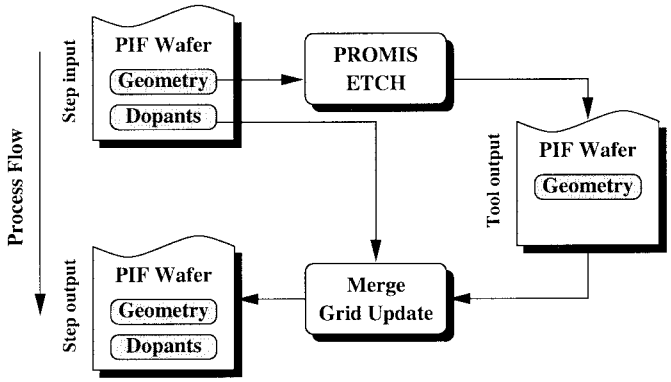

Fig. 3. Data flow for creating a consistent wafer description after a PROMIS etch operation.

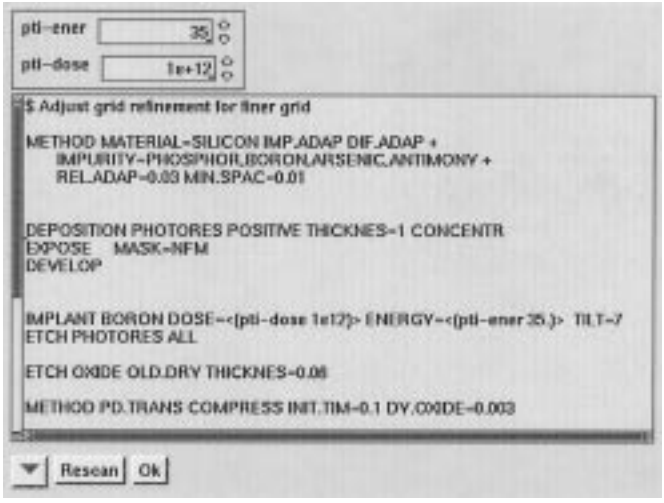

Fig. 4. Visual editor generated automatically from template file. Expressions enclosed in matching angle-bracket—parenthesis pairs are replaced by values supplied by VISTA/SFC to generate the simulator input deck.

Tool specialization has led to expert tools dealing with isolated aspects of a wafer only. For example, the SAMPLE

[1] A prominent example for a native tool format is TMA's technology interchange format (TIF) used by TSUPREM4 [13]. In VISTA/SFC, new data types can be defined to specify conversion routines and tools.

TABLE I
SIMULATION TOOLS AND PROCESSES FOR PROCESS SIMULATION

Simulator	Process	Symbolic Name
PROMIS Analytical Implant	Ion Implantation	promis-implant
PROMIS Monte Carlo Implant	Ion Implantation	promis-mc-implant
PROMIS Diffuse	Diffusion	promis-diffuse
PROMIS-NT	Diffusion	promis-nt-diffuse
PROMIS Etch	Hemispherical Deposition	promis-hemi-depo
	Isotropic Deposition	promis-iso-depo
	Unidirectional Deposition	promis-uni-depo
	Anisotropic Etching	promis-aniso-etch
	Isotropic Etching	promis-iso-etch
	Reactive Ion Etching	promis-ri-etch
SAMPLE	Conical Deposition	sample-conical-depo
	Dual-Source Deposition	sample-dual-source-depo
	Hemispherical Deposition	sample-hemi-depo
	Isotropic Deposition	sample-iso-depo
	Planetary Deposition	sample-planetary-depo
	Unidirectional Deposition	sample-uni-depo
	Anisotropic Etching	sample-aniso-etch
	Isotropic Etching	sample-iso-etch
SKETCH [a]	Expose	sketch-expose
	Spin-On	sketch-spin-on
	Strip	sketch-strip
SUPREM4	Diffusion	ts4-diffuse
	Ion Implantation	ts4-implant

[a] SKETCH is an auxiliary geometry manipulation tool used for "faking" lithography, deposition and etch processes where no actual numerical simulation of the physical process is required.

[14] and PROMIS [15] etch and deposition modules operate only on the wafer topography, causing inconsistencies between the grid-based dopant distribution data and the geometry boundary information. To ensure a correct, consistent, and concise wafer representation after each process simulation step, regridding operations have to be performed to reflect geometry alterations in the grid structure, to purge superfluous grid elements, or to merge dopant information from before and after a simulator call.

Fig. 3 sketches the VISTA/SFC data flow in the case of a PROMIS etch operation. For physically based simulations, the regridding operation causes an negligible amount of additional CPU time. However, in the case of simple geometrical operations, which are very fast in comparison, regridding times are comparable to simulation time.

The selection of an appropriate gridding tool potentially poses problems similar to the choice of the right simulator. Various approaches exist, with new developments constantly challenging established solutions. VISTA/SFC provides a choice of three grid generators, TRIANGLE [16], TRIGEN [17], and VORONOI [18], for regridding purposes, with a clear interface provided for the integration of additional gridders.

TRIANGLE has proven the most reliable and robust gridder of the three alternatives. It is based on a highly flexible delaunay gridder and takes care of enforcing a consistent wafer state with respect to grid and dopant data, using local grid refinement with respect to dopant concentrations. An enhanced refinement technique delivers rapid variations of grid density and thus minimizes the global number of grid points. As it reuses existing grids, numerical inaccuracy is reduced to a minimum and the creation of additional grid points is restricted to areas where it is necessary.[2]

[2] In a typical multitool process flow simulation scenario, numerous calls to the gridding utility are made to reconcile tool output with pre-existing data. We have found it to be extremely helpful to differentiate between "new" data generated by a simulator, and "old" data that already is the result of a regridding operation.

