Linking TCAD to EDA — Benefits and Issues

G. Chin [•] W. Dietrich, Jr.[*] D. Boning [†] A. Wong [‡] A. Neureuther [‡] R. Dutton [•]

[•] Stanford University, Stanford, CA [*] IBM Research Center, Yorktown Heights, NY
[†] Texas Instruments, Dallas, TX [‡] University of California at Berkeley, Berkeley, CA

Abstract

This paper shows the benefits of a tighter coupling between Technology CAD and Electronic Design Automation (EDA) for the design of high speed digital and analog circuits, for the design of more manufacturable systems with improved reliability, and for future technology and circuit development. Requirements for incorporating TCAD and EDA into a framework are presented.

1 Introduction

Although chip and technology design are intimately related, they have traditionally been separated at the circuit level. As a result, Computer-Aided Design has been partitioned into Electronic Design Automation (EDA) and Technology CAD (TCAD). Technology's role is becoming more important in an overall electronic design because of its effect on system performance and manufacturability (e.g. yield). To give TCAD's benefits to the circuit designer, EDA's boundaries must be expanded to include TCAD.

This paper describes the capabilities and benefits that TCAD can bring to circuit and chip design. The role of TCAD in device and process design is considered elsewhere[1]. Examples in Section 2 show how TCAD can improve traditional design automation for chips. This includes providing essential information for the design of high-performance chips that are reliable and economically manufacturable. Section 3 presents key issues in the integration of TCAD and EDA frameworks. Two key requirements unique to TCAD and necessary for TCAD frameworks are new design representations for wafers and fabrication processes. An overview of a new approach to wafer representation based on an object oriented approach is presented and the fabrication process representation used in the MIT CAFE system is described. Finally, progress towards TCAD standards is summarized in Section 4.

2 Using TCAD in the Design Process

Using TCAD in the design process can improve chip performance, reduce design cycle time, and improve reliability and manufacturability. In all of these areas, TCAD's goal is to provide circuit designers with more accurate information early in the design process.

2.1 Designing for High Performance

To meet aggressive design goals, high speed circuit design relies heavily on accurate simulation, requiring precise models in two areas: parasitics and devices.

In today's complex chips, interconnect delay is a major contributor to parasitic delay and approaches the significance of intrinsic device delay[2]. A great deal of EDA research has focused on efficient and accurate methods of extracting capacitance values from layout using finite-element analysis [3]. By simulating fabrication, TCAD provides topography information, which is not available from layout alone, but is necessary to support interconnect simulation[4].

TCAD can also be used to provide more physically-based information on device behavior, necessary for accurate modeling of small geometry transistors. Typical circuit simulation models inaccurately predict the transient response of digital gates [5] and the frequency response of analog circuits [6] because simple analytic models ignore many important effects such as high-level injection. Two approaches to improve circuit modeling are the use of subcircuits to describe a single transistor and the development of more elaborate models based on device physics. Both of these approaches rely heavily on the use of TCAD (device simulation) to describe device behavior at locations other than the external electrodes. Knepper[7] illustrates the former approach by constructing subcircuit models to model single devices. A complex distributed model that uses many internal nodes to describe circuit behavior is automatically generated based on information from device simulation. The distributed model includes behavior of parasitic elements not usually included in traditional circuit modeling. For more efficient use in

©1991 ACM 0-89791-395-7/91/0006/0573 $1.50

circuit simulation, the distributed model is reduced onto a simpler predefined circuit topology. Marash[8] demonstrated the second approach by simplifying the complex device equations into an analytic form. Device simulation was necessary to verify assumptions made in both of these approaches.

2.2 Reduced Design Cycle Time

Products such as ASIC require short design cycle times and depend on CAD tools to achieve this goal. Technology development cycles, however, tend to be long. In the absence of real wafers, designers must use simulation to deduce key device parameters. While the ability to extract model parameters has previously existed [9], use of these tools has been limited to technologists and TCAD experts, due to poor user interfaces and the need for explicit knowledge of device and process simulation.

Figure 1 introduces a new, intuitive approach to the problem. This integrated system is an extension of SIMPL-IPX[10]. Using a process "recipe", the system runs a series of simulations to generate the doping profiles and topographic features that describe the device. It extracts the device structure from the cross section and numbers electrodes based on contact mask locations in the layout. The user can specify the circuit-driving biases and displayed output variables using a front panel similar to the HP4145's. Grid generation, physical model selection, and solution method selection are automatically performed by the system. The system can generate a set of model parameters for circuit simulation by running a parameter extraction program.

TCAD tools also provide the ability to generate design rules earlier. For example, when using titanium silicide technology for local interconnect in high-density circuits, one must allow adequate spacing between NMOS and PMOS transistors to prevent threshold degradation due to lateral diffusion of dopants out of the P^+ gate material counter-doping the N^+ gate. Coupled process and device simulation [11] show that the minimum P^+ poly to N^+ poly spacing for thermal processing at 900C is $2.0\mu m$. Typical spacings for high density circuits may be on the order of $1.5\mu m$, resulting in device failures and a subsequent change in the design rules late in the design process. To understand the density-degradation trade-offs, circuit designers can experiment with transistor placement using TCAD process and device simulation.

2.3 Design for Reliability

Coupling of circuit design and technology also manifests itself in terms of chip reliability. Circuit simulation alone cannot simulate effects such as electromigration, ESD (electro-static discharge), hot electron effects, oxide breakdown, and latchup without information on device structure. TCAD provides the ability to predict reliability associated with a particular design *before* fabrication. Circuits can be checked for effects such as ESD, alpha particle, and latch-up beforehand to determine if additional precautions are necessary in the layout.

Layout can have a dramatic effect on the reliability of a standard cell. Three-dimensional device simulation has been used to investigate the effect of a butting contact on the amount of current necessary to induce latchup (trigger current) in a standard cell NAND gate[12]. Using a butting contact to the supply instead of a conventional well-contact increases the trigger current by a factor of 2.5. Qualitatively, the latchup mechanism can be explained using a circuit model comprised of *parasitic* elements that are not normally considered by conventional EDA extraction software. However, *quantitative* analysis of latchup using a simple circuit model is extremely difficult due to the problems of determining model parameters for the bipolar transistors and the resistors. The resistance of the resistors is current dependent and isolation of these *parasitic* bipolar transistors for characterization is extremely difficult unless TCAD tools are used.

2.4 Design for Manufacturability

Even when a technology that provides adequate performance has been designed, wafers are subject to variations in manufacturing. Because of this, circuit designers currently use worst-case circuit models and layout design rules. Instead, statistical simulation[13] can quickly determine the sensitivity of circuit parameters to processing parameters. Quantifying the effect of process sensitivities on design parameters gives a better estimation of circuit yield at a particular performance level. It also gives the designer more confidence in model accuracy, allowing him or her to trade yield for performance.

TCAD may also enable circuit designers to investigate the feasibility and impact of process modifications to increase the performance or capability of a technology. Consider adding a poly/diffusion capacitor with low voltage coefficient ($\frac{1}{C}\frac{dC}{dV}$) to an existing digital CMOS process to support analog circuits such as amplifiers. The CAFE[14] system provides the designer with a description of the existing baseline process and a set of manufacturing constraints associated with the process (using the process representation described in Section 3.2) — allowing for both process enhancement (adding or modifying steps in the process sequence), and optimization (adjusting parameters of the process step for performance/yield). Knowing the sensitivity of the capacitor circuit parameters (capacitance and voltage coefficient) with respect to capacitor implant parameters aids in generation of a manufacturable extension to the existing process.

3 Integrating TCAD into EDA Frameworks

The previous examples have shown that access to information provided by TCAD tools (particularly through improved tool integration and user interfaces) can facilitate the design process. Limited use of TCAD data for design is sometimes due to the organizational separation between the fabrication facilities and design centers. This "real-life" barrier is manifested in the CAD world by specifying the interface between TCAD and EDA as lumped values such as circuit model parameters and design rules. EDA Framework-based design allows tools to access data from many levels of abstraction (including layout, schematic, netlist, etc.) through well defined interfaces — giving the tool freedom to determine necessary input from all aspects of the design process. Extending EDA frameworks to include TCAD is essential for integrating TCAD tools into the design process.