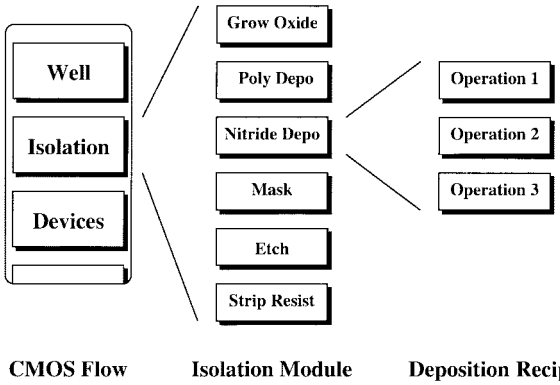

Fig. 5. Hierarchical process flow representation. A recipe consists of a sequence of operations and corresponds to a step in a process module. Process modules are combined to form a process flow.

TABLE II
BASIC FABRICATION PROCESSES

Process	
Deposition	process-depo
Etch	process-etch
Expose	process-expose
Furnace	process-furnace
Ion Implantation	process-ion-implantation
Spin On	process-spin-on

B. Tool Control Level

The control level takes care of all aspects of tool invocation, including the generation of input decks and command lines, the connection of input and output streams, the interpretation of return codes, and the extraction of results from output files. A fundamental problem in TCAD tool integration is caused by the fact that many simulators use complex input deck languages, which are rather difficult to represent on the framework level in a uniform manner. Therefore, VISTA/SFC treat each functional subset of a simulator as a separate task-level *application*. For example, the TSUPREM4 [13] simulator is capable of simulating a number of different fabrication processes, which are specified in an input deck. For each fabrication process, we define an application that appears to the framework as an independent tool.

Applications in VISTA/SFC are either based on external executables or on functions internal to the environment. All operations necessary to invoke an application and to return its output data are encapsulated by a *binding function*, which establishes a uniform interface between the environment and the application.[3] Applications can be defined and added to the framework during sessions without need for interrupting any active tasks. A simple GUI is generated automatically from the definition of the binding function. If complex interdependencies of an application's parameters have to be reflected in the GUI, more sophisticated interfaces can be defined by the user via an object-oriented GUI generator.

[3] Using a standardized VLISP function pointer as interface between the run controller and external tools keeps the interface as flexible as possible, while still allowing for the automatic generation of the function code from a more abstract *tool description* to minimize manual coding efforts. As VLISP does not differentiate between code and data, it is ideally suited for code generation applications.

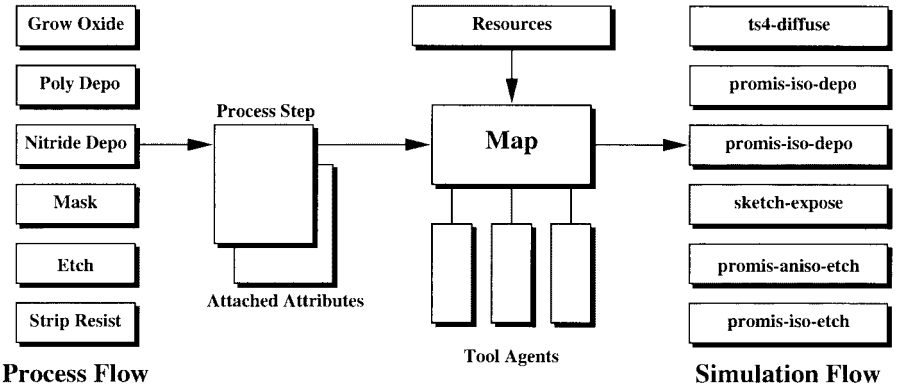

Fig. 6. Mapping from process steps to tool steps: tool agents generate simulator input statements for each step in the process flow.

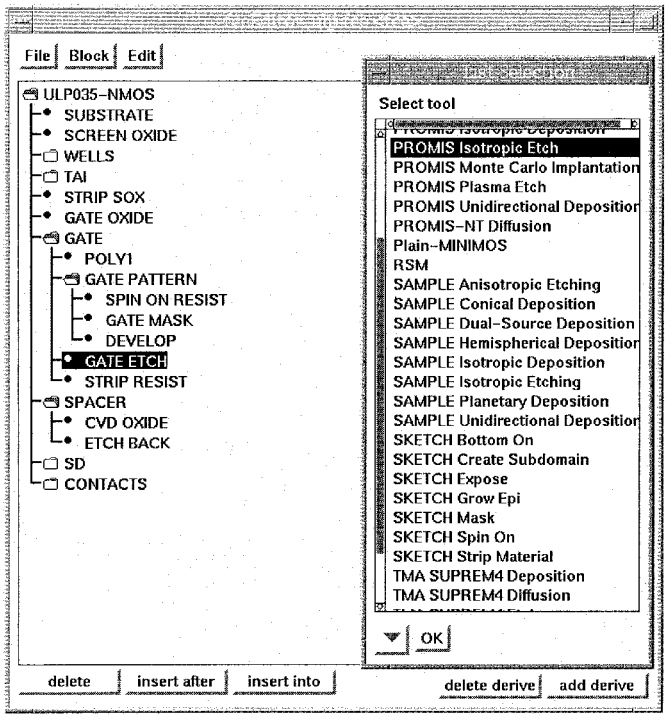

Fig. 7. Visual flow editor for hierarchical flow definition and tool selection.

Specialized simulators often understand a rich input deck syntax, enabling the experienced user to formulate complex analysis tasks. From the process engineer's point of view, however, only a few variables in the input deck may be of interest. In VISTA/SFC, a simulator together with a specific input deck *template* may be defined as an application by itself. The template is derived from an existing input deck by simply marking relevant fields, optionally specifying physical units, value ranges, and arbitrary expressions and function calls to derive a field's value. For interactive editing, an input panel is automatically created from the field definition in the template file (Fig. 4). In this fashion, pre-existing process and device simulation input deck libraries can be easily integrated without rewriting in a new language.

IV. Process Flow Representation

Modern VLSI manufacturing processes consist of several hundred fabrication steps [19], with a wide variety of individual process steps. A standardized representation of process flows forms the basis for process development, process optimization, process centering, and yield improvement, and facilitates process data exchange between TCAD, process development, and fabrication. Attempts to define a standardized vocabulary for representation and interchange of process flow information have met several challenges from the complexity of the physical processes and the variety of equipments and recipes in a rapidly evolving technology [20]–[25]. Some process simulators provide process flow specification mechanisms in their input languages [13], [14], [26]. Tailored to a particular tool's view of semiconductor technology, these process flow descriptions cannot be communicated between different tools. The existence of multiple formats for multiple tools is a significant barrier to the integrated use of these tools [27].

Throughout the semiconductor industry, process flow representations (PFR) of various formats are being used to capture, communicate, and store technology data. Unfortunately, these standards are not shared across different companies, as a lot of implicit information is required to actually translate a design given as process flow and mask data to silicon. These *hidden data* include silent agreements on proprietary procedures, recipes, and etchants, as well as detailed knowledge of the fabrication equipment available in a particular fab.

The VISTA/SFC process flow representation describes process flows as sequences of *simulator-dependent* and *simulator-independent* statements. A set of simulator-dependent statements forms the core layer for calling process and device simulation applications. They provide direct access to simulator parameters and are primarily used during tool development and tool calibration. Table I gives a list of presently integrated simulation tools, respective fabrication processes, and symbolic names.

Each step in a process flow is identified by a name. This name is used to reference split points and output data during simulation. All input parameters as well as generated output data items are accessed by symbolic names for use in process analysis and optimization tasks. The process flow representation supports hierarchical and parameterized process modules that allow the definition of large process flows in terms of predefined blocks (Fig. 5). These blocks can be nested

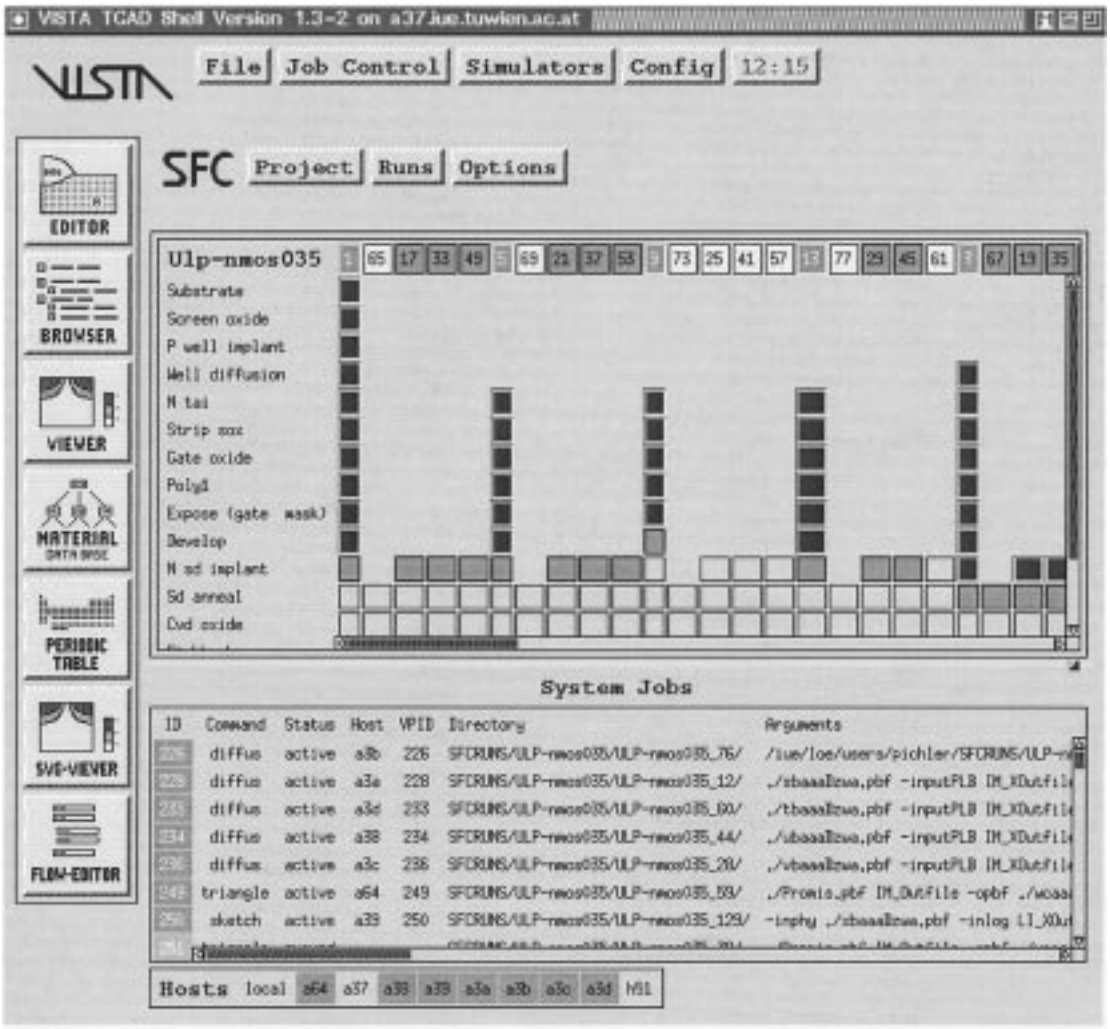

Fig. 8. Visual user interface showing experiment split tree with 93 experiments (24 visible) and monitor window for active and queued system jobs.

to arbitrary levels, reflecting the subdivision of industrial process flows in process modules, process submodules, process steps, and operations.