TCAD frameworks must address both deployment (integrating existing tools) and development (creating new tools). The use of common representations address both areas by minimizing the number of translators needed from $(O(n^2))$ to $(O(n))$. It also encourages the creation of shared libraries of functions and services (e.g. I/O routines). By using these library routines within a framework context, developers can focus on the development of the physically-based models that provide the intrinsic benefit of TCAD to EDA. Two representations will be presented — one to describe wafer state and one to describe fabrication process flow.

3.1 Semiconductor Wafer Representation

Wafer state describes the structures resulting from the fabrication and simulation of integrated circuits. Examples of wafer state information include topography, dopant profiles, and current density. This wafer state is created and modified via processing steps (e.g. deposition and implantation), and analyzed by device/circuit simulation.

This section describes the representation that the CFI TCAD Framework Group is developing[15]. Although the group has not finalized a representation, a conceptual view can be given. The Semiconductor Wafer Representation (SWR) is based on an object oriented approach. Tools query and modify the representation using a functional interface. The representation can be partitioned into two major components — Geometry and Fields. Geometry provides structural and topological information about the wafer while Fields provide information about properties that vary within geometric regions, such as dopant concentration.

The basic unit of Geometry is the cell, a collection of zero-, one-, two-, and three-dimensional point sets. Operations on cells include create, destroy, query, and modification. Cells can be created by combining primitive cells (e.g. unioning rectangular or triangular cells) or by combining lower-dimensional objects (e.g. constructing a face from a set of pair-wise adjacent edges). Because adjacency information must be readily available in TCAD simulators, sets of nonoverlapping cells can be grouped into special sets called "cell complexes". Within complexes, most queries about cell relationships (e.g. adjacency) can be answered quickly and accurately. Queries include classification (e.g. is this point inside this cell complex) and sectioning (e.g. return a 1D slice out of a 2D cell). Modification functions include inset (for deposition), and subtraction (for etching).

Fields represent mappings from a domain, usually a cell associated with the geometry, to a range. A specialization of field is "field on mesh". Numerical methods for solving equations that do not possess closed form solutions typically use meshes. Facilities are provided for automatically creating and refining several types of commonly used meshes. Meshes can also be constructed from lower-dimensional elements in a manner similar to the creation of geometry cells. An important operation is evaluation of a particular field at a point in space (e.g. find the boron concentration at this location). Modifications of fields usually result from application specific numerical operations (e.g. solve the diffusion equation).

Inconsistency between fields and geometry is allowed to enable client applications to define their own set of consistency functions. For example, a string-based etching simulator can modify the Geometry but has little knowledge of which algorithm should be used to force the existing "fields on mesh" to follow the new boundary. A subsequent diffusion client, however, could make the old mesh consistent with the new boundary in a way that minimizes discretization error.

The following program segments shows how a tool simulating the photolithography process would manipulate the wafer representation through the functional interface. Figure 3 shows snapshots of the wafer between function calls.

```
/* retrieve the existing wafer (Fig. 3a).
 */
cell_complex1 = getCellComplex(wafer);

/* deposit_layer uses the geometry
 *  component to create a cell based
 *  on the photoresist deposition
 *  process.  A reference to the new
 *  cell is returned (Fig. 3b).
 */
dcell1 = deposit_layer(cell_complex1);

/* incorporate the cell into the
 *  existing wafer (Fig. 3c).
 */
wafer.insetCell(dcell1);

/* set the material of the cell
 *  to photoresist.
 */
dcell1.setMaterial(Resist);
```

Note that the tool developer needs only to provide code for deposit_layer, since the additional functions necessary to access the wafer boundary and create the new geometry are provided through the programming interface.

Specialized functions such as insetCell are extremely complex and should be written by computational geometry experts. The next code sequence continues processing by having another tool perform photolithography on the deposited photoresist.

```
/* createMeshedField creates a new "field
 *  on mesh" associated with a particular
 *  cell.  The cell is automatically meshed
 *  if no tensor mesh is present (Fig. 3d).
 *  Hints is a data object containing
 *  information for the meshing algorithm,
 *  such as desired mesh density.
 *  A reference to the field is returned.
 */
exposure =
    dcell1.createMeshedField(TENSOR, hints);

/* the photolithography client takes
 *  information provided by the cell complex
 *  and fields, determines exposure
 *  intensity as a function of position,
 *  and puts values in the field  (Fig. 3e).
 */
perform_lithography(exposure, wafer);
```

Again, the tool developer is only required to provide code for the **perform_lithography** routine. Mesh generation and management of field information are provided through the programming interface. ''CreateMeshedField'' replaces a long sequence of steps that developers traditionally used to update wafer state. One of the more important aspects of the createMeshedField routine is the ability to automatically mesh a cell, an operation that requires knowledge of both computational geometry and numerical analysis. Meshers are difficult to write and should be provided to the typical developer.

As can be seen, the SWR interface makes the manipulaiton of wafer state transparent to the tool developer. In the past, each modeller typically had its own implementation of the operations it needed, instead of reusing existing code. Because many programs can use the same implementation of the SWR, SWR providers can concentrate on improved programs for these core services. This frees SWR users to concentrate on the physics being modeled.

3.2 Semiconductor Process Representation

The sequence of steps needed to manufacture a chip is specified by a fabrication *process*. The manufacturing process is thus a crucial last link in the spectrum of abstraction levels and data necessary for electronic design and analysis. The manufacturing process representation is needed to support TCAD tools and activities in a TCAD environment (e.g. process simulation, synthesis, optimization, and diagnosis), and to achieve any integration between design and manufacturing. The process representation, however, poses a number of problems.

First, the information about a semiconductor fabrication process has traditionally been widely varied and often unstructured (including input files for specific process simulators, printed run sheets, recipe manuals, and operator instructions for specific equipment. Second, this information is often incomplete and incompatible for different uses. The transfer of a process from design to manufacturing, or even from one fabrication facility to another, is complicated by the loss and necessary manual translation of design data. The analysis of processes during fabrication (e.g. for process diagnosis or yield improvement) is complicated because the description of the process used in manufacturing is different than that required by simulators or other tools.

The approach used in the MIT Process Flow Representation (PFR) addresses these problems in two ways. First, integration of design and manufacturing is achieved via a single, unified description of the process for use by both TCAD and computer integrated manufacturing tools and systems [16].

Second, in order to achieve integration, accessibility, and sharing of data, a great deal of structure is imposed on process information by the PFR. An *operation* performed on a wafer may consist of any number of ordered constituent operations (to as fine a detail as needed). Three categories of information for each operation are especially important. The *change in wafer state* describes the effect an operation has on the wafer. The *treatment* expresses the physical environment (temperature, gases, dopants) surrounding the wafer during the operation. The *machine settings* describe the controls or knob settings on the manufacturing equipment.

A semiconductor process representation (or SPR) does not by itself provide physical or other simulation capability (just as as SWR by itself does not perform device simulation). Instead, an SPR, coupled with a storage mechanism, can act as a repository and provide an interface for access, storage, and manipulation of process information by other tools. For instance, tools that rely on the use of an external process simulator may use as input both treatment and change in wafer state information for some process flow and generate the input commands for the simulator. A *fabrication system* might access the same SPR in order to schedule the motion of wafers through a fabrication line.

The SPR contains essential data, and is an important "starting point", for many of the kinds of investigations and activities that often arise during process design or analysis. These usually involve sequencing and iteration of process, device, circuit and other simulators or tools. As a result, the SPR is expected to be used not only by TCAD tools such as process simulators, but also via user defined, task oriented programs (perhaps using a framework extension language). For example, a designer might express a sensitivity analysis by writing an extension language program which varies (using an iteration loop) the energy and dose in an implant step (via calls to an SPR database containing the entire fabrication process). A standard SPR is fundamental to achieving such interconnectivity and accessibility in an EDA/TCAD framework.