Process statements in the flow description specify fabrication process steps in a simulator-independent way. They form a technology-focused vocabulary that is used as common representation for TCAD and manufacturing. Table II gives a list of available basic process steps. To accommodate additional processes and to specify process steps in greater detail, the process flow statement vocabulary is extensible by defining new entries. For a given statement, any number of specializations can be added as new statements with certain parameters set to the desired values without need for additional coding.

For example, a *CVD oxide* step is derived from a more general CVD step by permanently assigning the value $Si0_2$ to the *material* parameter.[4] To capture all relevant technology information, each step can carry any number of attributes such as the precise type of equipment or additional parameters not available in the basic process step. By assigning a symbolic name, a new statement can be defined as a specialization of a modified step with additional attributes.

Mapping from process statements to simulator steps is accomplished by *tool agents*, which reconcile equipment and process views with simulator models and input requirements. Fig. 6 shows the basic mechanism for converting a tool-independent process flow description into a simulator-specific flow. Each step can have attributes attached to specify the simulator, and to supply values for simulator-specific parameters for grid-manipulation and calibration purposes.

At each step, project-specific resources are merged with these attributes to yield a complete set of simulator input data. If a sub-sequence of steps can be mapped to the same simulator, these steps are grouped together to form a single simulation step in order to minimize the number of system calls.

Defining and maintaining process flows is supported by an interactive GUI-based flow editor (Fig. 7). In addition to a comprehensive set of basic editing capabilities, advanced features such as including and parameterizing process modules are provided. For programming purposes, an object-oriented interface has been implemented that simplifies creation and

[4] Possible ambiguities in material names are resolved by a material data base [28] that uses an inheritance-based strategy to identify material specifications.

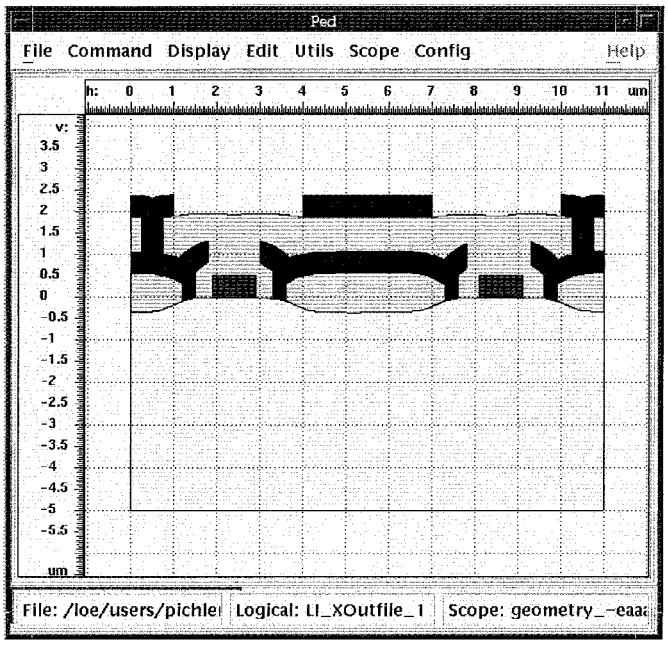

Fig. 9. The final CMOS structure including two metal layers.

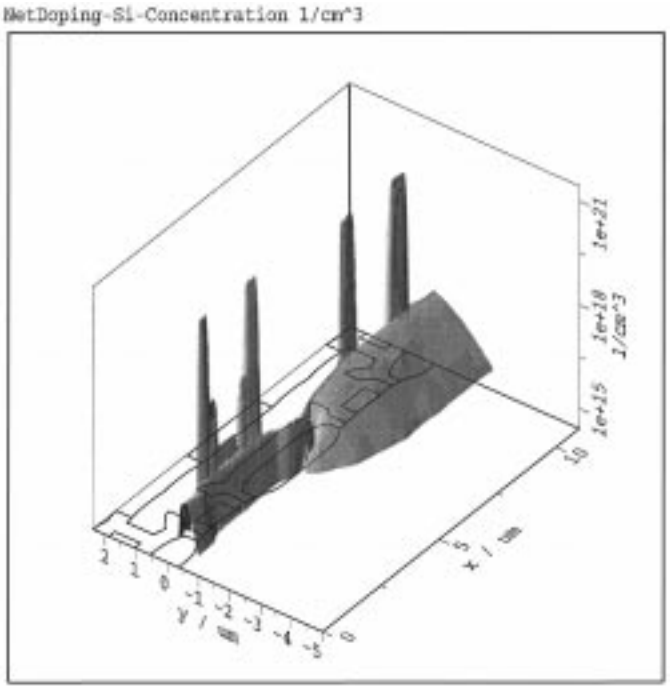

Fig. 10. Net doping distribution in CMOS inverter after simulation of metal 2. The N-device on the left side uses an LDD structure and an anti-punch-through implant.

modification of flow descriptions by user-written applications.[5] It also provides interactive editing functions for plain text terminal operation.

The interface between process simulation and device design is based on lithography mask data that are either specified as numerical coordinates or derived directly from the device layout. Cut-lines and cross sections for defining simulation areas are specified in an integrated layout editor [29]. Lithography masks are specified by name, boolean operations are used to derive a mask from a set of layout layers.

[5]For example, an automatic conversion tool has been implemented, which uses this interface to produce a VISTA flow description from input deck specifications in a NORMAN [4] input file.

V. Run Control

The run control module handles the automatic execution of multi-step process flows. It has been designed to support different stages of TCAD process development, from tool development and process design to automatic scheduling and parallelization of split-lot experiments. Great emphasis has been put on aiding in process flow debugging during early design stages. *Single-step*, *stop-and-resume*, and *rework* modes efficiently support the exploration of the design space.

Process flows are submitted for simulation either from the visual user interface or from a batch file. All user interactions are fully available during active simulations. Moreover, process flow simulations can also be requested by other applications in a client-server manner, e.g., a number of process simulation runs are started by a design-of-experiments module or an external optimizer, which are notified upon completion to read the generated wafer data and extracted parameters for further processing.

Fig. 8 shows the visual user interface during parallel execution of a large split-lot experiment on a UNIX workstation cluster. Ninety-three experiments were generated with a full-factorial center circumscribed (CCC) design for seven process parameters of a 0.35-μm ultra low power CMOS process [30]. The upper part of the screen displays the experiment split tree generated by the CCC design.

The leftmost column contains a list with process step names. Runs appear from left to right, the step sequence flows from top to bottom. Split-branches appear to the right of their parents. In this example, runs 5, 9, 13, and 3 split from run 1, runs 69, 21, 37, and 53 split from run 5. No rigid split-tree exists, but split-points are generated dynamically by searching for the best match among existing runs. Colors are used to clearly mark the state of runs and steps. At each run entry, a pop-up menu is available to directly access run information, to remove run data, and to stop and resume execution. Post-processors are launched automatically to display run output.

VISTA/SFC supports the simultaneous execution of multiple projects. The user uses the *Project* menu to select the project to display in the main window. All other active projects are disconnected from the GUI and continue to run in the background.

VI. Applications

In the following section, the application of the VISTA/SFC environment to the modeling and analysis of VLSI fabrication processes is demonstrated by the simulation of a complete standard silicon-gate LOCOS CMOS process [31] using a heterogeneous set of simulation tools. Based on this capability, we demonstrate how client applications on the task level can be defined.

TABLE III
CMOS PROCESS SEQUENCE USING 55 SIMULATION TOOL STEPS

Module	Mask	Process Step		Tool
Substrate			(100)-oriented, p-doped silicon (ρ = 10mΩcm) with a 12μm epi layer (ρ = 10Ωcm)	SKETCH
N-Tub	NW	Lithography		SKETCH
		N-Tub Etch		PROMIS
		Strip Resist		SKETCH
		Screening Oxide	$t_{ox} = 50nm$	PROMIS
		N-Tub Implant	Phosphorus (150keV,$5 \cdot 10^{12}$cm^{-2})	PROMIS
		N-Tub Diffusion	1150°, 150min, N_2	PROMIS
		Strip Oxide		SKETCH
Nitride		Oxide	O_2, $t_{ox} = 25nm$	TSUPREM4
		Nitride deposition	$t_{nitride} = 150nm$	PROMIS
	OD	Lithography		SKETCH
		Nitride Etch		PROMIS
		Strip Resist		SKETCH
Channel Stop	NWI	Lithography		SKETCH
		Oxide Etch		PROMIS
		Channel Stop Implant	Boron (16keV, $5 \cdot 10^{13}$cm^{-2})	PROMIS
Anti Punch-Through		Anti-Punch-Through Implant	Boron (180keV, $8 \cdot 10^{11}$cm^{-2})	PROMIS
		Strip Resist		SKETCH
Field Oxide		Field Oxidation	960°, H_2, O_2, $t_{ox} = 850nm$	TSUPREM4
		Etchback	$t_{ox} = 500$nm	PROMIS
Gate		Gate Oxidation	$t_{ox} = 25$nm	TSUPREM4
		Screening Poly	$t_{poly} = 100nm$	PROMIS
		Threshold Adjust Implant	Boron (45keV, $9 \cdot 10^{11}$cm^{-2})	PROMIS
		Gate Poly	$t_{poly} = 400$nm	PROMIS
	PS	Lithography		SKETCH
		Poly Etch		PROMIS
		Strip Resist		SKETCH
N-S/D	SN	Lithography		SKETCH
		N-LDD Implant	Phosphorus (100keV, $2 \cdot 10^{13}$cm^{-2})	promis
Spacer		TEOS Oxide	$t_{ox} = 300$nm TEOS-oxide	PROMIS
		Anisotropic Etch		PROMIS
		Screening Oxide	$t_{ox} = 30nm$	PROMIS
	SN	Lithography		SKETCH
		N Source/Drain-Implant	Arsenic (100keV, $5 \cdot 10^{15}$cm^{-2})	PROMIS
		Strip Resist		SKETCH
P-S/D	SP	Lithography		SKETCH
		P Source/Drain-Implant	Boron (30keV, $3 \cdot 10^{15}$cm^{-2})	PROMIS
Contacts		TEOS oxide	($t_{ox} = 800$nm)	PROMIS
		Reflow	900°, H_2, O_2	TSUPREM4
	CO	Lithography		SKETCH
		Oxide Etch		PROMIS
		Strip Resist		SKETCH
Metal 1		Al Deposition		PROMIS
	IN	Lithography		SKETCH
		Al Etch		SKETCH
Planarize		Plasma Oxide	$t_{ox} = 1.2\mu m$	PROMIS
		Resist Spin-On		SKETCH
		Etchback		PROMIS
Vias	COS	Lithography		SKETCH
		Oxide Etch		PROMIS
		Strip Resist		SKETCH
Metal 2		Al Deposition		PROMIS
	INS	Lithography		SKETCH
		Al Etch		PROMIS
		Strip Resist		SKETCH

A. LOCOS CMOS Process

The process flow contains all types of fabrication process steps occurring in modern VLSI technology and uses eleven lithography masks. Simulation was carried out up to the second metallization layer, using 55 tool steps. The simulated structure is a CMOS inverter, containing most of the relevant intrinsic and parasitic devices. The final topography is shown in Fig. 9.