4 TCAD Standards Efforts

Work toward TCAD standards began in 1985 and resulted in 1988 with PIF[17], the Profile Interchange Format. Based on feedback from industry and universities, development of the next generation of representations is occurring simultaneously in both Europe and in the United States. The European effort by the EDIF Device Modeling and Verification Technical Subcommittee is focusing on designing a topological model of semiconductor devices for intersite communication, using the EXPRESS information modeling language[18]. This information model will be used to generate new EDIF syntax for expressing device models. The wafer and process representation effort presented above is spearheaded by the CFI TCAD Framework Group. The CFI and EDIF TCAD organizations, which have many members in common, are working together to ensure that their models and interfaces will be compatible.

5 Conclusion

The importance of Technology CAD in the design of integrated circuits and technologies to achieve high performance, reduced cycle time, enhanced reliability, and increased manufacturability has been shown. Based on this need for TCAD capability, TCAD should be integrated into conventional EDA systems. An essential step toward such integration is the development of new, standard design representations for TCAD. An object oriented approach to the representation of device structures based on the use of geometry and fields has been described. The representation of process information emphasizing abstraction levels within the process has been described. Development and standardization of these representations, coupled with integration of TCAD tools and traditional EDA environments, will benefit both technology and circuit design.

6 Acknowledgments

Many person-years of effort have been devoted by dedicated members of the industrial and academic community to create a TCAD Framework. These include the original PIF Working Groups, the SRC TCAD Framework Task Force, and the CFI TCAD Framework Group. Contributions to development of the SWR standard have been made by Martin Giles, Robert Harris, Michael Karasick, Lee Nackman, Sani Nassif, V. T. Rajan, Robert Termain, and D. M. Hank Walker. Collaboration with the EDIF Device Modeling & Verification Technical Subcommittee has provided a great deal of stimulating discussion. Work by G. Chin was partially performed under SRC Contract 90-SP-101. Work by D. Boning was performed while attending the Massachusetts Institute of Technology under DARPA contract MDA-872-88-K-0008. Work by A. Wong was performed under SRC Contract 90-DC-008. SRC and DARPA have also played a large role in the establishment of the TCAD Framework effort.

References

[1] D. C. Cole, E. M. Buturla, S. S. Furkay, K. Varahraymyan, J. Slinkman, J. A. Mandelman, D. P. Foty, O. Bula, A. W. Strong, J. W. Park, T. D. Linton Jr., J. B. Johnson, M. V. Fischetti, S. E. Laux, P. E. Cottrell, H. G. Lustig, F. Pileggi, and D. Katcoff, "The use of simulation in semiconductor technology development," *Solid-State Electronics*, vol 33, pp. 591-623, 1990.

[2] K. C. Saraswat and F. Mohammadi, "Effect of scaling of interconnections on the time delay of VLSI circuits," *IEEE Trans. Electron Devices*, vol. ED-29, pp. 645-651, 1982.

[3] A. H. Zemanian, R. P. Tewarson, C. P. Ju, and J. F. Jen, "Three-dimensional capacitance computations for VLSI/ULSI interconnections," *IEEE Trans. Computer-Aided Design*, vol. CAD-8, pp. 1319-1326, 1989.

[4] K. Lee, Y. Sakai and A.R. Neureuther, "Topography Dependent Electrical Parameter Simulation for VLSI Design," *IEEE Trans. Elec. Dev.*, vol 30, pp. 1469-1474, 1983.

[5] G. P. Rosseel and R. W. Dutton, "Influence of Device Parameters on the Switching Speed of BiCMOS Buffers," *IEEE J. Solid-State Circuits*, vol. JS-24, pp. 90-99, 1989.

[6] P. J. V. Vandeloo and W. M. C. Sansen, "Modeling of the MOS transistor for high frequency analog design," *IEEE Trans. Computer Aided Design*, vol. CAD-8, pp. 713-723, 1989.

[7] R. W. Knepper, "Modeling advanced bipolar devices for high performance applications," *IEDM 1990 Technical Digest*, pp. 177-180, 1990.