To achieve accurate and realistic modeling of etching and deposition processes, the PROMIS ETCH module [15], [32] was used. It is based on extremely stable cell-based algorithms and contains a number of physically sound models for a variety of topography processes. Lithography steps are performed by the SKETCH tool, which operates as a simple geometry manipulator and provides mask pattern transfer and material strip capabilities where no accurate simulation of optical phenomena is needed. All implantation steps have been carried out with the PROMIS implantation module [26]. Diffusion has been realized with the PROMIS-NT diffusion module [33],

[34], which offers the possibility of solving different models on each material segment and therefore greatly facilitates the investigation of advanced effects in state-of-the-art devices. Oxidation and diffusion in reactive environment have been carried out with TSUPREM4 [13].

An overview of the process is given in Table III, a detailed description can be found in [31, p. 370]. Fig. 10 shows the resulting net doping concentration in both the N and P devices. Despite the interaction of fundamentally different simulation tools and the large number of simulation steps, fully automatic simulation has been performed without the necessity for user interaction at any point of the simulation.

Using the framework's built-in dynamic load balancing mechanism on a DEC 3000 600 workstation cluster, the total computation time for all process simulation and regrid operations amounts to less than 17 h of real time, 35 min were spent on regridding. 3.3% of the total time are used for file I/O. The overhead caused by regridding and writing of intermediate results amounts to approximately 7%.

Fig. 11 shows a detail of the resulting triangular grid after the last process step. 13 000 grid nodes are used to accurately resolve arsenic, boron, and phosphorus distributions generated by PROMIS, PROMIS-NT, and TSUPREM4. All grid operations, from the generation of the initial grid to merging and updating grid data at various points in the process flow, have been carried out by TRIANGLE.

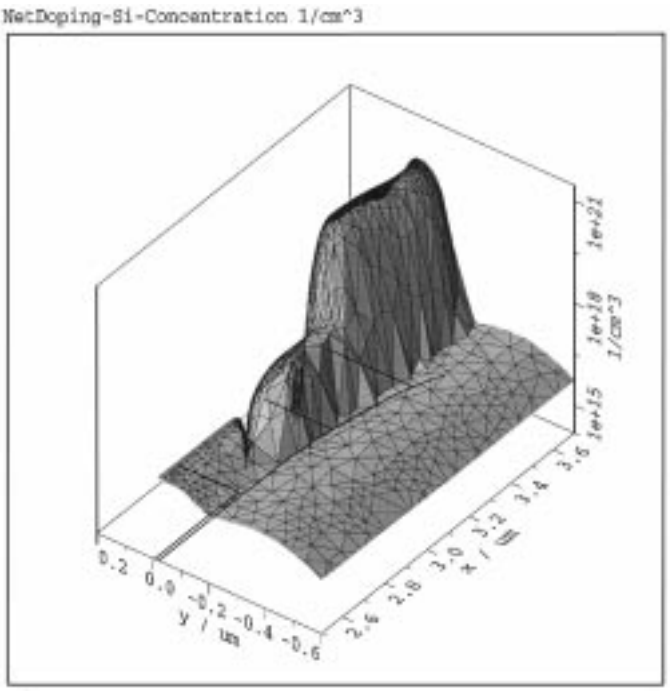

Fig. 11. Grid detail of N-device spacer area with LDD, channel and anti-punch-through implants after simulation of metal 2.

B. Task-Level Applications

On the task level, VISTA/SFC represents the entire process flow simulation as an object. This object can be used to solve more complex problems. The following lines of code give a glimpse of possible applications.

To assess the short-channel effect in the NMOS device, the statement below loops over a list of gate lengths and runs the current process flow model for each value of `lg`.

```
(dolist lg (0.3 0.4 0.5 0.75 1.0 1.5)
  (Run - Process : gate - length lg)).
```

To create a response-surface model describing the dependence of the threshold voltage on the halo implant parameters, two task-level commands are used.[6] First, a single statement is evaluated to run the necessary simulations using the central composite inscribed (CCI) design for the dose, energy, and angle parameters of the halo implant:

```
(Run - Doe CCI : parameters
  '(halo - dose halo - energy halo - angle)
    : range'((halo - dose 1e12 1e13)
     (halo - energy 15. 25.)
     (halo - angle 15. 45.))).
```

Subsequently, the RSM is extracted using the `Create-RSM` statement. To add an additional response for optimization to the RSM,[7] an expression is defined on the response surface:

```
(Define - Response 'halo - target
                  : expression '(and vth (abs
                  (/ ib vth)))).
```

In this fashion, arbitrarily complex applications can be created, combining the simulation capabilities of the run-control module and the full range of expressions of the extension language. Graphical user interfaces to assist in the interactive generation and visualization of RSM's and are under development.

VII. Conclusion

The VISTA/SFC environment offers powerful support for various TCAD tasks that require a smooth cooperation of a multitude of heterogeneous simulators and applications. Its scope spans a broad range of TCAD aspects from data level integration and tool control to process flow representation and simulation data management. Special emphasis is put on the fact that *openness* toward the integration of arbitrary simulation tools is a key issue in the design of VISTA/SFC. This includes straight-forward definition of tool bindings as well as the capability to handle different data formats for wafer representation. All functionality is not only available via comfortable and problem-oriented graphical user interface, but is also accessible in batch-mode operation. Therefore, large-scale experiments and optimizations can be run as detached processes somewhere on a remote machine or across phone

[6] For the sake of conciseness, some minor details have been omitted in the example; see [9] for a more extensive discussion.

[7] Note that response-surface models are represented in the same way as process simulation tasks. VISTA/SFC does not differentiate between the evaluation of a response surface model, the evaluation of a simulation task, or the evaluation of any other user-defined application.

lines without need for maintaining an interactive session. In addition to built-in capabilities for split-lot experiments and automatic parallelization of computation tasks on workstation clusters, a clear high-level programming interface is available for defining more complex TCAD applications, using simulation services provided by VISTA/SFC. The increasing degree of autonomous operation thus achieved liberates the user from attending the TCAD tools and makes human time and energy available for more creative tasks.

REFERENCES

[1] H. Masuda, "T^{2}CAD: Total design for sub-μm process and device optimization with technology-CAD," in *Simulation of Semiconductor Devices and Processes.*, H. Ryssel and P. Pichler, Eds. Vienna, Austria: Springer, 1995, vol. 6, pp. 408–415.

[2] Silvaco International, Santa Clara, CA, *VWF Interactive Tools, User's Manual,* 2nd ed., 1995.

[3] Technology Modeling Associates, Inc., Sunnyvale, CA, *TMA Work-Bench IC Technology Design Environment, Version 2.2, User's Manual,* 1997.

[4] R. Cartuyvels, R. Booth, S. Kubicek, L. Dupas, and K. M. 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. Vienna, Austria: Springer, 1993, vol. 5, pp. 29–32.

[5] D. S. Boning and P. K. 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, pp. 233–244, May 1994.

[6] F. Fasching, W. Tuppa, and S. Selberherr, "VISTA—The data level," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 72–81, Jan. 1994.

[7] S. Halama, Ch. Pichler, G. Rieger, G. Schrom, T. Simlinger, and S. Selberherr, "VISTA—User interface, task level, and tool integration," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 1208–1222, Oct. 1995.

[8] Ch. Pichler and S. Selberherr, "Process flow representation within the VISTA framework," in *Simulation of Semiconductor Devices and Processes.*, S. Selberherr, H. Stippel, and E. Strasser, Eds. Vienna, Austria: Springer, 1993, vol. 5, pp. 25–28.

[9] Ch. Pichler, "Integrated semiconductor technology analysts," dissertation, Technische Universität Vienna, Austria, 1997. http://www.iue.tuwien.ac.at/diss/pichler/diss/diss.html.

[10] Institut für Mikroelektronik, Technische Universität Vienna, Austria, *VISTA Documentation 1.3-1, VLISP Manual*, 1996.

[11] F. Fasching, C. Fischer, S. Selberherr, H. Stippel, W. Tuppa, and H. Read, "A PIF implementation for TCAD purposes," in *Simulation of Semiconductor Devices and Processes*, W. Fichtner and D. Aemmer, Eds. Konstanz, Germany: Hartung-Gorre, 1991, vol. 4, pp. 477–482.

[12] S. G. Duvall, "An interchange format for process and device simulation," *IEEE Trans. Computer-Aided Design*, vol. 7, pp. 741–754, July 1988.

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

[14] Electronics Research Laboratory, Univ. California, Berkeley, *SAMPLE 1.8a User Guide*, 1991.

[15] E. Strasser and S. Selberherr, "Algorithms and models for cellular based topography simulation," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 1104–1114, Sept. 1995.

[16] J. R. Shewchuk, "Triangle: Engineering a 2D quality mesh generator and delaunay triangulator," in *First ACM Workshop on Applied Computational Geometry*, 1996.

[17] R. E. Bank, "PLTMG: A software package for solving elliptic partial differential equations," in *Frontiers in Applied Mathematics*, Users' Guide 6.0, vol. 7, SIAM, Philadelphia, PA, 1990.

[18] S. Halama, "The Viennese integrated system for technology CAD applications-architecture and critical software components," dissertation, Technische Universität Vienna, 1994. [Online] http://www.iue.tuwien.ac.at/diss/halama/diss/diss.html.

[19] D. Widmann, H. Mader, and H. Friedrich, "Technologie hochintegrierter Schaltungen," in *Halbleiter-Elektronik.* Berlin, Germany: Springer, 1996, vol. 19.

[20] Ch. J. Hegarty, L. A. Rowe, and Ch. B. Williams, "The Berkeley process-flow language WIP system," Tech. Rep. UCB/ERL M90/77, Univ. California, Berkeley, 1990.

[21] J. S. Wenstrand, "An object-oriented model for specification, simulation, and design of semiconductor fabrication processes," Integrated Circuits Lab., Stanford Univ., Stanford, CA, 1991, Tech. Rep. ICL91-003.

[22] D. S. Boning, M. B. McIlrath, P. Penfield, Jr., and E. M. Sachs, "A general semiconductor process modeling framework," *IEEE Trans. Semiconduct. Manufact.*, vol. 5, pp. 266–280, Nov. 1992.

[23] D. Durbeck, J.-H. Chern, and D. S. Boning, "A system for semiconductor process specification," *IEEE Trans. Semiconduct. Manufact.*, vol. 6, pp. 297–305, Nov. 1993.

[24] A. S. Wong, "Technology computer-aided design frameworks and the PROSE implementation," Ph.D. dissertation, Univ. California, Berkeley, 1992.

[25] P. R. Kristoff and D. P. Nunn, "The process specification system for MMST," *IEEE Trans. Semiconduct. Manufact.*, vol. 8, pp. 262–271, AUg. 1995.

[26] G. Hobler, P. Pichler, and K. Wimmer, *PROMIS 1.5 User Guide*, Institut für Mikroelektronik, Technische Univ. Vienna, Austria, 1991.

[27] TCAD Framework Group and SPR Working Group, Semiconductor Process Representation Requirements VO.4-1, CAD Framework Initiative, Austin, TX, 1992, Tech. Rep. CFI Doc. TCAD-91-G-4.