[8] V. Marash and R. W. Dutton, "Methodology for submicron device model development," *IEEE Trans. Computer Aided Design*, vol. CAD-7, pp. 299-306, 1988.

[9] C. P. Wan, B. J. Sheu, and S. L. Lu, "Device and circuit simulation interface for an integrated VLSI design environment," *IEEE Trans. Computer Aided Design*, vol. CAD-7, pp. 998-1004, 1988.

[10] E. W. Scheckler, A. S. Wong, R. H. Wang, G. Chin, J. R. Camagna, K. K. H. Toh, K. H. Tadros, R. A. Ferguson, A. R. Neureuther, and R. W. Dutton, "A utility-based integrated process simulation system," in *IEEE 1990 Symposium on VLSI Technology, Dig. Tech. Papers*, pp. 97-98, 1990.

[11] C. L. Chu, private communication.

[12] G. Chin, Z. Yu, and R. W. Dutton, "A tool towards integration of IC process, device, and circuit simulation," *Proceedings of the Custon Integrated Circuits Conference*, May, 1991.

[13] S. R. Nassif, A. J. Strojwas, and S. W. Director, "FABRICS II: A statistically based IC fabrication process simulator," *IEEE Trans. Computer-Aided Design*, vol. CAD-5, pp. 40-46, 1984.

[14] M. B. McIlrath, D. E. Troxel, D. S. Boning, M. L. Heytens, P. Penfield, Jr., and R. Jayavant, "CAFE: The MIT computer-aided fabrication environment," *Proceedings of the International Electronics Manufacturing Technology Symposium*, 1990.

[15] CFI TCAD Framework Group, Semiconductor Wafer Representation Working Group, editors A. S. Wong, W. C. Dietrich, Jr., and M. S. Karasick, "Semiconductor wafer representation architecture - version 0.1," *CAD Framework Initiative, Inc.*, 1991.

[16] M. B. McIlrath and D. S. Boning, "Integrating process design and manufacture using a unified process flow representation," *Proc. Second Intl. Conf. on Computer Integrated Manufacturing*, May, 1990.

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

[18] W. Drangmeister, G. Hinrichs, D. Schroeder, and P. Conradi, "Data interchange between technology development and circuit design CAD tools," *Proc. 6th Annual USA EDIF Conference EDIF World 1990,* pp. 11-19, 1990.