[28] Institute for Microelectronics, Technical Univ. Vienna, *VISTA Documentation 1.3-2, PIF Application Interface Manual*, 1996.

[29] R. Martins and S. Selberherr, "Layout data in TCAD frameworks," in *Modeling and Simulation*, Soc. Comput. Simul. Int., 1996, pp. 1122–1126.

[30] G. Schrom, D. Liu, Ch. Pichler, Ch. Svensson, and S. Selberherr, "Analysis of ultra-low-power CMOS with process and device simulation," in *24th European Solid State Device Research Conference—ESSDERC'94*, C. Hill and P. Ashburn, Eds., Gif-sur-Yvette Cedex, France: Editions Frontieres, 1994, pp. 679–682.

[31] G. Schumicki and P. Seegebrecht, *Prozeßtechnologie.* Berlin, Germany: Springer, 1991.

[32] E. Strasser and S. Selberherr, "A general simulation method for etching and deposition processes," in *Simulation of Semiconductor Devices and Processes.*, S. Selberherr, H. Stippel, and E. Strasser, Eds. Vienna, Austria: Springer, 1993, vol. 5, pp. 357–360.

[33] H. Puchner and S. Selberherr, "An advanced model for dopant diffusion in polysilicon," *IEEE Trans. Electron Devices*, vol. 42, pp. 1750–1755, Nov. 1995.

[34] H. Puchner, "Advanced process modeling for VLSI technology," dissertation, Technische Universität Vienna, Austria, 1996. [Online] http://www.iue.tuwien.ac.at/diss/puchner/diss/diss.html.

[35] H. Ryssel and P. Pichler, Eds., *Simulation of Semiconductor Devices and Processes.* Vienna, Austria: Springer, 1995, vol. 6.

[36] S. Selberherr, H. Stippel, and E. Strasser, Eds., *Simulation of Semiconductor Devices and Processes.* Vienna, Austria: Springer, 1993, vol. 5.

Christoph M. Pichler (S'93–M'97) was born in Vienna, Austria, in 1966. He received the Dipl.-Ing. degree in control theory and industrial electronics from the Technical University of Vienna in 1991.

During the time of his studies, he held several summer trainee positions at electrical engineering companies in Austria and in Switzerland. In November 1991, he joined the Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA, where he worked on rapid prototyping and automation in manufacturing. He joined the Institut für Mikroelektronik, Technical University of Vienna in November 1992, pursuing a doctoral degree. From January to May 1995, he held a research position at the University of California, Berkeley, working on systems aspects of three-dimensional lithography simulation. In October and November 1995, he was with the Hitachi Device Development Center, Tokyo, working on automatic process model calibration and process optimization. In 1997, he joined National Semiconductor Corporation, Santa Clara, CA. His scientific interests include engineering design, semiconductor technology, and systems integration.

Dr. Pichler is a member of the Österreichische Verein für Elektrotechnik (ÖVE).

Richard Plasun was born in Vienna, Austria, in 1969. He studied electrical engineering at the Technical University of Vienna, where he received the Dipl.-Ing. degree in 1994. He joined the Institut für Mikroelektronik, Technical University of Vienna, in March 1995, and is currently pursuing the doctoral degree.

His scientific interests include linear and nonlinear optimization, TCAD framework aspects and software technology. From October to December 1996, he was with the National Semiconductor Corporation, Santa Clara, CA, where he concentrated on tool binding and process optimization aspects.

Rudolf Strasser was born in Ried im Innkreis, Austria, in 1970. He studied electrical engineering at the Technical University of Vienna, where he received the Dipl.-Ing. degree in 1995.

From spring 1992 to autumn 1993, he held a research position at the campus-based Engineering Center of Digital Equipment Corporation, Vienna, Austria. He joined the Institut für Mikroelektronik, Technical University of Vienna, in April 1995. In summer 1996, he was with the Advanced Products Research and Development Laboratory, Motorola, Austin, TX. He is currently working for his doctoral degree. His scientific interests include semiconductor technology, grid generation, and software engineering.

Siegfried Selberherr (M'79–SM'84–F'93) was born in 1955. He received the Dipl.-Ing. degree in control theory and industrial electronics from the Technical University of Vienna, Austria, in 1978. He received the doctorate degree on "Two Dimensional MOS-Transistor Modeling" in 1981 from the Technical University of Vienna.

He has been holding the "venia docendi" on "Computer-Aided Design" since 1984 as Professor. He is the head of the Institut für Mikroelektronik, Technical University of Vienna, since 1988. In 1994, he was appointed head of the "Computing Services Center" of the Technical University of Vienna. His current research topics are modeling and simulation of problems for microelectronics engineering. He has authored and coauthored more than 250 publications in journals and conference proceedings. He also authored the book *Analysis and Simulation of Semiconductor Devices*.

Dr. Selberherr received the "Dr. Ernst Fehrer" Award in 1983. In 1985, he received the award of the "Nachrichtentechnische Gesellschaf." In 1986, he was honored with the "Dr. Herta Firnberg Staatspreis," and in 1987, he received the "Heinz Zemanek" Award. In 1993, he was elected Fellow of the IEEE for *pioneering work in numerical analysis of semiconductor devices and their processes*. In 1994, he was honored with the "Wilhelm-Exner-Medal." He is a member of the Association for Computing Machinery, the Society of Industrial and Applied Mathematics, and the Verband deutscher Elektrotechniker. He is Editor of The Transactions of the Society for Computer Simulation (1983–present), of *Electrosoft* (1986–1991), of *Mikroelektronik* (1988–present), and of the Springer-Verlag book series *Computational Microelectronics* (1985–). He serves on the "Programme Advisory Board" of *Electron Technology* (1994–present).