Figures

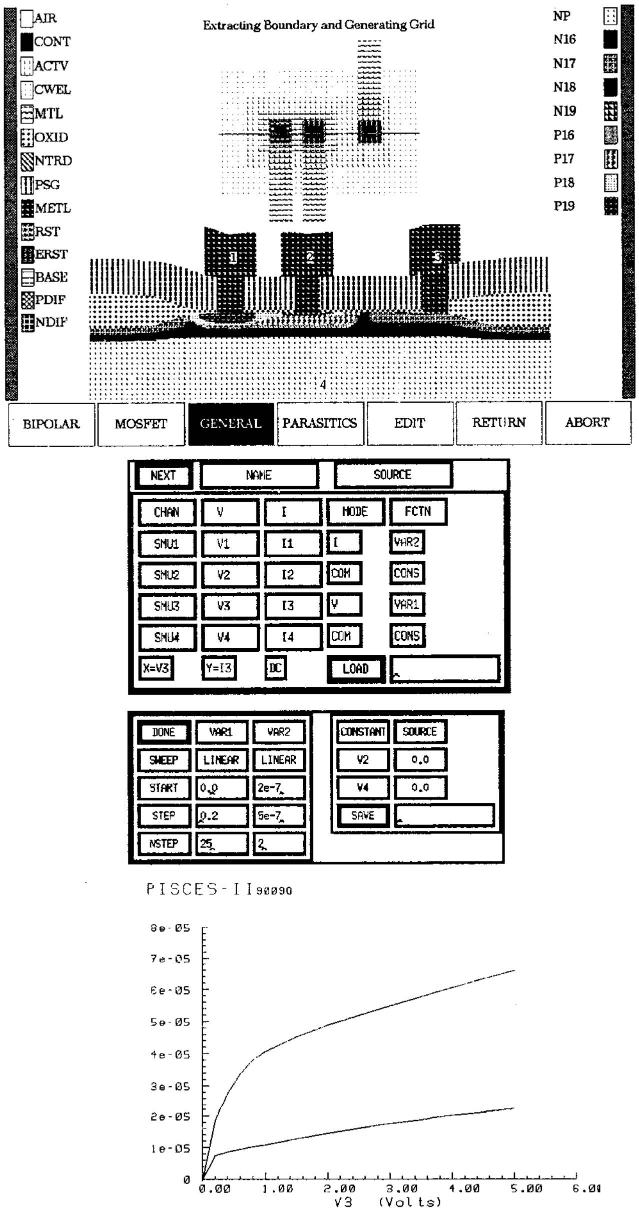

Figure 1: Use of SIMPL-IPX to extract Model Parameters. From top to bottom — simulation window showing cross section and layout, measurement panel for biasing device, and IV curves produced by device simulation.

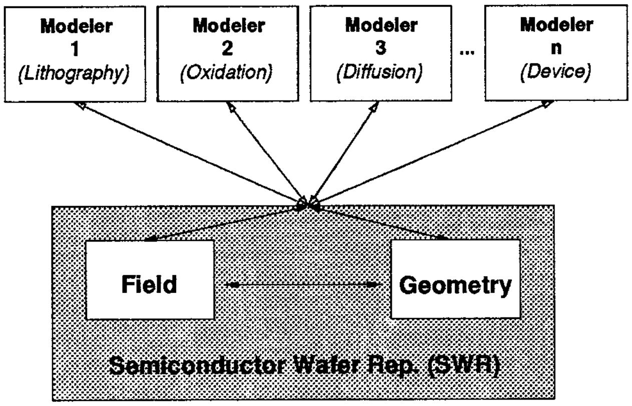

Figure 2: The SWR architecture showing 1) how tools communicate directly only through the SWR, 2) the partitioning of the SWR, and 3) direct communication between field and geometry components (Figure taken from [12]).

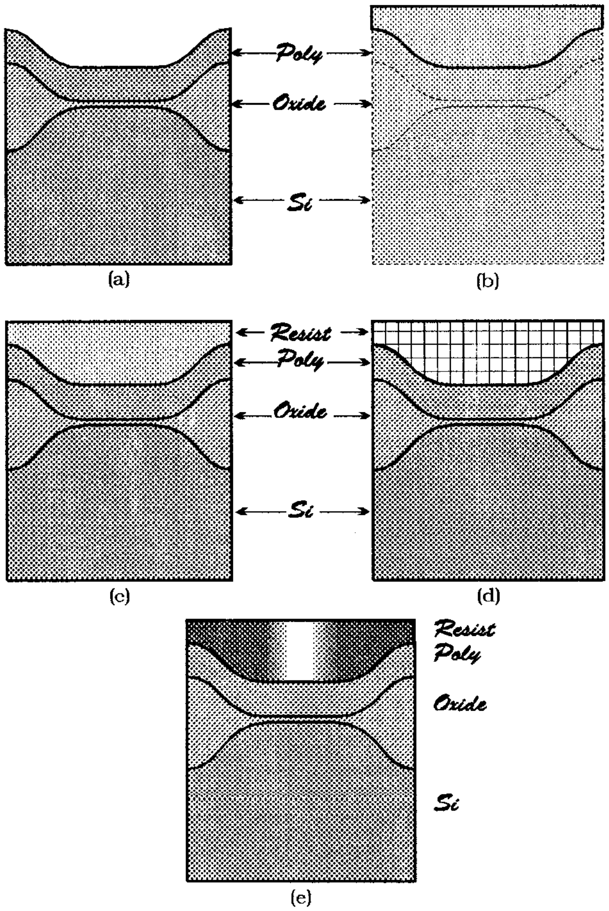

Figure 3: "Snapshots" during SWR example execution: Before (a) and after (b) calling deposit_layer, after calling setMaterial (c), after calling createMeshedField (d), and after calling perform_lithography, showing the field values for illumination per square micron (e).